# Multiple Imputation in Predictive Modeling of Arthroplasty Database

*Author: Øyvind Svenning Berge*

*Supervisor: Ankica Babic*

Masters Thesis


Department of Information Science and Media Studies
University of Bergen
Norway

June 1, 2019

**Abstract**

This thesis presents a method of imputing missing values in data, creating a simple data mining tool and using data mining to see whether such imputed data can be used to predict failures of hip prosthetics in smaller databases.

The data set used for this thesis is based on explanted prosthetics from total hip arthroplasty revision surgeries. It is in the early phases and is rather a small data set with many missing values. Multiple imputation was used to estimate missing values in an attempt to build a more complete dataset to perform predictive modelling. A simple linear regression and multiple linear regression were used with a prediction function for linear models. While the initial results of the imputation looked promising, comparisons with the original data and the imputed data did not show much improvement.

The data was also used in a prototype application for data mining that allows the users to input their data and select the variables for analysis, and present a plot and summary of the model. The application, which is the artefact of this research is fully functional, but simple. Creating larger and more general applications in R can get complex, and other technologies might be more suitable. However, it is a very powerful statistical tool for special tasks and modelling.

Data mining was used to explore the potential to make predictions with the data. Using linear regression on both the original and imputed data showed that the results were similar overall, but with some significant differences. The methods of validation indicated that, while the model was not great, there was something to gain from it. Predictions were run by a multiple linear regression model on both sets of data, displaying some difference but not enough to draw conclusions about the effect and contribution of data imputation.

Currently, the methods in question will have to be further refined, preferably in collaboration with experts. The application can be expanded, but a different approach should be considered based on the scope of any future research. The data mining, even when applied on limited data sets, shows potential and encourages applications of data mining methods from an early stage of research when data collection begins.

# Acknowledgements

First of all, my sincere thanks goes to my supervisor Ankica Babic, whose knowledge, guidance, motivation and endless positivity has been invaluable to the completion of this thesis. Not just for me, but for all the Master's students in the Department of Information Science and Media Studies. Your continued support and generosity is very much appreciated.

Thank you to Doctor Peter Ellison whose assistance, patience and knowledge helped lay the groundwork for this thesis. It would not have been possible without your help.

Another thanks goes to my friends and fellow Master's students who have contributed to a fantastic working environment. Their support through the entire process is appreciated, as is their ability to provide a good distraction when needed.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Introduction

This thesis aims to explore ways to use information technology in pursuit of prediction for patient outcomes in hip arthroplasty, using data collected from patients who have undergone the procedure. Worldwide, more than one million surgeries of this kind are performed every year, with a not insignificant percentage being on already operated patients. Some prosthetics only lasting a few years before needing to be replaced [Zhao, 2016].

The data for this thesis is based on data from total hip arthroplasty surgeries with the Spectron EF hip implant collected by Huakeland Hospital in Bergen. The goal is to see if the collected data can be improved and used for analysis and prediction. This particular data set is in the early stages still, and contains a limited number of observations. However, it is still important to gain knowledge based on growing data.

## 1.2 Motivation

Total hip arthroplasty surgeries are used to help alleviate pain for the patient. The outcome of the surgery is usually very good and improves the quality of life for the patient. With time however, the implants change and the prosthesis will wear down, which can cause it to become loose and painful. This means that in time the patients might need a revision surgery.

These repeat surgeries are a consequence of the wear and tear of the implanted prosthetics and can vary greatly from patient to patient. Total hip arthroplasty is a procedure that replaces the damaged hip joint with a metal stem and cup that has the same basic functionality of the hip joint, to help relieve pain for the patients and improve their overall quality of life.

This thesis seeks to examine data from explanted Spectron EF implants to analyse and see whether it is possible to predict the longevity of the implant and which factors lead to failure. This data is hard to come by and is not always complete with as much information as it would be good to have. As such, the thesis is also looking at ways to fill in the missing values and thereby expanding the usefulness of the data for analysis.

## 1.3 Research question

The work done on this thesis was done in an attempt to answer the following research questions:

1. *Can a relatively small and incomplete data set be prepared and expanded, to apply data mining techniques and still produce reliable results?*
2. *Can this data be utilized by well known methods of data mining?*
3. *Can this data be used to help predict outcomes for patients that have undergone arthroplasty surgery?*

## 1.4 Outline

This section outlines the overall structure of the thesis, the chapters and their contents.

**Chapter 1:** Introduction

This chapter contains the general introduction to the project, the research questions and motivation behind it.

**Chapter 2:** Theory and related work

Explaining the theory behind the project. The practical aspects of the development itself and the technologies that are being used. This chapter also includes a look at related work.

**Chapter 3:** Methods

Laying out the methods used for the thesis itself, as well as methods and technologies used for development.

**Chapter 4:** Development

Going into detail about the development of the project, including planning, design, prototyping and implementing data mining methods.

**Chapter 5:** Results

Presenting the results of the work done with the data itself, the development of a prototype and applying data mining.

**Chapter 6:** Discussion

A discussion on the research performed in this project, the methods used and results from the development and analysis.

**Chapter 7:** Conclusions and future work

Drawing conclusions based on the work done in the thesis and looking at what could be improved upon in the future.

# Chapter 2

# Theory and related work

## 2.1 Total Hip Arthroplasty

Total hip arthroplasty is a surgical procedure that entails removing the head and neck of the femur and inserting a manufactured head and stem with a ball joint to replace the old one. This can significantly reduce the pain experienced by patients undergoing the procedure and is a very common surgical procedure. [Siopack and Jergesen, 1995]

It is made up of three main parts: A femoral stem, a femoral head and an acetabular cup. The head of the stem is set in the cup allowing for the joint movement that the hip requires [Hallan, 2007]. An illustration of this can be seen in Figure 2.1. Some of the data used in this thesis is about the wear on these parts.

The wear of the implanted prosthetic can cause several problems. The breaking down of the material can cause osteolysis. This can occur when particles from wear of the prosthetic cause inflammation and start to destroy the bone the implant is situated in. Eventually this could cause the prosthetic to loosen and cause pain for the patient, requiring a revision surgery. Wear can also cause a mechanical failure, with the parts themselves dislocating or not working as intended. These issues are not separate but related and

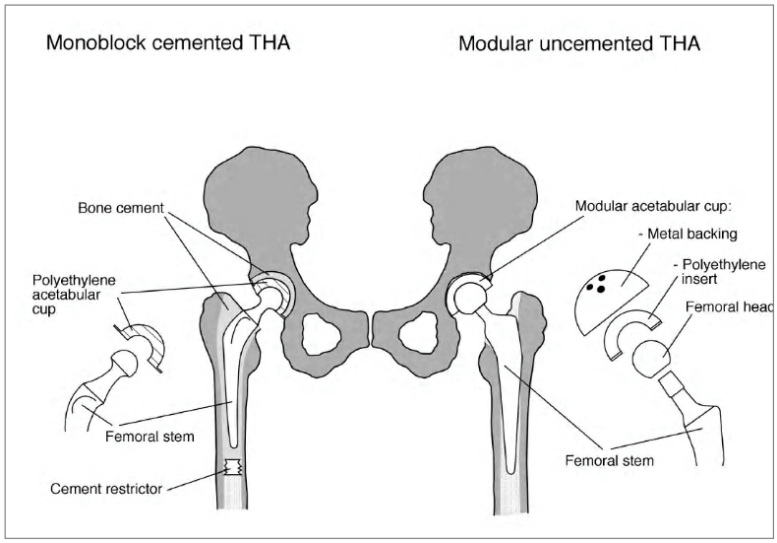



Figure 2.1: Illustration of the basic concepts of total hip arthroplasty [Hallan, 2007].

can hasten the other. The wear of the join that can cause the mechanical failure it also a contributor to the particles and resulting osteolysis. When the implant becomes loose without there being an infection, this is called aseptic loosening and the causes are not necessarily clear.

A revision surgery, while often necessary, is not desirable for several reasons. Any surgical procedure has some element of risk to it, and larger ones usually have more. It takes up time and resources for the staff and the patient will have to go through another round of recovery. In addition to the risks of surgery in general, a revision surgery has other risks as well. The surgeon has to be careful to avoid doing any further damage when removing the old implant. This can be fractures or otherwise damaging the bone. All the materials from the previous surgery has to be removed as well and there will be even less bone to work with for implanting the new prosthetic. On top of this, risks of infections increase and the overall risk of loosening increases substantially. This makes having the prosthetic last as long as possible an important issue, with the preferable outcome being not having to do revision surgery at all [Siopack and Jergesen, 1995], [Zhao, 2016].

## 2.2 Related work

### 2.2.1 Investigation of mechanisms leading to early aseptic loosening of hip prostheses

This is a masters thesis that looks at the possible causes for aseptic loosening of the prosthetic after total hip arthroplasty surgery. It is from the Faculty of Medicine and Dentistry at the University of Bergen and goes into grater detail on the medical and technical aspects of this issue. This work was done using a similar dataset based on the explanted implants after revision surgery.

The data includes the prosthetic itself, blood samples, biopsies and X-rays. It places a lot of focus on the wear of the implant, which also causes the particles found in the blood and is thought to be one of the important factors in aseptic loosening. The thesis finds that these particles are caused by mechanical wear. The particles can also cause further wear themselves, releasing even more particles leading to an eventual loosening of the prosthetic.

This thesis is also useful in that it provides a background for the issue of total hip arthorplasty surgeries. Both on a general basis in presenting the overall concept of the prosthetic itself, and the problems with loosening and revision surgeries. The background provided by the thesis also help provide background for this one in the introduction [Zhao, 2016].

### 2.2.2 Designing an e-learning platform for patients undergoing hip replacement surgery

This masters thesis also focuses on arthroplasty surgeries, but more on the user experience side of things and on developing an application for patients undergoing the procedure. It starts off with introducing and explaining the concept of total hip arthroplasty and the challenges that the patients may face. The artefact that is the goal of the thesis is an e-learning platform

to assist the patients in educating themselves in preparation for the procedure.

The main focus is on user experience and evaluation, as well as the development process itself. Carlsen uses the Design Science research methodology, following the steps in the process to create an artefact, the e-learning platform, and performing user evaluations. As the end user for this application are the patients, there is a large focus on usability and user experience. Going through several iterations to improve upon the prototype and using well known evaluation methods to evaluate the end product.

One of the main goals of this project was to improve patient safety though the e-learning platform. Giving the patient more insight into the subject and informing them on both preparation and rehabilitation was could help increase their safety. The e-learning platform could assist the patient in providing this information in an interactive and easy to use way and be a boon for both patients and the medical personnel working with them [Carlsen, 2018].

### 2.2.3 HALE, the Hip Arthroplasty Longevity Estimation system

This is a masters thesis detailing the development of a system called HALE, short for the Hip Arthroplasty Longevity Estimation system. The same data is used here, but with a focus on using machine learning methods to develop a system for biomedical engineers and physicians. It could then be used for analysis both for individual patients and patient groups to give the best possible treatment.

The HALE system was made using the Python programming language for the back end and machine learning parts. The front end was made with standard web-technologies and by using a framework to allow the front end and back end to communicate and be able to run Python code from a web browser. The end result is a web application that allows inputting information through

forms, and an analysis part that lets the user choose some parameters for analysis before being presented with the results.

There is also a thorough explanation of how the analysis was performed and validated through various methods. Longberg used a combination of open source software like scikit-learn for Python, and IBMs SPSS software. He discusses the pros and cons of the methods he used and the results of his analysis of the data using those methods. This is a similar project in that is uses the same data, but with a focus on the user experience and design side of things in addition to the technical aspect [Longberg, 2018].

## 2.3 Literature review

Before beginning development of the project, a literature review was conducted to look at similar projects that could be relevant to the development of this. There are papers that deal with similar subjects such as recommender systems and using data mining in medical informatics which can help to serve as inspiration.

### 2.3.1 Healthcare information systems: data mining methods in the creation of a clinical recommender system

This paper proposes a recommender system for healthcare based on data mining. Their reasoning for this is that they can mine historical data to automatically extract rules, instead of using an expert system. Additionally, data mining would be able to deal with changes and different practices for each hospital, by detecting patterns that are unique to them. Much of the paper is dealing with how to interpret and weigh recommendations as well at the data structures they use.

They also describe an experiment they ran on data from a real world hospi-

tal and how their system fared. They conclude that their system performs well and does so in real time, and complements an expert system and other decision support, making systems more connected and quick to update. This is a good example of how data mining can be used in healthcare [Duan et al., 2011].

### 2.3.2 Report Generation and Data Mining in the Domain of Thoracic Surgery

This paper describes a reporting system for a decision support system called AssistMe. The software was made to help thoracic surgeons do analysis and research, and uses data mining techniques like case based reasoning and cluster analysis. This allows them to make decisions based on data from previous cases and help them gain a better understanding of how to treat patients with similar cases, among other things. This paper focuses on the reporting component, which is also based on data mining techniques. When determining requirements for the system, they ended up with three user groups. Surgeons or physicians, administrative users, and patients. Each group have a different use for the system and with different level of permissions or access to sensitive data. Surgeons need all available information, including sensitive information. Administrative users generate reports, find info and preforms analysis on the data. And patients or other users can use it to learn more about about it and have the information presented to them in an easily accessible way.

There are many ways to make a report, but they have condensed it down to two main types: Predefined reports, and customized reports. Predefined reports are already programmed for their intended use, making it easy for the user to input the data. Customized reports add more interactivity and gives greater control over it, but with the downside of added complexity. They both have advantages and disadvantages, and they argue that the problem can be solved by including both in the system. They also emphasize the importance of a well presented report. The viewer should easily be able

to tell what the data says. Additionally, statistics displayed in tables and graphs helps visualize the data and differences between them. Making it well organized and easy to understand and digest is a critical part of making reports. Another point they make, that is important for data mining in general is how to handle missing values in the data. In a perfect world, one would not need to worry about this, but this world is far from perfect and missing values is always an issue. Working with medical data requires extra attention to detail as well. The solution would be to either use probability to make a new value based on previous ones, fill it in somehow, or ignore it. Given the sensitive nature of the data, that latter two were used here.

As a result of this research, they ended up with a set of predefined reports useful for instance for reports that are generated often and used for simplicity. They also support custom reports that allow interaction on a deeper level to help customize the report for their purpose. This combining more data for analysis and discovery of new knowledge. Their solution is web-based, allowing for real time updates of the data, as well as easily displaying it in graphs and tables that will also update accordingly. They also mention in the final section that the ability for the user to interact with and customize the system and make discoveries on their own is important for them to have trust in it and willingness to use it.

A final not is their step by step description of their data mining process. They used KDD, talked about in Section 2.4.2, as their process and provide a relatively short but well explained explanation for each step. While some parts of this paper might be dated simply due to its age and the rapid development of informatics, the data mining process and techniques are still very relevant [Voznuka et al., 2004].

### 2.3.3 Case Based Reasoning in a Web Based Decision Support System for Thoracic Surgery

Similar to the paper in Section 2.3.2 this paper deals with a decision support system for thoracic surgery, but this time focusing on the reasoning itself. They make the case that due to significant differences in reports of mortality rate, and choosing the right treatment for each patient is important. To help with this they developed what they describe as a web-based Clinical Decision Support System, or CDSS that utilizes case based reasoning.

This system allows them to find patients that have been treated previously and with similar medical background, and show information about them. Reports from their treatment and any other relevant information about them is displayed and can provide a picture of what has worked before, and can then be used in new cases with similar circumstances, which will give the physician a better basis to work with. To find similar cases, they employed the nearest neighbour algorithm, with a customizable number of variables where each variable has a value determining the importance of that variable relative to others.

When using the system, the user enters a web page that gives access to the database of current patients. Here they can enter identification for a patient and get a list of the variables and values which they can add to and edit. They can retrieve patient data for similar cases by selecting the desired variables, tweaking some options and then running the algorithm. They system will then give a list of similar cases which is presented in a user interface on the web page. To evaluate the system they had three physicians study cases to compare results with it. They had good results with two of them, but one did not match up well.

One of the major issues was missing values, and the one that did not do as well had tried replacing the missing value based on other similar cases. How to handle missing values is important in such a system, and shows an interesting difference in how human and algorithm handles it. They also commented on

how showing operation records together with a patients clinical records was useful, which further emphasizes the importance of good design and user experience.

They concluded that the background and experience of the user affects the functionality of the case based reasoning engine. While the method of choosing similar cases is mostly the same in both cases, the number of variables considered can differ, as well as how they are weighted. Additionally, people with different background might focus on different variables in their evaluation. Some might feel a variable is more important than others, and will assign a different importance to it. These are useful things to take into account when making a system, and can explain certain differences when evaluating.

They further discuss the issue of missing values and how to handle it as well. Substitution with a mean value appeared not to be appropriate, and one should attempt to make the replacement value as close to the true value as possible. Several possible methods are mentioned, one being to find the closest related case using the system and taking the value from the one most close to the case with the missing value. They explain that further work on the system in regards to missing values would enhance the system [Babic et al., 2014].

## 2.4 Data mining

### 2.4.1 Data mining

Data mining is a process used to find patterns and gain information and knowledge from data. It has been used for many years under various names, and is an ever evolving and growing field today, in large part due to the significant increase in the amount of data that is produced as technologies changing and new ones are developed. The fact that is has been around for so long means that there is a lot of relevant research, and ample documentation

to back it up. You can find papers, articles, books and blogs about data mining in all kinds of scientific fields, as well as for business purposes and even recreationally. The various tools and techniques used in data mining have evolved and improved over the years and continue to do so as our digital society is rapidly moving forward.

One of the fields that data mining is very useful in is medical informatics. In the world of medicine, vast amounts of data is produced all the time, much faster than one can put it all to use. Each patient has their own medical history and all the information that encompasses. The doctors and other medical workers leave a long paper trail in the system that contains various information that could be useful later, which is then stored somewhere. Much of this is of course used later on, to provide an overview of the patients' medical history or other information medical workers might need at a later time. Much of this clinical data can and will likely go unused, or at least underutilized. With the rapidly evolving technologies at our disposal, the potential for analysing the data and putting it to even better use is greater than ever.

There is a large crossover in the fields of data mining and statistics. Many statistical techniques can be and are used in data mining, they are essential to the process. The wide applicability of many of these techniques means that there is a lot of literature on them from a great variety of sources and a lot of documentation to be found. The task it to know of or find and adapt these techniques for data mining of the chosen data. Some methods are commonly used and are easy to find documentation for and examples of use, while others might be more obscure or specialized and require some digging to unearth and properly apply.

There there is also the issue of the field of data mining always evolving and new techniques being developed. Machine learning has become much more prevalent in recent years due to a combination of factors, and is quickly becoming a household term. But due to it's relatively recent popularity, it can also be more difficult to find the most relevant or thoroughly tested

methods. And it will of course also depend on if it is suitable for the data at all.

Another important aspect of data mining and this project will be the tools and technologies used for the actual data mining process and presentation of the data and findings. Choosing the right tool for the job is important. There can be a lot of overlap between tools used for data mining, so choosing the one that is most suited is also a part of the process. The technologies and tools that have been considered and used are explored further in Section 3.3.

### 2.4.2 The data mining process

Data mining has been around for a long time, and has many different methods and practices. New technologies and improvements emerge all the time, that can change and enhance the process. So while the core concepts remain the same, some of the finer details might change in time. For example, some older papers on the subject recommend using lower and less intensive settings to compensate for the relatively low computing power at the time. This could potentially result in less accuracy, but saving a lot of time. This is of course still something to take into consideration, but computing power has increased by orders of magnitude in recent years, and the tools used have become more efficient and advanced.

There are also different approaches to data mining as a whole to consider. These formalized processes are usually divided into several steps and these might differ depending on the process. The main steps can be generally divided into processing and preparing the data, the data mining itself, and finally validating and reviewing the results. While these processes have much in common as they in large part do the data mining similarly, they can be made for different purposes and differ in the details.

### 2.4.3 KDD: Knowledge Discovery in Databases

The process chosen for this thesis is Knowledge Discovery in Databases, or KDD [Fayyad et al., 1996]. According to Fayyd KDD is "the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data." It is also described as an interactive and iterative process. There is user interaction at most of the steps in the process, which can greatly influence the results. From the selection of data, to selecting algorithms and models and interpreting the data. It is also iterative in that you can learn from it and run it again to improve upon it as needed.

**Selection** is the first step, where you familiarize yourself with the project domain and gain a greater understanding of the subject, the requirements and the goals.

**Creating target dataset** to use for your data mining either by using an existing dataset, or selecting part of a dataset.

**Pre-processing** involves collecting the data into something usable. It must be large enough to be useful for discovering patterns, but not so large that the time it takes to process is prohibitively long. It also includes fixing any errors like missing values and noise in the data.

**Transformation** uses various methods to make the data better suited for the goals and prepare the data for use, for example by removing extraneous variables.

**Choosing data mining method** that matches the goal set for the project in the earlier steps.

**Analysis** and selecting algorithms and models suitable for finding patterns in the chosen data.

**Data mining** using the chosen methods best suited for the goal and the actual searching for patterns in data happens here. The previous steps lays the groundwork for this.

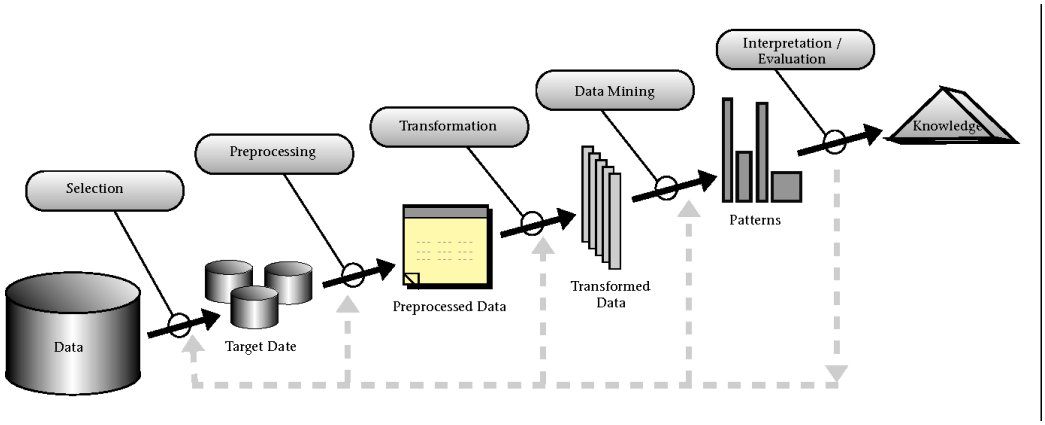



Figure 2.2: Knowledge Discovery in Databases [Fayyad et al., 1996].

**Interpretation/evaluation** after the data has been mined. Any discoveries have to be interpreted and evaluated to see if the goals have been fulfilled. You might want visualisations of the data, or a report summarizing the findings and anything else needed to present the findings. The end result after this is hopefully new knowledge.

**New knowledge** is hopefully the result of the process, which can then acted upon and used, presented to interested parties, added to the pool of existing knowledge and used for further work.

These steps detail all of the KDD-process from learning about the domain to getting new knowledge from the process. One does not need to follow each step exactly, but it provides a general guide for how to perform data analysis that is useful for this project. A visual representation from the Fayyad paper can be seen in Figure 2.2. This visualizes the most important steps in the KDD process: Selection, pre-processing, transformation, data mining and interpretation.

The first and second step of KDD has already been done by others when collecting data for this project. The first data set provided is a preliminary selection that was a good candidate for data mining, but with much potential for improvement. The second data set was longer but narrower which in this case is a good thing. It is a larger sample size with fewer variables to consider

where the chosen variables are more likely to be relevant.

For this data set, the first and second steps were also performed by others before receiving the data. The transformation step with the raw data consisted mostly of removing observations with too many missing variables to be of use, and was also done in the form of multiple imputation in an attempt to improve the data for use.

The data mining step was done with well known techniques and algorithms to look for any patterns and informations that could be gleamed from the data. Visualizations were produced and interpreted to evaluate the results which the final discussion and conclusion is based on.

# Chapter 3

# Methods

## 3.1 Design science

Design Science research is a methodology used in information technology that focuses on the development of a novel artifact to solve a problem and provides guidelines for the process [Hevner et al., 2004]. This more practical and hands on approach is in contract to another research paradigm, behavioural science research, which takes a more theoretical approach to explain or find solutions to a problem. Hevner et. al. argues that while the goal of behavioural science research is truth, and the goal of Design Science research is utility, but both can be useful for the other. A short summary of the guidelines for Design Science Research can be found in Table 3.1.

**Design as an Artefact:** An artefact in relation to Design Science research as defined by Hevner et. al. can be construct, a model, a method or an instantiation. While they acknowledge that the organization and implementation of the artefact is an important part as well, they maintain that the creation and capabilities of the artefact are crucial for Design Science research.

In this case, the artefacts are both the data that is the result of various applied methods and the application that is built for utilizing the

| Design Science Research Guidelines | |
|---|---|
| **Guidelines** | **Description** |
| Design as an Artifact | Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation. |
| Problem Relevance | The objective of design-science research is to develop technology-based solutions to important and relevant business problems. |
| Design Evaluation | The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods. |
| Research Contributions | Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies. |
| Research Rigor | Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact. |
| Design as a Search Process | The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment. |
| Communication of Research | Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences. |

Table 3.1: Design Science Research Guidelines [Hevner et al., 2004].

data. These can be seen and used as separate artefacts as they are not dependent on each other and made separately, but they could enhance each other when used together. There could also be an argument for the collection of methods used to be seen as an artefact as well.

**Problem Relevance:** This is the second guideline for Design Science, explaining how the artefact and research process should be approached in relation to the problem at hand. This is, according to Hevner, "the construction of innovative artefacts aimed at changing the phenomena that occur.

Expanding the data that can be used to work on the problem for this thesis is one part of it, another to put this data to use in a system to assist those working on it, researchers and doctors, in discovering patterns that ultimately can help the patients.

**Design Evaluation:** Using well thought out evaluation methods to evaluate the design of an artefact is important as well, preferably testing with the artefact integrated in the environment that is is intended for at some point in the process, though general usability is also important.

For this project, the evaluation is only of the analytical tool as that is the only artefact that has any direct user interaction and design. Though the new data produced can be of some relevance as well, this is mostly for researchers themselves and not intended for an end user.

**Research Contributions:** This guideline specifies that the artefact must provide clear contributions to at least the design artefact itself, the foundations of the design or the methodologies used.

This project is built upon proven methods for data analysis and data mining, as well as using a very well known statistical programming language to built the design artefact. The methods used for extending and verifying the data, plus the tool itself are together contributing to these areas.

**Research Rigor:** Rigorous methods in both the construction and evalua-

tion of the artefact is also an important part of Design Science.

This project has gone through several iterations with improvements along the way, and well known and documented methods have been used in the development of the artefact. Both in the technical aspect and for validation.

**Design as a Search Process:** This guideline focuses on the iterative process of Design Science research. Creating and improving upon designs within the bounds of the project through iteration to ultimately end up with a good design.

For this thesis, this part lies mostly in the iterations of various methods uses to explore and expand the data, and designing an artefact within the bounds of the chosen development environment.

**Communication of Research:** The final guideline details how the research and findings should be presented, both to a regular audience and to those who are more technologically minded.

This thesis in itself provides the documentation of the project and presentation of the results. It should be detailed enough for allowing others to understand what you have done on a base level, for those interested, as well as easy enough to get the gist of without much previous understanding of the subjects.

## 3.2 Development Methodology

When it came down to choosing a development methodology, there were several factors to consider. First of all, this would be developed by one person, not a team. As many development methodologies are aimed at aiding communication and work in teams, there are a lot of redundant steps in the process. This provides the luxury of being able to choose how to develop very freely.

There is also a focus on the business side of things which is not particularly relevant here. While it is developed towards an end "product", this will only be an early prototype at best and not something that it intended to be handed off as complete. Additionally, the requirements are not set in stone and will evolve as research continues and the project develops.

The parts that were used from common methodologies were for instance something akin to a Kanban board to keep track of tasks that needed doing and what had been completed. It was also an iterative process as many methodologies use, using prototypes and iterations to make progress towards the end goal. In the end, it shared the most similarities with a Kanban project, but working solo. It was useful for organizing and keeping an overview of what needed doing, but not something that was followed religiously.

## 3.3 Technologies

This section contains information and explanations for the various technologies that have been used when working on this thesis. The main tool that was used is the R programming language, which was used with the RStudio Integrated Development Environment (IDE). Additionally, there were many packages for R that were used to a larger or smaller extent. There were several other tools used throughout the development as well, which are explained below.

### 3.3.1 R

R was chosen as the main tool for working with the data and developing the artefact. There are several reasons for this choice. The main reason for choosing R is that it is a tried and true language used for statistical modelling and analysis for many years. It provides many ways to handle and make useful operations on the data to make it better suited for whichever

purpose you require it for. In addition to this, R makes it very easy to produce high quality graphical representations of data and generate graphics based on the analysis you perform. This includes plots, graphs, diagrams and much more, all with a high level of customizability. This makes R well suited as an all in one package for the data mining process. All the way from importing the data to preparing and performing analysis and producing the final graphics for publication [R Project, b].

Another major advantage of R is the community and packages and plug-ins made for it. R has been around since the 90's and was released under the GNU General Public License allowing anyone to modify and distribute it freely [GNU Project]. This has allowed R to build up a substantial community of users and developers over the years, and with it comes many packages for all kinds of purposes. In many cases you can find a packages that already accomplishes what you are trying to do, saving you the time and effort needed to code it from the ground up. These packages can be found on the Comprehensive R Archive Network (CRAN) which has mirrors in many parts of the world at various organizations and universities [R Project, a]. Installing a package is as simple as writing a single line of code, making it very efficient to extend the capabilities of the base R software.

### 3.3.2 RStudio

RStudio is a powerful open source Integrated Development Environment, or IDE, for the R language. It allows for a much more efficient and easy to learn work flow as it significantly simplifies many operations in R. The graphical user interface allows you to make changes that would normally be done with typed commands. You get a file explorer and project overview, console and logs, overview of project variables, direct graphics output and many other things that makes working with R much easier.

It also includes a package manager which is helpful when a project grows and uses many packages, and many other quality of life features. Another thing

of note is that RStudio seamlessly integrates with other relevant software like Shiny, which is also made by RStudio Inc. and is talked about in Section 3.3.3. [RStudio Inc., 2018b]

### 3.3.3 R Shiny

Shiny is a package for R that allows for the creation of interactive web applications based on R. It is created by RStudio Inc. and is fully integrated with the RStudio IDE. Shiny makes it simple to publish interactive applications to the web that update both graphics, text output and data in real time. It uses relatively simple code to make the layout, similar to how Bootstrap for CSS works. As a consequence, most Shiny applications are very similar in appearance as the components are standardised. It is a simple, but functional design that makes it easy to use even if you have little to no experience with web development. It is also possible to customize it further using JavaScript and CSS should you want more control over the web functionality and design [RStudio Inc., 2018a], though this is more of an advanced feature and not required.

It uses reactive functions to run R code in real time, and updating the reactive values in the user interface accordingly. This only requires some code on the developer's part, and the creators of Shiny and the community provide ample documentation to get started, as well as many examples of applications others have made and the source code for them. One drawback is that Shiny requires that you run R on the server hosting the application, which means you will have to use a web host offering this feature, or use your own virtual or dedicated server. Once R is installed, running the application is as simple as putting it in a folder to be accessed by the website and a single line of HTML code on the page you want to run it. Should you not have access to your own server, RStudio offer hosting on shinyapps.io for this specific purpose. They have several tiers of pricing depending on your needs, with the most basic and limited plan being free.

The Shiny package is another big reason R was chosen for this thesis, as the relative ease of use and simplicity allows for the creation of interactive applications with only a little bit of code. It provides a framework that gives you the ability to create something that would otherwise require a great deal of coding and programming experience. There are also many other functions like being able to writing R code to generate HTML code, allowing you to essentially create entire web pages in R, but this is less relevant for this project.

### 3.3.4 VirtualBox

While RStudio comes complete with a way to run the programs and Shiny applications you write in the IDE itself, it was also useful to test it in something akin to a real world scenario. In order to do this, Oracle's VirtualBox was used. VirtualBox is an open source virtualization software that allows you to run unmodified operating systems and software in a Virtual Machine (VM) [Oracle, 2018]. This virtual machine is for all intents and purposes a fully functional computer that runs on virtual hardware.

This makes it ideal for testing as it gives the user full control over every aspect, and it can be customized to closely match the real world conditions you might be working with later. It also allows the testing to occur locally without using a server exposed to the internet, which is useful when potentially sensitive data is involved.

The operating system on the virtual machine was a standard installation of Ubuntu Server 16.0.4 LTS. Linux servers are commonplace and are realistically how you would set up a web server that can run R in the real world. For the web portion of the server, the open source Apache HTTP Server 2.4 was used. This is one of the most used web servers today and has been around for over 20 years. It makes it relatively easy to set up a functioning website that you have complete control over, unlike many other web hosting options that put limitations on what you can change and install.

This set-up made it easy to test the Shiny applications in a simulated real world situation while keeping everything local. And it would be trivial to move everything over to an actual server should it be desirable to put it on the web in the future.

# Chapter 4

# Development

## 4.1 Prepearing the data

Preparing or pre-processing the data, is a crucial part of the data mining process, especially when it is prone to inconsistencies, errors, missing values and other factors. If you use poor data for the analysis, you cannot expect good results either. This was a concern with this data set which has several potential issues. This section describes those issues and what was done to attempt to counteract them.

The pre-processing detailed in this section is divided into two parts; the first part deals with the first data set that was provided, and the second part with the improved and somewhat expanded second data set.

The first dataset that was used for this thesis is a good example of real world data with a multitude of issues that complicate the process of working with it. That does not mean it is not useful, but it does take up time and effort to further prepare it for use.

This dataset is very small compared to what is usually used for data mining and analysis. It has a relatively large amount of missing data. The small number of observations make every variable count if something useful is to

come of it. Some data mining techniques might not require all the variable to be present, but others do and will not work with missing values. R's default response to a missing variable is to remove the entire row. Losing a whole row of data is not desirable, especially considering the already small number in this data set. Therefore, multiple imputation has been used as an alternative to avoid losing too much data due to missing variables, as explained in Section 4.2.

The original data was in Microsoft Excel XLSX format, which demands some more work due to the proprietary nature of it and many different options to get the data in the right format. It is also more work to import the data directly from the file if you do not convert it to another format before importing. Therefore, the document was converted to a Comma Separated Values, or CSV file instead, which is very lightweight and standardised and can be used in most programs without additional plug-ins.

The data was messy from the start, with much excess data outside the main table which that did not seem to belong or have an explanation, and located in places it should not be. Other clutter made it difficult to get an overview. This is something that can be expected in real world situations, where the collection of data is often not ideal. Doctors and other researchers do not have time to consider the requirements of analysts and how to best store the data for use when they are preoccupied with the patients and their jobs.

In the Excel document there were several graphs scattered around, smaller tables of data in various places outside the main table, which doesn't make much sense to a layperson as no context or explanation was provided for that data. Some parts of the table extend beyond the rest, but only for a few of the variables which can be difficult to understand if it is even intentionally added. There are also various color coding on the text and background that probably makes sense to the people who made it, but with no real explanation it was hard to make sense of coming into it later.

When converting and cleaning the data data from the main table in the initial analysis, the rest of the data and notes were discarded, as it would

likely not help with the data mining. There were also at some point going to be a cleaner and easier to work with version of the data set that would come in the future, which made this process more of an initial exploration, preparation and testing, and not the main analysis.

After extracting data from the original file, the result was a CSV file with 27 observations and 47 variables. A visual representation of the data can be seen in Figure 4.1. This shows the different types of data in the data set as well as a "missingness map" of which variables are missing data and how much for each, represented in grey. As it is evident from the visualization, there is a lot of grey, meaning a lot of missing data. The amount of variables relative to the amount of observations also makes it somewhat impractical to fit onto one page while keeping it large enough to read.

Using R, we can calculate the percentage of missing data in the supplied data set. In this case, that comes out at about 22%, a very high percentage. More than a fifth of the values is missing, which means a lot less to work with than apparent at first, and even more so in a data set this small.

Instead of returning the percentage of missing data for the entire set, we can adjust the code in R to show it for each variable, the results of which can be seen in Table 4.1. Here we can see that some variable do not have anything missing, like the Record Number (REK), sex, year of birth, inserted and removed. These are variables that are easy and expected to be recorded, and are not likely to be missing. But the more interesting variables, like the wear on the implant or various trace metals in the blood, have quite a few missing values. The worst case is "Al", aluminium in the blood, which has almost 90% missing and had to be excluded.

We can also see that many variables have the same values missing, meaning the same number of values missing. Given that this is a trend for several variables, it might indicate that they are missing from the same observations; something a quick look at the data confirms. This is most likely due to no sample being taken or recorded.

It is worth noting that the number of variables greatly exceeds the number of

| Variable | Missing % | Variable | Missing % |
|---|---|---|---|
| REK | 0.00000000 | cup.grade | 0.29629630 |
| Sex | 0.00000000 | REK.2 | 0.00000000 |
| Year.of.birth | 0.00000000 | tissue | 0.00000000 |
| Inserted | 0.00000000 | macrophage | 0.14814815 |
| Removed | 0.00000000 | giant.cell | 0.14814815 |
| Years.in.vivo | 0.00000000 | lymphocyte | 0.14814815 |
| Linear.wear.rate | 0.37037037 | poly.count | 0.18518519 |
| Total.linear.wear..calc... | 0.37037037 | darkfield.count | 0.14814815 |
| Total.linear.wear..real. | 0.37037037 | poly.count.from.all | 0.22222222 |
| Osteolysis.area | 0.48148148 | darkfield.count.from | 0.14814815 |
| Osteolysis.score | 0.51851852 | darkfield.density.from | 0.14814815 |
| Osteolysis.percentage | 0.51851852 | ED.median.um. | 0.14814815 |
| Paprosky.acet | 0.00000000 | REK.3 | 0.00000000 |
| Paprosky.femur | 0.00000000 | blood | 0.00000000 |
| REK.1 | 0.00000000 | Al | 0.88888889 |
| side | 0.00000000 | Ti | 0.33333333 |
| loose.component | 0.00000000 | V | 0.62962963 |
| stem | 0.00000000 | Cr | 0.33333333 |
| stem.grade | 0.37037037 | Mn | 0.66666667 |
| stem.wear.percentage | 0.37037037 | Co | 0.37037037 |
| offset | 0.62962963 | Ni | 0.62962963 |
| offset.value | 0.07407407 | Zr | 0.37037037 |
| stem.size | 0.00000000 | Mo | 0.33333333 |
| cup | 0.00000000 | | |

Table 4.1: Percentage of missing data for variables in the first data set.

observations, which is decidedly less than ideal. Many of these observations have missing values and "bad" variable values, as well. There were quite a few cases of non-standard values for variables. For example having text notes on numeric variables, and other special cases that makes it more difficult to work with as those will have to be cleaned up or excluded for the analysis. Many R methods completely ignore observations with missing or non-standard values, so this has to be done to preserve as much data as possible.

It is also of note that that data set only contained data from failed cases, and with no control group data which, again, would come at a later time. Many variables are likely not not be useful at all, and would probably skew the data for the worse. So while some could be useful, others might be detrimental to the analysis and should be removed or at least ignored in the program.

The second data set provided was much more suitable from the start. This data was already in CSV format, so there was no need to convert it and check to see if there were any additional data scattered around the document. It could be directly imported into R and used immediately. Many of the unnecessary variables from before were removed, in fact, the number of variables were more than halved. This left only the variables that could potentially be relevant, removing a lot of the noise from incomplete and superfluous data.

There are still quite a few missing values for several of the variables, not nearly as many as the first. Again, we can use R to find the percentage. This gives us a little over 16% missing data, which is still a significant amount, but better than previously. Using the second method to again display the percentages for each variable, gives us the results in Table 4.2. In this table, we can see that the percentage overall has gone down quite a bit, as expected. "Ni", nickel blood values, is the worst case at almost 30%. With all the others below that, there is a much better chance of the variable having enough data to actually be useful. We can also observe the same trend in this data as in the first one, with several values missing from the same observations, consistent with the previous data. And given that this is an expanded version, it comes

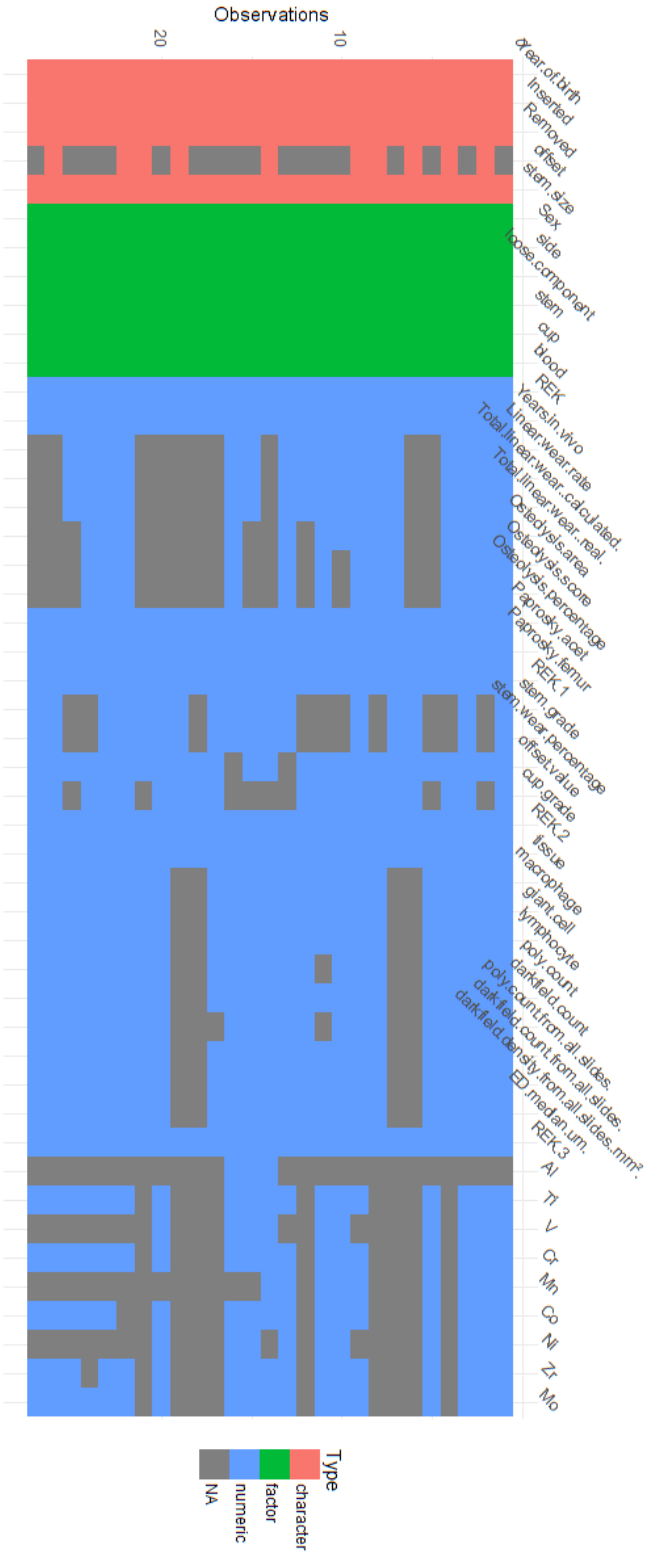



Figure 4.1: Visualization of the data types and missingness of data in the first data set.

| Variable | Missing % |
|---|---|
| id | 0.00000000 |
| Case | 0.00000000 |
| cupLoose | 0.00000000 |
| stemLoose | 0.00000000 |
| sex | 0.08163265 |
| years.in.vivo | 0.00000000 |
| Cr | 0.22448980 |
| Co | 0.22448980 |
| Zr | 0.22448980 |
| Ni | 0.28571429 |
| Mb | 0.22448980 |
| linWear | 0.26530612 |
| linWearRate | 0.26530612 |
| volWear | 0.26530612 |
| volWearRate | 0.26530612 |
| Inc | 0.18367347 |
| Ant | 0.18367347 |
| CupX | 0.18367347 |
| CupY | 0.20408163 |

Table 4.2: Percentage of missing data for variables in the second data set.

as no surprise.

Another improvement in this data set is that the missing values are left blank which means they can be interpreted by R properly as NA values, which it knows how to handle. There are also no text notes and other inconsistencies like text in numeric variables, so the data is clean and ready for use. Consequently, the data needed little to no preprocessing before it could be used, though the same process as on the first data set was still used to check it for potential errors and other issues.

This data was also more extensive than the previous data set with a total of 48 observations, and 19 variables. That is still a very low number for use in data mining, which benefits from having as much data as possible. Some of the new observations in this data are the control cases, where the implant did not fail, which opens up more possibilities for analysis. All in all, this is

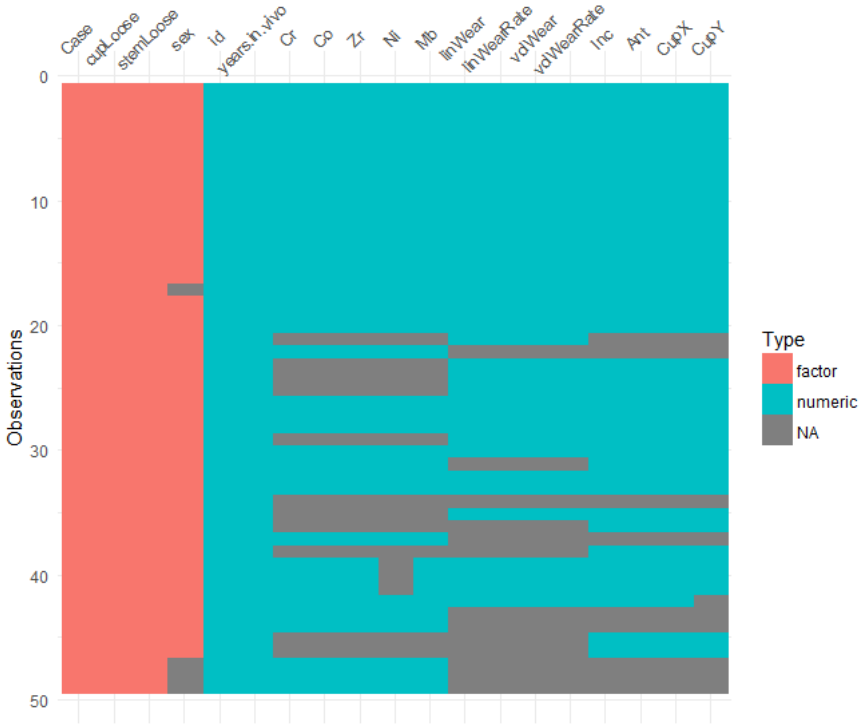



Figure 4.2: Visualization of the data types and missingness of data in the second data set.

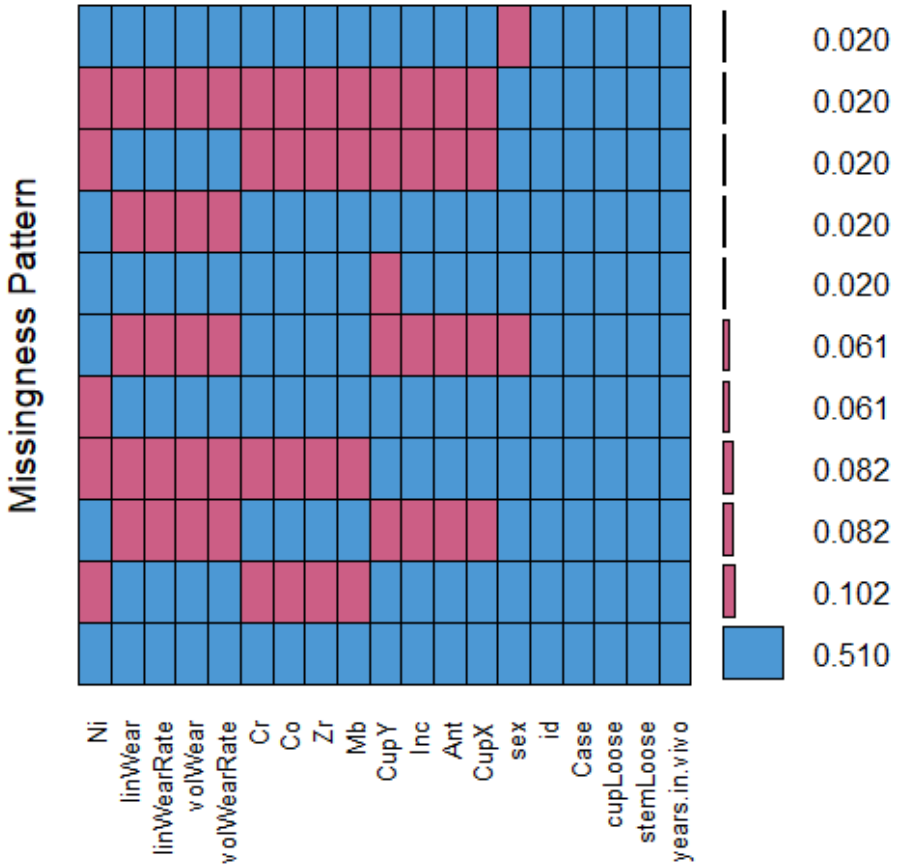



Figure 4.3: Missingness pattern

much easier to work with and more useful than the first data set, though it is still limited by its small size.

## 4.2 Imputation of missing data

This section covers the use of multiple imputation as an attempt to counteract some of the downsides of missing data to the extent that is it possible. It starts of with an explanation of what multiple imputation actually is and the specific tool used for it in this thesis.

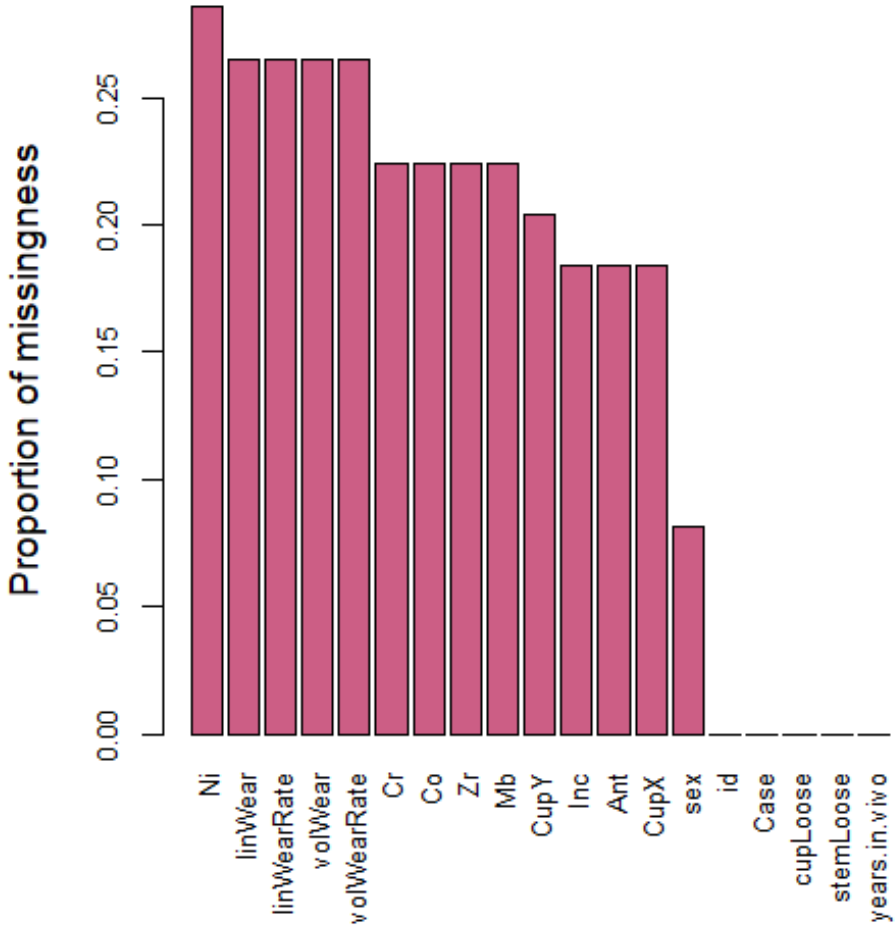



Figure 4.4: Proportion of missingness

This is followed by an explanation of how the process of multiple imputation was applied for the first and second data sets; which parameters were used, issues encountered along the way and what considerations were taken into account for each data set. And finally the results of the imputation are presented, using various methods of comparing and visualising different aspects of the imputed data sets.

### 4.2.1 Multiple Imputation

To handle the missing data in the original data set, multiple imputation, or MI, was used. Multiple imputation is a technique for analysing data sets where missing data is replaced by new calculated values. The strength of MI is that it simulates the data set at hand and replaces the missing values $m$ times, resulting in $m$ data sets with different, but plausible values filled in.

It is then possible to pool the results of the imputations and run analysis on them [Schafer, 1999]. Or to use all the data sets as a longer version of the original, take an average of the imputed data sets, or simply choose one of them to work with. In general when working with data, the more data the better results, and this is the case here too. In R, multiple imputation can be done with a package called MICE, which is short for Multivariate Imputation by Chained Equations.

### 4.2.2 Multiple imputation with MICE

The actual process of imputing with MICE is relatively simple, though some preparation is needed, especially with a complex data set. When importing the data to R, after cleaning and preparing it, several parameters in the import function are used to ensure that the data types were correct, as this can affect which methods MICE will use to impute the data.

To run MICE, a command is used where the only necessary parameter is to

specify the dataset to run it on. If not specified, other parameters will use their default value. There are several other parameters that might be of use, though. The 'm' parameter is the number of imputations to be run. The default value for this is 5, and that is what was used for this first test with the initial data. This has been considered in older literature as sufficient for most purposes [Schafer, 1999].

But the number of imputations has been reconsidered in more recent times both due to more imputations producing even more accurate results, and the fact the the increased computational power makes time and speed less of an issue. It still takes time to run, especially for a large and complex data set. It is easier to run more imputations in smaller data sets like this one as the computational requirements and time needed are a lot less intensive, but it is still worth consideration. As a rule of thumb, the number of imputed data sets should be set around the percentage of missing values in the data you are analysing [White et al., 2011].

The 'maxit' parameter is the maximum number of iterations that MICE should run. A number around 30 should be sufficient for most purposes, but using a larger number of iterations should still provide more accurate results, even if the improvement is marginal.

And finally the 'method' parameter, which allows you to choose the method of imputation, overriding the default. MICE will by default pick the method that it deems best suited for that type of variable.

Once the imputation finishes it results in a 'Multiply Imputed Data Sets' object, or MIDS. This object contains all the imputed data sets and is very useful for comparing the imputed data with the original, to see how similar the imputed data is to the original and assess whether the imputation was good. There are multiple plots that are useful for this purpose. This can also be of great help if choosing one of the imputed data sets to work with further.

### 4.2.3 Imputation of the first data set

The first data set that imputation was tested on had a fair share of problems initially, as explained in Section 4.1. There are variables of different types; numeric, dates, characters and factors. There are numerous variables as well, a total of 47, though not all of them have missing data. Many of them are likely not relevant and could be excluded from the imputation, so that only relevant variables are taken into account.

Given that this is such a small data set, cleaning it was done manually in a spreadsheet editor. Superfluous strings were removed from values, comma decimals in numeric values were replaced with points to work better with R, and other things that would give errors or produce issues in R was fixed. Additionally, the missing values that are empty strings were set to be NA, so that R will understand that the value is actually missing, and tat the blanks are not an intended feature of the data.

This was possible to do manually due to the small size of the data and was done this way in the interest of saving time. Otherwise, it would preferably be done in R, which is well suited to handle cleaning the data with various built in methods, or some other assisted or automated tool. In a larger data set, manually doing this would not be feasible in a short amount of time.

This first data set was mainly used early on for testing purposes, as it was far from ideal to analyse due to the issues mentioned previously. But it was useful to try out the methods, to prepare for a more refined data set later on. Because this was more of a trial run, it was first tested with a lower amount of imputations and iterations than is recommended by more recent literature, but more in line with the older literature on the subject.

This was in part to save on computational time while experimenting and changing things often, and also because the visualizations used to assess the imputation afterwards can quickly become difficult to read and interpret due to the sheer amount of data to display in a single plot. It was therefore useful as an exercise to run MICE with the 'old' recommendations for parameters

```
'data.frame':   27 obs. of  47 variables:
 $ REK                                    : num  1 7 15 20 42 140 142 147 151 152 ...
 $ Sex                                    : Factor w/ 2 levels "1","2": 1 1 2 1 2 1 1 1 1 1 ...
 $ Year.of.birth                          : chr  "1926" "1949" "1971" "1929" ...
 $ Inserted                               : chr  "1999-09-30" "2002-05-27" "1997-12-09" "2003-05-05" ...
 $ Removed                                : chr  "2007-10-30" "2008-10-16" "2009-06-18" "2011-01-27" ...
 $ Years.in.vivo                          : num  8.08 6.39 11.52 7.73 3.32 ...
 $ Linear.wear.rate                       : num  0.244 0.154 0.201 0.272 NA ...
 $ Total.linear.wear..calculated.         : num  1.731 0.982 2.313 2.101 NA ...
 $ Total.linear.wear..real.               : num  581 922 2.2 1709 NA ...
 $ Osteolysis.area                        : num  384 74473 900631 379836 NA ...
 $ Osteolysis.score                       : num  2 2 3 3 NA NA 3 1 2 NA ...
 $ Osteolysis.percentage                  : num  34 28 64 77 NA NA 51 22 37 NA ...
 $ Paprosky.acet                          : num  6 4 5 5 2 5 2 1 2 3 ...
 $ Paprosky.femur                         : num  2 9 3 9 9 9 9 1 2 9 ...
 $ REK.1                                  : num  1 7 15 20 42 140 142 147 151 152 ...
 $ side                                   : Factor w/ 2 levels "1","2": 2 2 1 1 1 1 2 2 1 2 ...
 $ loose.component                        : Factor w/ 3 levels "cup","cup & stem",..: 2 1 2 1 1 2 2 1 2 1 ...
 $ stem                                   : Factor w/ 2 levels "0","1": 2 1 2 1 1 2 2 2 2 1 ...
 $ stem.grade                             : num  0.424 NA 0.402 NA NA ...
 $ stem.wear.percentage                   : num  42.4 NA 40.2 NA NA ...
 $ offset                                 : chr  NA "high" NA "high" ...
 $ offset.value                           : num  38 47 36 42 39 42 41 51 41 42 ...
 $ stem.size                              : chr  "115mm collared" "125mm Prim Hi Offs" "125mm collared" ...
 $ cup                                    : Factor w/ 2 levels "0","1": 2 1 2 2 1 2 2 2 2 2 ...
 $ cup.grade                              : num  31 NA 26 33 NA 21 23 26 24 23 ...
 $ REK.2                                  : num  1 7 15 20 42 140 142 147 151 152 ...
 $ tissue                                 : num  8 4 3 3 3 0 4 2 3 3 ...
 $ macrophage                             : num  2.5 2.5 2 2 2 NA NA 1.5 0.5 2 ...
 $ giant.cell                             : num  2 1.5 3 2 3 NA NA 1 0 3 ...
 $ lymphocyte                             : num  1 1 1 1 1 NA NA 1.5 0 1 ...
 $ poly.count                             : num  3 0 4.17 5.83 11 ...
 $ darkfield.count                        : num  990 54.2 1250.2 122.5 134 ...
 $ poly.count.from.all.slides.            : num  3 0 2.42 3.08 11 ...
 $ darkfield.count.from.all.slides.       : num  990 54.2 701.7 74.8 134 ...
 $ darkfield.density.from.all.slides..mm².: num  43103 2358 30550 3255 5834 ...
 $ ED.median.um.                          : num  0.373 0.541 0.392 0.852 0.426 ...
 $ REK.3                                  : num  1 7 15 20 42 140 142 147 151 152 ...
 $ blood                                  : Factor w/ 2 levels "0","1": 2 2 2 1 2 1 1 1 2 2 ...
 $ Al                                     : num  NA NA NA NA NA NA NA NA NA NA ...
 $ Ti                                     : num  1.52 4.86 0.21 NA 1.3 NA NA NA 0.92 0.95 ...
 $ V                                      : num  0.01 0.07 0.03 NA 0.01 NA NA NA NA 0.11 ...
 $ Cr                                     : num  1.05 1.52 1.26 NA 0.36 NA NA NA 1.8 0.82 ...
 $ Mn                                     : num  6.52 6.8 5.01 NA 11.63 ...
 $ Co                                     : num  6.02 0.42 0.78 NA 0.33 NA NA NA 2.5 0.34 ...
 $ Ni                                     : num  1.22 1.14 0.99 NA 0.31 NA NA NA NA 0.74 ...
 $ Zr                                     : num  2.08 0.6 0.02 NA 0.08 NA NA NA 1.08 0.5 ...
 $ Mo                                     : num  1.3 0.78 0.44 NA 0.86 NA NA NA 0.65 0.7 ...
```

Figure 4.5: Data types of variables in the first data set.

to make assessing it easier, and having something to compare with the newer and more updated data set.

When starting the imputation for the first data set, the cleaned up version of the data was used to avoid the issues mentioned previously. It was imported as a standard CSV file, and the data type for each column was set manually as an input parameter to ensure that all the values were correctly handled. There were a lot of variables to account for, as seen in Figure 4.5. Then, the process of imputation described in Section 4.2.2 was followed.

The m parameter, the number of imputations, was left at the default value of 5, which is not very high, but should be enough to get results [van Buuren and Groothuis-Oudshoorn, 2010]. The maxit parameter for number of iterations

was set to 30, which is around the recommended number as well, though more is better for accuracy.

The method chosen for this imputation was Classification And Regression Trees, or CART. This is able to handle both numeric and categorical variables and still produce results, while being simpler and less computationally heavy than some other more specialized methods. With this many different variables and data types, it runs the risk of MICE not working with more complex methods so something like CART which can handle the complexity can be preferable.

After applying the imputation the methods used for each variable was checked to see if it performed correctly. The newly imputed data is then ready to be analysed. For that, one of the imputed data sets was selected using the MICE 'complete' command, which takes an imputed data set and makes it into a data frame which R and R packages can use. Most visualization work well with data frames, as is it one of the standard data structures in R.

### 4.2.4 Imputation of the second data set

The second data set was better to work with for many reasons as described in Section 4.1. With the number of variables significantly decreased, the number of observations almost doubled and the data cleaned up and prepared from the start, there was a lot less work needed before running MICE. As with the first data set, the CSV file was imported into R while making sure the data types were correctly handled. This was again set manually in the imputation parameters to avoid issues. As can be seen in Figure 4.6, this data consists of fifteen numeric variables and four categorical variables, each of them with two possible values.

When running the imputation the second time around with the new data, some more parameters were used to try to get better results. Firstly, the 'id' variable was excluded from the imputation so that it would have no effect. It is just an identifying number the case has been given in the data and is

```
'data.frame':   49 obs. of  19 variables:
 $ id           : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Case         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ cupLoose     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ stemLoose    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ sex          : Factor w/ 2 levels "1","2": 2 1 2 1 2 1 1 2 2 2 ...
 $ years.in.vivo: num  10.19 10.49 10.02 7.11 7.58 ...
 $ Cr           : num  1.95 0.67 0.35 0.37 0.05 0.77 0.39 0.4 0.63 0.1 ...
 $ Co           : num  0.16 0.96 0.54 0.62 0.05 0.25 1.29 1.02 0.2 0.14 ...
 $ Zr           : num  0.4 0.15 0.51 1.23 0.03 0.56 0.6 3.68 0.43 0.14 ...
 $ Ni           : num  1.14 0.1 0.17 0.08 0.2 0.37 0.51 0.33 0.25 0.16 ...
 $ Mb           : num  0.89 0.46 0.49 0.8 0.44 0.9 1.27 0.93 0.62 0.48 ...
 $ linWear      : num  1.75 2.29 0.05 1.02 0.87 2.73 0.59 1.71 1.08 1.88 ...
 $ linWearRate  : num  0.17 0.22 0 0.14 0.11 0.26 0.05 0.17 0.11 0.19 ...
 $ volWear      : num  1077.3 1412.2 29.2 628.7 535 ...
 $ volWearRate  : num  105.69 134.57 2.91 88.45 70.63 ...
 $ Inc          : num  45.7 45.5 57.7 54.2 49.3 ...
 $ Ant          : num  7.25 12.88 6.09 22.74 3.49 ...
 $ CupX         : num  32.6 38.3 35.3 19.8 26.8 ...
 $ CupY         : num  9.67 19.98 12.72 35.74 9.8 ...
```

Figure 4.6: Data types of variables in the second data set.

| **Variable** | **Method** |
|---|---|
| id | - |
| Case | - |
| cupLoose | - |
| stemLoose | - |
| sex | logreg |
| years.in.vivo | - |
| Cr | pmm |
| Co | pmm |
| Zr | pmm |
| Ni | pmm |
| Mb | pmm |
| linWear | pmm |
| linWearRate | pmm |
| volWear | pmm |
| volWearRate | pmm |
| Inc | pmm |
| Ant | pmm |
| CupX | pmm |
| CupY | pmm |

Table 4.3: Method used for imputation of each variable.

not relevant at all for the analysis itself and should therefore not be used by MICE in the imputation.

Secondly, the 'm' parameter, the number of imputations, was set to 16 based on the percentage of missing values in the data set. While early documentation recommended to put the number of necessary imputations for a good result around 5 imputations, this was in large part due to efficiency in a time where computing power was more of a concern. Now, following the rule of thumb suggested in more recent literature, the number of imputations was set around the percentage of missing values in the data [White et al., 2011].

The 'maxit' parameter was also changed to 50 this time, for a total of 50 iterations. While a lower number could be sufficient, there is no real reason not to go for a higher number for extra accuracy [van Buuren and Groothuis-Oudshoorn, 2010]. Both the higher computing power of modern computers and the small size of the data set makes the impact on efficiency and speed negligible.

The 'method' parameter was left blank this time around, meaning MICE will use the default method, which is to choose what it considers the best method for each variable. In this case, that is predictive mean matching, or 'pmm', for the numeric values and 'logreg' or logistic regression for the categorical variables. Additionally, the 'seed' parameter was used to allow for reproducibility of the results.

After running MICE, the result was again a MIDS object containing all of the imputed datasets. Due to the number of data sets, not all of the visualizations are as easy to see as there are many overlaying lines simply due to the number of data point to be drawn. Additionally, the small size of the data and relatively few variables to be imputed can make some graphs seem to diverge more than would be the case otherwise.

### 4.2.5 Using imputed data

There are several ways to make use of the data once the imputation has been completed. The MICE package offers many tools for working with the imputed data. As the end result is a MICE-specific format, some level of post processing will usually be needed. Perhaps the easiest method is to use the built in 'complete' function. It takes one of the imputed data sets and saves it to a data frame which is usable in the standard R application; essentially filling in the missing values from the original data set. This is also the method used for further work in this project.

You can also use the 'complete' function with the 'long' parameter to achieve a different result. Instead of filling in the missing values of the dataset, but not changing anything else, it creates a larger data set by using all of the imputed data and stacking them on top of each other. Creating a very long data set that is made up of one data set per imputation. There could be some cases where this would be desirable, but much of the data would be similar or even identical and there was no need for it in this case. It would likely no produce any better results, but might rather be a source of more noise.

Another way of using the data is by so called 'pooling' of the imputed data sets. The pooling method in MICE allows you to run analysis separately on each of the imputed data sets and pool the resulting estimates for further analysis. While this might produce similar results to only using one data set, it can increase the likeliness of achieving a good result, and might indicate if there are issues with the data as you are using more of it. This can be a powerful tool for statistical analysis, but is somewhat outside the scope of this thesis. Nevertheless, it is something that could be of use in future work.

## 4.3 Planning and prototyping

### 4.3.1 Establishing requirements

For this project, talks with expert users in the early stages of the project provided the groundwork for the requirements and a better understanding of what the system should provide in terms of functionality and usability.

Requirements describe what a product should do or how it should perform. It should also be unambiguous to make it clear what the requirements are as to avoid misinterpretation. This also helps to see when a requirement has been fulfilled, if there is less room for interpretation. In software engineering, there are mainly two types of requirements that are used. These are functional requirements and non-functional requirements. Functional requirements describe what the system should do, whale non-functional requirements describe constraints for the system and its development [Preece et al., 2011].

### 4.3.2 Functional requirements

Functional requirements describe what the basic application should be able to do and should allow the user to do in it, while also describing the scope of the system. These are requirements for a bare-bones version of the prototype application, but allows for further expansion in the future.

1. The system should allow the user to input data.
2. The system must allow the user to perform analysis on data using predefined methods.
3. The system must allow the user to customize the parameters for the analysis.
4. The user must be able to validate the results of the analysis in the system to some extent.

5. The system should provide the user with sufficient instructions/documentation for use.

### 4.3.3 Non-functional requirements

Non-functional requirements can be harder to define as they might be less concrete than the specifics of functionality. They can be related to the look and feel of the system, performance considerations, maintainability etc.

1. The system should be easy to use
2. The system should be fast and responsive
3. The system must run in a web browser

### 4.3.4 Design

For the initial planning of the design, adobes design and user experience/user interface tool Adobe XD was used. It provides a lightweight and easy to use tool for creating designs quickly and easily, perfect for early wireframes and low fidelity prototypes. These are simple designs, but thanks to the Adobe XD software, they are in fact interactive, allowing user input and testing from the start.

The first design for the application was a very basic version, a low fidelity prototype. It was as simple as you can get, just to have a starting point to build from. This design which can be seen in Figure 4.7 featured a single method of analysis, linear regression in this example. The dependent variable was already selected from the start and checkboxes added for the user to select the independent variable.

While this is definitely an easy way to do it, it doesn't really work for several reasons. It is extremely limited in that it only allows one option, the variables take up an unnecessary amount of space on the screen, everything is

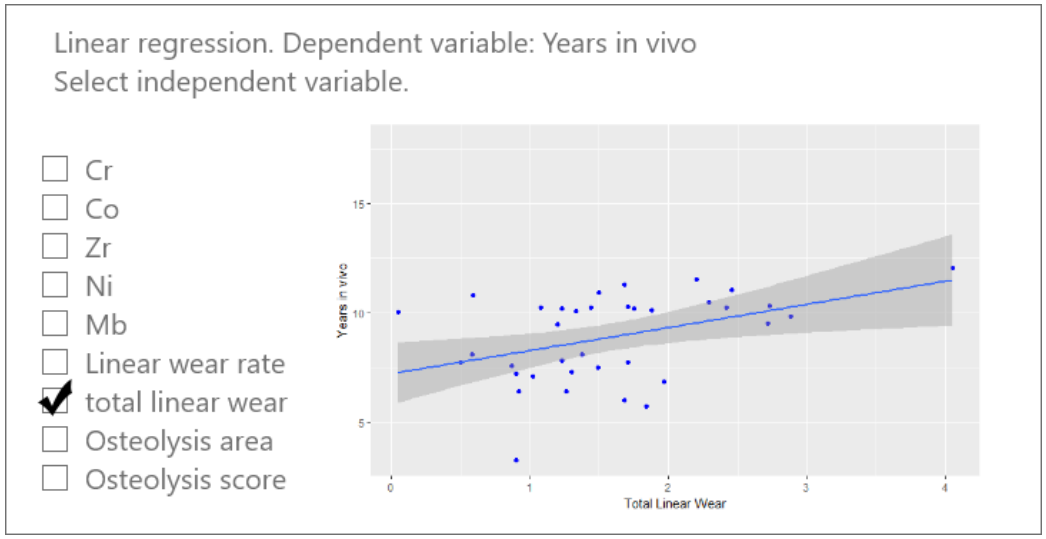



Figure 4.7: Early low fidelity prototype.

hardcoded and it not suitable for further expansion or development. There are better ways of doing it, but it worked as a first design and test run.

The second design presents more options for different methods of analysis as can be seen in the example in Figure 4.8. The first tab would allow the user to upload a file with data to analyse. The selected tab shows how you could do a simple linear regression. Simple drop down boxes were added for choosing the variables for the regression and displaying the plot directly in the window.

At the bottom of the window are two boxes displaying the R squared value and Standard error of the regression as an example of additional values that could be shown. Both the plot and the values would change as you chose new variables. This is a minimalistic way of doing it that can more realistically end up as a functional design that can also be expanded upon later.

More tabs can be added to the top of the window to support more methods or options that are quick and easy to switch between, or could contain additional information that could be relevant for the analysis. Alternative methods of handling the general layout can be changed as development progresses as well. The code written in R can be reused for different layouts and ways

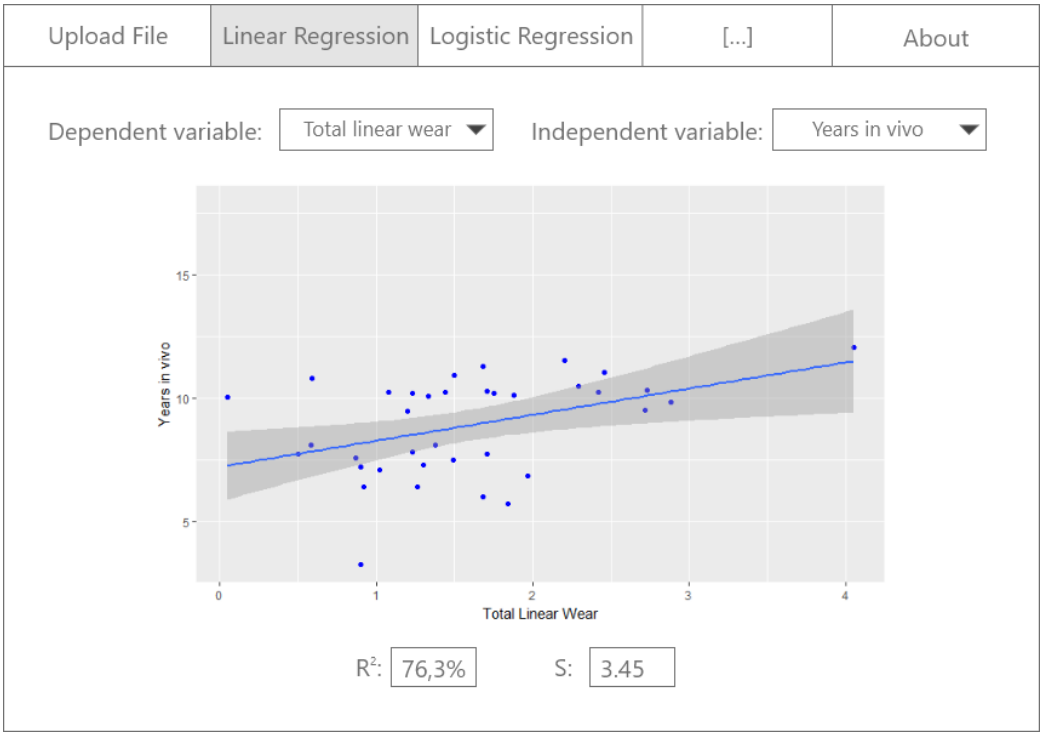



Figure 4.8: Improved low fidelity prototype.

of using the application, as the underlying code usually doesn't need to be modified much to change the design.

## 4.4 Prototype

The prototype was developed using the Shiny framework for R, introduces in Section 3.3.3. Shiny provides a set of standard components with default behaviour. This makes it easy to put something together without having to think too much about design. This has both advantages and disadvantages, expanded upon in the discussion in Section 6.1.3. The development of the prototype was therefore more focused on functionality than design. Modifying the components of Shiny is possible, but can be time consuming and should be done with a concrete plan and direction for the design. This is somewhat outside of the scope for this project.

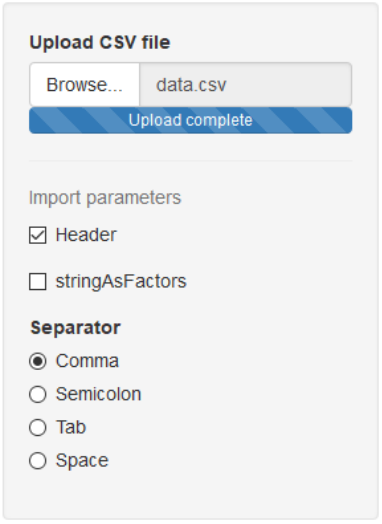
The prototype itself was developed more through continuous development and implementation than concrete iterations. There are no explicit versions to reference, as components can be easily added and removed or re-purposed in the same application. If you need a new feature or page, and can be added to a new tab or drop down menu which changes to the new layout or functionality.

The development started with what is essentially a blank canvas that can be filled with components that are tied to functionality made in R. Once a feature was completed, you can simply move to the next tab and start on the next feature there. That is one of the advantages of using this framework. You get the base functionality and navigation for 'free', and tie the components to the custom functionality you code in R.

The basic Shiny application is a simple window that you put the components in. This can be a tab panel on the top, a sidebar, various drop down menus and check boxes, text fields etc. The first thing that was implemented was the tabs in the wireframe, for changing between different windows. Clicking on one of the tabs changes the content of the rest of the window. There are many ways to implement multiple views like this, but this is easy and simple and works for this purpose. Should there be a need for more, you can make drop down menus with multiple alternatives for each tab as well.

Once basic navigation was done, the next feature was the ability to input data, as seen in Figure 4.9. This view allows the user to select a CSV file to upload their data. Once imported, the data fills the window. On a web browser this will scale to the size of the window and allows the user to scroll through it horizontally and vertically. The 'Header' parameter should be checked if the columns are named in the .csv file to make R interpret it as the column names and not values. The 'stringsAsFactors' determines if strings should be converted to factors by R. Whether or not it is necessary will depend on the data set. In this specific case it is not needed as all the variables are numeric.

Figure 4.10 shows a full view of the application as it look when running it




| id | Case | cupLoose | stemLoose | sex | years.in.vivo |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 2 | 10.19 |
| 2 | 0 | 0 | 0 | 1 | 10.49 |
| 3 | 0 | 0 | 0 | 2 | 10.02 |
| 4 | 0 | 0 | 0 | 1 | 7.11 |
| 5 | 0 | 0 | 0 | 2 | 7.58 |
| 6 | 0 | 0 | 0 | 1 | 10.33 |
| 7 | 0 | 0 | 0 | 1 | 10.80 |
| 8 | 0 | 0 | 0 | 2 | 10.28 |
| 9 | 0 | 0 | 0 | 2 | 10.26 |
| 10 | 0 | 0 | 0 | 2 | 10.11 |
| 11 | 0 | 0 | 0 | 2 | 10.21 |
| 12 | 0 | 0 | 0 | 1 | 10.08 |

Figure 4.9: Data imported from .csv file.

directly in RStudio. The top menu bar allows for switching between the different views and also includes an about section for additional information and instructions. This can easily be expanded to include more tabs and other menu items.

It also shows the variable selection where you have a drop down menu for the dependent variable and independent variable. Both lists contain the variables in the data set so you can choose any combination. To the right of it there is a scatter plot of the data with a regression line fitted to the model. Below this is a text area that shows information about the regression using the summary function in R. This provides plentiful information about the regression, but does not do any interpretation on its own. Thus the user will have to have some experience or explanation of it to be able to benefit from it.

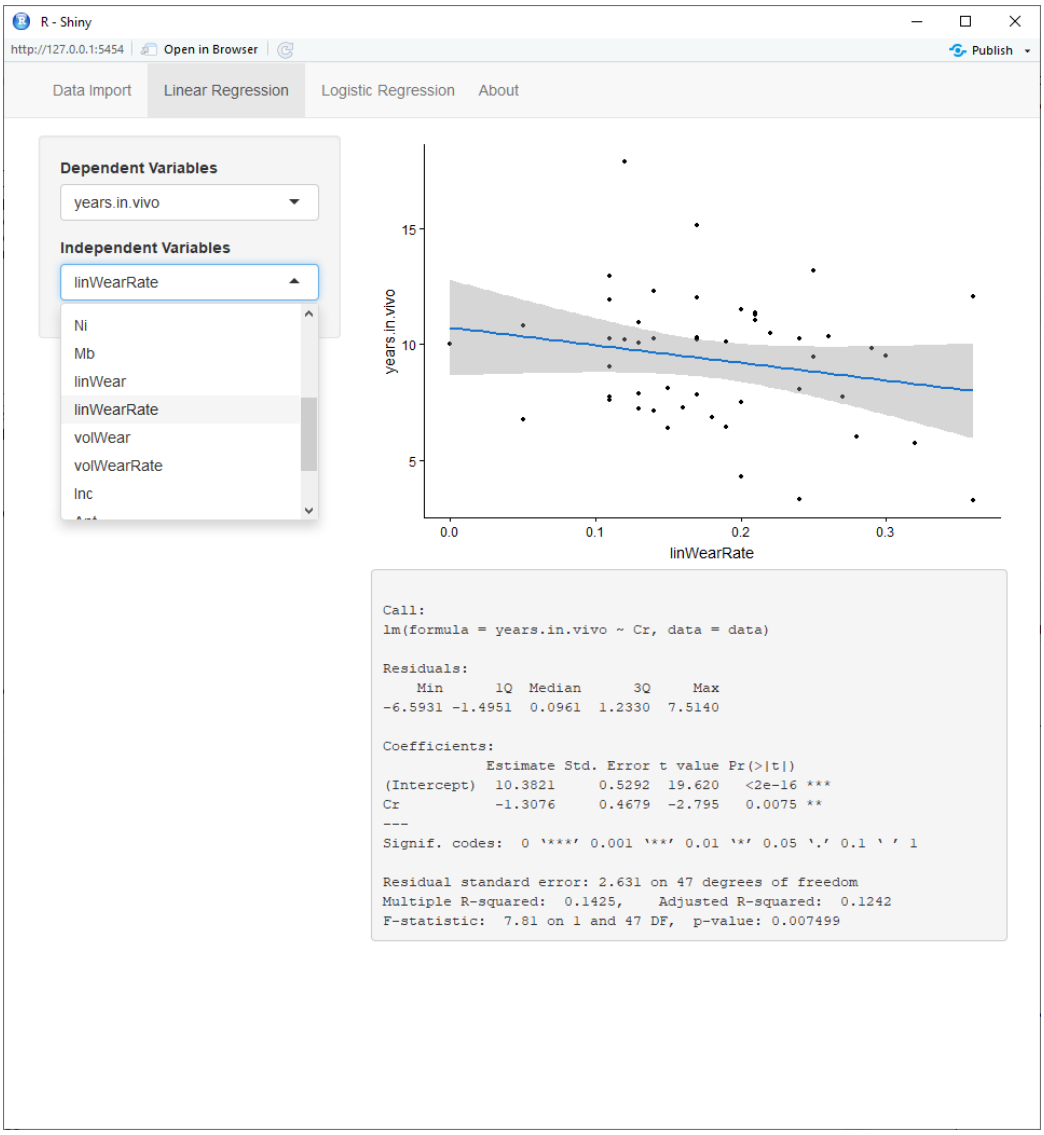



Figure 4.10: Application interface.

## 4.5 Linear Regression

The linear regression was implement using the linear model function 'lm' in R. It lets you enter the independent and dependent variables and the data set to be used as parameters in the function. Using it with Shiny allows for the input parameters to be replaced with reactive values. These are values that can be changed based on user input, which is what allows Shiny to send the new parameters to R and update the input in real time to produce the new results.

For the linear regression in the application, the data set was set to be uploaded by the user, and the variable lists were filled based on the variables in the data. The user can then select values for the independent and dependent variables, which will update the scatter plot with regression line in real time as well as the model summary provided in the text area below the plot. This is a basic, but fully functioning implementation of a simple linear regression. This was also used to examine the effects of the multiple imputation on the data. The results of this are detailed in Section 5.5.1.

## 4.6 Prediction

Predictions were run on the data in R to further evaluate the usefulness of the data for prediction, as well as the effects of the imputation on the predictive power of the data. Before running the prediction, the rows in the data were randomized to distribute the case and control data randomly, as these were ordered in the original data. This was done to get both case and control data when spilling the data set, and avoid skewing the prediction. The data was then split into a 'train' set for training the model and a 'test' set for testing it afterwards with an approximate 80:20 split.

The predict function in R takes the model to be used for prediction and the data to test it on as parameters. In this case, a multiple linear regression model was used with all the variables to predict the years.in.vivo variable,

how long the implant lasted before needing to be explanted. The result of this is a data object with predicted values that we can then test for accuracy using various methods.

One quick and easy way to get an impression of the result of the prediction is to compare the predicted values with the actual values. This returns a data frame with the two columns side by side allowing for a 'manual' inspection of the results. Another method is to check the correlation accuracy between the actual and predicted values. This checks whether or not the values have similar movement, that the predicted values move in the same direction as the actual values, and the other way around. A higher value is better.

Minimum-maximum accuracy is another measure we can use to check the accuracy of the prediction. This function takes the minimum predicted value and the maximum predicted value, calculated the average of them and does the same with their actual counterparts before comparing them. In this case, a higher value is better and indicates more accurate predictions. With a value of 1, or 100%, it would be as close to a prefect prediction as one could get.

The final validation method that was used is Mean Absolute Percentage Error, or MAPE. This is a measure of the accuracy of a prediction. It gives a percentage value for the accuracy, where a lower value is better. These methods were applied on each of the data sets, both the imputed and original data. The results of the analysis can be found in Section 5.5.2.

# Chapter 5

# Results

This chapter presents the results that have been produced in this project. The main subjects are the Multiple imputation the Application developed based on that data, and the data mining utilizing and comparing the data. The Multiple Imputation is presented in two parts: A brief overview of testing out methods on the first data set and the issues that arose, and a second round of using what was learned from that on the second data set to improve upon the results.

## 5.1 Results of multiple imputation

This section details the results of the multiple imputation for both data sets. Using R, the imputed data can be used in plots to assess the success of the imputation. In MICE there are specialized methods using the previously mentioned MIDS object that is the result of a multiple imputation, that can plot the original data and the imputed data together to get an idea of how the data sets compare. This can potentially show whether or not the new data is similar to the original and the resulting values are plausible.

There are several types of plots that can be used to examine this for both of the data sets. Because of the gap in time between working on the data sets, as

well as the difference in quality between them, the second has received more attention and thorough analysis using more methods than the first.

## 5.2 First data set

The first data set was mostly successful when running the imputation, but with some caveats. While most variables completed without any severe issues, some of them did not. MICE simply couldn't make enough sense of the existing data to get something worthwhile out of it, leaving those variables out completely to be able to finish the rest. So while most of the missing values were filled with new ones, some were ignored and there remained a big hole in the data that was of no use, but rather detrimental to the success of the algorithm attempting to fill in the blanks.

The data also produced some errors when plotting the data for analysis afterwards due to this, and was generally not easy to work with in this way. It did end up producing some results though and these are shown below. These results will also provide a contrast to the second and more refined data set, showing the improvements over the first and refinement of the methods.

### 5.2.1 Density plot

One of the first plots one might look at after imputation is a density plot. This shows the distribution of values in the data much like a histogram, but without the limitation of having it distributed into bins. The number of bins in a histogram can greatly affect the look of the end result and lead to misinterpretations. In the case of MICE, the density plot function uses the original data as one plot, and only the imputed data in the other plots. This is done to more easily help you see the difference between the new data without having the original data as noise.

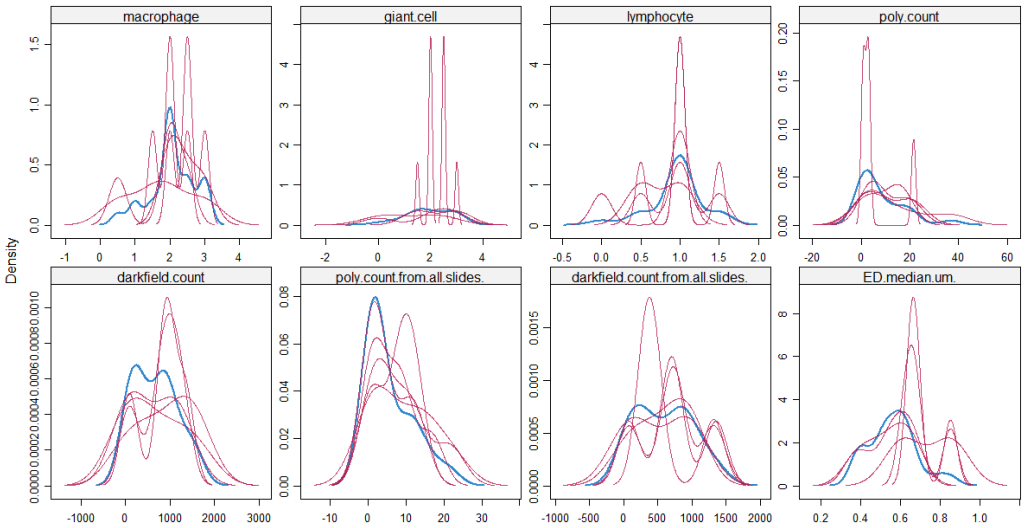



Figure 5.1: Density plots of some of the imputed variables in the first data set. The original data is show in blue, and the imputed data shown in pink.

But this also means that the plots for the imputed data can vary greatly depending on the number of missing values. With fewer values missing and subsequently imputed, the plots can look very different with extreme outliers, even thought they are well within the range of the original data. This can be seen clearly in Figure 5.1, which shows density plots for some of the imputed variables in the first data set.

In Figure 5.1 we can see the original data in blue and the imputed data in pink. We would expect the imputed data to look quite similar to the original if the imputations was good. But as explained previously, due to how MICE makes density plots using only imputed values, at first glance it looks like the imputation has resulted in very different densities compared to the original data for some of the variables.

But if we look more closely, it does somewhat follow the shape of the original, even if there are several spikes and dips outside what might be the expected range. This effect is exaggerated by the small number of variables in the data set. Because there are so few values for the variables, there are also few missing values to replace, even though the percentage of missing values

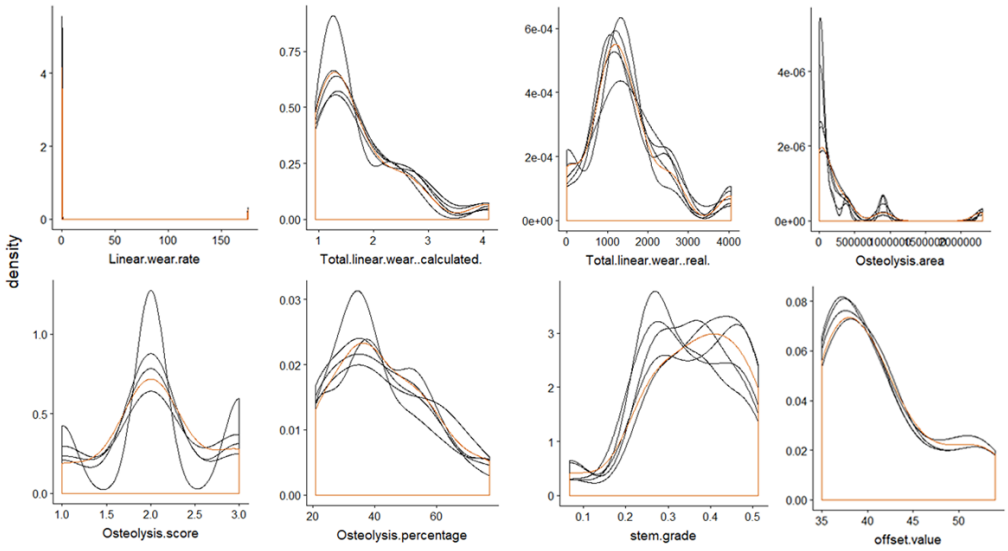



Figure 5.2: More density plots of some of the imputed variables in the first data set. Including all values, not just imputed values. The original data is show in orange, and the imputed data shown in black.

is high. This means that each value has a significant impact on the plot, causing the seemingly large discrepancies.With this in mind, the plots seem to be somewhat similar in shape for most variables and not necessarily not as bad as the first impression makes it out to be.

### 5.2.2 Modified density plot

In order to properly assess this another function was used to make a density plot that includes all the values in the imputed data sets, not just the new imputed values. An example of this plot can be seen in Figure 5.2, with a selection of variables from the first data set.

In this density plot we can see much more clearly how the imputed data more closely aligns with the original data when taking all values into account. There is still some variation, but this is to be expected given the small amount of observations we have to work with. Had the data set consisted of hundreds or thousands of values with the same amount missing, there would likely be

little to no discernible difference between the plots. So while this shows that the imputation is on the right track, there as still room for improvement

This first imputations on the first data set were useful for testing multiple imputation with MICE and its limitations, but not particularly suited for further analysis. The data was already of limited use as explained in Section 4.1, and while the imputation helped fill in some of the missing values, there were also some values that is could not do anything about. The total number of variables combined with the small number of observations and the amount of data missing from them made it difficult to get a good result. A larger data set would help, however, for both this thesis and a real world scenario one has to work with what one has.

## 5.3 Second data set

The second data set, as described in Section 4.1, was much better to work with, and produced better results. For one, the MICE finished the imputation with no errors, and all the variables with missing values were able to be imputed. With the complexity of the data being significantly lower, less data missing and more data in general, the results are more encouraging. This also showed when trying to plot the data, as all the plots were generated with no errors, unlike with the first data set. This allowed for a more thorough analysis of the imputation as well. Because this imputation was run with different parameters for a better result, there are more plots as the number of imputations increases. This results in many more plots per chart and might be more difficult to tell apart at times.

### 5.3.1 Density Plots

As before, the density plot is a good starting point. Figure 5.3 shows a density plot for each of the imputed variables in the second data set. At first it is evident that there are many more plots this time, making it hard to

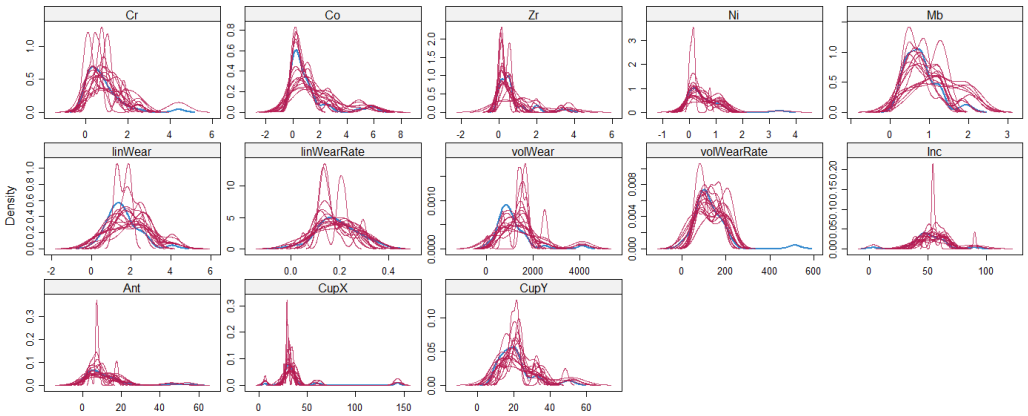



Figure 5.3: Density plots for the imputed data for each variable in the second data set. The original data in blue, imputed data in pink.

distinguish the individual line from each other. But as before, we are looking for similar shapes in the plots of the imputed data to the original, which does largely seem to be the case. But there are some significant differences here as well. Due to the number of imputations run, some plots were generated as a larger version to look more closely at.

Figure 5.4 shows the density plot the the 'Mb' variable. Here we can see that while the plots do somewhat follow the general shape of the original data, there is quite a bit of variation which doesn't leave too good an impression initially. In Figure 5.5 we can see the density plot for the 'Zr' variable which, while still deviating from the original data, follows it more closely except for a couple of extreme outliers. In fact, most of the density plots look to be similar to the original even if there is also a lot of variation, which does indicate that the imputation what somewhat successful.

### 5.3.2 Modified density plots

The second density plots utilizing all of the data were also made for the second data set, to get a better look at the big picture. Figure 5.6 and Figure 5.7 show the same 'Mb' and 'Zr' variables, but using all of the data for plotting

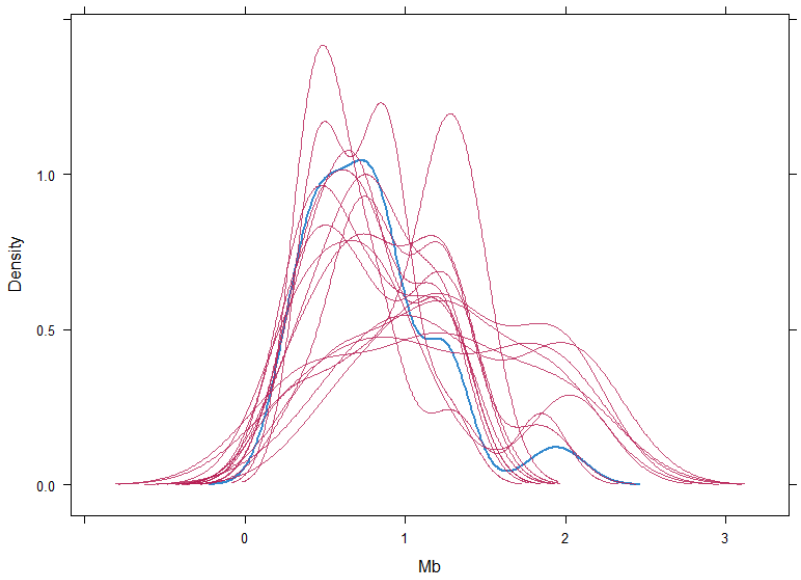



Figure 5.4: Density plot for the Mb variable in the second data set.

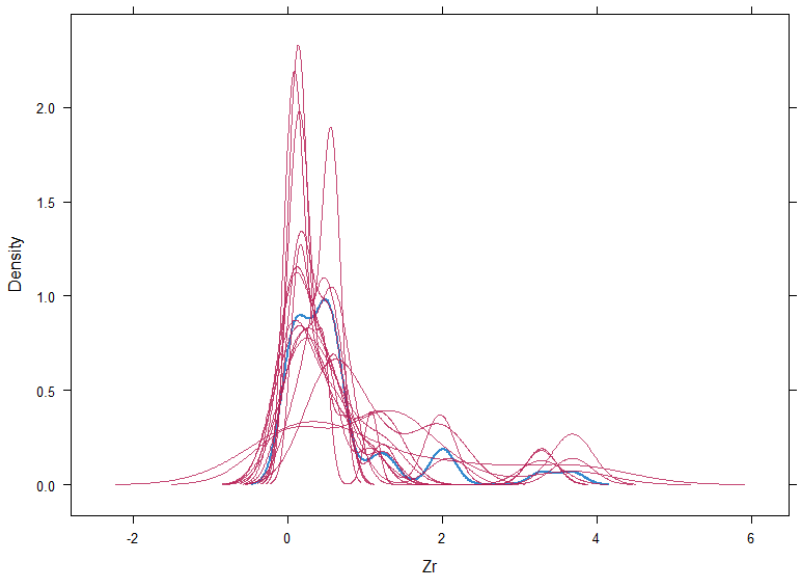



Figure 5.5: Density plot for the Zr variable in the second data set.

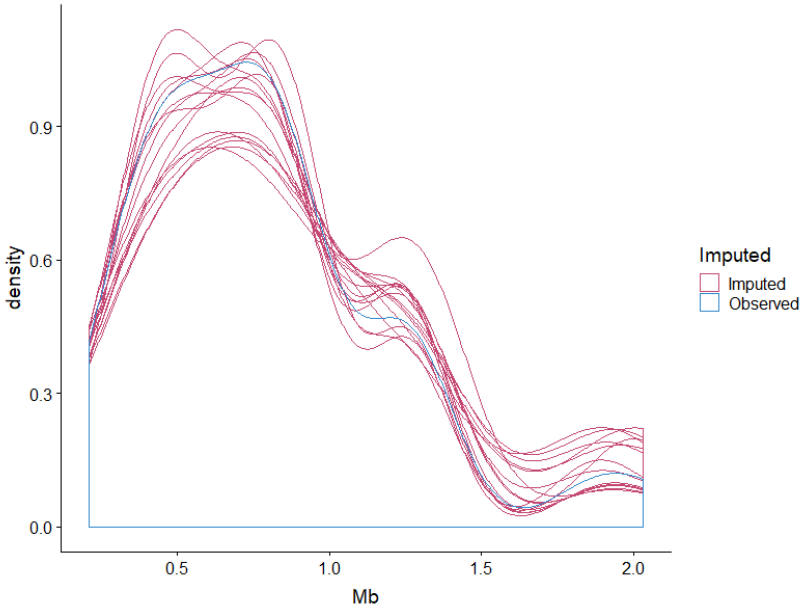



Figure 5.6: Density plot for the Mb variable in the second data set, using all the available data.

instead of just the imputed data. This gives us a better overview of the entire data set instead of just the new imputed values. The 'Mb' variable looks a lot better now, mostly following the original data. And the 'Zr' variable makes a significant improvement as well, closely following the plot of the original data apart from a couple of outliers. This appears consistent with the rest of the variable as well, giving a better impression of the overall quality of the imputation. In fact, the 'Zr' variable appears to have the biggest single outliers, while some like 'Mb' have more variation overall. They are mostly consistent with the original data.

### 5.3.3 XYPlot

Another plot that can be useful the what in MICE is called an XYPlot. This is a scatter plot that like the density plot separates the imputed and observed data to allow for comparison. This plots each imputation in a separate chart and arranges them in a grid. This also uses blue for the observed data

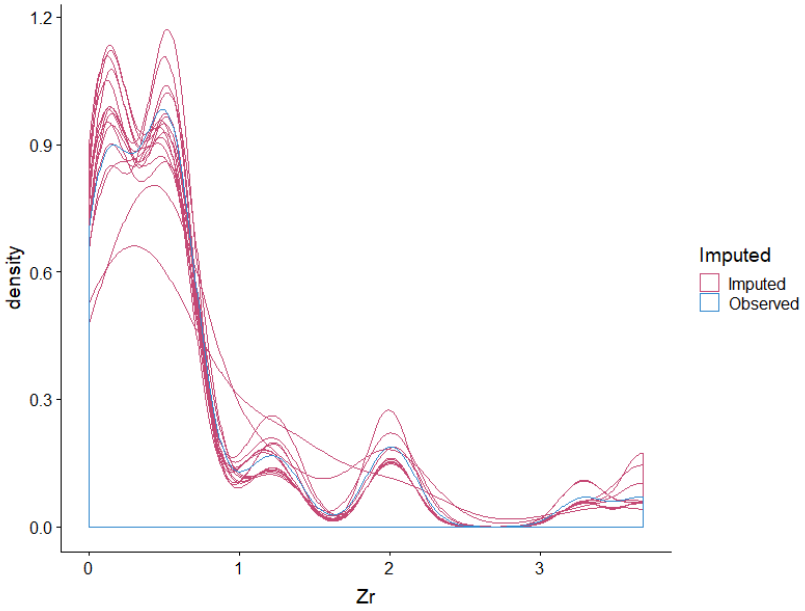



Figure 5.7: Density plot for the Zr variable in the second data set, using all the available data.

and pink for the imputed data, the standard colors in the MICE package. The first graph in the grid is simply the original data plotted. And each subsequent graph uses an imputed data set to plot along with it, making it a total of 16, one per imputation, and the 17th for the original data.

We are looking for the same thing here as with the density plots. In Figure 5.8 the 'volWear' and 'volWearRate' variables have been plotted against each other. We can see that they have mostly similar shapes which is what we want. The imputed values are within a plausible range of values from the original. This is true for all of the plots of all the variables here as well. Though there are outliers, some are to be expected, especially given the small data set. Most values are still well within a plausible range, and the data set had outliers to begin with which can also affect the result.

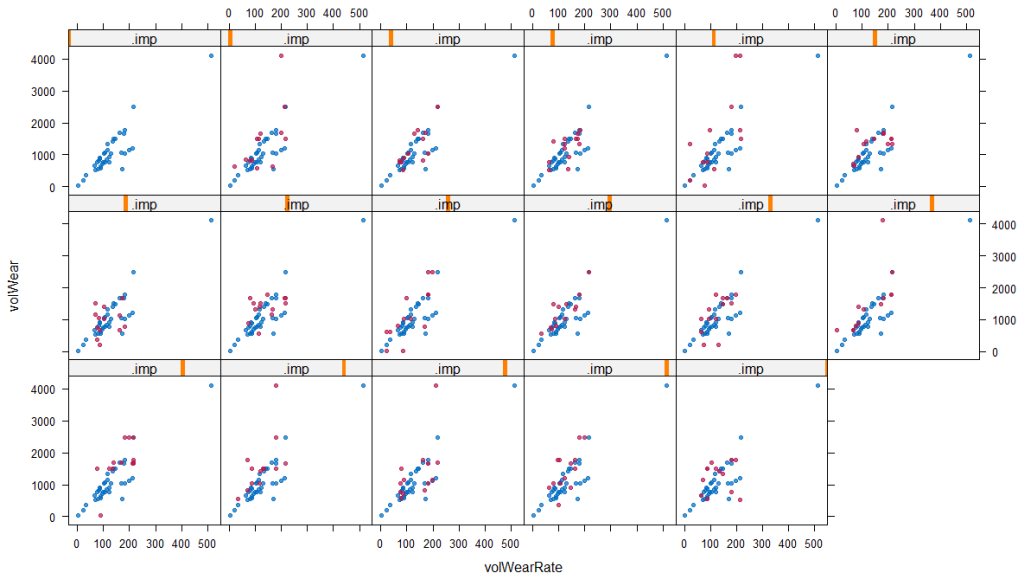



Figure 5.8: 'XYPlot' of the imputation of the volWear and volWearRate variables.

### 5.3.4 Stripplot

Yet another plot that can be used in much the same way as an XYPlot is the Stripplot. These are also scatter plots, but the are one dimensional, and only use one axis. This allows R to distribute multiple axis in the plot; one for each data set. So each stripplot represents a single variable in each of the 16 imputed data sets plus the original, making it a total of 17. As with the other plots, the original data is in blue and the imputed data is in pink.

The first column with all blue dots is the original data only. Figure 5.9 shows a stripplot of the linWear variable, showing the same trend of closely aligning with the original data, which is what we want. The same goes for the other variables as well, indicating like the other plots that the imputation has been at least somewhat successful. In Figure 5.10 you can see a smaller version of all the stripplots for each of the variables. Note that the two first variables had no missing values, and therefore shows up in all blue as there was no missing data to fill with the imputation.

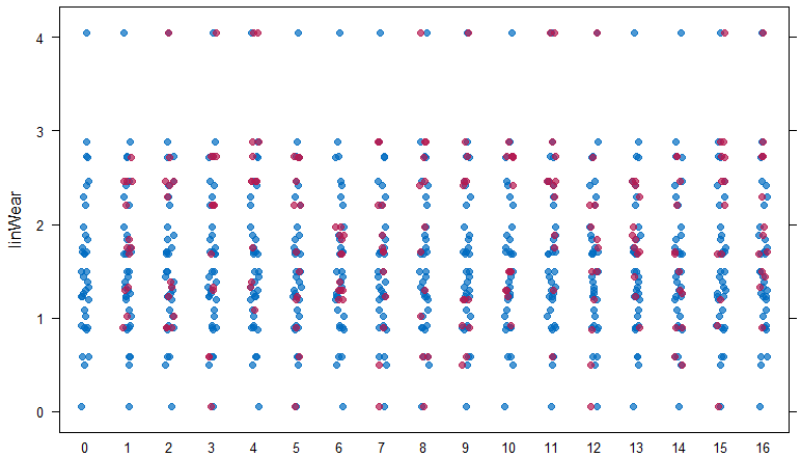



Figure 5.9: Stripplot for the linWear variable in the second data set.

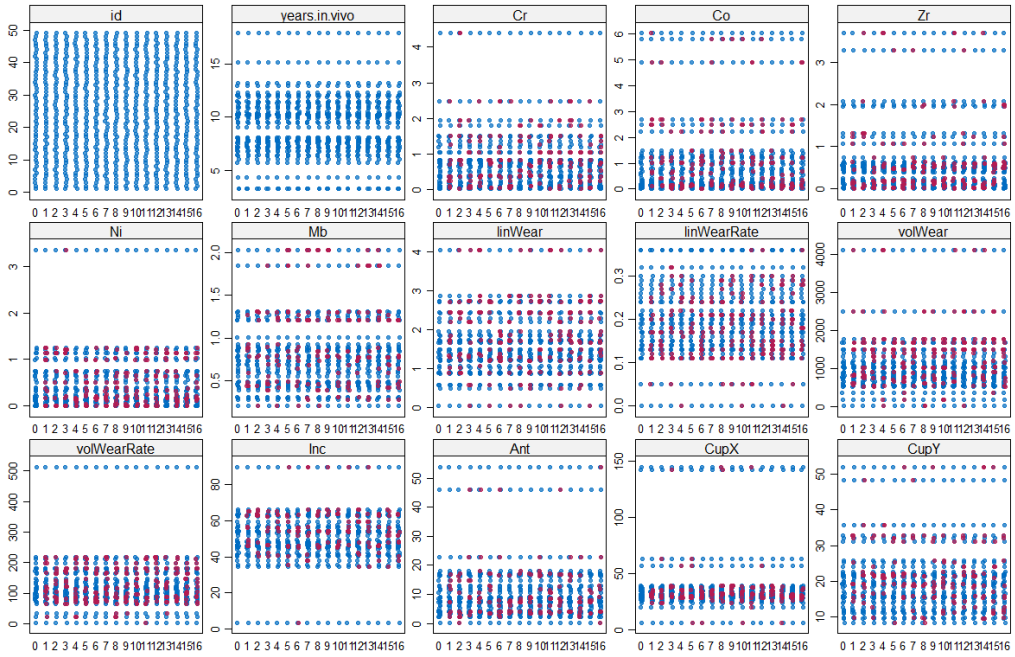



Figure 5.10: Stripplots for all variables in the second data set.

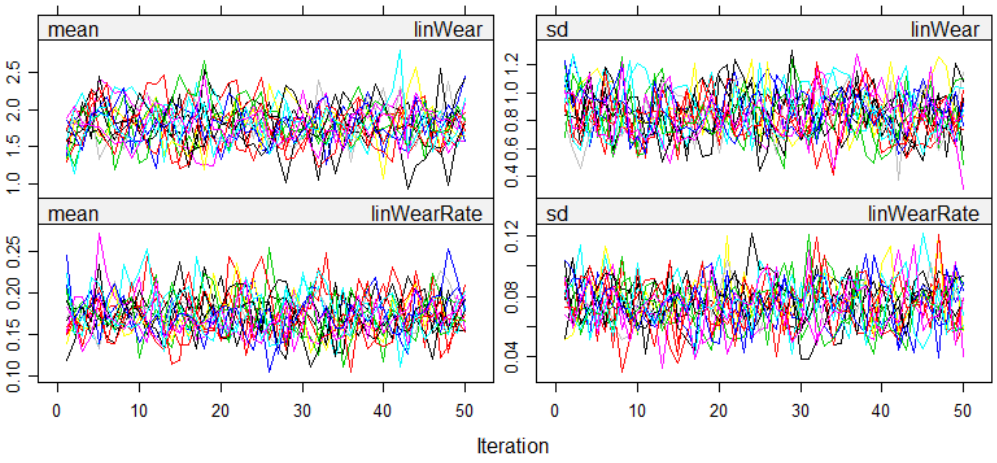



Figure 5.11: Convergence plots for the linWear and linWearRate variables.

### 5.3.5 Convergence

Convergence is another way of visually assessing the results of the imputation. A good result when looking for convergence is that there are no real trends in the plot, and the data mingles well. Even if there is a trend initially, this can quickly change as the algorithm iterates, and begin mingling quickly. A healthy convergence is reached when the plots freely intermingle with each other [van Buuren and Groothuis-Oudshoorn, 2010].

In this regard, the imputation seems to have done quite well also. Figure 5.11 shows the convergence plots for the linWear and linWearRate variables. It can appear quite chaotic and random, which in this case is exactly what we want. Using many imputations makes the plot quite a bit more messy, but will also ensure that there are no trends forming further along the graph.

The general trend for the variables seems to be an overall lack of trends, which is desirable. The plots seem to freely intermingle for the different variables, indicating a healthy convergence.

## 5.4 Results of prototype

The first view of the prototype application gives you the option to upload the data to be analysed. It is a simple but functional screen that gives you the bare minimum of functionality needed. This worked just as intended, much thanks to the file browser functionality in Shiny. This opens up a standard file browser allowing you to browse to a CSV file on your computer to upload. It can also be useful to be able to view the data after uploading, which is the default behaviour.

Switching over to the Linear Regression view will allow you to choose the dependent and independent variable to the regression from the data set. It will then update the regression plot live as you change the variables thanks to the reactive components of Shiny. This also works very well and is fast and easy to use. The summary of the model below the plot can be used to further analyse the model.

Overall, this is a fully functional albeit a very simple application that allows you to upload data an analyse it using the pre defined methods. The possibilities are many should you put in the necessary work, but this is a good start and proof of concept. How the technologies performed and the overall impression will be further talked about in the discussion section.

## 5.5 Results of data mining

### 5.5.1 Linear regression

Using the prototype to run linear regression for all the variables, a table of results was compiled using R squared and p-values in an attempt to see if the model was good and the data could be used for predictions. Three different measures were used to attempt an evaluation of the model. The values for the imputed data set and the non-imputed data set can be seen in Table 5.1 and Table 5.2 respectively.

| Imputed data | | | |
|---|---|---|---|
| Regressor | R squared | p-value | Std. error |
| Cr | 0.1425 | 0.0075 | 0.4679 |
| Co | 0.0106 | 0.4816 | 0.2472 |
| Zr | 0.0445 | 0.1456 | 0.5076 |
| Ni | 0.1794 | 0.0024 | 0.6172 |
| Mb | 0.0171 | 0.3707 | 1.0637 |
| linWear | 0.0886 | 0.0378 | 0.5255 |
| linWearRate | 0.0427 | 0.1541 | 5.237 |
| volWear | 0.1019 | 0.0254 | 0.0004 |
| volWearRate | 0.0244 | 0.2834 | 0.0052 |
| Inc | 0.0055 | 0.6112 | 0.0334 |
| Ant | 0.0128 | 0.439 | 0.0414 |
| CupX | 0.0009 | 0.8366 | 0.0173 |
| CupY | 0.0765 | 0.0543 | 0.0427 |
| Sex | 0.00201 | 0.7594 | 0.8195 |

Table 5.1: Linear regression with imputed data.

**R squared** measures how well the data fits along the regression line measured. A lower number indicates that it does not explain the variability well, while a higher number indicates that it does.

**p-value** is a measure of the significance of the variables, with a lower value being better. In general, a p-value lower than 0.05 can be considered a significant relationship.

**Standard error** is similar to R squared in that it measures the distance from the regression line. It does so in the unit of the dependent variable, and a lower number is better as it means the point is closer to the line.

From these tables we can see that while the imputation itself was a success in that the imputed values seemingly fit with the rest of the data, it has not necessarily helped the predictive power of the data. When looking at the R squared and comparing the two sets of data, we can see that some of the values did increase, however slightly, indicating a better fit. However, the opposite is true for some of the other values where it has actually decreased, indicating a worse fit. In fact, it seems like a little over half have a lower

| Original data | | | |
|---|---|---|---|
| Regressor | R squared | p-value | Std. error |
| Cr | 0.1309 | 0.0256 | 0.5507 |
| Co | 0.0163 | 0.4444 | 0.3252 |
| Zr | 0.0788 | 0.0877 | 0.5562 |
| Ni | 0.1746 | 0.0125 | 0.7578 |
| Mb | 0.0337 | 0.2695 | 1.185 |
| linWear | 0.1735 | 0.0115 | 0.396 |
| linWearRate | 0.0375 | 0.2581 | 4.1821 |
| volWear | 0.0512 | 0.1843 | 0.0005 |
| volWearRate | 0.0253 | 0.3544 | 0.0041 |
| Inc | 0.0464 | 0.1819 | 0.0271 |
| Ant | 0.0163 | 0.4321 | 0.0335 |
| CupX | 0.0017 | 0.9595 | 0.0138 |
| CupY | 0.0815 | 0.0781 | 0.0365 |
| Sex | 0.0075 | 0.5704 | 0.7569 |

Table 5.2: Linear regression with original data

R squared in the imputed data. Not by much in most cases, but still lower. Looking at the p-value and standard error looks to be telling the same story. Around half of the regressors have better values and the other half has worse values. In both cases, lower is better.

To get a better idea of what is causes this, scatter plots with the regression line overlaid can be used. Figure 5.12 shows the regression line of the Co variable from original data set, while Figure 5.13 is with the imputed data. Here we can see that it indicates a better fit with the model. The data point added by the imputations fits well enough with the rest that the confidence has somewhat increased.

On the other hand, we have the plots shown in Figure 5.14 and Figure 5.15. They are displayed in the same way with the original data and the imputed data and the regression line, this time with the linWear variable. In this case we can see that the imputation has actually introduced more noise in the data and decreased the confidence. Looking through the rest of the variables in the same way show that this is the case for most. Some have introduced

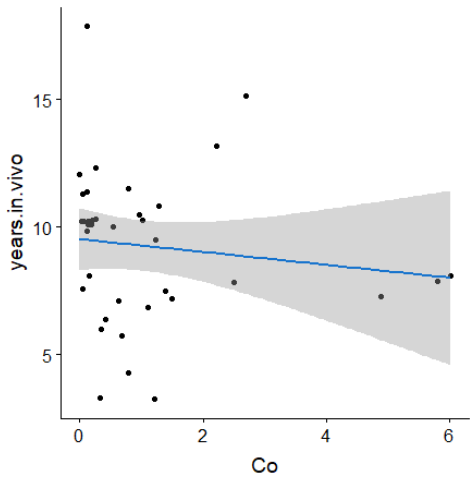



Figure 5.12: 'Co' with original data.

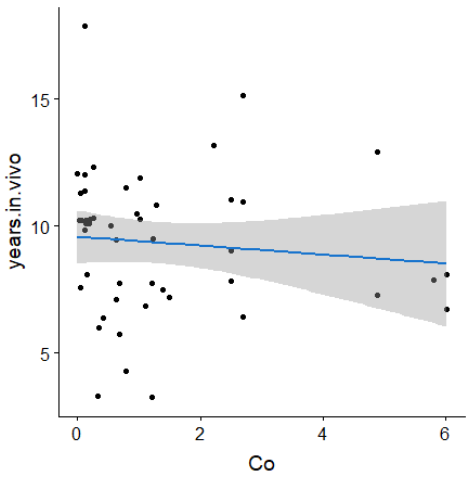



Figure 5.13: 'Co' with imputed data.

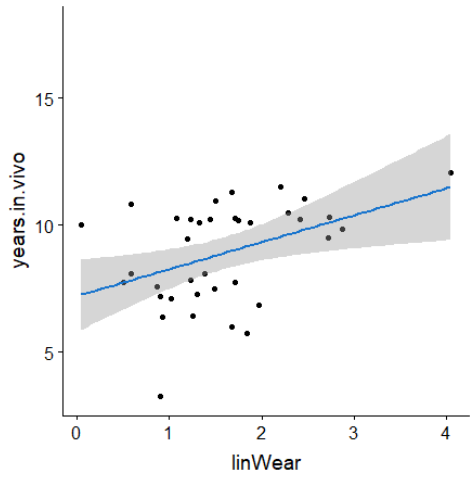



Figure 5.14: 'linWear' with original data.

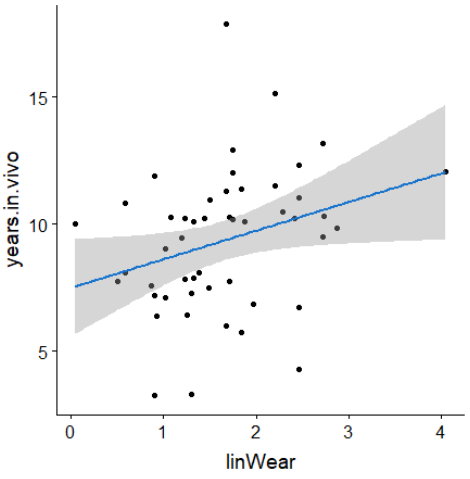



Figure 5.15: 'linWear' with imputed data.

| Imputed data | |
|---|---|
| **Actual** | **Predicted** |
| 12.071184 | 11.855055 |
| 10.236824 | 11.678582 |
| 6.414784 | 7.745044 |
| 6.839151 | 5.538713 |
| 10.329911 | 10.031885 |
| 7.507187 | 7.012504 |
| 11.523614 | 8.548507 |
| 10.494182 | 9.670746 |

Table 5.3: Actual and predicted values with imputed data.

| Original data | |
|---|---|
| **Actual** | **Predicted** |
| 9.834360 | 10.698058 |
| 7.575633 | 9.001381 |
| 10.236824 | 12.112383 |
| 10.020534 | 8.127149 |
| 10.083504 | 9.557858 |

Table 5.4: Actual and predicted values with original data.

| | **Imputed** | **Original** |
|---|---|---|
| Corr. accuracy | 59.8% | 36.3% |
| Min/Max | 85.8% | 87.2% |
| MAPE | 16.6% | 14.0% |

Table 5.5: Prediction validation results

more outliers and introduced more noise which in turn leads to less confidence and worse R squared, p-values and standard error. Others are the opposite where the data fares better after imputation with more confidence and better values. Overall it's difficult to say if there is an actual pattern to this, given the ambiguity of the results.

### 5.5.2 Prediction

To test the potential for using the data for predictions, the 'predict' function in R was used as described in Section 4.6. The two data sets were split into training and testing data, the prediction was run and methods of validation that can give an indication of the accuracy of prediction were applied. The actual and predicted values for imputed data can be found in Table 5.3 and for the original data in Table 5.4. The variable predicted is years.in.vivo, how long the implant lasted before needing to be removed.

As we can see from the tables, the predicted values are relatively close to the actual values in both the imputed data and the original. The greatest difference being about three years. Overall the prediction seem to be relatively accurate just from looking the results. But there are more ways to get indications for the accuracy of the predictions. Three other techniques were used: Correlation accuracy, minimum-maximum accuracy and mean absolute percentage accuracy. Each of these returns a percentage values that gives an indication of accuracy. The results of this can be found in Table 5.5.

Correlation accuracy is a measure of the correlation between the actual and predicted values, indicating whether or not the values have the same kind of directional movement, which can be used as an indication of accurate predictions. A higher percentage is better, and the imputed data handily beats the original data in this case with 59.8% over 36.3%. Minimum-maximum accuracy averages the max and min values of the prediction and actual data and compares it to output the result as a percentage. Higher values are better here as well. With percentages of 85.8% and 87.2%, the original data is marginally better, but not my much. It is however a rather favourable result in terms of implied accuracy.

The final method used is Mean Absolute Percentage Error which measures the accuracy of the prediction. In this case, a lower value is better as it calculates the error of the prediction and not the accuracy of the prediction itself, but is still useful as an indicator of accuracy. Again, the original data is better than the imputed data by a small margin. And again, the calculated percentage indicated that the prediction can be somewhat accurate. The general impression is that it looks promising. The predictions are not too far off from the actuals values, and the methods of measuring accuracy indicate that there is potential there, though perhaps not as much one would hope for.

# Chapter 6

# Discussion

## 6.1 Methods and methodologies

This section of the discussion is about the methods and methodologies used for this thesis. It covers the process all the way from planning and preparing to the actual development and technologies that were used.

### 6.1.1 Design Science

Design Science was the chosen methodology for this project. As explained in Section 3.1, Design Science is often used in information technology and the focus on designing an artefact as in integral part of the process seemed suitable for a masters thesis in information science. It provides guidelines and structure for the project which is helpful both to ensure that it has sufficient thought behind the decisions made, and that is does not grow beyond the intended scope.

The focus on developing an artefact also gives something concrete to work towards and base the thesis on. The practical aspect combined with the theoretical aspect of literature and research make up the bulk this thesis. The hands on approach gives a greater understanding of the technologies

and principles involved in the development, and the challenges one might face.

The guidelines also have a focus on the research and contribution of the work, which is important in academia. This project has been following these guidelines as much as it was suitable. Not following it strictly step by step, but using them as what they are, guidelines, and bring structure to the entire process. In this, they are very helpful and provide a good framework for the thesis.

### 6.1.2 Development

As explained in Section 3.2, there was no specific development methodology that was followed for this project. Picking and choosing the suitable methods from different methodologies and development processes proved to work well. The main purpose was to keep organised in terms of tasks to be completed and using the planning tool as a roadmap of things than needed doing and were done. With just one person working on a project, the collaboration tools and ways of communication are not really of use as it would be in a team. But using it to streamline the development process and keep it structured was something that worked well.

### 6.1.3 Technologies

For a thesis in the field of information technology, the technologies are naturally an essential aspect of it. Part of the thesis is testing out the suitability of the chosen technologies for this purpose. The most important one, and the basis for most of the thesis is the R programming language. The rest are tools that support or complement it in some way. R itself is, as explained in Section 3.3.2, a very powerful statistical programming language. And it did perform very well in working on this project. Working with data and analysis is made relatively easy by giving access to powerful functions that

quickly give results. However, it was not a problem free venture.

While it is indeed powerful language that can satisfy most users and statisticians, it is limited by the knowledge of the user. Being able to run the functions in R is one thing, but perhaps more important is to know when to use which methods. A background in statistics makes it a much more powerful tool. So for this thesis much has been learned from other papers and tutorials on the web to provide the necessary information and knowledge to correctly use the tools provided by R. It is safe to say that it is a programming language for statisticians, and not a statistical language for programmers.

This issue also applies the other way around. In the hands of a statistician with only a basic understanding of programming, you can do very powerful statistical analyses. But for a programmer looking to use it for statistics, the language can seem oddly limiting. There are too many differences to list between this and a "traditional" programming language, but there are several things that makes it more challenging to work with.

The main issue though is that it is simply not made for traditional programming tasks like creating an application with complex interwoven functions and systems, but focuses on single tasks. It is a precision tool that can be used for many different specific tasks, but is not as flexible in how it can be applied like other languages. Some of this is helped by plug-ins and extension you can add to it, but this only somewhat rectifies the issue.

R Shiny is the other major technology that is used. It is a plug-in for R and integrated with the RStudio IDE. It is at the same time both very powerful and limited. It allows for the creation of interactive statistical applications with relatively little coding experience as explained in Section 3.3.3. But on the other hand, this more streamlined creation of simple applications makes it more difficult to create the complex ones.

Another thing is that because it is standardised and simplified to open it up more for non-programmers, it it limited in how you can customize it without significant effort. For instance controlling the size of the elements on the

screen and their position can be an endeavour if you are used to having more control in other languages. In general it is harder to customize the user experience and user interface as you are limited to the components already made for you. This works very well for simple concepts, but if creating a fully fledged system that you would like to customize the look and feel of, it might not be the best choice. For something like this, using python like Longberg [Longberg, 2018] did would provide access to a similar set of tools, but would be easier to extend beyond the limits placed on it by R and Shiny.

So while R and Shiny are powerful in many ways, they are limited in others. If this project were to be worked upon further, a consideration on whether to continue with R or switching to Python would be high on the list of priorities. R seems more suited to smaller applications, where you do not want to have to set up the entire framework with Python, and get a decent looking application together fast. But for the more involved projects with many facets, something like Python might be more suitable.

Another option could be to use R for the back end to do the "heavy lifting" and using standard web technologies with the great variety of already existing plug-ins the present the results. This requires more infrastructure and planning, but is an alternative that lets you harness the power of R while still being more user friendly and design oriented like many modern websites are.

## 6.2 Preparing the data

The process of preparing the data, which is detailed in Section 4.1, was twofold. The first data set was supplied quite a long time before the second and was not easy to work with. There were too many variables to be able to use everything and only some would likely be usable. A lot of time was spent sorting through it and trying out and testing various techniques to see what might work and yield results. This first set of data should preferably have been prepared further before being used for analysis to take out the su-

perfluous material and try to eliminate as much noise as possible. If this was done by or in collaboration with experts in the field who have the necessary domain knowledge to determine what should be used, it would be very useful as the next data set showed.

The second data set was supplied later in the project, and was a significant improvement over the first. Applying the same methods could be done without much preparation at all. This shows the importance of delivering a 'clean' data set if you do not want to make more work when preparing it. This had the relevant variables already chosen from the multitude in the other data. It also contained more data than the first.

Making the preparations for working with the data is relatively simple using R. There are many helpful functions and visualizations then can give you a better understanding of it. What kind of data there is, what kind of data types is consists of and how much data is missing for instance. In this respect, R really shines. There are methods for all kinds of ways of looking at data built into R, and with the CRAN repository of R packages you can find almost almost anything you can think of that you would need. A background in statistics would still be extremely useful here, but the supplied documentation and online tutorials can alleviate some of the harsh learning curve.

It is worth mentioning however that while R was good in this regard, it is a very manual process. You have to type in all the commands and interpret the results yourself. At times, just getting the correct syntax can be time consuming with some of the more complex functions. Creating a data analysis tool for this purpose could be done using Shiny, but would probably be quite complicated to make work in an intuitive manner. Having the data prepared beforehand and not in the application itself might be preferable to keep the complexity down.

## 6.3 Multiple imputation

Multiple imputation was run with the MICE package for R to fill in missing data, further explained in Section 4.2.2. It also provides the tool needed to analyse the imputed data and to work with it to get the most out of the process. Working with MICE was somewhat of a mixed bag, though mostly positive. It can be a powerful tool if used correctly. The various documentation and articles describing the use haven't always agreed on the best way to do things, but the overall recommended process was rather simple to apply.

The basic parameters to run MICE with do not leave much up to interpretation. The number of imputations could probably be turned up much more and get marginally better results as well. On thing to note is the Predictor Matrix parameter that exists in MICE which lets you control how variables are used as predictors. It can further refine the results if you for instance have some expert knowledge about which variables are likely to be relevant for prediction. This is a more advanced method and not implemented here, but could be used in the future. Modern computers have no issue with the computations, especially not with this small data set. On the other hand, the size also makes it harder to verify. The built in methods of verifying if the imputation was successful also worked well, and was thoroughly documented in various papers on the subject.

The first data set was difficult to work with and would crash the program if run with more advanced methods than CART. It was simply too complex and lacking, and even then it struggled to complete it. This again underlines the importance of having good data to work with. The second data set was much better here. Overall the tests indicated that it was indeed a successful imputation with the second data set. The limitations on the data does make it harder to verify though, which has been a pervasive issue throughout the project.

There are several routes to go after imputing the data. To keep it simple

and being able to use it in the prototype application, 'completing' the data was used. This returns a finished set of data with the missing values filled in. It essentially fills in the holes in the data with a data set from one of the imputations. From a purely analytical standpoint, the pooling function described in Section 4.2.5 would likely provide better results, but this is also a more advanced feature that could be used in the future should work continue.

## 6.4 Development

This section of the discussion covers the topics related to the development process. Defining the requirements for the application, creating the prototype and the data mining aspect of the thesis.

### 6.4.1 Requirements

The requirements for the prototype were established using guidelines from the field of interaction design, and described in Section 4.3. Both the functional and non-functional requirements were used to set a goal for what features should be implemented in the system for the prototype. These worked well as both guidelines and to limit the scope, as was intended. Feature creep, where new features are added often and can hinder the overall progress, is easier to avoid when setting clear goals like this. Overall, the system does fulfil the requirements. Granted, it is not an extensive list of requirements but as a starting prototype it does what it needs to.

Creating a simple system first makes it easier to build upon and expand it later, while also keeping the original system in place, at least with R and Shiny. The non functional requirements are more subjective, but given the rather simple nature of the application there is not as much room for misunderstanding as would be in a more complex system. These requirements might be more relevant for a project that is intended for release to a customer

or the public or at least intended to be released in some form, but it is still useful to keep in mind during development.

### 6.4.2 Prototype

Developing the prototype with R and Shiny was an interesting experience. The overall impression was as has been mentioned previously that it is very powerful for specific purposes, but can fall somewhat short when making more complex systems. That is not to say that it cannot be done, but it can get complicated to add and maintain a large number of features.

There are several ways to deal with getting the data into the application. One way is to hard code the data into the application itself, but this defeats the purpose and you might as well run it from R directly. Perhaps the most ideal way would be to have the data on a server where it could be requested through an API and imported into the application. There are several issues with this approach though. You need the infrastructure, the server and hosting, and you also have to take into account security and privacy. Some kind of authentication would be needed so that the public would not be allowed access to it.

Another way would be to make the calculations and analysis server side, and never actually make the data itself available, only the results which could then be used for further analysis and visualization. All of this would require significantly more infrastructure, but would likely be a better solution overall and easier for the end user as they do not have to worry about the data themselves. In the current version of the system, a simple file chooser was implemented to handle data import. This is a part of the Shiny framework and works well as a manual alternative.

Implementing the individual methods in R was a relatively simple process, but stitching it together to make reactive values that the user can update at will takes more work. And even more work should you want to have multiple options and different views. The code base can easily grow quite a

lot larger when trying to implement several features like this, and R might not be the most suitable language for these kinds of systems. That said, the prototype does work, and complex systems can be made like this, but some considerations regarding the maintainability and expandability should be made before beforehand.

The current iteration if the system takes all the information and outputs it to a text area. This is not really that user friendly and would be better served in a more organised table or list with each value clearly marked. The raw output from R is not easily readable and should be improved upon. There is also the issue of hardcoding features or making them dynamic. The variable list for the linear regression is currently being generated from the data chosen in the import. Hardcoding makes it easier to adapt new features as there are less things to take into account that can go wrong, but making it dynamic ensures that it can be used even with changes in the data.

## 6.5 Data mining

### 6.5.1 Linear regression

After the imputation was done, a simple linear regression was done to try and see how the data held up when used for data mining. This was done by running the linear regression in the prototype application and making notes of the values in the summary, adding it to a table for each variable. One table was made for the original data and for the imputed data. Additionally, the scatter plots with regression lines were used to further assist in evaluation.

The results were somewhat ambiguous and not really encouraging in regards to the effects of the imputation on the data. It looked as though many of the variables had more noise introduced as a result and thus less confidence in the regression. This is naturally not the desired outcome, and should be explored further to find out whether or not this was a fluke, or an issue with

he method or the data. There could be several reasons for this problem.

This analysis was done on a single imputed data set extracted from the set of all imputations. If it were to be done on another imputed data set instead, it might produce different results as each imputation is different. This is not a very efficient way of working though. A more natural way to examine this is to use the Pooling function of MICE mentioned in Section 4.2. This function allows you to run analysis on each of the imputed data sets and pool the results together, in essence analysing each set separately and taking the mean of the results to end up with a more realistic outcome as there can be a lot of variation between single data sets. These somewhat more advanced methods are outside the scope of this thesis, but is a natural next step in this work.

### 6.5.2 Prediction

In order to explore the potential for using this data to make predictions, a multiple linear regression model was used in conjunction with the predict function in R. The results from this analysis can be seen in Section 5.5.2. The results indicate that the imputation has had some positive impact in regards to one of the measures, correlation accuracy, but not Min-Max accuracy or MAPE. Looking at the predicted values compared to the actual values, we can see that none hit the mark exactly, though a couple came pretty close.

The data looks to perform fairly well on its own without the added values from multiple imputation, though it did improve one of the metrics. It also resulted in the benefit of more results, simply due to having more data to work with. R will remove observations with missing values by default. This is one of the benefits of imputation, allowing the use of the full data set without having to exclude parts of it.

The actual process of prediction was fairly straightforward. Using the previous linear regression model and adding more variables for a multiple regres-

sion that could be used as a model for the prediction. The exact same code was used for the original and imputed data, with only the input data being different to keep the process the same and the results comparable. This is by no means an exhaustive analysis, but an initial indication on the usefulness of this data for prediction. There are several issues that should be solved before proceeding, and many other data mining methods that could also be applied here.

Overall the results look promising with some trepidation in regards to accuracy. In order to properly evaluate this the uncertainty of imputations have to be taken into account. This is something that needs to be explored more if it is to be used on the data for predictions. Another issue is the simple fact that this is a very small data set. Making accurate predictions based on observation numbers in the tens and not hundreds or thousands is optimistic, especially with machine learning methods that generally perform better the more data you have.

## 6.6 Answering the research questions

1. *Can a relatively small and incomplete data set be prepared and expanded for use with data mining and still produce reliable results?*

   **It could be possible**. The data set was limited in size and with a relatively high percentage missing. This will often be the case when working with real world data and is a relevant issue to tackle in regards to data analysis. The missing values were filled in with multiple imputation which, when looking exclusively at the methods of validation for the imputation itself, looks to have been successful (Section 5.1). The missing values were replaced by new values created based on the rest of the data. However, when using this imputed data for linear regression analysis, the methods of validation indicated that there was no marked improvement (Section 5.5.1). Some variables scored slightly worse and some slightly better, but overall it does not appear to have had much

effect. There was some improvement on the use for predictions, but only for one metric. If this technique is to be utilized effectively, more advanced methods of imputation and validation should be explored.

2. *Can this data be utilized in well known methods of data mining?*

   **Yes**, although it is of limited value due to the small size of the data set. In this case, linear regression was used as a simple and very well documented method of data mining. The data set is small enough that when using the imputed data with about 16 per cent of the values filled in, it is enough to noticeably change the results (Section 5.5). In some cases for the better and others for worse, but mostly to a small extent. The same is true for the prediction, where it did positively affect it, but not by that much. While these are only two methods, the results indicate that this could be the case for other methods as well, and should be rectified or at least more thoroughly explored before proceeding with further analysis.

3. *Can this data be used to help predict outcomes for patients that have undergone arthroplasty surgery?*

   **Yes**. Despite the limited amount data and the uncertainty of the effect of multiple imputation on the data, both data sets performed relatively well in the initial prediction, as indicated by the calculations of accuracy. The tests were generally positive in that they indicated that the predictions were potentially somewhat accurate (Section 5.5.2). This is despite the small size of the data set, so having more data would be very useful to be able to further validate the performance of the prediction.

# Chapter 7

# Conclusion and Future Work

## 7.1 Conclusion

The goal of this thesis was to explore the possibilities of using data mining methods to analyse and make predictions regarding patient outcomes of total hip arthroplasty surgery. Using clinical data from Haukeland University Hospital with failed cases, there was an attempt to improve the data by filling in missing values as well as using the data for an application that can be used for analysis through data mining. Design Science research was used as a framework for the development of an artefact, and Knowledge Discovery in Databases used as a basis for data mining. Both proved useful as guides for the project as a whole.

Using the R statistical programming language and a plug-in called MICE, a technique called Multiple Imputation was used to calculate likely values to fill in missing values based on the existing data. It is a well known and tested technique for filling in missing values, and is supported by literature and thorough documentation. Through various methods of validation, the imputation looks to have been a success, though the size of the data set and large amount of missing values might have had an adverse effect on the end result and made it more difficult to verify the results directly.

The next step was to create an application that utilized the data to allow the user to perform simple data mining techniques. This was achieved using a plug-in for R called Shiny. An application was made that allowed the user to upload a CSV file so that could then be used to run a linear regression and analyse the data. Shiny proved to be both powerful and limited at the same time; it is possible to quickly make simple systems that do complex tasks like data mining, but it might not be as suited for more complex systems with many parts.

Utilizing the application to run linear regression analysis on the data to compare the original and imputed data gave varying results. In some cases it improved the fit of the regression, but in others it decreased it. The results overall were inconclusive, and casts doubt on the value of using multiple imputation in this case. The relative amount of missing data combined with the small size of the data set is likely to play a part, given the techniques' significant backing in literature. More accurate results could likely be achieved with more advanced techniques, as well.

Using a prediction method in R, there was an attempt to assess the predictive power of the data and comparing the original and imputed data sets. This fared much in the same way as the linear regression with mostly ambiguous results in comparison, but cautiously optimistic in regards to using the data for prediction. It is by no means a sure predictive model, but with more data and more work it could potentially be used for predicting the longevity of the implants.

Overall the results show some promise, and the techniques used are backed by previous literature and research. However, the results here are inconclusive and requires more work to reach a conclusion on the usefulness of in on this data set. The technologies used should also be reconsidered to fit the purpose and scalability of the desired system. This thesis explores possibilities of using the limited data available, and while they do not yield conclusive results in this case, it is worth revisiting with more knowledge and more data in the future.

## 7.2 Future work

Future development on this project would go in two directions. The data itself and the application for data mining. Improving the data could be done by utilizing more of the available methods in the MICE package for Multiple Imputation to further optimize it, especially selecting the best variables to be used for imputation in cooperation with an expert. And also using the pooling method to use the full range of imputed data for analysis, which can lead to even better results.

Second is the application itself. In its current form, it can be expanded with more methods of data mining. Exploring well known methods and more experimental ones, and looking more into machine learning is a natural next step in this regard. But the limitations of Shiny become apparent as one expands the application. Other tools like Python could be better suited for a more complex system, and this should be further explored, alternatively with R running the back end. Switching to a back end / front end model would relieve the user of several tasks and increase efficiency and security, as well as being more user friendly. It would however require more infrastructure.

Exploring the data set further and setting up data mining modules using more advanced techniques could be one way of developing good predictions, as well as other models in the early phases of gathering data, and aid understanding of the feasibility of data mining.

# Bibliography

A. Babic, B. Peterzen, U. Lönn, and H. C. Ahn. *Case Based Reasoning in a Web Based Decision Support System for Thoracic Surgery*, pages 1413–1416. Springer International Publishing, Cham, 2014. ISBN 978-3-319-00846-2. doi: 10.1007/978-3-319-00846-2_350. URL http://dx.doi.org/10.1007/978-3-319-00846-2_350.

Tor Aimar Carlsen. Designing an e-learning platform forpatients undergoing hip replacementsurgery. Master's thesis, december 2018. URL http://bora.uib.no/handle/1956/18769.

L. Duan, W. N. Street, and E. Xu. Healthcare information systems: data mining methods in the creation of a clinical recommender system. *Enterprise Information Systems*, 5(2):169–181, 2011. doi: 10.1080/17517575.2010.541287. URL http://dx.doi.org/10.1080/17517575.2010.541287.

Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: an overview. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in knowledge discovery and data mining*, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996. ISBN 0-262-56097-6. URL http://dl.acm.org/citation.cfm?id=257938.257942.

GNU Project. The GNU General Public Lisence. URL https://www.gnu.org/licenses/gpl-3.0.en.html.

Geir Hallan. Wear, fixation, and revision of total hip prostheses. 2007.

Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *Management Information Systems Quarterly*, 28(1), 2004. URL https://aisel.aisnet.org/misq/vol28/iss1/6/.

Per Niklas Longberg. HALE, the Hip Arthroplasty LongevityEstimation system. Master's thesis, december 2018. URL http://bora.uib.no/handle/1956/18783.

Oracle. VirtualBox, 2018. URL https://www.virtualbox.org/.

Jenny Preece, Yvonne Rogers, and Helen Sharp. *Interaction Design: Beyond Human Computer Interaction*. 2011.

R Project. The Comprehensive R Archive Network, a. URL https://cran.r-project.org/.

R Project. R: What is R?, b. URL https://www.r-project.org/about.html.

RStudio Inc. Shiny, 2018a. URL https://shiny.rstudio.com/.

RStudio Inc. IDE features – RStudio, 2018b. URL https://www.rstudio.com/products/rstudio/features/.

Joseph L Schafer. Multiple imputation: a primer. *Statistical Methods in Medical Research*, 8(1):3–15, feb 1999. ISSN 0962-2802. doi: 10.1177/096228029900800102. URL http://journals.sagepub.com/doi/10.1177/096228029900800102.

Jorge S. Siopack and Harry E. Jergesen. pages 243–249, march 1995. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1022709/.

Stef van Buuren and Karin Groothuis-Oudshoorn. *MICE: Multivariate Imputation by Chained Equations in R*. [UCLA Statistics], 2010. URL https://dspace.library.uu.nl/handle/1874/44635.

Natalja Voznuka, Hans Granfeldt, Ankica Babic, Markus Storm, Urban

Lönn, and Henrik Casimir Ahn. Report generation and data mining in the domain of thoracic surgery. *Journal of Medical Systems*, 28(5):497–509, 2004. ISSN 1573-689X. doi: 10.1023/B:JOMS.0000041176.58311.29. URL http://dx.doi.org/10.1023/B:JOMS.0000041176.58311.29.

Ian R. White, Patrick Royston, and Angela M. Wood. Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine*, 30(4):377–399, feb 2011. ISSN 02776715. doi: 10.1002/sim.4067. URL http://doi.wiley.com/10.1002/sim.4067.

Ruiting Zhao. *Investigation of mechanisms leading to early aseptic loosening of hip prostheses. A retrieval study of 27 failed cases of cemented Spectron EF stem in combination with Reflection acetabular cup.* PhD thesis, may 2016. URL http://bora.uib.no/handle/1956/12880.