University of Twente
Behaviour Management and Social Science
Industrial Engineering and Management

Master Thesis

# Data driven solution to predictive maintenance

*Authors:*
Xinyi Chen

UNIVERSITY OF TWENTE.

*Supervisors:*
Prof. Jos van Hillegersberg
Dr. Engin Topan

TATA STEEL

*Supervisors:*
Steve Smith CEng MIET
Matthew Roberts CEng MIMechE

January 16, 2020

# PREFACE

Here it comes, another mile stone of life. I will not say this is the end of my student life, as I wish it is not. Learning and exploring has become an essential part of my satisfactory that I can not live without.

Plenty to be grateful for. First of all, for the opportunity provided by my first supervisor Jos van Hillegersberg so that I was able to do my thesis in the UK. Jos was very hard to grab before this project kicked off as he is a very busy person. However, when the thesis officially starts, I can feel how supportive he actually is. I appreciate that Jos is always innovative and ask me to think about more on how to contribute to the academic world not only to practice. His comment is always focus on the content and process and barely on writing(This is also the comment from his other students from BIT and that's part of the reason I chose him to be my supervisor.) Thus to the reader(fellow student) of this work, if you want your thesis to be done efficiently, you'd better go to him. Secondly, Engin Topan, my second supervisor is able to critically assess my work in technical perspective that influence the way I do the experiment in a more systematical way.

Thirdly I want to thank Tata Steel Shotton and my company supervisors. TATA Steel Shotton is like a family. I have spend a fantastic time here. Both of my supervisors Matthew Roberts and Steve Smith have been given me plenty of trust and freedom. Every report meeting I had was full of encouragement and complement that drive me to deliver better result every time. I'm also very grateful for the inclusive environment in the company where I get to hangout quite often with colleagues and get to know their families.

Finally, I want to thank my parents to always been supportive to me and have faith in me. My loving boyfriend who has been accompanied me in the UK for 3 months now and has taken up 99% of the housework for me to focus on my work.

Very last word, University of Twente is the very first place that lead me to discover my full potential, to let my work speak for me, to make life long friends and unforgettable memories. I have had a brilliant time there and if I were to be given a second chance I will choose it again.

January 16, 2020
Xinyi
TATA Steel Shotton, United Kingdom

# MANAGEMENT SUMMARY

This project investigates predictive maintenance using data driven methods in the context of industries that are in transition to Industry 4.0. The study is conducted in TATA Steel UK (Shotton) as a case study.

The problem lies in one of the production line called Hot Dip Galvanising Line in TATA Steel Shotton. The function of the line is to coat zinc on steel strips to protect the strip surface. The critical part of the Galvanising line is the pot gear which is filled with melted zinc. Three rolls(sink roll, stabilizing roll and correcting roll) are sunk in the zinc pot to carry the strip. The three rolls together are replaced approximately 4 weeks because the bush that is connected to the sink roll is likely to wear out at around 4 weeks. This decision for the maintenance interval is purely based on experience and sometimes the part is overly maintained that when pulling the rolls out the bush hasn't been worn out. Thus the engineers are keen on getting insights on the wearing pattern of the bush such that the bush can be replaced just in time. Moreover, plenty of data has been logged through sensors in distributed systems but has never been used for decision support, thus the project is aiming at predicting the bush wear with currently available data.

Although the ultimate goal is to improve the maintenance of the pot gear, in this thesis, our goal is only to predict the bush wear and thus the wearing data is the target variable. The wearing data started to be measured during the preparation time of this project at the end of every maintenance cycle when the component is replaced. The wearing data is measured manually by operators. This leads to the first challenge: small sample size due to small size of target variable. By the end of this project 9 samples are available in total.

There are 4 data sources exists for data logging namely: Set up sheet, IBA, EMASS and Data Warehouse. While the target variable is being recorded, all data sources within the company have been investigated to understand the meanings of those data and check the qualities. Intensive literature review has been conducted, aiming to find the vital variables to be used as predictor where massive amount of literature suggests using vibrations as predictor, however, component itself is in a zinc pot, and there were no sensors connected directly to the bush so the vibration data is not available. This is the second challenge faced during the project.

Investigation on metal contact wear was then conducted aiming to find alternative predictors. The Archard's law is found to be the most common mathematical formula for metal contact wear where sliding distance and force are suggested the two most critical parameters. After investigating into the indicator of sliding distance and force, related variables are extracted from different sources. Time indicator(Number of Days) and environmental indicator(bath temperature) are also extracted for further selection. In the end 14 features are selected as shown in Table 6.7 including the bush wear

measure. The feature set is selected based on model performance.

| **Features** | **Data Source** |
|---|---|
| Total Length | Data warehouse |
| Scrape Length | Data warehouse |
| Total Surface | Data Warehouse |
| Mean Tension | IBA |
| Minimum Tension | IBA |
| Maximum Tension | IBA |
| Median Tension | IBA |
| Skewness Tension | IBA |
| Kurtosis Tension | IBA |
| Standard Deviation Tension | IBA |
| RMS Tension | IBA |
| Remaining Bush Width | Set up sheet |
| Days | Set up sheet |
| Roll Diameter | Setup sheet |

Table 1: Features selected based on PLSR performance

Three modeling techniques that have been used and compared are Partial Least Squared Regression (PLSR), Artificial Neural Network (ANN) and Random Forest (RF). PLSR is chosen because according to literature it is suitable when the number of independent variables is larger than the sample size and when the dependent variables and independent variables are forming a linear relationship. The linear relation is suggested by literature and by fitting a linear regression to the existing samples we found that the independent variables and the dependent variables did form a linear relation. ANN is selected because it is the most used technique in literature and always produces good results. Some literature suggests that the minimum sample size for ANN is 10. This requirement has not been met yet but will be accomplished in the near future thus ANN is worth investigating. RF is selected because it has also been used in the literature to predict component wear and produces good prediction accuracy. However, both ANN and RF are used when vibrations data are available as predictors in literature.

The models are evaluated using cross validation due to small sample size. In order to maximize the training set variance, every time we train the model we use only one sample as test set and the rest as the training set. RMSE is used to evaluate the prediction power of the models and and R squared value are used to see how much variance of the data can the model present. Learning curves are plotted to see how the modeling performance will change when adding more samples to the training set. The result can be found in Table 6.9 and Table 6.10. Learning curve can be found in Figure 6.21 and Figure 6.22. We can see from the result that PLSR is the most suitable model for the current available data. Further more, a monitoring web page has been developed for the engineers to see the bush wearing behaviour online. The web page is based on PLSR model. The user only need to insert the date to see the predicted wearing pattern from the starting date of the corresponding maintenance cycle.

In conclusion, predictive maintenance using current available data in TATA Steel Shotton is feasible. The modeling performance can be improved by improving the data quality of the corresponding vari-

| Training samples | PLSR | ANN | RF |
|---|---|---|---|
| 5 | 2.23 | 0.39 | 7.16 |
| 6 | 3.53 | 4.39 | 5.65 |
| 7 | 5.74 | 6.58 | 6.43 |
| 8 | 4.14 | 5.26 | 6.09 |
| Average | 3.91 | 4.15 | 6.33 |

Table 2: Modeling RMSE comparison

| Training samples | PLSR | ANN | RF |
|---|---|---|---|
| 5 | 0.85 | 1.00 | -0.53 |
| 6 | 0.61 | 0.40 | 0.01 |
| 7 | 0.18 | -0.07 | -0.02 |
| 8 | 0.53 | 0.25 | -0.01 |
| Average | 0.55 | 0.39 | -0.14 |

Table 3: Modeling $R^2$ comparison

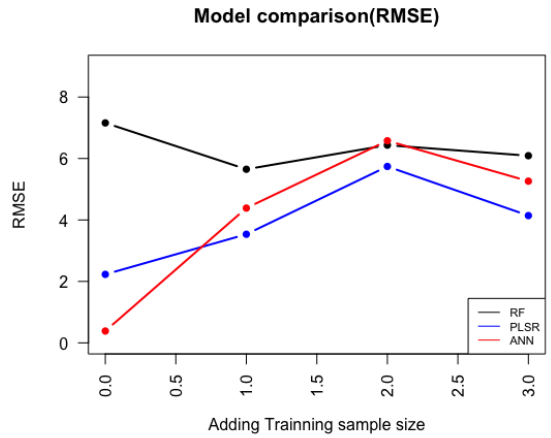


Figure 1: Modeling comparison RMSE

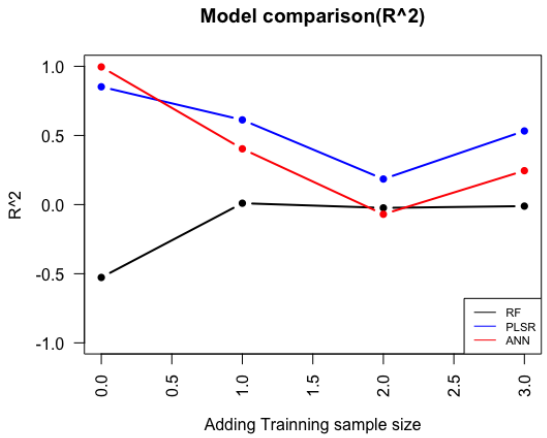


Figure 2: Modeling comparison $R^2$

ables. As a industry in transition period, we discovered challenges when doing predictive maintenance using data driven method. Challenges mainly come from unbalanced and limited samples and the quality issue of data in real life setting.

The limitations are that the current model is built upon a very small sample size thus the model maintenance is critical. The whole modeling process(from feature selection to technique selection) should be conducted repetitively in certain time interval as the performance is subject to change when more data is coming in. The current model is predicting the current wear instead of future wear. Thus the developed web page can be viewed as a monitoring tool instead of a prediction tool. Furthermore, this study has only been looking at the technical perspective of predictive maintenance however there are many other aspects such as economic, regulation and employment etc. All these aspects might bring challenges for implementing predictive maintenance in industries that are in transition period.

Follow-up projects to this study can be conducted in many different directions. The current model can be tested to see whether it can predict future bush wear by selecting a time window in the past as predictor. The current model can be expanded onto similar problems on different lines and even if the problem is different, the methodology used in this study can still be referred to. In addition, unsupervised learning techniques are also worth investigated to be implemented on large sensor based data.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

CHAPTER
# ONE

# INTRODUCTION

The latest industrial revolution proposes varieties of smart products such as smart cities and smart grid. While smart industry concept is referred to as "Industry 4.0". The concept includes smart manufacturing, smart factory, lights out manufacturing and internet of things (IOT). (Sniderman 2019) The essential idea of industry 4.0 are automation, connectivity and big data exchange in manufacturing process. Automation leads to not only automotive production but also automotive decision making system. One of the application is predictive maintenance.

Rotating metal to metal contact wear prediction is one of the critical area in predictive maintenance as rotating mechanical components such as bearing and bush are widely implemented in machines and failure of those causes down time of machinery and production line. Existing prognostic methods can be classified into three categories namely are "Model-based prognostics", "Data driven prognostic" and "Reliability-based prognostics" (Tobon-Mejia 2012). Model-based prognostics requires deep knowledge in system functions. Mathematical models are built to represent the system behaviour including component degradation process. However, systems are often complex in reality thus mathematical modeling is often computationally expensive and assumptions are required when building models. Reliability based prognostics can also be referred to as "experience-based prognostics" which uses historical data during a significant period of time and discover the statistical distribution for each parameter. Poisson, exponential, weibull and log-normal distribution have been proposed in the literature for failure time distribution. This approach is easy to implement when historical data from significant period of time is available, however, the prediction result is less precise than those with model-based and data driven method. Data driven prognostics aiming at getting information from the raw data mainly from sensors. It uses mainly artificial intelligence tools or statistical models to learn the wearing behaviour and to predict the condition in the future. The model operates automatically without considering the explanatory power to the real system or parameters. Although data driven method is not computational expensive it still provide good prediction results for systems where it is easy to monitor data representing wearing behaviour or system failure. However, critical data such as failure related data are often missing in industry as failure has been prevented in every way possible due to the huge cost of down time (Zschech 2019).

This project conducts a data-driven bush wear prediction based on steel production line in Tata steel Shotton (UK). A critical part of the production line is maintained by replacing the component every four weeks which leads to the fact that there were barely any failures occurred. As the bush operates in melted zinc pot and there is currently no sensors connect to it, the bush remaining width is measured every four weeks thus the wearing data is only available at the end of each maintenance cycle from

15/05/2019.

The main contributions of this study will be two folds as follows:
To academic environment:
-This study provides some characteristics for industries that is in transition to industry 4.0 in terms of predictive maintenance.
-This study finds a possible existing model to use in industry context that was barely used in predictive maintenance before.
-This study discusses the difference between theory and practice and presents challenges could face when doing predictive maintenance and the guidelines to deal with the challenges in industry in transition to industry 4.0.

To practice:
-This study investigates different data source in TATA Steel Shotton reflect on data quality and improvement regarding data logging.
-This study discovers gaps within currently logged data and essential parameters needed for wearing prediction.
-This study gives conclusions on the feasibility of predicting bush wear and prediction results on proposed methods that help TATA steel Shotton gain insight on the big data and the ways to better fit in industry 4.0 concept.
-This study developed a tool to monitor the wear using selected model(s) such that the research result is able to be implemented in operations.
-This study presents business opportunities and following up projects for TATA steel to further improve the current maintenance process as well as expanding the improvement to other sites and other problem area.
Current situation and problem description are presented in the rest of this chapter. Section 1.1 will introduce the current case situation, section 1.2 describing the problem, section 1.3 outlines the strategy and method to solve the problem. A report structure is proposed at the end of this chapter.

## 1.1 Current Situation

### 1.1.1 Tata Steel Shotton

Tata steel shotton is located in Deeside, North Wales, with its annual production of approximately 500,000 tonnes of steel for building envelope, domestic and consumer applications. The plant in Shotton has existed for over 120 years, the colour coated steel products have been produced for over 50 years and are backed by guarantees of up to 40 years. Differentiation has been its strategy with innovative products occupies 75% of the order.(Tata Steel at Shotton fact sheet).There are in total 22 lines in Shotton, the project is based on the No.6 Galvanising line.

### 1.1.2 No.6 Hot Dip Galvanising Line

Galvanising is the process of coating the steel substrate with zinc to protect the steel from atmospheric contaminants such as water, oxygen and salts such that the steel corrodes slower. The No.6 Galv Line consists of the following sections: Entry, Furnace, Cooling, Bath, After Pot Cooling, Water Quench, Temper Mill, Tension Leveller, Chemical Coater Section, Oiler and Exit Section. These sections are presented in Figure D.2.

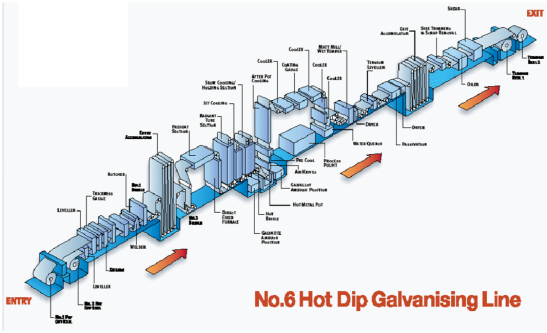


Figure 1.1: No.6 Galvanizing line diagram(TATA)

All coils arrive on site by rail and transported on shuttle car from rail head to entry. The raw coils are cropped to remove any off-gauge or damaged steel before being put on entry section. The Furnace section is to clean the strip and prepare it to be suitable for galvanizing. The aim of the cooling section is to reduce strip temperature to about 520°C and stop any further micro-structural changes. There are two bath on No.6 Galv namely Galv Bath and Galfan Bath that galvanize different products. After the strip has passed through the bath, the zinc coating need to be solidified before touching any roll surface. That's why after pot cooling exist to lower strip temperature to below 280°C. The strip is cooled further in the Water Quench to achieve a strip temperature of 50°C. The Temper mill and tension leveler applies a to the galvanized strip and stretches the strip respectively to improve strip shape, surface finish, mechanical properties and remove yield point elongation.

The Chemical Coater Section applies precisely metered amount of coating to both sides of a moving strip before the Oiler applys a film of oil on one or both sides of the strip. The inspectors can inspect the coil throughout process at the Exit Section. An exit inspection sheet is filled out by exit operators with quality details of the strip. Once the coil has finished it is transported to the packing or storage area for customer or further processing.

### 1.1.3 Bath gear equipment characteristics

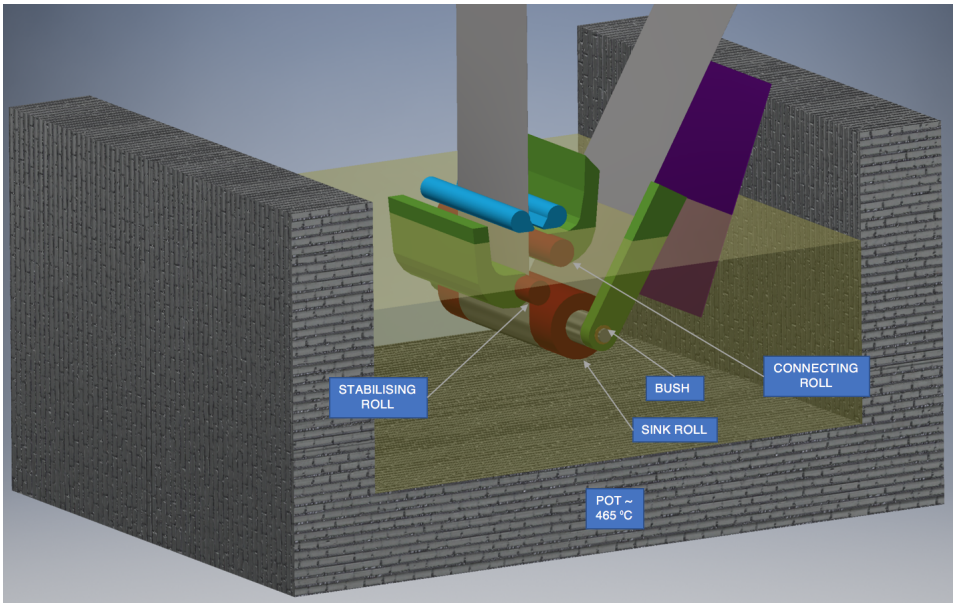


Figure 1.2: Layout of Bath Gear Equipment Submerged in No.6 Pot(TATA)

The pot gear is a key part of the NO.6 Galvanising Line which consists of three major parts: the sink roll, stabilizing roll and the correcting roll. The layout graph can be found in Figure 1.2. All three rolls are inspected and replaced after 3-4 campaigns(maximum 100 days in bath).

The Sink Roll is always coated with tungsten carbide, the diameter of the sink roll is in between 560mm and 600mm. The roll is connected to standard sleeves that are replaced every Bath campaign(4 weeks). Two sets of legs are are connected to sink roll frame that are fitted with concentric bushes. Bushes are replaced after every bath campaign.

Same as the sink roll, the stabilizing roll is tungsten carbide coated as well, with maximum diameter 230mm and minimum diameter 210mm. Roll can be fitted with either standard or coated sleeves based on what is identified on the set up sheets. When the rolls are new from stores they are to be fitted with coated sleeves. The fitted standard sleeves are changed after every bath campaigns. Bearing blocks are fitted with standard length bushes. Blocks are replaced after every campaign and bush can be reused for another campaign.

The correcting rolls has similar characteristics to the stabilizing roll. Only the bearing blocks to be fitted with reduced length bushes except when no reduced length bushes available.

The distance between the correcting roll and stabilizing roll as well as the size of all the rolls in the bath can play an important part in the bow of the strip. Generally it is very hard to produce a flat strip and normally the strip is slightly bowed which causes unbalanced coating on strip.

In addition to the rolls, the Air Knives plays an important part to control the strip shape. The knives should be operated low to the bath surface when running higher coating weights or at low line speeds which provides better strip stability, less cross bow and less dross on the edges. If the knives are too high, the distance between strip and knife is tend to be high. As a result, pressure are high, cross bow

in strip and un-stability of the strip increases. Owing to this, the strip is likely to be faulty coated such as non uniformly distributed coating and more dross. The knife shouldn't be too low to the bath either as it causes splash from the liquid spelter.

## 1.2 Problem Description

The maintenance cost of the pot gear is increasing drastically during the past years. In 2009, the maintenance cost of pot gear is under £50000 while in 2019 this number has already been increased to £400000. The correcting roll and the sink roll occupies the most of the maintenance cost (TATA).

One of the equipment that is connected to the sink roll is the bush. The bush are replaced after every bath campaign which is expected to be 4 weeks. This is due to the bush wear, by experience, the bush will wear into the legs after 4 weeks. As shown in Figure 1.3, the bush the upper part of the bush is wider than the lower part. According to domain expert, the wearing process is happening to the upper part from the inner circle to the outer circle. When unexpected events happened such as the strip breaks then the bushes also are replaced even if the campaign is not completed. This is an expensive process because every time the bushes are replaced, the whole line has to be shut down, the pot gear is taken out of the bath and replaced with a new set. The old one is inspected, bushes are replaced. Despite the fact that it cost 1507£/hr for the line shut down and it takes 7 hours to replace the pot gear, it is also dangerous assignment for the operators. Thus it is valuable to investigate the cause of the bush wear and potential ways to predict the wearing condition of the bush.

Although plenty of sensors and data loggers are installed in the process and different data storage servers are available, the decisions are still made based on experience. From process point of view, the engineers and the management teams have a brief picture (different prospective) of what may be the reason of the bush wear, however none of those reasons are confirmed or validated. Thus this project has been set up to discover the correlation among different parameters and the relation between a combination of parameters and the bush wear with limited amount of wearing samples. An ideal project out come could be given a set of parameter value in a specific time window, the wearing condition of the bush can be predicted. Such that the operator will only replace the pot gear when it is just necessary and maintenance cost is reduced.

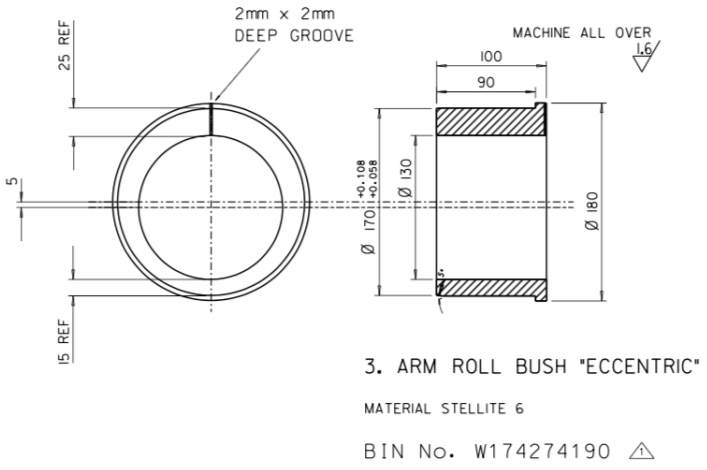


Figure 1.3: Bush(TATA)

## 1.3 Research Design

### 1.3.1 Research Problem

Based on the current situation described in the previous chapter, the goal of this project is to investigate the feasibility of predicting bush wear with current available data and to propose potential methods to predict bush wear using data driven method while wearing data are in small sample size. By gaining insight on the bush condition, the ultimate goal is to prolong maintenance cycle such that maintenance frequency is reduced and so as the maintenance cost. In other words the following question will be answered by the end of this project:

*"How can data driven methods be applied to predictive maintenance in industries that are in transition to industry 4.0?"*

### 1.3.2 Scope

This study is mainly looking at data-driven methods to predictive maintenance in terms of predicting the bush wear. Mathematical models may be used to assist on modeling performance but it is not essential. In terms of modeling, artificial intelligence models are used and evaluated. We use bush wear prediction in Tata Steel Shotton as a case representative of predictive maintenance in industries that are in transition to industry 4.0. By solving the specific problem we are likely to be able to reflect on industries in transition period in general.

### 1.3.3 Research questions

In order to be able to answer the main research question, sub-questions are defined and the corresponding approaches of answering each sub questions can be found in Figure 1.4. Steps within the approach are explained in the next section.

**Research question 1**: What is the current situation in Tata Steel Shotton as an industry that is in transition to industry 4.0?
**1a.** What is the current maintenance process in Tata Steel Shotton?
**1b.** What are the existing data that are ready for collection?
**1c.** What are the meaning of the existing data?
**1d.** How can the data from different sources be integrated?

**Research question 2**: How is predictive maintenance fit in industry 4.0 concept and what is the current state of art regarding predictive maintenance using data driven methods?
**2a.** Which data driven models have been used to predict metal contact wear in industries?
**2b.** What are the steps that has been used in literature in terms of data prepossessing and model building?
**2c.** What are the measures to evaluate predictive power of the models?
**2d.** Based on prediction power and current context which models are the most suitable for predictive maintenance?
**2e.** What are the gaps and limitations of current literature on predictive maintenance in industry?

**Research question 3:** How is the quality of industrial data and what are the challenges?
**3a.** What are the characteristics of industrial data?
**3b.** What are the challenges while pre-processing the data and how to deal with them?

**Research question 4:** What are the most suitable and feasible machine learning models for wearing prediction in Tata Steel Shotton ?
**4a.** What features can be extracted from the data set?
**4b.** What models are suitable to use based on the current available data?
**4c.** How can the model be trained and how can the modeling performance be evaluated?
**4d.** How can the predictive model(s) be implemented to the production operation?

**Research question 5:** What are the business insights that can be extracted from the modeling performance?
**5a.** Is it feasible to predict the bush condition ?
**5b.** How can the model performance and current situation regarding data logging and maintenance process be improved based on the findings?

### 1.3.4 Method

CRISP-DM(Shearer C,2000) has become the most applied and referred approach for data mining expert.(Forbes,2015) However it is a generalized methodology for big data analytic not specifically for predictive analytic. A guideline for predictive analytic has been proposed in (Shmueli & Koppius 2011). During execution of this project, the two methods mentioned above has been combined and modified to fit in real industry context. The detailed steps taken is shown in Figure 1.4

**Business understanding and data collection**
Step1 to 3 were conducted to answer research question 1a-1d. Meeting were organized to meet line expert and help understand the problem area. It was noticed that different people have their own understanding of the problem and what may causes the problem and most of the them don't have scientific knowledge support. Instead, most of the opinions were based on experience. Moreover, nearly everyone suggested the cause of the problem from process point of view and have very limited knowledge on the existing data. Experts exist for the data source but not for the existing data. However, by communicating and measure suggested missing variables, we were able to generate knowledge on data availability and current maintenance process.

**Literature review**
Batch knowledge was generated by literature review in step 4. The goal is to learn as much as possible on predictive maintenance in industry and existing modeling techniques. Techniques were investigated in a general level of industrial context and also to solve the case problem. By the end of this step, research questions 2a - 2f should be answered.

**Data preparation and exploration**
Data are cleaned and explored at this stage. By understanding the data and comparing data to its norms data quality is reflected. Some descriptive techniques for example unsupervised learning were used to explore the data. A data cleaning guide and the challenges were presented specifically to the industrial context. Research questions 3a - 3b are answered by the end of step 6.

**Modeling and result analysis**
The modeling phase corresponds to steps 7 - 9 to answer research questions 4a - 4d. Features suggested by literature were used in this stage. The modeling techniques were selected based on case specific situation. Features were further selected based on predictive power of selected models. That is different feature sets were put into the same model to compare prediction results. Modeling performance were validated after the best models were selected. The selected model is integrated into operations

by tool development.

**Conclusion and recommendation**

Meetings with different function teams were organized during the process to discuss the contribution of this project in terms of project/business development economic and the academic world to answer research questions 5a - 5d. The case context is considered as industry environment that is in transition to industry 4.0.

**Generalization**

By answering the main research question based on the case study we are able to provide a guideline on how to do predictive maintenance in industries in transition period to industry 4.0 with certain characteristics. Challenges occurs not only when using industrial data but also some theoretical models that may not work in practice. The difference between theory and practice may bring more value for future research in real industry context. The method in general is working in an agile way that steps are interconnected as a feedback loop. Outcome of the next steps are reflecting and validating the previous steps and when errors are detected it is always required to go back to the previous steps and fix the error. Project result in most cases will generate new follow up projects and problems to solve and gradually improve the current business situation.

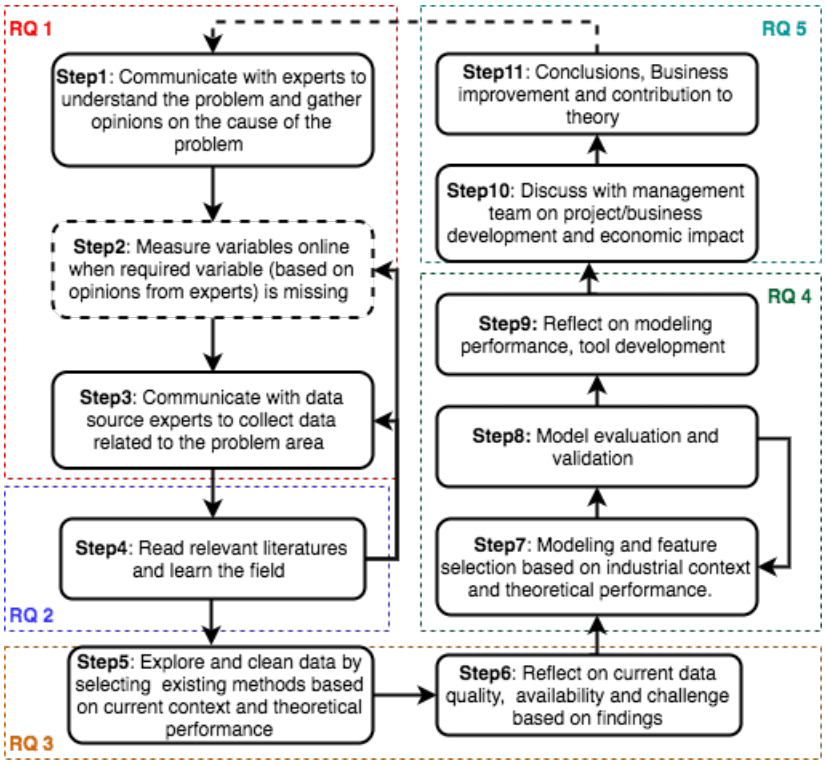


Figure 1.4: Method

### 1.3.5 Report structure

Each chapter of this report will be correspond to multiple sub questions. Chapter 1 and 2 answer research questions 1a - 1d. They presenting the background information, methodology and current business understanding. Chapter 3 answers research question 2, regarding current theoretical status of the problem. Chapter 4 will answer research question 3 data will be cleaned and explored based literature and industrial context for the first impression some insight and opportunities can already

be discovered by the end of this chapter. Research question 4 corresponds to chapter 5, models are built and compared. Modeling results are analyzed and validated. In chapter 6, conclusion are drawn to answer research question 5 together with the main research question. Summarized report structure can be found in Table 1.1.

Table 1.1: Report structure

| **Chapter** | **Research Questions** |
|---|---|
| 1.Introduction | 1a |
| 3.Data Collection | 1b-1d |
| 4.State of art | 2a-2d |
| 5.Data Exploration | 3a-3b |
| 6.Modeling and evaluation | 4a-4c |
| 7&8.Conclusion and recommendation | 5a-5d and main question |

CHAPTER
# TWO

# BACKGROUND KNOWLEDGE

Techniques that are used in this project regarding data pre-processing and modeling are explained on a high level in this chapter. Each section corresponds to one technique.

## 2.1 Principle component analysis

Principle component analysis is a dimensional reduction technique that project high dimensional data to different directions into vectors. The total number of data dimensions is the total number of principle components where the first principle component representing the most variance of the data set and the second principle component representing the second largest variance etc. A simple figure example can be found in Figure 2.1. This technique is used in this project to select the variables that represent the most variance of the whole variables. This is done by calculating the correlation between the variables and the first principle components. The higher the correlations are, the more important the variable is.

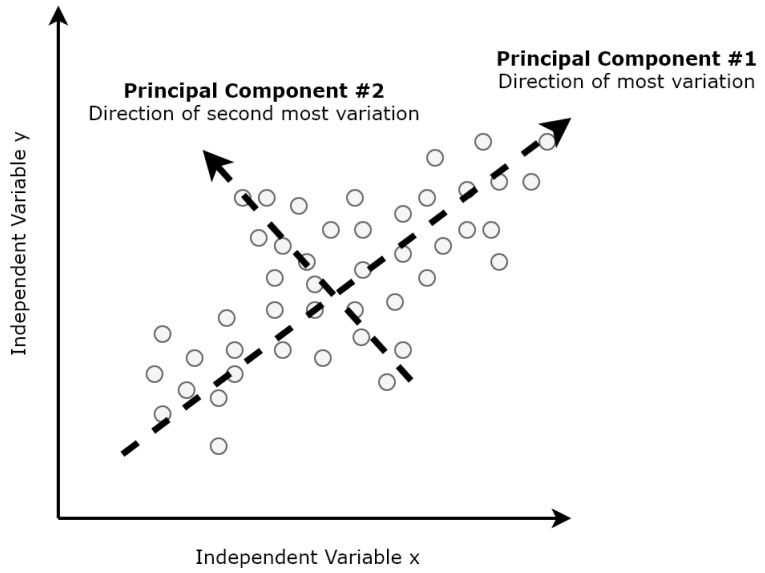


Figure 2.1: Principle component analysis

## 2.2 Artificial Neural Network

Artificial Neural Network is an interconnected system that can learn from samples. It is inspired by biological neural network. An example graph is shown in Figure 2.2. Each neural network consists of an input layer, output layer, hidden layer and biased nodes. Each node has its activation function and on each neuron there is an assigned weight. In our case, the initial weights are chosen at random. The weights are updated each time it get through a node with activation function. The activation function we use here is the following logistic function:

$$Sigmoid(z) = \frac{1}{1 + e^{-z}} \tag{2.1}$$

Sigmoid function is widely used because it always return a value that is between 0 and 1 thus it is a good representative of probability used for binary step function. A binary step function means that if the value is above a certain value known as the threshold, the output is activated, otherwise it is not activated. The final prediction value is the sum of the weights multiplied by the corresponding input

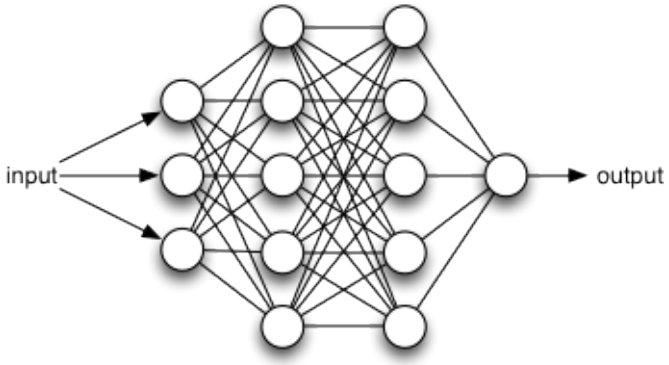


Figure 2.2: Artificial Neural Network

plus bias value formulated as follows:

$$Prediction = \sum (weight * input) + bias \tag{2.2}$$

## 2.3 Random Forest

Random forest is an extension of decision tree. The model aggregates the prediction made by multiple decision trees of varying depth. Each tree is trained on a subset of the data set. The portion of samples that were left out all named Out-Of-Bag data set which is used for the model to evaluate itself. Random forest in R deciding on the criteria to split a tree by measuring the impurity produced by each feature. The impurity is indicated by Gini index or entropy. The prediction result is produced by taking the average of the predictions made by each decision tree in the forest.

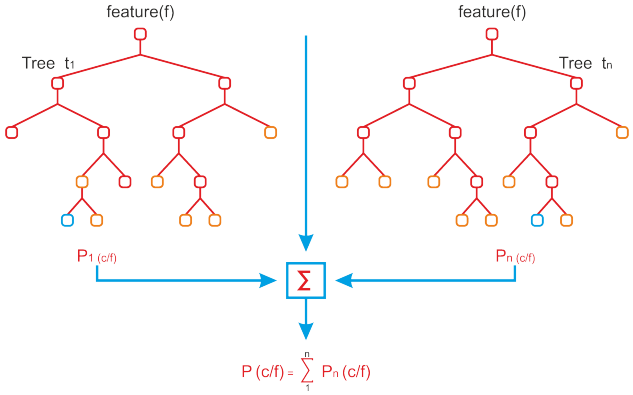


Figure 2.3: Random Forest

## 2.4 Partial Least Squared Regression

Partial Least Squared Regression (PLSR) is a statistical method that finds a linear regression model by projecting the independent variables and dependent variables to a new space. It belongs to the Partial Least Squared (PLS) models family. According to (Heberger, 2008), PLS model is a bi-linear method where information in the original data set X is projected into a small number of latent variables to ensure that the first components are those that are most relevant for predicting Y variables. This make it close to the idea of principle component analysis and principle component regression. The mathematics formulation of PLS model is the following:

$$X = T * P^T + E \tag{2.3}$$

$$Y = U * Q^T + F \tag{2.4}$$

Where X is independent variable matrix, Y is dependent variable matrix. T and U are projections of X and Y. P and Q are orthogonal loading matrices. E and F are the error terms. For detailed algorithm of PLSR in R, one can refer to (Helge Mevik and Wehrens, 2007) for further reading.

CHAPTER
# THREE

# DATA COLLECTION

In this chapter we will introduce the data source and parameters that is currently exist and potentially usable for this project. Data are primarily cleaned and integrated for first glance. Investigation on the meaning of the data has been conducted. In addition, part of the relation among parameters has been checked to verify the quality of the data source. A full data dictionary can be found in appendix A. Four separated data logger systems are currently logging data for the Galv line namely are Set up sheet(section 2.1), Data warehouse(section 2.2), IBA(section 2.3) and EMASS(section 2.4). Data integration tool is introduced in section 2.5.

## 3.1 Set up sheet

Setup sheet is the only data source where data are manually measured and logged. It is used for engineers to track parameters of maintenance cycles. The data are component related data such as roll diameter, whether the bush is new when installed and whether the sleeves are coated. The bush wear has been recorded after each campaign (approximately every 4 weeks). By the end of the data collection phase 6 samples of bush wear measurement is available from 15/05/2019 to 12/09/2019. By the end of the project 3 additional samples becomes available that are used for validation purpose. 15 variables are logged including date and time. The data logged on set up sheet are not continuous and need to be further processed into continuous data points before implementing. Furthermore, as the data are logged manually, human errors are not avoidable especially the bush wear measurement that is being used as target variable.

## 3.2 Data warehouse

Data warehouse is logging procurement and product property related data. Two interface within data warehouse called "SIMPPS" and "NEMO" has been investigated as these two interface has been suggested the most relevant by domain expert. Data from 15-05-2019 to 12-09-2019 has been extracted corresponding to the maintenance cycles. Respectively 95 variables and 110 variables are extracted and they are with different number of observations, 4232 observations in "SIMPPS" and 20232 observations in "NEMO". As there are some parameters with no data logged at all, or only few observations are logged, we have excluded them and only leave the parameters with continuously logged value. In total 93 variables are left for further investigation. After investigated the meanings of these variables, we select the variables that eventually used for modeling based on literature survey findings.

Time spots from different interface are logged differently only very few data are logged at the same time. Some parameters share the same meaning but with different names such as coil width. There are in total 4 parameters that indicate coil width. They are however can't be interpreted logically as the most of the finished coil width are logged larger than the ordered and received coil width from supplier. This according to firm expert is not logical as the coil are stretched during the process thus the width should be narrower. Conclusion can be drawn that if it is not the coil width that is logged wrong, it is then the coil ID was messed up. As this flaw is happening in a large margin and the fact that the widths data are all sharing the same trend the possibility that the coil IDs are logged wrong is low. This further reflects the data quality of data warehouse has a large space for improvement. In the end coil length, scrape length and surface area related variables are collected and used for prediction purpose. The reasoning for this is explained in Chapter 6

## 3.3 IBA

Data from IBA are system related corresponds to the the air-knives and the bath. The data are coming from sensors that has been installed within the system. We have extracted all the parameters available from the IBA starting from January 2019 to September 2019. The data file is over 10 GB in text file format.

As a first glance, we discovered that from 01-01-2019 to 17-03-2019, the data weren't logged correctly, as the data are either logged as "1" or "nan"(not a number). From 17-03-2019, the data are all logged in numerical format. 12 data points are logged per minute and that is the main reason why the data file is huge. Moreover, 694 variables are logged representing various blocks in the control loop. According to the domain expert, only variables that marked as input are relevant for the air-knife behaviors. Base on that, 81 variables are left for further investigations. 3,119,156 observations are logged from March to September 2019. As a preliminary judgement, there are few missing data points each parameter and few parameters with the same data but different labels. These quality issues will not effect the analysis as they can be easily cleaned.

## 3.4 EMASS

EMASS system is a brand new system that has just been implemented from January 2019 to stabilise the strip. The system logged features from the strip such as strip width, length and vibrations. The system current are also logged which indicates how hard the system works. 16 sensors exist on EMass with 8 sensors on each side of the strip logging distances of the strip to the sensor. Currents in the sensors are also logged. Data files per day are logged into 7Z file format. Thus these data files has to be compressed into a folder before opening. After open the folder, there are approximately 200 text files were logged per day. There are in total 5 million to 6 billion observations per day and the total data size in EMASS is 43GB. Thousands of data points are logged per minute with inconsistent logging intervals. More over, the operators shut down the system every 12 hours to clean the machine. Depending on the shut down time span, it make sense that the total number of observations per day differs.

## 3.5 Data cleaning and integration

Several data cleaning tools has been tried out to clean the data especially for data from EMASS system, because the data size is the largest of all data source. Open refine is one of the cleaning tools that was tried out. It is a tool designed especially for messy data. It does work well with small data

sets, however, the memory space is not large enough to handle data from EMASS, not even half day data. For the same reason other data cleaning tool wasn't good enough for our case either.

In the end data are integrated from different data source using statistical software R. Specific variables from Set up sheet, Data Warehouse and IBA are selected for predictive model building. Detailed data reading, selection and integration process can be found in Chapter 5 and Chapter 6.

## 3.6 Conclusion

To conclude the report so far, the first research questions (question 1a-1d) is answered as follows:

1a. *What is the current maintenance process in Tata Steel Shotton?*
In general maintenance decisions have been made based on experience in Tata Steel Shotton. Specifically, the current maintenance cycle of the pot gear, current maintenance cycle is approximately four weeks when there are no other faults such as strip break occurs. At the end of every four weeks, the line will be shut down and the pot gear will be replaced even if there hasn't been any failure. Engineers has been trying out larger bush diameters to prolong the maintenance cycle.

1b-c. *What are the existing data that ready for collection and what what are the meaning of the existing data?*
In total four separate systems are in use to collect data from different perspective. The four data sources are: Set up sheet, Data warehouse, IBA and EMASS. Set up sheet records data regarding component properties such as roll diameter, roll condition before in line and bush condition. Bush wear data are logged after each maintenance cycle from 15-05-2019. The data in set up sheet are logged manually thus human measurement errors are hard to avoid and should be considered when further analyzing

Data warehouse records mainly procurement related data, such as the product the customer ordered, the product delivered, the team ID and batch ID etc. The data from different interface in data warehouse are not logged at the same time point. Some parameters such as "coil width" appear repetitively but can't be interpreted logically, for example, the finished coil width is supposed to be narrower than ordered and received coil width but they are logged larger. Such phenomenon to certain extent reflects the poor data quality in data warehouse.

1d. *How can the data from different sources be integrated?*
Statistical software "R" has been used for data reading, cleaning and integration. Because R is capable of handling the data size from EMASS system while the rest of the tools such as OpenRefine has been tried out but failed to do so.

CHAPTER
# FOUR

# STATE OF ART

Over fifty papers has been reviewed in this chapter to present a view on the current research status in predictive maintenance in terms of component wear. Section 3.1 explores the general position of predictive maintenance in industry 4.0 concept. Section 3.2 explores physics behaviour of metal to metal contact wear to extract insight on the cause of wearing. Section 3.3 presents a broad review of data driven methods that have been used to predict component wear. Section 3.4 is focused on techniques for independent variables selections and feature extraction. The required data structure for modeling has also been reviewed. Current research gaps in predictive maintenance in component wear has been presented in section 3.5.

## 4.1 Predictive maintenance in industry 4.0

To understand the positioning of predictive maintenance in industry 4.0 concept and the requirement of achieving industry 4.0, literature survey has been conducted to provide an overview of what to achieve in industry 4.0 and how can predictive maintenance assist in getting there. A categorical framework of manufacturing has been provided in (Qin, 2016). Smart Factory, Self-organized business sections, Smart product new customer ordering process, smart vehicle etc are all parts of the concept. Some researchers proposed a '5C' structure to guide the development of industry 4.0. The '5C' namely are 'Connection Level' focusing on hardware development for wireless and sensor network connection. 'Conversion Level' indicates the level of information discovery by data analytic. The 'Cyber Level' emphasizing the level of automation. The 'Cognition Level' is to be aware of problems early and thus this is where predictive maintenance fit in. As a result of all the levels mentioned, 'Configuration Level' aims at achieving intelligent production as the accomplishment of industry 4.0. It is evaluated in(P.O'Donovan, 2015) that industrial equipment maintenance activity account for over 30% of a facility annual operating cost and between 60% and 75% of machine lifetime cost. Benefits and advantages of developing comprehensive predictive maintenance through the concept 4.0 focus on remote monitoring and self-diagnosis function. The major challenges is addressed as machine sensoring and monitoring, predictive modeling, cloud solutions and intractability. (Zhang, 2019) conducted a survey for data-driven methods for predictive maintenance of industrial equipment. It addressed major steps of predictive maintenance using data driven methods namely: operational assessment, data acquisition, feature engineering and modeling. Logistic Regression(LR), Neural Network(NN), Support Vector Machine(SVM), Random Forest/Decision Tree(RF/DT), and Auto-Encoder(AE) have been investigated and compared based on the application and prediction power. The result shows that LR has the lowest complexity level and still can achieve 100% accuracy in some applications. SVM is good at classification task and also performs well in fault diagnosis. DT and RF are strong in

explanatory power and suitable when data set has high dimensions,however, it is prone to overfit. ANN, DNN and AE have achieved more than 97% accuracy in 80% literature. It also concludes that when using deep learning(DL) algorithms, deeper network architecture and higher dimensional feature vectors are more significant in improving the performance.

## 4.2 Mechanical contact wearing behaviour

As the goal of this project is mainly investigating the predictive maintenance on wearing condition, literature surveys have been done to explore the cause of mechanical degradation, especially metal to metal contact wear. In (Cubillo, 2016), metal-metal contact wearing behaviour is classified into two stages: adhesive wear and fatigue wear. Adhesive wear is due to surface contact that leads to material transfer or loss from surface(Bayer, 2004). Fatigue wear however, generates surfaces cracks that after a critical number of cycles resulting in a severe damage. Adhesive wear can be modeled with the Archard's law :

$$Volume[wear] = K * \frac{Load * Slidingdistance}{3 * Materialhardness} \tag{4.1}$$

Where K is wear coefficient that can be adjusted to fit more complex wear models. Fatigue wear however can be modeled in different ways based on friction coefficient and lubrication factor. The main cause of fatigue wear is the force taken per unit area. Modeling formula for bearing wear with lubricating factor can be found in (Ocak, 2007). Bearing degradation behaviour has been investigated in (Wang, 2017) where the behaviour has also been classified into two stages, the first stage has been simulated with a normal random variable $\alpha$ with mean $\mu$ and variance $\alpha^2$ the second stage of degradation has been simulated with Geometric Brownian motion. Analysis of variance(ANOVA) was used to determine the effects of the machine parameters of surface roughness and flank wear. Linear and quadratic regression were applied to predict the outcome of the experiment. (Kivak, 2014). Similarly, (Hwang, 2015) did experiments and numerical study of wear in cross roller thrust bearings. In which the first stage of wear was concluded as linear wear with the Archard's law. Failure such as spalling was concluded as non-linear wear. (Wang, 2014) applied enhanced particle filter to predict tool wear and the relationship between tool wear rate and the changes of applied load has been investigated which is only a modified differentiation equation of the the Archard's law. In addition, tooling force was found to be increased as the tool gets worn. Normal load is proportional to the wear width, thus normal load can be approximated as linear relationship of wear width.

Summarising the findings, mechanical contact wear, material hardness, load, sliding distance and lubrication factor cause wearing and determine wear volumes(B K N Rao et al, 2012). The wearing behaviour mostly are classified into two stages, the first stage is linear wear whereas the second stage is nonlinear. It is found that load changes are proportional to the wear, indicating that the wear width forms a linear relation with the force at the stage. The second stage is more complex and usually leads to failure.

## 4.3 Data driven methods for wearing prediction

Methods for predictive maintenance in terms of wearing condition can be generally classified into three categories (Tobon-Mejia, 2012), namely mathematical model based method, experience based method and data-driven method. Mathematical modeling based method produces good results, but requires deep knowledge on system functions, which is a viewed as a disadvantage as it is often

hard to obtain. Experienced based method makes decisions based on empirical events and often produces poor prediction results. Data driven based methods investigate the problem mainly from data perspective. Machine learning is often used to learn data patterns. Predictive maintenance using data driven method requires failure data as target variables and featured variables to describe input data(Gouriveau et al, 2013), however, failure data is often not available in industries as down times are prevented with the best efforts due to high down time cost.

Artificial neural network, support vector machine and random forest are the most implemented data driven methods. Besides, hidden Markov model(HMM) with strict data structures and reliable computing performance has also attracted lots of attentions. In recent years, many researchers have applied HMM in tool condition monitoring and achieved good results.(Liao, 2016) however, the data used for wearing prediction are mostly experimental data, which means machines are set up and run to failure for a number of times and data are collected by the sensors that installed on the machine. Thus critical parameters are fully obtained. Such ideal data sets are hard to get from real life.

Limited amount of study has concentrated on predictive maintenance with missing labels. (Zschech, 2019) investigated in prognostic model development with missing labels. It first uses unsupervised learning such as clustering techniques to create labels then use supervised learning namely Recurrent neural network(RNN) to predict future labels. In (Amruthnath, 2018) unsupervised learning has been investigated to detect early fault in predictive maintenance on experimental data. T statistics, k-means clustering, c-means clustering and hierarchical clustering are implemented and compared. The result confirms that unsupervised learning can detect faulty behaviour and clustering results are similar. (R.Langone, 2015) proposed a least square support vector machine(Ls-SVM) framework for maintenance strategy optimization based on real-time condition monitoring. It used both clustering (unsupervised learning) and a supervised leaning method namely nonlinear auto-regression(NAR), however, with different data set, and conclude that supervised learning can achieve better result and meanwhile it is more computational expensive. Two methods combined may result in the optimised maintenance cost configuration.

Deep learning is also starting to be implemented for wearing prediction. In (Martinez-Arellano, 2019), convolutional neural network(CNN) in combination of time series imaging are implemented to predict wearing that can effectively detect the wearing condition and location.

Other studies such as (Zhao, 2017) and (Zhong, 2016) proposed using correlation analysis among different sensor signals to detect early anomaly as they discovered that the distribution of sensors signals is likely to change when there is a faulty behaviour. Thus the correlation between sensors will change before anomaly truly happens. This way faulty behaviours can be detected in time for maintenance activities.

Evaluations of modeling performance are mostly based on prediction power. Classification models are evaluated with True positive(TF), False positive(FP) rate and accuracy(Li, 2004)&(Li, 2003)&(Susto, 2014). Regression models are often evaluated with mean squared error(MSE), residual and R squared value,(Wu, 2017). Some of the studies also put training time as one of the evaluation criteria.(Wu, 2017 and 2016). R square is calculated with the following formula:

$$R^2 = 1 - \frac{SumOfSquaresOfResiduals}{TotalSumofSquares} \tag{4.2}$$

Detailed calculations for Sum of Squares of Residuals and Total Sum of Squares can be found in Chapter 6.

## 4.4 Data pre-possessing

A critical overview of wearing prediction using artificial neural network has been presented in (B K N Rao et al, 2012) where articles from 1997 to 2011 has been reviewed. Different data prepossessing techniques that described in the article are dimensional reduction techniques, data fusion methods, statistical analysis and optimization algorithm for feature selection. Principle component analysis(PCA), multivariate methods etc are in the category of statistical analysis. PCA is the most implemented dimensional reduction technique.

Feature extraction and selection has been an active research area for deployment of artificial intelligence. In terms of prognostics, a dataset is expected to have a minimum sample size around 10 in order to perform data-driven modeling effectively.(Eker, 2012) Preferably, dataset for predictive maintenance should be able to describe several instances of the same fault in various different but similar components (Klein, University of Trier). In (Veltan, 2000) 72 wear volumes are generated in the datasets with experimental settings. PCA was used to reduce its dimension. In (Bolon-Canedo, 2011) two main approaches for feature extraction are presented as individual evaluation and subset evaluation. However it was claimed in the article that there is no best method. Efforts are focused on finding a good method for a specific problem area.

In (Javed, 2012) 315 wear measurements were produced with 16 features extracted from each measurement. In (Wang, 2015) 300 files were produced with each file corresponding to one cut. Force and vibrations in three directions are stored in the files. Pearson correlation coefficient is selected as a feature since, according to the author, a good feature should present consistent trend with wear propagation. Fisher's discriminant ratio is also a feature selection technique that was implemented in (Xie, 2019) to extract features from cutting force. Statistical feature was used and wearing conditions were given labels: 'initial wear', 'median wear' and 'severe wear'. Some studies even skipped feature extraction step such as (Zhao, 2017), raw sensory signals are directly put into a CNN for wearing prediction. However, normalization is commonly used for data reprocessing (Kolodziejczyk, 2010) to first put data into a same scale. Genetic programming is used as an optimization algorithm to choose the best features and/or describe wear volume (Kolodziejczyk, 2010). Statistical features are widely used in wear prediction.(Zschech, 2019) extract time domain features: peak value, root mean square, standard deviation, kurtosis value, cxest factor, clearance factor, impulse factor and shape factor.

A table summarizing essential variales used for wearing predictions can be found in Table 4.1. Majority of wearing predictions are based on vibration analysis that is distinguished between time domain and frequency domain. Time domain vibration analysis extract time series features such as peak, rms, kurtosis and clearance value whereas frequency domain deploys spectrum analysis, envelope analysis and high frequency resonance technique(Eker, 2012).

Vibration signals are used as indicator of the condition. Vibration of 0.012 is considered as a reference value to decide the change of tool.(Krishnakumar, 2015) Real time vibration based on structural damage was detected using one dimensional CNN in (Abdeljaber, 2017). The damage was detected and the location of the damaged joint was able to be detected. In (Y Shan, 2000) different prediction methods were applied to different bearing running stages to predict remaining life. Variables characterizing the state of deterioration are extracted for neural computation. Time domain vibration signals of rotating machinery with normal and defective bearings are used for wearing prediction in (Samania, 2001). Acoustic signals are used in(Xu 2002) and optimal statistical features are selected using a genetic algorithm. Current signature analysis are used in (O..Nel, 2006).

Table 4.1: Survey of essential variables

| Literature | Analyzed variables |
|---|---|
| Ocak 2007 | Vibration |
| Nectoux 2012 | Rotating Speed, Force, Temperature, Vibration |
| Krishnakumar 2015 | Vibration |
| Abdljaber 2017 | Vibration |
| Si et.al 2011 | Pressure, Temperature, Vibration, Moisture, Humidity, Loading, Speed, Oil |
| Liu and Mengel 1992 | Vibration peak amplitude, frequency domain and peak RMS |
| Alguindique 1993 | Vibration |
| Baillie and Mathew 1994 | Vibration |
| Aiordachioaie 1995 | Vibration |
| Wang 1996 | Data created by mathematical model of vibration signal, frequency spectrum |
| Samanta 2001 | Vibration |
| Peng Xu 2002 | Acoustic signals, DWT coefficient |
| Samanta 2003 | Vibration |
| O Nel 2006 | Current |
| Ghafari 2007 | Vibration |
| Li Yun 2008 | Frequency domain characteristics of vibration |
| Vijay 2011 | Vibration |

## 4.5 Gaps

Data cleaning and huge data set handling, according to (Bulletin 2000), should include: data analysis, definition of transformation workflow and mapping rules, schema-related data transformation, verification, transformation and back-flow of cleaned data. These steps are provided as data cleaning guide to detect errors and inconsistency, transforming data into standard format with least possible manual inspections and verify the correctness of data transformation. Ideally, the cleaned data should replace the data in the original source. The approaches for data cleaning have never been presented in literature regarding predictive maintenance. Some results in articles are based on huge sensor-based data sets, for example, data sets with size 4TB(Li 2013). But how these data sets are read and cleaned are not presented. However, there are some techniques to clean sensor data such as weighted moving average(Zhuang, 2007) but it is implemented especially on small samples. In real life, sensor-based data sets are often huge and difficult to read. Thus it should require a long time to implement these techniques. In this report we present some techniques used to read large data sets in a reasonable amount of time and discuss how to reduce the file size and integrate data in chapter 5

Comparison of prediction result using different variables and reasoning of using vibration data in comparison with others are missing in literature. Almost all the papers were using vibration data to predict wearing condition without reasoning. Although using vibration data did produce good prediction results it is worth investigating that when vibration data is not available, are there other alternative parameters that could potentially replace vibrations and be used to predict wearing condition? Few papers analyzed on force and some combined several parameters(Kolodziejczyk, 2010) but no comparison study in terms of the prediction power with different parameters were conducted. Thus in this report we will also do some comparison in terms of predictive maintenance using different parameters and explore differences among results.

There are limited amount of studies on real industry cases where there is a lack of failure and wear-

ing measurement. As mentioned before, most of the studies were conducted in labs, used data are experimental data. Most of the studies used multiple milling machines and let them run to failure and meanwhile the sensors installed on the milling machines are able to measure the wearing on the milling point, log vibrations and log forces from different dimensions. However, this ideal situation almost never happens in real industry setting. Take the cased used in this study as an example, there are no sensors can be installed around the bush as it is in a zinc pot. Thus vibrations and force both are not available, not to mention that the wearing measures are only able to be measured after every campaign. Some literature combines unsupervised and supervised technique to simulate target, but the result is hard to validate as there are only simulated labels exist. This study investigates the following: to what extent can predictive maintenance on wearing conditions be done with only partly available wearing measures and what other feasible predictive maintenance activities can be done with current available data.

## 4.6 Conclusion

By this end research question 2 is able to be answered to conclude predictive maintenance using data driven methods.

2a. Which data driven models have been used to predict metal contact wear in industries?
Data driven methods are mostly machine learning models. The most implemented machine learning models are random forest, support vector machine and artificial neural network. They all produce good prediction results on predicting failure(classification) or remaining use of life(regression). Artificial neural network are the most implemented and is often modified into different structures. Other statistical based models can be considered as the overlap of empirical and data driven methods such as detecting correlation changes and distributions from historical data.

2b. What are the steps that has been used in literature in terms of data prepossessing and model building?
Classical method for predictive analytics is data collection, data cleaning, feature extraction and model building. Data sample required for model building in predictive analytics are samples covering variety of cases, with many predictive variables corresponding to one target variable.

Limited amount of studies concentrate on missing target issue as limited amount of studies are based on real life industrial data. Some studies compare different clustering techniques that are unsupervised machine learning models such that no target variables are required. However it is also falls out of predictive analytic category as unsupervised machine learning has descriptive nature. Some studies combine unsupervised learning and supervised learning, specifically using clustering techniques to detect abnormal behaviour and making different labels for training set and then training predictive models using training set with simulated labels. The limitation of this method is that it is hard to evaluate the predictive power as it can only be evaluated with simulated labels.

Feature extraction is an active research area as features extracted are likely to influence prediction result. Statistical features are the most widely used. Many algorithms are suitable to select optimal features such as principle component analysis and genetic algorithm. It has been concluded that no best technique exist for feature extraction it all depends on specific situations.

2c. What are the measures to evaluate predictive power of the models?

Predictive maintenance using data driven methods can be categorized into two categories as follows:
-Classification: Predict system/component failure, the labels then can be set as: fail or not fail. Prediction power evaluation for classification models are True Positive rate, False Positive Rate and Accuracy measure.
-Regression: Predict system/component remaining use of life, wearing millimeters etc. Prediction is a continuous variable. Prediction power is evaluated using MSE(mean squared error) and R squared value.
Some studies also take training time as one model selection criteria.

2d. Based on prediction power and current context which models are the most suitable for predictive maintenance?
When data availability is not a problem, random forest, support vector machine and neural network all show good prediction power. Random forest and support vector machine have their advantages that they are computational less expensive. Whereas neural network is more complex but then it is more flexible to suit different systems and conditions. It can even be used as an automatic feature extraction model and also powerful in deep learning for more detailed prediction such as recognizing the specific wearing location.

2e. What are the gaps and limitations of current literature on predictive maintenance in industry?
Current studies on predictive maintenance barely describe the size of data set and how to read huge data set. Most of the data cleaning techniques are tested on small data samples which is far from real life situation. As in real life, sensor data are logging at a high frequency and the data file are huge and not even be able to read by software. Not to mention implementing techniques to clean them.

Most of the predictive maintenance on wearing conditions only use one parameter namely vibration signals as it is always produce good prediction result. Few studies uses other parameters based on physical property of the component. However, sometimes none of these parameters are available, thus alternative parameters to predict bush wear are worth investigating.

Speaking of real life situation, almost all the literature that have been investigated in this study are conducted in laboratory settings. The data are experimental and ideal as sensors are installed and able to capture all relevant features while the machines are run to failure. That is one of the reasons that prediction accuracy is high. Studies based on real industry data will face more challenges as essential parameters may not be available.

CHAPTER
# FIVE

# DATA EXPLORATION

In this chapter, data cleaning and prepossessing techniques are presented in section 4.1 and 4.2. Then we combine principle component analysis and unsupervised learning techniques to explore the existing data further without considering target labels (wear measurements) in section 4.4, 4.5 and 4.6. As a result, using existing sensor data, other predictive maintenance activities can already been done even without predicting bush wear condition.

## 5.1 Reading Large data sets

It has been addressed that we have 4 separated data sources namely: IBA, EMASS, Setup sheet and data warehouse. Among which, IBA and EMASS are logging sensor-based data whereas set up sheet and data warehouse are logging component and product related data thus the data are logged manually. As sensor based data source in this case is logging much more frequently than human, IBA and EMASS data sizes are much larger than the other two and the data format is also more inconsistent, thus requiring more time when cleaning the data.

Data cleaning process is more like data engineering, as the cleaning algorithm should be built based on the specific raw data structure and problem will keep incurring until the data are fully structured. As a general guideline for data cleaning: it may take a long time only to read the data due to memory limit of PC/Software. Software used here is R as mentioned before and it can read 2GB to 4GB data at a time with reasonable speed. Maximum 2 billion indices can be store in memory. It is important to split the file into small pieces and clean them separately. When the sub-files are still not able to be read because it is over-sized, it is important to do operations before reading the file such as skipping lines while reading to keep the data size fit for the software.

IBA data from 17/03/2019 to 13/09/2019 stores 3,119,156 observations and 687 variables in one single text file. It wasn't feasible to read it as a whole so the text file was split into 10 sub-files using R and read again. After deducting the variables into 27, as explained before, it becomes feasible to read them as a whole. Time to read the data set is around 15 minutes and after reading the data set, simple cleaning technique is hard to implement on the data due to the limitation in the memory of vector.

EMASS data which is the largest of all stores over 5 million observations per day. At some days over 6 billion observations. The EMASS is logging over 200 text files per day and zip them automatically. R is able to read a 6-billion observation text file in one and a half hour but no additional operations

are able to be executed. As the data size is already large for one-day data and there are thousands of observations within one minute, the data were read by reading one row every 80-500 rows depending on the data size that day.

Setup sheets and data warehouse are logged manually by human, thus logging frequency aren't as high as that of the sensor-based data. 20232 observations were read from data warehouse using R and the parameters logged in setup sheet were manually transferred into Excel files.

## 5.2 Data cleaning

Table 5.1: Variable norms

| Variable | Norm | Data Source |
|---|---|---|
| Al content | 0.18 - 0.23 | IBA |
| Tension | 1577( Reference value) | IBA |
| Roll Diameters | 560mm-600mm | EMASS and Set up sheet |

Efforts are mainly made to make the data format from different data source consistent. Since each data source contains a different data format, there was no general algorithm for cleaning all data sources. Code for cleaning can be found in Appendix C. Rows contain missing data are deleted. Duplicative variables were detected by taking average value per day for variables with the same labels and then deduct each other. By plotting the deducted values, a horizontal zero line is shown when duplicative variables were detected. Logical check was conducted to specific variables to make sure the logged values are within its norm as shown in Table 5.1. It is surprising to see that Al content data are never logged within its norm which means that the data are logged wrongly. By summarizing the variable, data quality can be intuitively reflected. As an example, a summary of 'Tension' variable can be found in Figure 5.1. We can see that neither mean or median value falls below the reference which means in general, the tension is higher than the given norm. Same analysis was conducted on Roll Diameters with given norm from a domain expert and we found that roll diameter is roughly around the norm.

```
> summary(Tension$IBA.input.stripTension)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  -6022    1539    1850    1762    2161    2725    5920
```

Figure 5.1: Summary Tension

## 5.3 Data prepossessing

For sake of simplicity, principle component analysis(PCA) was conducted on each data source in order to reduce data dimension. All variables were normalized before doing PCA. Only continuous variables were included in this study because PCA can only be implemented on continuous variables. Some variables from data warehouse are selected due to their explanatory power.

## 5.4 Variable selections using PCA

32 variables from Emass has been reduced to 9 variables as shown in Table 5.2, namely Distance.A2-A5, Distance B2-B5 and Current B4. The first 13 principle components were taken into account as they represent 82.5% of the variance of all 32 principle components. We select variables based on

the correlations between variables and principle components and select the variables only when the absolute values of correlations to any of the principle component PC1 - PC13 are larger than 0.8. Similar operations has been done on IBA variables. We select the first 6 principle components as they

Table 5.2: Variables selected from EMASS

| Variable | Correlations to principle components |
|---|---|
| Distance.A2 | 0.96 |
| Distance.B2 | 0.96 |
| Distance.A3 | 0.94 |
| Distance.B3 | 0.86 |
| Distance.A5 | 0.96 |
| Distance.B4 | 0.96 |
| Distance.A5 | 0.96 |
| Distance.B5 | 0.96 |
| Current.B4 | 0.80 |

represents 81% of the total variance. We select variables with absolute correlation to any of the first six principle components larger than 0.8 as shown in Table 5.3. Statistical features are extracted from

Table 5.3: Variables selected from IBA

| Variable | Correlations to principle components |
|---|---|
| input.controlPress.TOP. | 0.85 |
| input.controlPress.BOT. | 0.85 |
| input.headerPressRaw.TOP. | 0.85 |
| input.headerPressRaw.BOT. | 0.85 |
| input.headerPress.TOP. | 0.85 |
| input.headerPress.BOT. | 0.85 |
| input.coatMass.TOP. | -0.84 |
| input.coatMass.BOT. | -0.83 |
| input.potTemp | 0.99 |
| input.corrRollPos.RIGHT. | -0.92 |
| input.potAlContent | -0.92 |

these selected variables, namely: Mean, Median Minimum, Maximum, Root mean square(RMS), kurtosis, skewness and standard deviation. These statistical features are chosen because they are widely used in literature. Kurtosis describes whether the distribution contains extreme values. Skewness describes the extent to which a distribution differs from a normal distribution in terms of asymmetry.

## 5.5 Variables with explanatory power

Since the ultimate goal of this project is exploring the possible ways to predict bush wear condition and we have summarized that the wearing is caused by "sliding distance", "force", "Lubrication factor" and "material hardness", some existing variables are selected for their physics-based explanatory power. New variables were created using existing variables from data warehouse and setup sheet as shown in Table 5.4.

Table 5.4: Created Variables

| | |
|---|---|
| Total Run Length | Data source |
| Total Scrape Length | Data Warehouse |
| Total Surface | Data Warehouse |
| Total Number of turn | Data Warehouse and Set up sheet |

"Total Run Length", "Total Scrape Length" and "Total Surface" were created by summing up the finished coil length, scrape length and finished coil surface values for each day. Total number of turn is calculated with the following formula:

$$NumTurn = \frac{TotalRunlength}{SinkRollDiameter * \pi} \tag{5.1}$$

## 5.6 Unsupervised learning

To explore hidden patterns and descriptive information within chosen variables, unsupervised learning techniques are used. Unsupervised learning techniques are descriptive models that are able to detect faulty behaviour by clustering data based on certain criteria(Amruthnath 2018). K-means clustering and hierarchical clustering are two of the basic clustering techniques that are used for online monitoring and fault detection purpose. Especially, K-means clustering is widely used as it is less computational expensive compared to other clustering techniques. As mentioned before, eight statistical features are extracted from each variable thus PCA is implemented again to reduce the increased dimension. Principle components are chosen only when it is representing over 90% of the variance.

### 5.6.1 Hierarchical clustering

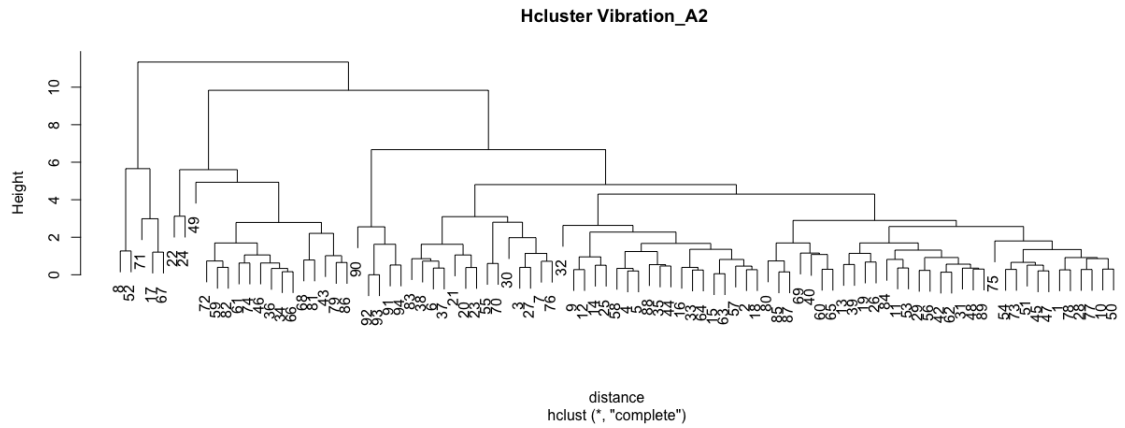


Figure 5.2: Hierarchical clustering Distance.A2

The most representative hierarchical clustering results are shown in Figure 5.2, Figure 5.3 and Figure 5.4. The reason why only these three clustering results are showing here is because clustering results of other variables are similar and lead to the same conclusion. Hierarchical clustering result shows that Distance.A2, Distance.B2, Distance.B3, and Distance.A5 all separate indices 8, 17, 52, 67, 71 from the rest of the data which indicates that abnormal events happened on the corresponding dates: 14/06/19, 24/06/19, 01/08/19, 16/08/19, 20/08/19. Hierarchical clustering on current B4

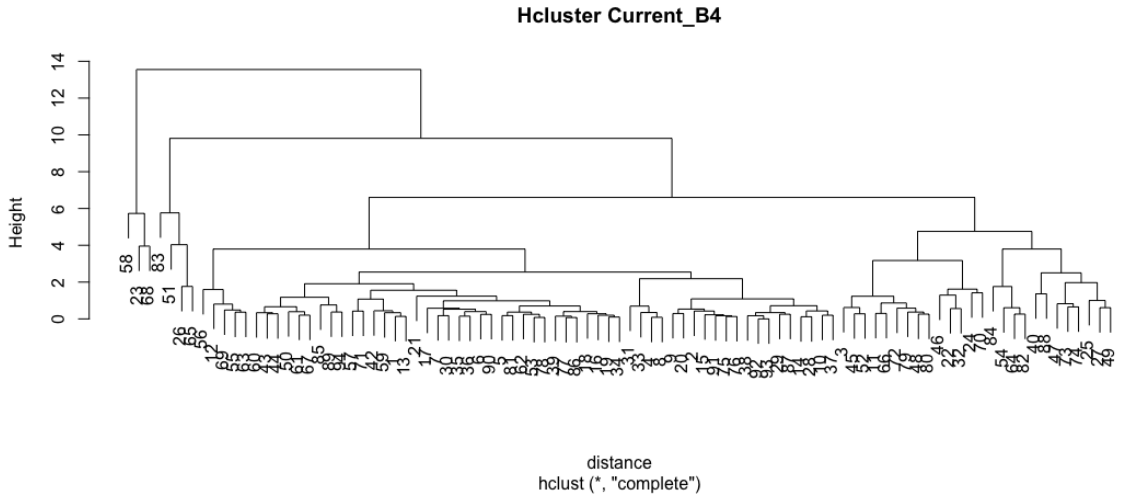


Figure 5.3: Hierarchical clustering Distance.A2

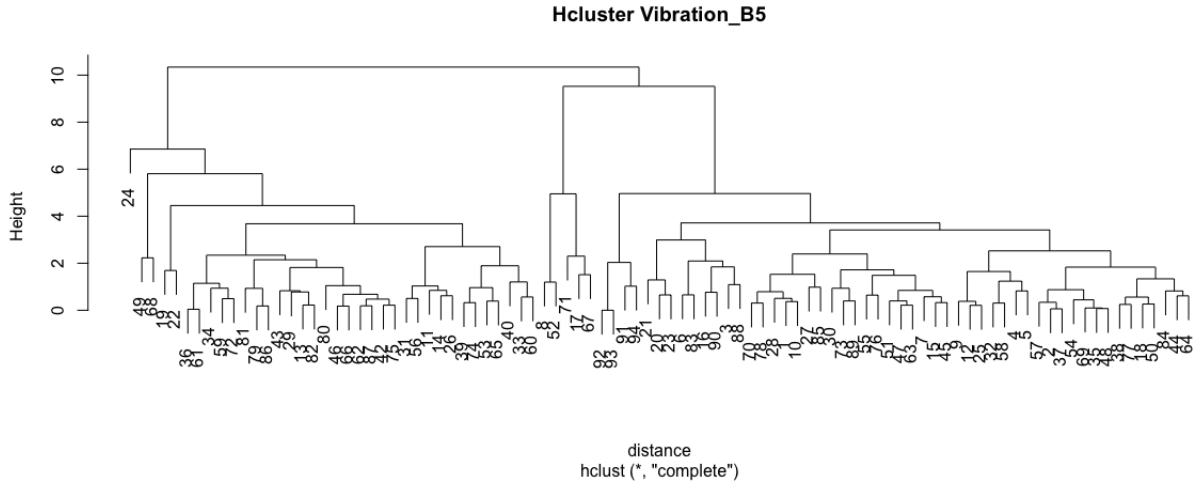


Figure 5.4: Hierarchical clustering Distance.A2

clustered out index 23, 58, 68, corresponding to dates: 30/06/19, 07/08/19, 17/08/19 respectively. Distance B5 clusters out index 24 corresponding to 01/07/19. Hierarchical clustering on other variables aren't explicitly clustering out certain dates, meaning abnormal event has been detected more in detail which will not contribute much to cost saving in preventive maintenance due to high frequency of maintenance activity.

The faulty behaviours detected on IBA variables with the same procedures differ among the variables. On some variables, both clustering results shows massive abnormal events, indicating that a significant amount of maintenance needs to be conducted. Extreme events have been detected from CoatMass.BOT on index 6, 109, 112, 162, 167, 168 corresponding to dates: 2019-03-22, 2019-07-03, 2019-07-06, 2019-08-25, 2019-08-26, 2019-08-30, 2019-08-31. Hcluster indictates from correcting roll position: index 21, 22, 26, 33, 34, 51 are abnormal, corresponding to dates: 2019-04-06, 2019-04-07,2019-04-11, 2019-04-18, 2019-04-19, 2019-05-06. etc. Clustering results can be found in Table 5.6.

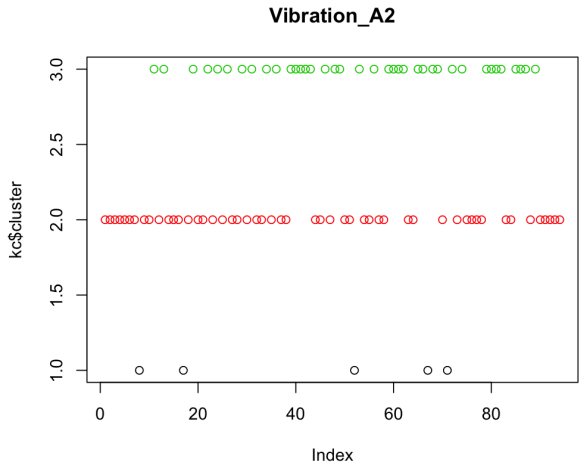


Figure 5.5: K-means clustering on Distance.A2

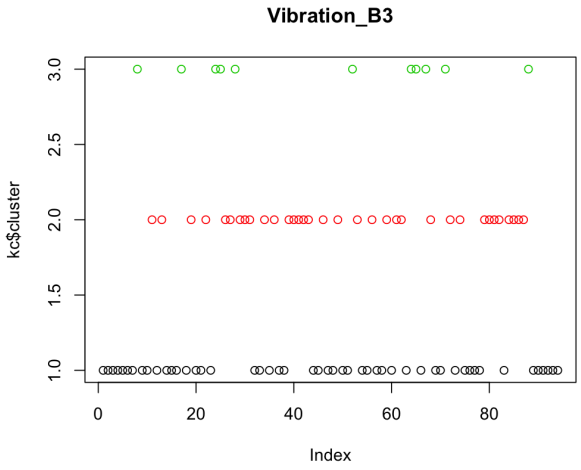


Figure 5.6: K-means clustering on Distance.B3

### 5.6.2 K-means clustering

Again, only the most representative clustering results are shown here. Distance.A2, Distance.B2, Distance.A5 and Distance.B4 were clustered out the same index as hierarchical clustering did corresponding to dates: 14/06/19, 24/06/19, 01/08/19, 16/08/19, 20/08/19. K-means clustering on Distance.B3 shows more abnormal events that happened on dates: 14/06/19, 24/06/19, 01/07/19, 02/07/19, 05/07/19, 01/08/19, 13/08/19, 14/08/19, 16/08/19, 20/08/19 and 06/09/19. For Distance.B5, K-means clustering shows the same result as hierarchical clustering that abnormal event has happened on 01/07/19. Both clustering techniques gives the same results on Current B4 where faulty behaviour has been detected on 30/06/19, 07/08/19 and 17/08/19. Two representative K-means result graph can be found in Figure B.11 and Figure B.60.

Both hierarchical clustering and K-means clustering have their pro and cons. We will reflect them here. The advantage of both techniques is that they are easy to compute and require short running time. However, hierarchical clustering can only make two clusters, which is not sufficient for clustering out extreme events. K-means clustering requires the number of clusters to be defined by human. The main disadvantage of K-means cluster is that, the clustering results are strongly dependent on the initial criteria, which means running the algorithm multiple times the result might change.

### 5.6.3 Validation

Unsupervised learning results are validated by looking into events recorded in shift report. After checking each abnormal date from clustering and the corresponding comments in the shift report we can see from Figure 5.7 that all dates that have been clustered out did have either planned delay or unplanned issues except for 12/04/2019 where nothing abnormal was happening. This further confirms that clustering technique is able to detect abnormal events. However, we also discovered that there are lots of delays that weren't recognized by unsupervised learning. Some of the delays happened at the entry and exit of the line which weren't detected probably because the data we have is only around the pot gear section. Some other delays were caused by operating error, in other words, human error or decision based delay; they were not detected because those were not system related delays. Apart from those mentioned before there were still some delays that should have been detected but weren't clustered out.

| Source | Date | Planned Delay | Unplanned issues | Comments |
|---|---|---|---|---|
| corrRollPos&ControlPressTOP | 06/04/2019 | | | **Continuation of strip break** |
| corrRollPos | 07/04/2019 | | | The CWT. Gauge failed with a thermal fault |
| corrRollPos | 11/04/2019 | | | D/s knife skew tripped start of shift tech reset ok. Knives wouldn't do a knife adaption until skew was reset. System then in auto. |
| PotTemp & AL Content | 12/04/2019 | | | no issue at all |
| PotTemp | 17/04/2019 | | | Coil 483815 left off due to damage on the outer 15 laps. |
| corrRollPos | 18/04/2019 | | | Line on o/s of strip on 1290 wides traced to ac . Roll changed at weld coil 151815/1 to go to No1 for levelling |
| corrRollPos | 19/04/2019 | | | PCM reported B1 gearbox knocking, tensions adjusted. Monitored throughout shift and has quietened down |
| PotTemp & AL Content | 26/04/2019 | | | **OP-unplanned 0.05hrs** |
| corrRollPos & ControlPressTOP | 06/05/2019 | | | **Continuation of strip break** |
| PotTemp | 10/05/2019 | | | Several coils left off |
| AL Content | 26/05/2019 | | | Product change to Galfan |
| Distance.A2-A5,B2,B4 & Distance.B3 | 14/06/2019 | | | Product change to Galfan 5Hrs- 14Mins |
| PotTemp & Distance.A2-A5,B2,B4 & Distance.B3 | 24/06/2019 | | | Product change to Galfan- 4 Hrs |
| Current.B4 | 30/06/2019 | | | Coating weight gauge went off over temp |
| Distance.B5 and B3 | 01/07/2019 | | | Scrap Coil Knife line/cleaning marks |
| Distance.B3 | 02/07/2019 | | | The barrels of zinc skimming's / dust have been taken down to the entry section weighed and banded 770kg gross weight. |
| CoatMass.BOT | 03/07/2019 | | | Product change to Galfan 5:43 Hrs |
| Distance.B3 | 05/07/2019 | | | Furnace cooling water temps a/c current high atmospheric temps and suspected blocked heat exchangers |
| CoatMass.BOT | 06/07/2019 | | | **Slack bore delay causing strip wander. 1Hr 37Min** |
| HeaderPressRawBot &ControlPressTOP | 18/07/2019 | | | Planned line stop |
| Distance.A2-A5,B2,B4 & Distance.B3 | 01/08/2019 | | | **line stopped - cold run - 07.30 --13.46 - line started ok** |
| ControlPressTOP | 04/08/2019 | | | Coating weight gauge curve has been updated but still reading out. |
| Current.B4 | 07/08/2019 | | | IBC of TF5 delivered, it is down the entry on a bund. |
| Distance.B3 | 13/08/2019 | | | Back knife cleaner u/s twisting and jamming on op side |
| Distance.B3 | 14/08/2019 | | | **Suddenly the Hoffman blowers stopped - unsure why 16.04 Op stopped the line - tech called---16.07 Line restarted** |
| Distance.A2-A5,B2,B4 & Distance.B3 | 16/08/2019 | | | Planned product change |
| Current.B4 | 17/08/2019 | | | **09.30 Line stopped--10.01 Line started** |
| Distance.A2-A5,B2,B4 & Distance.B3 | 20/08/2019 | | | B_rota_Linestop Removing coil |
| CoatMass.BOT | 25/08/2019 | | | Operator side back knife cleaner blade has become away from housing and is now bent touching the baffle frame its now isolated |
| CoatMass.BOT | 26/08/2019 | | | **05.24 Line stopped---05.28 Line restarted** |
| CoatMass.BOT | 30/08/2019 | | | **Hot bridle roll 13 tripped out causing the line to come to a stop. Length of delay - 54 mins** |
| CoatMass.BOT | 31/08/2019 | | | **03:55 run out of time. Length of delay: 4 mins.** |
| Distance.B3 | 06/09/2019 | | | Leveller stand 2 bottom cassette changed- linear line on strip. |
| PotTemp | 13/09/2019 | | | Baffles failed to open cause damage to the strip |

Figure 5.7: Validation for clustering results

## 5.7 Correlation and regression

```
> summary(linearMod)

Call:
lm(formula = `Total NumTurn` + sd_T ~ label, data = Cycle)

Residuals:
cycle1 cycle2 cycle3 cycle4 cycle5
 48310 -33879 -66198  -5847  57614

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   518575      87606   5.919  0.00963 **
label         -16802       4573  -3.674  0.03490 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 61150 on 3 degrees of freedom
Multiple R-squared:  0.8182,    Adjusted R-squared:  0.7576
F-statistic:  13.5 on 1 and 3 DF,  p-value: 0.0349
```

Figure 5.8: linear regression total number of turn, tension std and remaining bush width

To discover some correlation between parameters with explanatory power and the bush wear we created a correlation table(Table 5.5) that shows the correlation between variable and the minimum bush wear from both sides. We only explore the correlation between run length and tension related variables and the remaining bush width because of the Archard's law we've discovered in the previous chapter. It is shown in the Archard's law that the wear volume is correlated with load, sliding distance and material hardness.

It is worth mention that the correlation test is only based on 5 samples as we only have 5 bush measurements thus the result lacks credibility. However, based on the 5 samples we have, we can see from the table that total run length, total surface, total length and total number of turn all have

| Variables | Correlation Label(Bush remaining width) |
|---|---|
| Total NumTurn | -0.90 |
| Scrape Length | -0.93 |
| Total Surface | -0.91 |
| Total Length | -0.91 |
| Maximum Tension | -0.23 |
| Mean Tension | 0.61 |
| Minimum Tension | 0.39 |
| Median Tension | 0.47 |
| Standard deviation Tension | 0.76 |
| RMS Tension | 0.66 |

Table 5.5: Explanatory variables correlations with bush wear

high correlation factors between bush wear. Among tension features, mean tension, tension standard deviation and tension root mean square value effect the bush wear the most where tension standard deviation seems to be the main cause apart from the run length.

As the Archard's law shows linear relationship in between sliding distance, force and wear, we fitted a linear regression model between total number of turn, tension standard deviation and remaining bush width as shown in Figure 5.8. Both p-Values are well below the 0.05 threshold, so we can conclude our model is indeed statistically significant which means the linear relation can be accepted. A qq plot is shown in Figure 5.9 which shows more intuitively how well it is fitting the linear regression model. Again it is only based on five samples, so we can't draw conclusions based on such limited amount of samples.

## 5.8 Conclusion

To conclude this chapter, the following research question is answered:
How is the quality of industrial data and what are the challenges?

3a.What are the characteristics of industrial data?
Specifically to TATA Steel case, data are highly unbalanced. Data size from different data sources differ a lot from each other. Sensor based data source such as IBA and EMASS produced huge amount of data with high dimensions(hundreds of variables) whereas manually logged data source(setup sheet) produced extremely small sample size (6 samples in 4 months). Quality of data from different sources also differs. For example, data from set up sheet is clean in format, however, since they are manually logged the values are rounded thus are not very accurate. On the contrast, sensor based data source produce data with high diversity. Text format is the most common in TATA case but the ways the data were recorded in the text files also differ per data source. In this particular instance of TATA, data are commonly logged off the norm and produces massive amount of missing data points. It may be an issue of the data quality but also can be the domain knowledge from expert wasn't comprehensive enough to cover diverse cases. From the case of TATA STEEL, we can have a general idea of the data situation in industry that data are often unbalanced and not well structured.

3b.What are the challenges while pre-processing the data and how to deal with them?
One of the biggest challenges we faced is the size of the data is so large that the data cannot be even read at the first place. Thus data are splitted into small data files and read by skipping lines. Check has been made to make sure that after skipping data points the observations are still spread out through the day rather than only data corresponds to certain time period were left.

The data are in very high dimension that makes it hard for us to extract information from it. Thus how to filter the variable and extract the most essential information is one of challenges. We have also found that some parameters are logged completely off their norms provided by expert. Thus assumptions had to be made that there are no batch errors within data sets. PCA is used to reduce data dimensions. Variables are filtered by only leaving the variables with the largest correlations with the principle components that represents the most variance. After filtering the data, only the data represents the most variance were left. We then extract 8 statistical features from each variables and did PCA on each variable again.

Unsupervised learning techniques namely hierarchical clustering and K-means clustering techniques are investigated to discover some hidden patterns filtered variables and features. The result shows that clustering techniques can recognize abnormal event from the data set that summarized in Table 5.6. However, depending on the data sets and techniques, the recognized dates will differ. Although hierarchical and K-means clustering are widely implemented in industry as they are less computational expensive, they still have some limitations, for example, K-means clustering is very dependent on initial criteria so that it produces inconsistent clustering result every time and hierarchical clustering can not cluster data set into more detail thus it is hard to capture every pattern within data.

Validation for unsupervised learning has been done by looking back at the events recorded in the shift report. Nearly all dates that have been clustered out can be confirmed with either planned or unplanned issues happened. However there are also quite some days with abnormal events that haven't been clustered out. The main reason is that the data set only contains data from one section of the line and the fact that some variables are logged wrongly decreases data quality. If data from the whole line can be collected(with good data quality), the clustering performance should improve

and this provides a new predictive maintenance project direction for the future. Moreover advanced unsupervised learning techniques may worth investigating to increase the clustering performance.

Correlation test and linear regression have been implemented on the variables that are strong in explanatory power to wearing condition according to literature. Correlation test and linear regression model can to certain extent validate the finding from literature that the wearing is correlated to total number of turn and tension. However, it is only based on five samples which hugely reduce the credibility of this finding.

| **Variables** | **Type** | **Dates** |
|---|---|---|
| CoatMass.BOT | Hcluster | 03/07, 06/07, 25/08, 26/08, 30/08, 31/08 |
| corrRollPos | Hcluster | 06/04, 07/04, 11/04, 18/04, 19/04, 06/05 |
| HeaderPressRawBot | Hcluster | 18/07 |
| ControlPressTOP | Hcluster | 06/04, 06/05,18/07, 04/08 |
| PotTemp | K-means | 13/09, 12/04, 17/04, 26/04, 10/05, 24/06 |
| AL Content | Hcluster and K-means | 12/04, 26/04, 26/05 |
| Distance.A2-A5,B2,B4 | Hcluster and K-means | 14/06, 24/06, 01/08, 16/08, 20/08 |
| Distance.B3 | K-means | 14/06,24/06,01/07,02/07,05/07,01/08,13/08, 14/08, 16/08,20/08,06/09 |
| Distance.B5 | Hcluster | 01/07 |
| Current.B4 | Hcluster and K-means | 30/06, 07/08, 17/08 |

Table 5.6: Clustering results

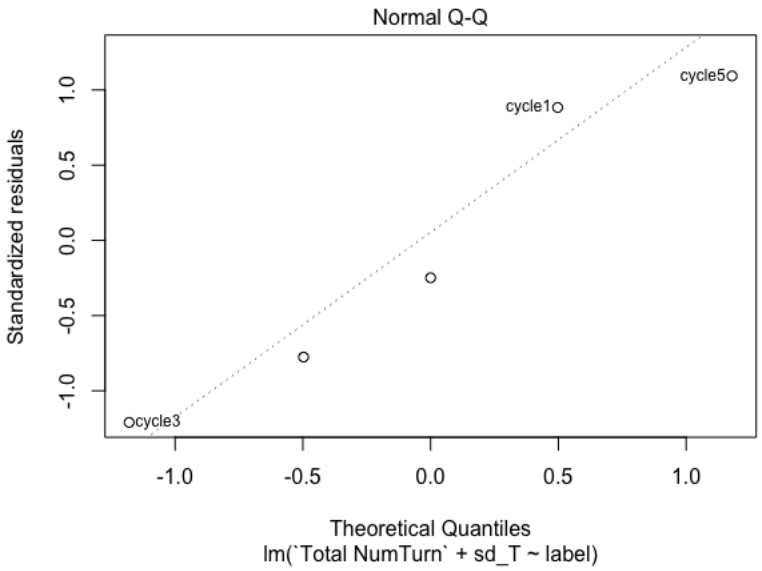


Figure 5.9: linear regression QQ plot

CHAPTER
# SIX

# MODELING AND EVALUATION

From literature review we found that support vector machine, neural network and random forest are the most implemented machine learning models for wear prediction. However, they are mostly based on experimental settings. In our case, the data are unbalanced with a sample size for target variables of six. Thus support vector machine is not applicable due to sample size limit. Thus we excluded SVM for further evaluation. We will motivate our modeling technique choice in Section 5.1. Section 5.2 introduces what features are extracted from the available variables and why they have been extracted. Section 5.3 introduced the data set we used and how to prepossess it before putting it into models. Model building and comparison can be found in section 5.4 - section 5.5. Data used are the maintenance cycle form 15/05/2019 on-wards.

## 6.1 Selection of modeling techniques

The current sample size has built a barrier for the building of machine learning models. Because the bush wear has been measured since 15/05/2019 and it is measured every 4 weeks while no other faulty behaviour happens. Till 12/09/2019, we only have 6 samples available. The samples size for support vector machine starts to be evaluated in (Figueroa, 2012) is 80. Neural network and random forest can handle smaller sample size but the minimum sample size requirements is recommended as 10. (Eker, 2012) In order to be able to deal with limited amount of samples, partial least square regression(PLSR) was investigated. PLSR is widely implemented in Chemometrics and Genomics for prediction purposes. It is used when the number of variables are more than the number of samples in the data sets and when the variables are correlated.(Swathik Claronicia) This is exactly our case when it comes to predicting bush wear measures as we have explored in the last chapter that run length, roll diameter, scrape length and tension features are all having strong correlation with bush wear based on existing samples. Moreover, PLSR has been investigated in maintenance context. For example, in (Li, 2009) a generalized PLS Regression forecast model is developed to predict maintenance cost of Warship. Data set with sample size 5 has been used, with the first 4 samples as training set and the last sample as test set. The model has been compared with normal PLSR models and see how good the prediction result for the test set is. As PLSR is well suited for our case(large number of variables and small sample size), we will focus on using PLSR modeling and interpret the results. We also try out artificial neural network and random forest because a sample size 10 should work for them. Although we don't currently have 10 samples we will have them in the near future. In the next sections we will explore on these three models with different settings and compare their prediction performance.

## 6.2 Feature selection

The ultimate goal of this project is to predict the bush wear to support decision making in maintenance policy. Thus explanatory power (the interpretability) from the predictors is preferred such that the operators are able to adjust parameters settings to possibly prolong the life time of the bush. Considering this, decision have been made to only investigate the parameters that has explanatory power which according to the Ardcher's law, are the variables related to sliding distance, force, and material hardness. However, material hardness is a set value thus we excluded it from our analysis. The variables selected are shown in Table 6.2. Sliding distance representative variables are Total Length, which is aggregated by parameter "Coil Length", Scrape Length and Total Surface are both aggregation of the corresponding variables from data warehouse. Force representative variables is "Strip Tension" from IBA. Eight features are extracted from this variable, namely Mean value, Minimum value, Maximum value, Median value, Skewness value, Kurtosis, Standard Deviation, and Root Mean Square value(RMS). The RMS value is calculated with the following formula:

$$RMS(x) = \sqrt{\frac{x_1^2 + x_2^2 + ... + x_n^2}{n}} \tag{6.1}$$

Standard deviation is calculated with the following formula:

$$s = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + ... + (x_n - \bar{x})^2}{n - 1}} \tag{6.2}$$

Where n is the sample size, x is the sample variable value. We also selected 3 additional variables that are Bath Temperature, Days and Roll Diameter as shown in Table 6.1. As the bush is always working in the bath, the chemical content and the bath can effect the bush life thus it is worth exploring parameters related to the bath. The temperature differs per product namely Galv and GalvFan. Thus these two variables will be both added when exploring this parameter. Variable "Days" is selected to explore the bush wear in relation to time. Roll Diameter represents the diameter of the sink roll. As the bush is installed and supports the turning of the sink roll, the roll diameter is effecting the number of turns of the bush which might effect the predictive power of the model.

| **Features** | **Data Source** |
|---|---|
| Total Length | Data warehouse |
| Scrape Length | Data warehouse |
| Total Surface | Data Warehouse |
| Mean Tension | IBA |
| Minimum Tension | IBA |
| Maximum Tension | IBA |
| Median Tension | IBA |
| Skewness Tension | IBA |
| Kurtosis Tension | IBA |
| Standard Deviation Tension | IBA |
| RMS Tension | IBA |
| Remaining Bush Width | Set up sheet |

Table 6.1: Features selected with explanatory power based on the Ardcher's law

| Features | Data Source |
|---|---|
| Bath Temperature | Data warehouse |
| Days | Setup sheet |
| Roll Diameter | Setup sheet |

Table 6.2: Additional features selected for further experiments

## 6.3 Data Processing

5 samples from the data set are used as training set and the remaining one sample is test set. This is to maximize the number of training samples for the model to learn. As the sample size is extremely small, we use Leave One Out(LOO) cross validation for model training. In LOO, the parameters of the chosen model is performed automatically on, in this case, 4 out of 5 samples in training set and test the prediction on the 5th sample. In this step the 5th sample is the test set while training the model. Repeating the process 5 times, each time leaving one sample out as test set and tune parameters of the model during this process.

Before putting data into models, data are scaled so that variables with different units are put into the same range such that variables are comparable. Formula for data scaling is as follow, where std(x) is the standard deviation of variable x.

$$Scale(x_i) = \frac{x_i - \bar{x}}{std(x)} \tag{6.3}$$

A scaled data set is shown in Figure 6.1. Labels are not scaled as they are variables to be predicted and validated. The label is calculated from manual measurement of the remaining bush width. We extract the minimum remaining bush of both sides from the original diameter of the bush such that we have the maximum wearing of both sides, because the decision of whether to change the bush or not should be based on the largest wear from both sides. Detailed formula is as follows:

$$Label = BushDiameter – Min(RemainingbushLeft, RemainingbushRight) \tag{6.4}$$

| | Total Length | Scrape Length | Total Surface | Temp GV | Temp GF | mean_T | min_T |
|---|---|---|---|---|---|---|---|
| cycle1 | 0.05067685 | 0.1240577 | 0.04644454 | 0.7865965 | 0.002681942 | 0.4041015 | 0.6526560 |
| cycle2 | 1.00129829 | 0.8984889 | 1.01285392 | -1.0807328 | 0.125670717 | 0.6441931 | -1.2596576 |
| cycle3 | -0.70828425 | -0.6722720 | -0.74368446 | 1.5804359 | 1.091817333 | 0.2423847 | 0.6365899 |
| cycle4 | -1.61485096 | -1.6789917 | -1.59158590 | -0.3081580 | -1.888982002 | -2.0230025 | 0.6519837 |
| cycle5 | 0.83242953 | 0.7730327 | 0.82733410 | -0.2224593 | 0.345055690 | 0.3999193 | 0.6402092 |
| cycle6 | 0.43873053 | 0.5556844 | 0.44863780 | -0.7556822 | 0.323756320 | 0.3324039 | -1.3217812 |

| | max_T | median_T | skewness_T | kurtosis_T | sd_T | rms_T | RollD | Days |
|---|---|---|---|---|---|---|---|---|
| cycle1 | -0.35467727 | 0.5156629 | -0.2456890 | 0.08036901 | -0.64246995 | 0.3534912 | 0.3465009 | 0.2042148 |
| cycle2 | 0.22412395 | 0.5300124 | -0.9917729 | 1.54362812 | -0.61999328 | 0.6967909 | 1.1197894 | 0.7697327 |
| cycle3 | -1.53767744 | 0.2194510 | -0.1258762 | -0.55614233 | 0.05127883 | 0.2632137 | 0.6634224 | -0.5498091 |
| cycle4 | 0.05079549 | -2.0196056 | 1.9416169 | -1.44874998 | 1.94852737 | -2.0179478 | -1.2381068 | -1.7750978 |
| cycle5 | 0.05079549 | 0.5214505 | -0.3285043 | -0.06454424 | -0.62520116 | 0.3506214 | 0.3465009 | 0.6754797 |
| cycle6 | 1.56663978 | 0.2330288 | -0.2497744 | 0.44543942 | -0.11214181 | 0.3538307 | -1.2381068 | 0.6754797 |

| | label |
|---|---|
| cycle1 | 12 |
| cycle2 | 19 |
| cycle3 | 12 |
| cycle4 | 1 |
| cycle5 | 15 |
| cycle6 | 17 |

Figure 6.1: Scaled Modeling data sets

## 6.4 Partial Least Squared Regression

In order to select the set of features that produce the best overall prediction and meanwhile the results are preferred to be interpretable. We will select the final feature set based on PLSR performance. We select the feature not only based on the predictive power on the initial samples but also the generalization ability of the model, that is, how the model performance changes when more samples coming in. To evaluate the generosity of the model, we plot a learning curve. A learning curve is how the prediction accuracy changes when adding more samples in training set. The final feature set is chosen after comprehensive evaluation of modeling performance and other modeling techniques will use the same feature set selected at the end of this chapter for sake of consistency.

### 6.4.1 Experiments on extended variables

By using the features selected according to the theory(sliding distance and force related variables) we use "Leave One Out" cross validation to evaluate the performance of PLSR model. Every time we run the model we will use a different cycle as test sample and the rest of the cycles as training samples. This is to make sure the model has the most variance of samples to learn from(as much as training samples as possible). As PLSR is an extension of principle component regression, the data are projected into lower dimensions as "latent variables". The number of latent variables are called "The Number of Components" during the experiments from now on. The number of components has to be chosen before predicting new results. Many techniques can be used to choose the most effective number of components, here we plot the number of components against Root Mean Square Error of Prediction(RMSEP) and choose the most effective number of component when the RMSEP is the lowest. Plots corresponds to each test set can be found in Figure 6.2 to Figure 6.7. These plots are different when using different feature set, here we only have presented plots corresponding to feature 3 as an example. When coding the model, the component selection part has been automated in the code so there is no need to select manually.

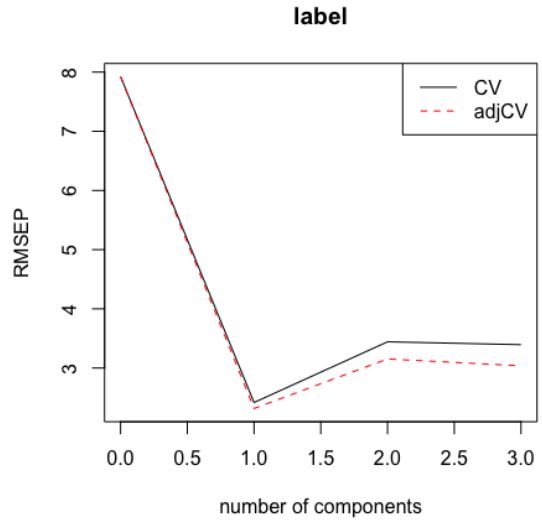


Figure 6.2: Selection cycle1 as test set

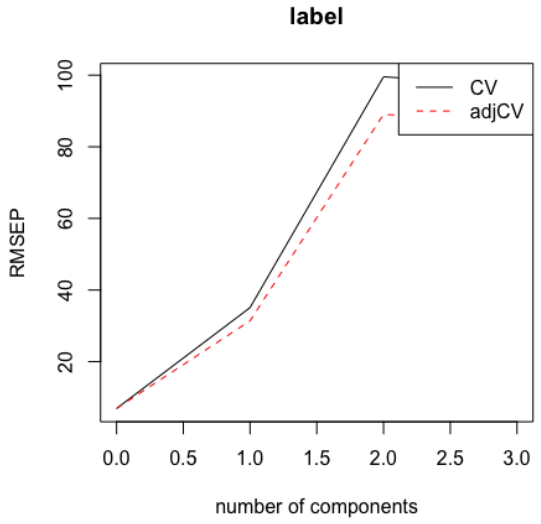


Figure 6.3: Selection cycle2 as test set

The corresponding prediction results on each feature set can be found in Table 6.3 - Table 6.6. We can see clearly from the table that when using Feature set 1, 2, 3 the prediction result is almost the same. Cycle1, Cycle7 and Cycle3 are have a error at around 3mm. Whereas other cycles the prediction

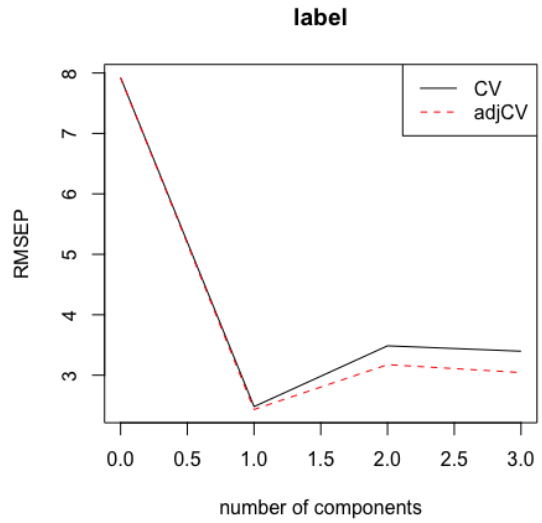


Figure 6.4: Selection cycle3 as test set

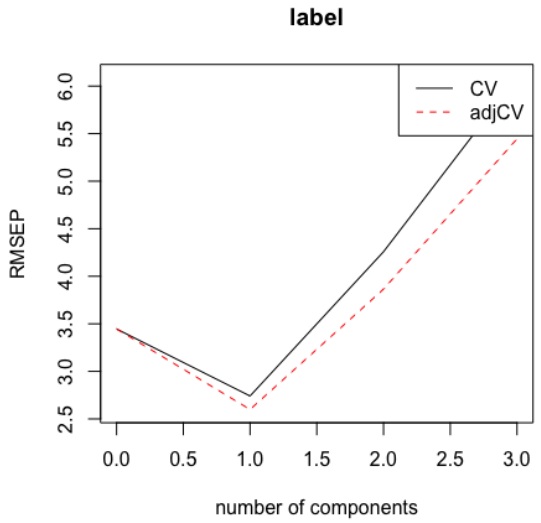


Figure 6.5: Selection cycle4 as test set

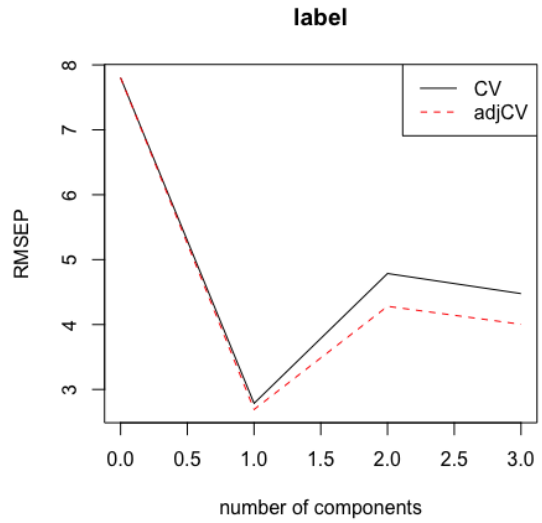


Figure 6.6: Selection cycle5 as test set

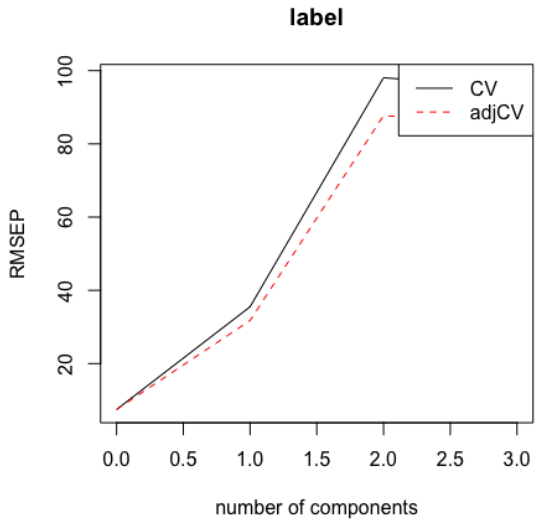


Figure 6.7: Selection cycle6 as test set

errors are within 2mm. When we do the experiment on Feature set 4 (Adding bath temperature) the result becomes a bit off. This is mainly because prediction on Cycle4 becomes worse with an error of 4mm.

A more intuitive comparison is shown in Figure 6.8. Root mean square error value has been used for modeling performance evaluation. We don't use MSE here as most literature did because it is a measure of difference between prediction value and true value and thus more intuitively shows the prediction deviation from the real value. It is calculated as follow:

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N} (x_p - x_r)^2}{N}} \tag{6.5}$$

| Test sample | Prediction | Real value |
|---|---|---|
| Cycle1 | 14.46 | 12 |
| Cycle2 | 20.48 | 19 |
| Cycle3 | 9.10 | 12 |
| Cycle4 | 2.57 | 1 |
| Cycle5 | 14.34 | 15 |
| Cycle6 | 14.56 | 17 |

Table 6.3: PLSR validation result with standard features(Feature set 1)

| Test sample | Prediction | Real value |
|---|---|---|
| Cycle1 | 14.51 | 12 |
| Cycle2 | 20.06 | 19 |
| Cycle3 | 8.98 | 12 |
| Cycle4 | 2.66 | 1 |
| Cycle5 | 14.64 | 15 |
| Cycle6 | 14.78 | 17 |

Table 6.4: PLSR validation result adding "days"(Feature set2)

| Test sample | Prediction | Real value |
|---|---|---|
| Cycle1 | 14.60 | 12 |
| Cycle2 | 20.24 | 19 |
| Cycle3 | 9.47 | 12 |
| Cycle4 | 2.69 | 1 |
| Cycle5 | 14.71 | 15 |
| Cycle6 | 13.51 | 17 |

Table 6.5: Adding "days" and "RollD"(Feature set 3)

| Test sample | Prediction | Real value |
|---|---|---|
| Cycle1 | 14.30 | 12 |
| Cycle2 | 19.75 | 19 |
| Cycle3 | 9.90 | 12 |
| Cycle4 | 5.17 | 1 |
| Cycle5 | 14.85 | 15 |
| Cycle6 | 13.72 | 17 |

Table 6.6: Adding "days","RollD" and "Bath Temperature"(Feature set 4)

Where xp is a measure of predicted value and xr is a measure of real value. N is sample size. As shown in Figure 6.8, all the prediction errors on different feature sets are very similar. Thus it is not wise to choose the feature set based on the prediction power on the six initial samples only. Thus a generalization test is done by plotting a learning curve of the PLSR model. The learning curve is a common technique when evaluating modeling techniques and comparing different models. It shows

whether the modeling performance will benefit from more training samples. (scikit-learn developers, 2019) The plot can be found in Figure 6.9. We can see from the plot that using feature set2 produces the best prediction result with the initial sample set. However, by adding one sample into training set at a time, the error continuously increases until the 9th sample(the last sample we have). However, by using the rest of the feature set the error increases at first and then drop when the last sample coming in. Due to the limited amount of sample size, we cannot know just now how will the performance change further, but it is likely to converge when we get enough samples. Thus choosing the feature set that produces the best prediction when all the samples are in should be the best choice at this stage. Therefore, the final feature set will exclude bath temperature but include the rest of the features.

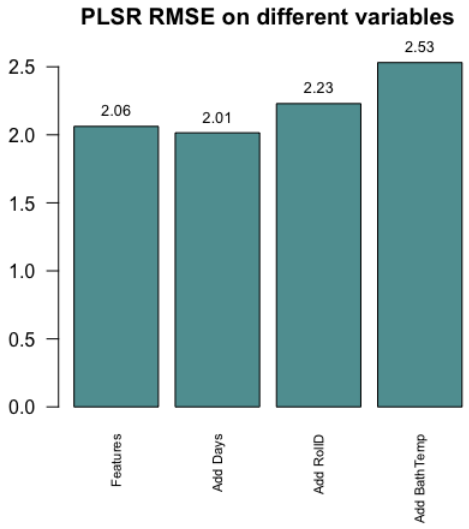


Figure 6.8: PLSR performance on different feature set

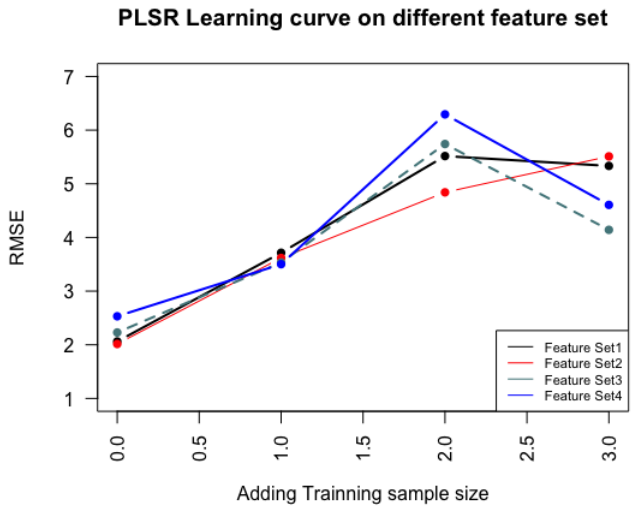


Figure 6.9: PLSR learning curve on different feature set

### 6.4.2 Result interpretation

One of the advantage of using PLSR is that the results can be interpreted by plotting the 'correlation loading' plots(Figure 6.10 to Figure 6.15). The plots show the correlation between each variable and the selected components. Each point corresponds to a feature. The squared distance between the point and the origin represents the fraction of the variance of the variable explained by the components in the panel.(PLS package, 2019). The correlation value represented for each variable shows how much this variable is contributing to the prediction result.

We can see from the plot that Maximum Tension and Minimum Tension contribute to most of the prediction result when we use cycle1, cycle2, cycle3 and cycle4 as test set. The rest of the features share similar correlations to the results. While predicting wear width of cycle 4, the sink roll diameter and median value of Tension also have a big effect on prediction. When predicting wear on cycle 5 and cycle 6, each feature has similar amount of correlation to the prediction.

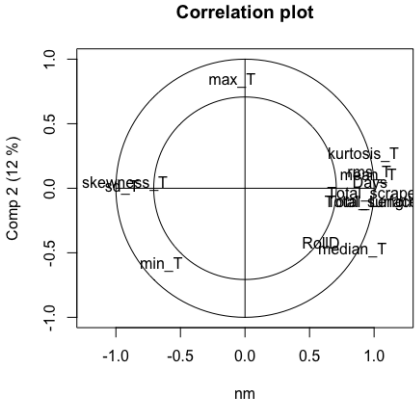


Figure 6.10: Correlation plot cycle1 as test

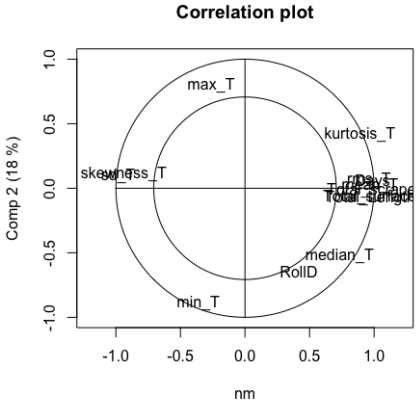


Figure 6.11: Correlation plot cycle2 as test

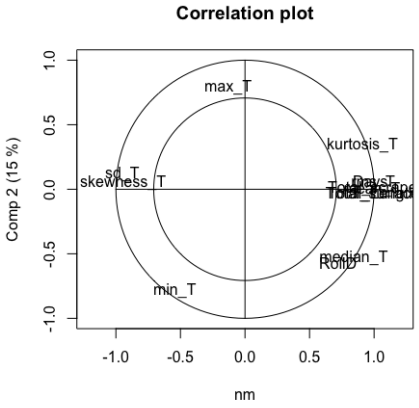


Figure 6.12: Correlation plot cycle3 as test

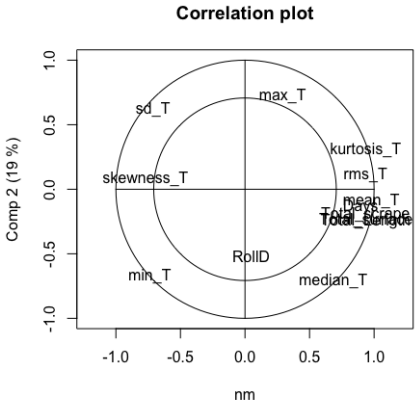


Figure 6.13: Correlation plot cycle4 as test

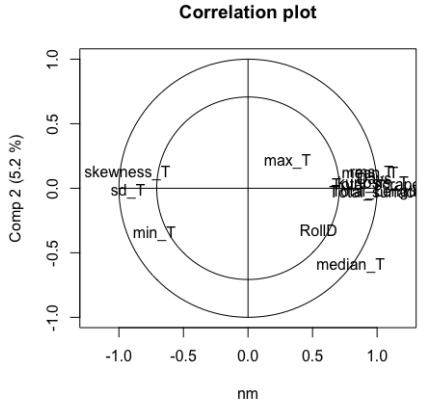


Figure 6.14: Correlation plot cycle5 as test

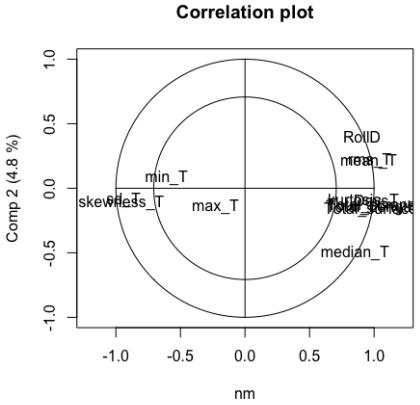


Figure 6.15: Correlation plot cycle6 as test

## 6.5 Neural Network

### 6.5.1 Inner layer and weights tuning

In order to maintain consistency in model comparison, models are trained on the features selected from the last section.(Table 6.7). Neural Network has lots of parameters to tune, namely number of layers, number of nodes on each layer, and initial weights on each neuron. Default training mechanism in R is that the model will stop training when sum of squared error is less than 0.01. Initial training weights are assigned at random. This causes problem as the prediction result may be different every time we run the model. In order to produce consistent result, whenever a good prediction result is produced, we will record the weight matrix and set it as initial weight then train the model using different training samples and validate it on test sample. A neural network contains a input layer

| **Features** | **Data Source** |
|---|---|
| Total Length | Data warehouse |
| Scrape Length | Data warehouse |
| Total Surface | Data Warehouse |
| Mean Tension | IBA |
| Minimum Tension | IBA |
| Maximum Tension | IBA |
| Median Tension | IBA |
| Skewness Tension | IBA |
| Kurtosis Tension | IBA |
| Standard Deviation Tension | IBA |
| RMS Tension | IBA |
| Remaining Bush Width | Set up sheet |
| Days | Set up sheet |
| Roll Diameter | Setup sheet |

Table 6.7: Features selected based on PLSR performance

on which the number of nodes is equal to the dimension of the data set. In our case the number of input layer is 13. The output layer is set with one output node of 1. Decision has to be made for the number of node on the hidden layer. It is suggested by (Jeff Heaton, n.d) that the optimal size of the hidden layer is usually between the size of the input and size of the output layers. It is also suggested that the situations in which performance improves by adding a second (or third, etc.) hidden layer are very few. One hidden layer is sufficient for the large majority of problems. Thus, some neural networks with 2 or 3 hidden layers and confirming that the prediction results did not vary much, decision has been made to use only one hidden layer. The number of nodes is decided by using one cycle as test set and adding up neurons from 1 to 13, and choosing the number when the prediction result is fluctuating around the label of the test sample instead of other random values.

The structure of the ANN is tuned using Cycle 6 as test sample where the real label is 17 and the prediction result is 16.999712. We do not change the number of layer and number of nodes on each layer anymore for further experiments. The Neural network structure we use is 12 meaning disregarding input layer, output layer and bias nodes(Blue nodes in Figure 6.16) there is 1 hidden layer with 12 nodes. After setting the NN structure, we use LOO cross validation again to select the initial weights by using one cycle as test set and training the model on the rest of the samples. The reason why we only use one sample as test set is that we want to maximize the the number of training samples.

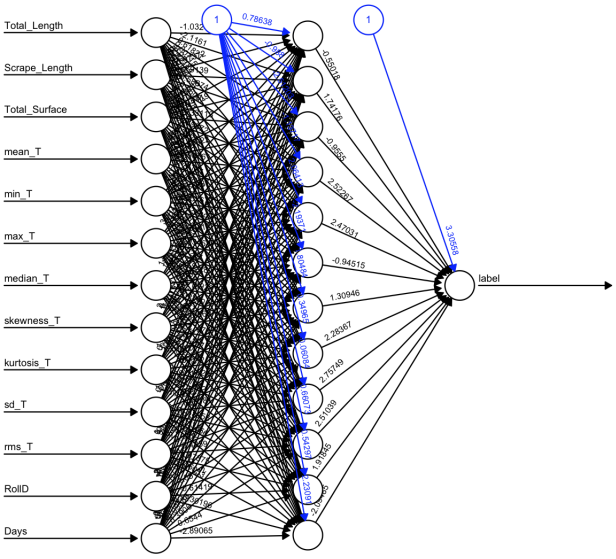


Figure 6.16: Neural Network Structure

Similar approach has also been used in literature. The model is evaluated based on RMSE value. A graph with a comparison of RMSE value when using different cycle as test sample can be found in Figure 6.18. We can see from the graph that ANN produce stable result on initial 6 samples no matter which sample was used as the test sample. All RMSE value is within 1mm. The prediction result is always close to the real value. However using cycle4 as test set does produce the best result.

In order to show more intuitively how close is the neural network prediction result to the real value, we plot Figure 6.17 showing the prediction result of ANN using different cycle as test set. However, although the prediction result is good on the initial data set, it still subject to change when more data is coming in. Thus we recorded all the initial weights(6 weights sets) that was used in the experiments, and test them one by one to predict new samples. In order to capture how the prediction changes when new samples are added into training samples, we add one sample into training samples each time and predict all the samples all over again. For example, when training sample size is 7 the cycle 1,2,3,4,5,6,7,8 are taken into account and each cycle is predicted as test sample with remaining samples in the training sets. The average prediction error changing with the number of training samples can be found in Figure 6.19. We can see that the learning curve shares similar trend as PLSR learning curve. Error first goes up and then drops when adding 3 samples to the training set(8 samples in the training sample). Overall when using initial weights1, the error is smaller than the others, thus we will compare the modeling result of ANN using initial weights1.

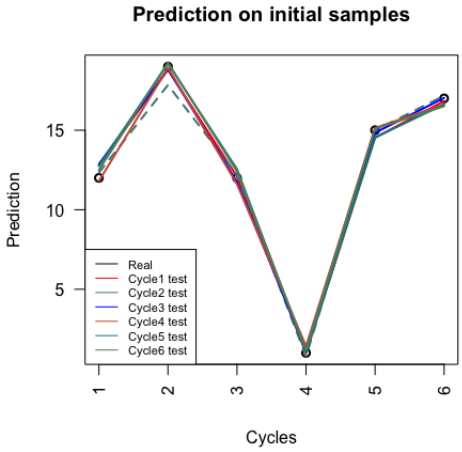


Figure 6.17: RMSE comparison using different test samples to find initial weights

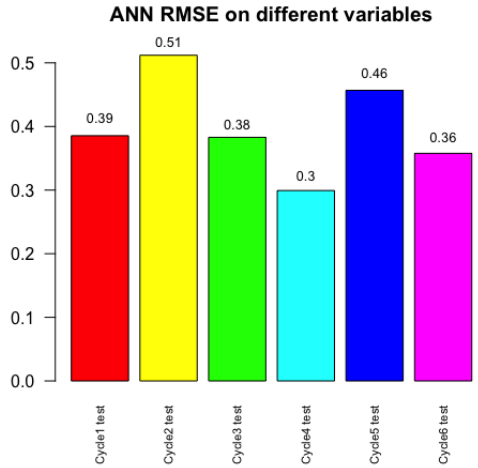


Figure 6.18: Model performance and selection

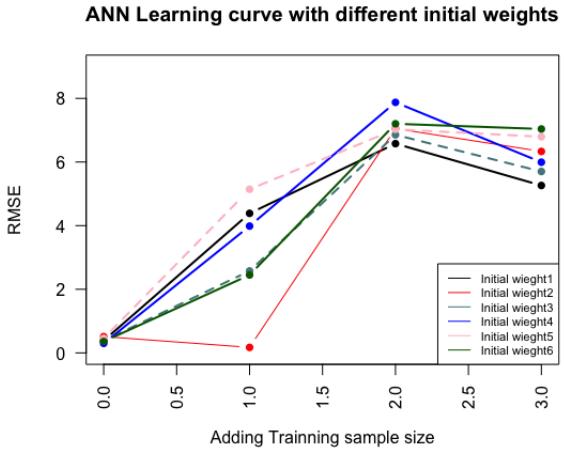


Figure 6.19: ANN learning curve

## 6.6 Random Forest

Based on(Gerard, n.d.),Random Forest should also be able to handle small sample size. Thus we also have tried out the random forest algorithm. The default number of trees in R is 500. Different from ANN, we don't have to tune the parameter in Random Forest as it is a random process. Samples chosen for each tree are different and the features chosen at each split are chosen randomly. Thus we expect the model to produce different result even with the same samples. This is considered one of the disadvantages of using Random Forest. The prediction on initial 6 samples using Random Forest can be found in Table 6.8. The model is not able to predict special cases(Cycle 4 and Cycle 2) and the RMSE is 7.16 mm which is rather large for wear monitoring as the total diameter of the bush is only 30mm. We also plot the learning curve to see whether the prediction gets more accurate when more samples are taken into consideration. We can see that the error slightly dropped.

| Cycles | Prediction | Real label |
|---|---|---|
| Cycle1 | 13.97 | 12 |
| Cycle2 | 12.14 | 19 |
| Cycle3 | 14.77 | 12 |
| Cycle4 | 16.24 | 1 |
| Cycle5 | 14.60 | 15 |
| Cycle6 | 12.95 | 17 |

Table 6.8: Random Forest result validation using LOO

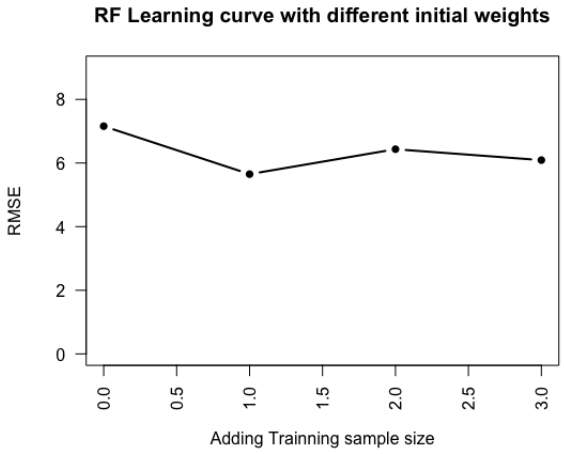


Figure 6.20: Learning curve random forest

## 6.7 Comparison and evaluation

The three models: PLS regression, artificial neural network and random forest are evaluated based on RMSE measure and R squared value. R squared value is also called coefficient of determination, which indicates the proportion of variation in the dependent variable that is predictable by the independent variables. It is calculated with the following formula:

$$R^2 = 1 - \frac{\sum_i (x_p - x_r)^2}{\sum_i (x_r - \bar{x_r})^2} \tag{6.6}$$

where xp means predicted value and xr means real value. A higher R squared value indicated that more variability is explained by the regression model and generally it is better.(Wu 2017). As the modeling performance changes when the number of training samples changes, it's hard to conclude only by comparing one single prediction result. Thus we put all the RMSE and $R^2$ value into tables with different number of training samples to have an impression of how they change and then average them to have a overall measure. The results are shown in Table 6.9 and Table 6.10. We can see from the table that PLSR produce comparatively stable result and get less effect from the number of samples in the training set. However, ANN shows extreme fluctuation in terms of the prediction power. Specifically, ANN predict very well when 5 samples in the training set but when new samples coming in it gets worse drastically which means that the model is not general enough. RF performance is stable with comparatively large error independent from the number of training samples.

When it comes to $R^2$ value, we can see that both performances of PLSR and ANN fluctuates when the number of training samples increases. Overall variances that can be represented by the model drops when more samples coming in especially ANN. However, we can not know so far whether the performance will continue dropping or it will rise again. However in theory the performance should converge when the training samples have represented enough variance. It is worth mentioning that the RF model can not capture any variance of the samples. This can be also intuitively seen from Table 6.13, all the prediction result is around 13.

Figure 6.21 and Figure 6.22 visualizes the change in both RMSE and R squeared value. We can

| **Training samples** | **PLSR** | **ANN** | **RF** |
|---|---|---|---|
| 5 | 2.23 | 0.39 | 7.16 |
| 6 | 3.53 | 4.39 | 5.65 |
| 7 | 5.74 | 6.58 | 6.43 |
| 8 | 4.14 | 5.26 | 6.09 |
| Average | 3.91 | 4.15 | 6.33 |

Table 6.9: Modeling RMSE comparison

conclude from the graphs that based on the current available samples, the PLSR out performs the other two models by having the overall lowest prediction error and overall better capability of capturing sample variance.

In order to give a intuitive view on the real prediction result, a record of result that included 8 samples in training set is presented in Table 6.11 to Table 6.13. The corresponding visualization of the tables is Figure 6.23. We can see from the result that ANN predict reasonably well on cycle 1 to 7. The error increase drastically when predicting cycle 8 and 9. PLSR prediction is very stable when predicting cycle 1 to 9 but also produce bad result when it comes to cycle 8. Thus cycle 8 is very likely to be a special case that is hard to be captured when it is not in the training sample.

| Training samples | PLSR | ANN | RF |
|---|---|---|---|
| 5 | 0.85 | 1.00 | -0.53 |
| 6 | 0.61 | 0.40 | 0.01 |
| 7 | 0.18 | -0.07 | -0.02 |
| 8 | 0.53 | 0.25 | -0.01 |
| Average | 0.55 | 0.39 | -0.14 |

Table 6.10: $R^2$ comparison of models

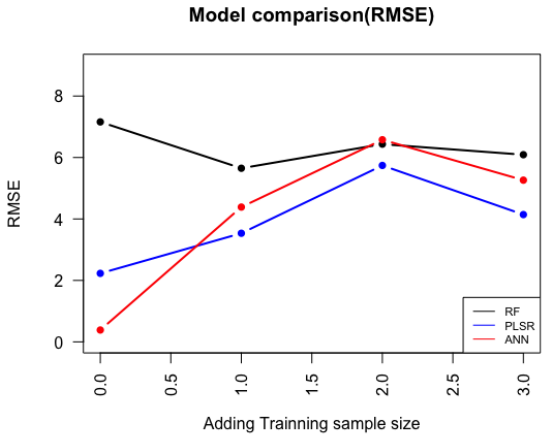


Figure 6.21: Modeling comparison RMSE

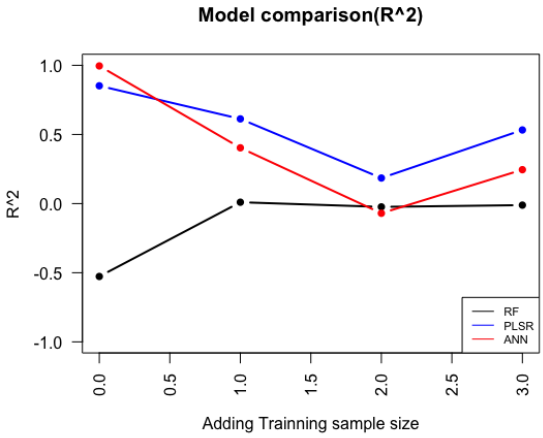


Figure 6.22: Modeling comparison $R^2$

| Cycles | Prediction | Real label |
|---|---|---|
| Cycle1 | 15.39 | 12 |
| Cycle2 | 19.59 | 19 |
| Cycle3 | 12.49 | 12 |
| Cycle4 | -0.42 | 1 |
| Cycle5 | 14.20 | 15 |
| Cycle6 | 17.00 | 17 |
| Cycle7 | 16.28 | 18 |
| Cycle8 | 11.60 | 24 |
| Cycle9 | 20.82 | 12 |

Table 6.11: ANN prediction result

| Cycles | Prediction | Real label |
|---|---|---|
| Cycle1 | 16.23 | 12 |
| Cycle2 | 20.40 | 19 |
| Cycle3 | 11.12 | 12 |
| Cycle4 | -1.44 | 1 |
| Cycle5 | 17.92 | 15 |
| Cycle6 | 14.55 | 17 |
| Cycle7 | 13.66 | 18 |
| Cycle8 | 15.12 | 24 |
| Cycle9 | 15.95 | 12 |

Table 6.12: PLSR prediction result

| Cycles | Prediction | Real label |
|---|---|---|
| Cycle1 | 13.88 | 12 |
| Cycle2 | 12.95 | 19 |
| Cycle3 | 13.88 | 12 |
| Cycle4 | 12.96 | 1 |
| Cycle5 | 13.88 | 15 |
| Cycle6 | 12.95 | 17 |
| Cycle7 | 13.63 | 18 |
| Cycle8 | 13.62 | 24 |
| Cycle9 | 13.64 | 12 |

Table 6.13: RF prediction result

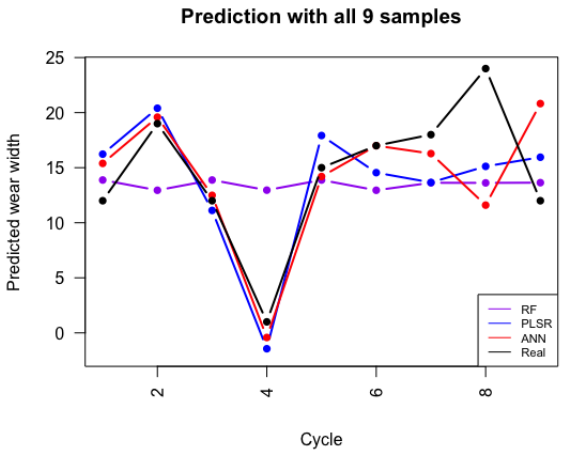


Figure 6.23: Prediction of 9 samples

## 6.8 Discussion

### 6.8.1 Sensitivity to data processing

The whole modeling process has been done 3 times due to data availability changing over time. It has been discovered that the way data is processed strongly effects the feature selection part of the modeling. Specifically, issue rises when the definition of a maintenance cycle is not consistent. For example when a cycle start at 15-05-2019 and ends at 06-06-2019, the next cycle can be defined starts at either 06-06-2019 or 07-06-2019. This should be defined consistently through every cycle. Depending on the definition, the prediction result is changing. Furthermore, the number of days in a cycle should be calculated in a consistent manner. The orginal date from the set up sheet was recorded manually thus the number of days was calculated differently. That is, sometimes, the number of days is EndDate - StartDate and sometimes it is EndDate - StartDate +1. The two different calculation effect the data quality from the set up sheet thus directly effect the feature selection result. Here in this paper, the starting and ending date of the cycle has been defined non repetitive. When a cycle ends at certain date, the next cycle has been defined to start on the next day. The number of days of a cycle has been calculated consistently by EndDate - StartDate +1. Whenever this definition changes, the whole modeling process should start all over again.

### 6.8.2 Overfitting

The ANN model suspect to have overfitting problems because of near perfect prediction on initial samples and the drastic dropping in prediction power when new samples are taken into account. This problem may be solved by continuously adding new samples and then retraining the model, when the model has learned enough variance from samples the result should be reasonable and stable. The ANN structure should also be modified when more samples coming in.

### 6.8.3 Model Evaluation

All three techniques chosen have their own advantages and disadvantages. Specifically to this case and current available data, PLSR is the most suitable model to implement for now. The stable prediction result produced by PLSR model in turn verified the Archard's Law that the predictor variables and the target variable are form linear relation based on the current available data samples.

The strength of PLSR model is the explanatory power and its ability to allow model builders see the contribution of each variables to the prediction result. This is significant when it comes to decision support. PLSR also produces consistent and stable results which is easier to implement and maintain when new data coming in. As the data size grow bigger, the modeling performance is subjective to change but it will likely to converge at some point.

Artificial neural network has the potential to produce prediction result with very high accuracy like with initial 6 samples but the prediction performance is unstable with current available samples. Meanwhile, with small sample size the ANN has to be tuned many times in order to fit the current sample. This need manual tuning and huge amount of iterations for updating the weights on each neuron. However this problem should gradually disappears when more data is coming in as the weight won't differ drastically every time when updating them. Another disadvantage of ANN is that it is a "black box" model thus it is hard to interpret the results as it is not able to provide insight on which variables effect the wearing condition the most.

Random forest model is so far not able to either fit or predict with current samples and more over the result is not interpretable. The model combined too many random process thus should requires huge data sets before it can produce decent results. However Random Forest is the least computational expensive among all three modeling techniques as it doesn't need parameters tuning or component selection. This is a technique that worth trying when the sample size grows bigger and it can potentially be the most user friendly model for online monitoring.

### 6.8.4 Challenges and Model maintenance

Small sample size brings challenge and critics from the very beginning of data exploration when we did correlation test between variables and bush wear. Because the sample size is too small, we can't really draw any conclusion on if tension and wear are truly correlated although it seems to have strong correlation based on the samples we have. The same with linear regression model. We can not be sure that the variables and bush wear follow linear relation based on only six samples. The linearity might be gone when more data become available. However, the good prediction results from PLSR model is to some extent validating the correlation and linear relationships between tension, run length and wear measures because the underlying theory behind PLSR is linear regression.

The modeling process from feature selection to model selection should be repeated in a set time interval. At the beginning when model performance is not stable the process should be repeated more frequently, say once per month. When the modeling performance converge, the maintenance interval can be longer, say every 3 months or half year depend on the specific situation. In the worst case scenario where the modeling performance dropping, more investigation into other techniques should be done in order to maintain the prediction accuracy.

## 6.9 Implementation

To implement predictive modeling into operation, a web page has been developed for monitoring purpose. By implementing the web page engineers will have insight on the current wear and predicted wearing pattern of the bush. The web page contains two main part, the first part is the input, including date, sink roll diameter and bush diameter(Figure 6.24). As each maintenance cycle sink roll diameter and bush diameter are subject to change, it is not possible to have them automated in the modeling process. Date is the essential input. When users select a date, the algorithm will automatically search for the start day of the maintenance cycle that contains the user selected date and predict the remaining bush width of each day within that cycle. The prediction result of the selected date is presented at the bottom of the graph, both the wear width and the remaining width(bush width - predicted wear width). Combining the information of predicted wearing pattern and the predicted wear(Figure 6.25), decision maker should be able to decide whether to replace the part or not and when should the maintenance activity be planned.

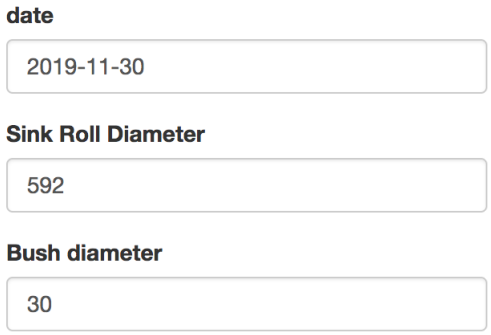


Figure 6.24: Web monitoring page input

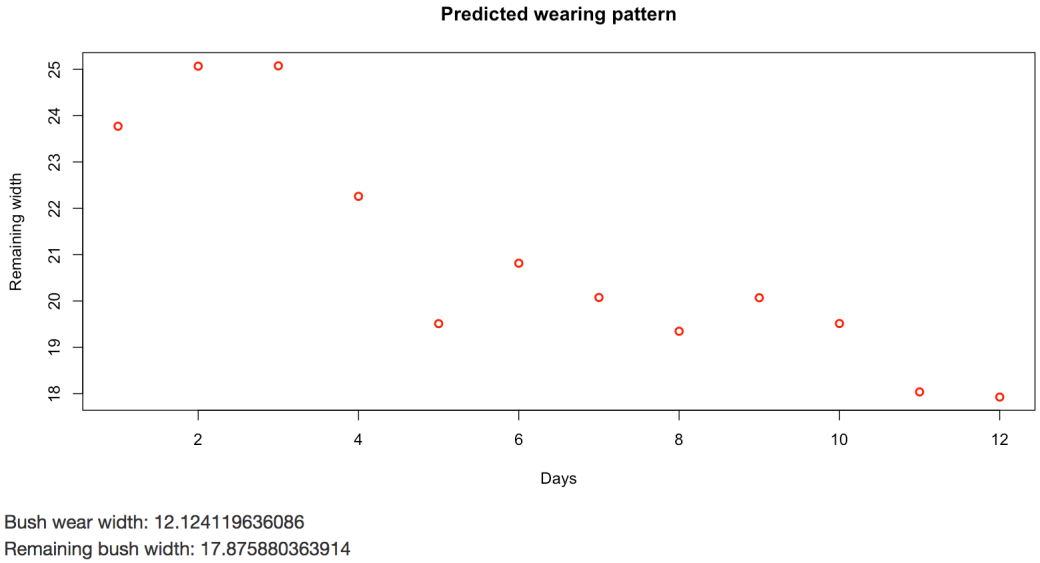


Figure 6.25: Web monitoring page output

## 6.10 Conclusion

To end this chapter, research question 4 is answered.

4a. What features can be extracted from the data set?
The features related to sliding distance, tension and time are extracted as independent variables. The sliding distance and tension related variables are extracted because of the Archard's law that repetitively appears in literature as a formula to describe the metal-metal contact wearing behaviour. In that formula, force and sliding distance are critical to wear volume. Modeling performance is compared by adding variables related to time, temperature and connected component characteristics, specifically the sink roll diameter. The final feature set is decided when PLSR model performs the best because modeling result is able to be explained by PLSR correlation plot so that insights on the importance of each variable can be obtained. In the end we use 14 variables in Table 6.7 as feature set to try on other models. The selected feature set containing sliding distance, tension, time and component characteristic. Temperature related features have been excluded.

4b. What models are suitable to use based on the current available data?
PLSR, ANN and RF are tried out for modeling because PLSR is suitable for data samples where the number of independent variables are larger than the number of dependent variables and when the variables are co-linear. ANN and RF theoretically can handle sample size of 10. As the characteristic of current data samples are: extremely small sample size and strong correlation between independent and dependent variables. PLSR is theoretically best suited for this case. ANN and RF should also work in near future as according to literature, they are suitable when sample size is at least 10.

4c. How can the model be trained and how can the modeling performance be evaluated?
We train and test the model using 'Leave One Out' cross validation and evaluate the model using RMSE and R squared measures. The reason why we use "LOO" is to maximize the number of training samples. RMSE represents the measure of prediction power and R squared value represents the measure of modeling variance. Learning curves are plotted to show how the prediction changes when more samples are taken into consideration. As a result, PLSR outperforms the other models with an average prediction error of 3.19mm and R squared value of 0.55. PLSR also has its advantage of explanatory power so is suitable to be used for decision support. ANN fits the initial samples very well but produce large errors when predicting new samples with an average error of 4.15. Random Forest produces large error of 6.33mm, because it is not able to recognize special samples thus is not suitable to predict with current available data.

4d. How can the predictive model(s) be implemented to the production operation?
As model comparison result, PLSR is so far the most suitable model in this case. Thus a monitoring web page is developed based on PLSR model. The web page is user-interactive where the user chooses a date and the corresponding sink roll diameter of that date. The web page will automatically predicted the bush wear of every day in the corresponding cycle and plot a graph with predicted wear on each day. The goal is to give decision maker the insights of the predicted wearing pattern such that he/she can decide when to replace the component.

CHAPTER
# SEVEN

# CONCLUSION AND RECOMMENDATION (TO PRACTICE)

This chapter focuses on the contribution to practice, namely the deployment plan of the project results, the following up projects and business development in the TATA STEEL Shotton. Research question 5 is answered in this chapter.

## 7.1 Feasibility

*5a. Is it feasible to predict the bush condition?*
By this end, the average error of the best model so far is 3.91mm. The bush that is used most of the time is 30mm. Thus the error rate is approximately 13% and therefore we can say it is feasible to predict the bush condition.

It is feasible to predict the bush condition using existing data that is around the component, namely the sliding distance, force, mechanical property(diameter) and environmental property(temperature) related data. Most importantly the wearing data were measured as the target variable. The availability of these data variables leads to the success of the prediction. Partial least squared regression(PLSR) predict the bush with a good accuracy and produces stable results. Thus PLSR is the recommended for the current available data. The value of the project mainly lies in the ability to monitor current wear and thus provide insight on the wearing behaviour of the bush for decision support.

## 7.2 Improvement

*5b. How to improve model performance and current situation regarding data logging and maintenance process based on findings?*
In terms of data logging, some quality issues occur in critical variables that should be improved. Data logged in IBA appears to have batch missing values continuously within a time span. 'Tension' is one of the critical variables and the data logging flaws effect the data quality of 'Tension' and thus very likely to influence prediction accuracy.

The procedure of a data driven predictive maintenance project can be simplified. The data in use varies from what literature suggests, as most of the literature use vibration data only to predict wearing behaviour. Unfortunately, vibration of the bush in this specific context was not able to be logged. Based on literature findings, vibration is the direct indicator of machine status and thus can be deployed on other problems in the production line. Some literature reports on the use of deep learning techniques and by feeding only the raw vibration data, it can successfully predict the wearing

width as well as the wearing location. Deep learning can simplify the modeling process to large extent because feature selection and extraction steps are no longer needed. The only step required before model building is to extract and clean the vibration data.

The robustness of the predictive model should be tested and improved by feeding in more training samples as the current model is trained to fit the existing samples and not generalized enough for different cases. Learning curve should be continuously plotted to see the change of the prediction accuracy changes while increasing the learning sample size. The convergence of the curve indicates that the prediction accuracy is fluctuating at the same level. If the the prediction ends up with bigger errors then new models should be investigated and selected. If the curve converges to a small error then the current model can be kept using.

The current prediction result is already adequate for the decision support with the developed tool because the current prediction reached a good accuracy. The value of the tool is the capability of monitoring the bush wear. This helps for 'lean operation' so that every time the pot gear can be replaced just in time when no other interruptions occur by online monitoring the wearing condition. The modeling process should be able to be repeated on similar sections of the line as long as the variable used in this project are logged. The wearing(target variable) width should be measured otherwise building predictive maintenance model will not be feasible in the first place. Predictive maintenance project can also be developed on other sections of the line however, the variables required should differ, depend on the goal of the project.

## 7.3 Limitation to practice

The biggest limitation of this project is the sample size. By the end of this project the sample size has been increased from the initial 6 samples to 9 samples. According to the plotted learning curve, the prediction error first rises and then drops in the end based on 9 samples. We can not know whether the error will continue dropping or it may increase again when more samples are taken into account. Thus it is necessary for the model to be retrained with new data. The correlation and linear relation among the variables and wear measure are also subject to change. Although in theory, they should follow linear relation and so far PLSR model performance can also in turn validate the linear relation.

In addition, the current tool is a monitoring tool which means it predict the current status but not the future. In other words, we are not predicting the bush wear of next week when we are still in this week. In stead, we are predicting the bush status at the user inserted date which is the current date.

## 7.4 Implementation

(Bousdekis, 2019) discussed the software structure and platform in steel industry. The author believes that the key issue of any design and system development in the context of Industry 4.0 is the proper implementation of Reference Architectural Model Industrie(RAMI)4.0. The predictive maintenance architecture in the frame of RAMI 4.0 is divided into 6 layers, namely assets, integration, communication, information, functional and business. Tata steel is currently having the first three layers meaning the machines are sensor monitored, data are logged and stored in servers. The other three layers requires data processing, data analytic, data monitoring(visualization) and allowing user interaction. Thus here we propose a hardware flow diagram for the information, functional and business layer based on current infrastructure in TATA. (Figure 7.1). AI and other data analytic tool can be used for data processing, modeling and visualisation whereas IBA is used for data logging, SQL for

data storage. Thus the developed web page should be connected to SQL and directly read modeling with data from SQL. Raw Data should be feed from IBA and data warehouse into SQL and the data should be cleaned and processed into desired format in SQL to be ready to use for the web page. To make decision process more automated,in Industry 4.0 concept, information extracted from the data should be fed back to assets and have the decision making system decide on the maintenance time.

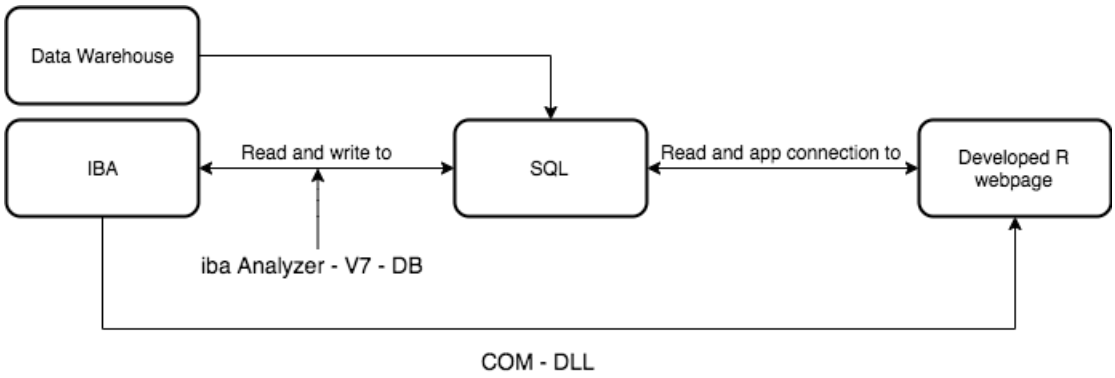


Figure 7.1: Flow diagram

## 7.5 Follow up project development

*-Predict the future bush wear*
As mentioned before one of the limitation is that the model is only predicting the current status but not the future. Thus a following up project can be investigation in whether predicting the future is feasible. To do this, a time window should be selected as predictor time span, where the data is used to build predictors and predict the label at the end of a maintenance cycle. For example, we use data of two weeks to predict the remaining bush width. It should be noted that the number of days the bush in the pot will also be an changing variable. Experiments should be done to select the most suitable time window.

*-Make use of unsupervised learning and correlation tests to detect the abnormal event happens in line*
Unsupervised learning technique can be used to build monitoring interface at each section of the line. By clustering, abnormal events that happens in this section can be detected and then depend on source of the error, it gives insights on what is possibly causing the defect/line delay etc. According to literature(described in Chapter 4), the correlation among sensors signals will change before an abnormal event actually happens and that is because the distribution of the sensors signal is changing once the faulty behaviour is about to happen. Thus it is feasible in theory to detect the faulty behaviour before it actually happens. Once this is tested on one section of the line, subsystems can be build at each sections of the line and a centralized monitoring system can be build eventually to monitor the whole production line.

*-Expand the model to other sites/lines*
Providing that critical parameters have been discovered for wearing condition prediction. For the purpose of predicting the wear of the component, under the similar context, component based variables can be extracted for modeling. Thus project can be expanded to other lines for the same component or different component but the component wear is essentially due to metal contact. It is worth mentioned that the whole modeling process should be conducted as a whole from feature selection to technique

selection as things may deffer based on the data.

-*Follow similar methodology build models for other problem areas*
For different problems of predictive maintenance for instance, the goal is to predict failure or Remaining Use of Life(RUL) instead of wearing condition, or it is not based on mechanical contact etc. The same methodology can be used to discover critical variables and select suitable models based on theory and the validation in practice.

CHAPTER
# EIGHT

# CONCLUSION AND RECOMMENDATION (TO THEORY)

In this chapter, contribution to the theory is discussed. We will focus on real industrial context in transition period that differs from theoretical settings, the challenge faced and a guideline of how to deal with the challenges. The main research question is answered:
*How can data driven methods be applied to predictive maintenance in industries that are in transition to industry 4.0?*

## 8.1 Characteristics of industry in transition period

As a industry in transition to industry 4.0, maintenance decisions are based on experience. Wireless Internet connection starting to be implemented in factories and not yet expanded to all offices and moreover installing new software such as R on PCs is a very expensive process and takes long time. Technologies are being investigated and vision maps are created but are far from being integrated into operations. Sensors and cameras are being installed gradually. Different data logging servers are working separately to log and store data. Thus in general, the industry is learning and gradually changing towards automation and smart manufacturing.

## 8.2 Theory VS Practice

In most of the literature, data used for predictive maintenance studies are public available data. These data set are well structured, balanced and cleaned with complete failure samples that covers different failure situations. Some of the data sets are obtained by running machines to failure in different settings and the data set often only contains maximum hundreds of samples. However, in real industry settings, machines and components are not allowed to be failed due to high down time cost. Visualization of the raw data are often used to get a first glance impression of the data pattern in theory. But in industries, sensors produces large volumes of data that are impossible to be visualized directly because the visualization is too computationally expensive and that the raw data is too messy to be analyzed.

In our case the data are also very unbalanced while some data sources contain large volumes of sensor data and others are manually logged. Manually logged data are rounded as human can't measure a variable very accurately. Also as the sample size of our target variable is too small, most of the machine learning techniques are theoretically inapplicable. Theoretically, vibration is the direct indicator of component failure and thus most of the predictive maintenance study use vibration signals only and extract different features from the vibration signals for modeling. In industry settings, vibration

data are not always available and not even can be measured online. In our case for instance, nothing on the component can be measured as the component is in a zinc pot.

However, high dimensional variables from the system that is around the component is available. That is why discovering physical behaviour of the component is essential. There is limited amount of predictive maintenance studies are based on a mixed methodology of mathematics modeling and data driven method. But in this study because failure indicator is missing, all the predictive parameters selected are based on theoretical physical metal-metal contact wearing behaviour. Two data characteristics, namely small sample size and high dimensional variables leads to the decision of investigating in PLSR model which is barely used in maintenance context in theory but it does produce good and stable prediction result. The reason why the PLSR model is suitable to be used here is that the core model behind PLSR is linear regression and the variables we selected also have a linear relation with the target variable. Thus we can conclude that data driven predictive modeling methods should really be selected based on existing data characteristics and relations.

Moreover, unlike theoretical studies, implementing predictive analytics into practice always requires model maintenance. Feature selection and technique selection process have to be repeated together with model training to be a dynamic process to maintain predictive performance. Human logged data source has to be consistent with calculation and rules. Small inconsistency will lead to different modeling result.

## 8.3 Challenges

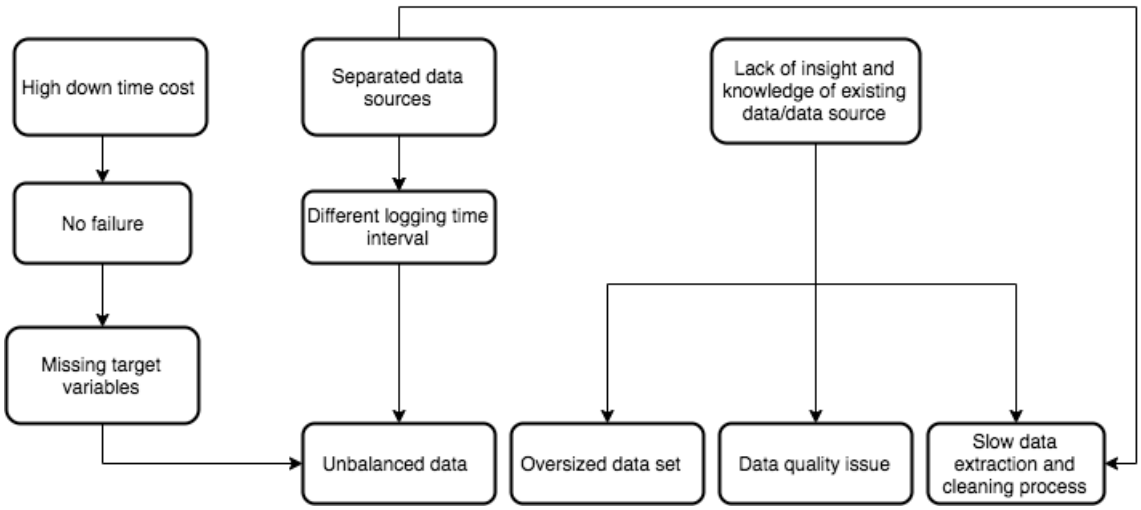


Figure 8.1: challenges

The challenges and the cause of those challenges are presented in Figure 8.1. No down time allowed in production directly leads to the first challenge: target variables missing. As solution, failure indicator can be investigated and measured as a target variable. In terms of wearing prediction of critical components, wearing measures were measured online. As the target variable is measured online, the time span of usable data is already reduced because historical data without corresponding target variable can't be used for predictive analytics. In addition, since the data sources are separated as well as the sensor logging difference, the data logging time interval from each data source is very different. Then it leads to the second challenge: Highly unbalanced data. In our specific case we have data source

that loggs billions of data per day and data source that log one observation per month. In transition period, industries don't have a big picture of the availability and quality of the data and employees tend to lack knowledge of the functions of newly implemented data logging devices, which leads to the third challenge: Data size and quality issues. Without a integrated data source, data has to be extracted from different sources separately and thus some additional softwares need to be installed for data extraction purpose. After that the IT department has to learn a new software before the data can be actually extracted. Moreover, data format from different data source differs. These leads to slow data extraction and cleaning process.

Over all, when applying data-driven method in industries that in transition period, data availability, data quality and data cleaning are the main part in which challenges may occur. Finding meaningful predictive variables is critical to the success of predictive analytics. Oversized sensor data makes cleaning process occupy most of the time of the whole project. Data cleaning process has to be carefully done in terms of data format and should look back to the raw data to make sure it is flawless. Because numerical raw data sometimes is not recognized by cleaning tools as "numerical" they can be read as "factors" or "character". When it is recognized as "factors" it can't be converted to numerical directly.

## 8.4 Limitation to theory and future research

The context of 'industry in transition period to industry 4.0' is merely based on case study of TATA Shotton which is hard to be generalized as different industries are in different stages. A broad survey of industry in transition is required before we can have a thorough view on the characteristic of the industries in transition. Furthermore, the research only looks at predictive maintenance in a technical perspective. There are more aspects to consider. In (Briefing, 2015) from EU Parliament, the following challenges are presented:
*-Investment and change*
*-Data ownership and security*
*-Legal issues*
*-Standards*
*-Employment and skills development*
It has been addressed by domain experts that the biggest challenge is to find the people with the right skills. These challenges must all be overcome before industry 4.0 is accomplished. Thus research should be conducted in different perspectives in terms of data-driven predictive maintenance to provide a clearer view of the current general status and how far before we can complete the transition.

# BIBLIOGRAPHY


[1] Tata Steel(16 May 2018), Statement of profit and loss

[2] Tata Steel Shotton Fact sheet

[3] Forbes(2015), What IT Needs To Know About The Data Mining Process

[4] Shearer C.(2000), The CRISP-DM model: the new blueprint for data mining, Data Warehousing,13—22.

[5] Shmueli, Koppius(2011), Predictive analytics in information system research p553-572.

[6] Sniderman, Mahto, Cotteleer(2019), Industry 4.0 and manufacturing ecosystems Exploring the world of connected enterprises

[7] Zschech, Heinrich, Bink, S.Neufeld(2019), Prognostic Model Development with Missing Labels - A Condition-Based Maintenance Approach Using Machine Learning

[8] Liao Z, Gao D, Lu Y, Lv Z (2016), Multi-scale hybrid HMM for tool wear condition monitoring. Int J Adv Manuf Technol

[9] Yu J, Liang S, Tang D, Liu H (2016), A weighted hidden Markov model approach for continuous-state tool wear monitoring and tool life prediction.

[10] Osama Abdeljaber a, Onur Avci a,n, Serkan Kiranyaz b, Moncef Gabbouj c, Daniel J. Inman (2017), Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks.

[11] R. Teti, K. Jemielniak, G. O'Donnell c, D. Dornfeld (2010), Advanced monitoring of machining operations

[12] Nagdev Amruthnath, Tarun Gupta (2018), Fault Class Prediction in Unsupervised Learning Using Model-Based Clustering Approach

[13] B.K.N Rao, P. Srinivasa Pai and T.N. Nagabhushana (2012), Failure Diagnosis and Prognosis of Rolling - Element Bearings using Artificial Neural Networks: A Critical Overview

[14] Veronica Bolon-Canedoz, Noelia Sanchez-Marono, Amparo Alonso-Betanz(2013), A review of feature selection methods on synthetic data

[15] Brad Cline, Radu Stefan Niculescu, Duane Huffman, Technology Corporation, Bob Deckel(2017), Predictive Maintenance Applications for Machine Learning

[16] Adrian Cubillo, Suresh Perinpanayagam and Manuel Esperon-Miguez (2016), A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery

[17] So Young Hwang, Na Ra Lee and Naksoo Kim (2015), Experiment and Numerical Study of Wear in Cross Roller Thrust Bearings

[18] Lukas Spendla, Michal Kebisek, Pavol Tanuska, Lukas Hrcka (2017), Concept of Predictive Maintenance of Production Systems in Accordance with Industry 4.0

[19] Rocco Langone, Carlos Alzate, Bart De Ketelaere, Johan A.K.Suykens1(2013), Kernel spectral clustering for predicting maintenance of industrial machines

[20] Turgay Kıvak(2014), Optimization of surface roughness and flank wear using the Taguchi method in milling of Hadfield steel with PVD and CVD coated inserts

[21] P Krishnakumara, K Rameshkumarb, K I Ramachandranc(2015), Tool Wear Condition Prediction Using Vibration Signals in High Speed Machining(HSM) of Titanium (Ti-6Al-4V) Alloy

[22] Zhe Li, Yi Wang, Ke-Sheng Wang(2017), Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario

[23] Giovanna Martinez-Arellanol, German Terrazas, Svetan Ratchev(2019), Tool wear classification using time series imaging and deep learning

[24] Jinjiang Wanga, Peng Wangb, Robert X. Gaob(2015), Enhanced particle filter for tool wear prediction

[25] Diego Tobon-Mejia, Kamal Medjaher, Noureddine Zerhouni, Ggerard Tripot(2012), A data-driven failure prognostics method based on mixture of gaussians hidden markov models

[26] Nagdev Amruthnath, Tarun Gupta(2018), A Research Study on Unsupervised Machine Learning Algorithms for Early Fault Detection in Predictive Maintenance

[27] Rocco Langone, Carlos Alzate, Bart De Ketelaere, Jonas Vlasselaer, Wannes Meert, Johan A.K. Suykens(2015), LS-SVM based spectral clustering and regression for predicting maintenance of industrial machines

[28] K.Velten, R.Reinicke, K.Friedrich(2000), Wear volume prediction with artificial neural networks

[29] Dazhong Wu, Connor Jennings, Janis Terpenny, Soundar Kumara(2016) Cloud-Based Machine Learning for Predictive Analytics: Tool Wear Prediction in Milling

[30] Zhengyou Xie1, Jianguang Li1, Yong Lu1(2019), Feature selection and a method to improve the performance of tool condition monitoring

[31] Jihong Yan, Yue Meng, Lei Lu, Lin Li(2017), Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance

[32] Pushe Zhao, Masaru Kurihara, Junichi Tanaka, Tojiro Noda, Shigeyoshi Chikuma, Tadashi Suzuki(2017), Advanced Correlation-Based Anomaly Detection Method for Predictive Maintenance

[33] Rui Zhao, Ruqiang Yan, Jinjiang Wang, Kezhi Mao(2017), Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks

[34] Patrick Zschech, Kai Heinrich, Raphael Bink, Janis S. Neufeld(2019), Prognostic Model Development with Missing Labels A Condition-Based Maintenance Approach Using Machine Learning

[35] Shisheng Zhong, Hui Luo, Lin Lin, Xuyun Fu(2016), An Improved Correlation-Based Anomaly Detection Approach for Condition Monitoring Data of Industrial Equipment

[36] Dazhong Wu, Connor Jennings, Janis Terpenny, Robert X.Gao, Soundar Kumara(2017), A comparative study on machine learning algorithms for smart manufacturing: Tool wear prediction using random forest

[37] Hongfei Li, Dhaivat Parikh, Qing He, Buyue Qian, Zhiguo Li, Dongping Fang, Arun Hampapur(2014), Improving rail network velocity: A machine learning approach to predictive maintenance

[38] Hongfei Li, Buyue Qian, Dhaivat Parikh, Arun Hampapur(2013), Alarm Prediction in Large-Scale Sensor Networks-A case study in railroad

[39] Gian Antonio Susto, Andrea Schirru, Simone Pampuri(2015), Machine Learning for predictive maintenance: A multiple classifier approach

[40] Abhinav Saxena, Ashraf Saad(2007), Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems

[41] T.Kolodziejczyk, R.Toscano, S.Fouvry, G.Morales-Espejel(2010), Artificial intelligence as efficient technique for ball bearing fretting wear damage prediction

[42] Kamran Javed, Rafael Gouriveau, Ryad Zemouri, Xiang Li(2012), Robust, reliable and applicable tool wear monitoring and prognostic: approach based on an Improved-Extreme Learning Machine

[43] Yongzhen Zhuang, Lei Chen, X.Sean Wang, Jie Lian(2007), A Weighted Moving Average-based Approach for Cleaning Sensor Data

[44] Karoly Heberger(2008), Chemoinformatics—multivariate mathematical–statistical methods for data evaluation in Medical Applications of Mass Spectrometry

[45] Swathik Clarancia Peter, Durai Sundar(2019), Quantitative Structure-Activity Relationship (QSAR): Modeling Approaches to Biological Applications

[46] Bjorn-Helge Mevik, Ron Wehrens Biometris, Wageningen (October, 2019) Introduction to the pls Package

[47] Xie Li,Weiru Xiang,Jiang Tie jun,Zhang Ping(2009), Generalized PLS Regression Forecast Modeling of Warship Equipment Maintenance Cost

[48] Rosa L Figueroa Qing Zeng-Treitler, Sasikiran Kandula and Long H Ngo(2012) Predicting sample size required for classification performance

[49] Gerard Biau, Erwan Scornet(2016), A Random Forest Guided Tour

[50] Jian Qina, Ying Liua, Roger Grosvenora(2016), A Categorical Framework of Manufacturing for Industry 4.0 and Beyond

[51] Jian Qina, Ying Liua, Roger Grosvenora(2016), A Categorical Framework of Manufacturing for Industry 4.0 and Beyond

[52] Susana Ferreiro, Egoitz Konde, Santiago, Agustin Prado(2016), INDUSTRY 4.0: Predictive Intelligent Maintenance for Production Equipment

[53] P.O'Donovan, K.Leahy, K.Bruton and D.T.J.O'Sullivan(2015), An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities

[54] Alexandros Bousdekis, Katerina Lepenioti, Dimitrios Ntalaperas, Danai Vergeti, Dimitris Apostolou, Vasilis Boursinos(2019), A RAMI 4.0 View of Predictive Maintenance: Software Architecture, Platform and Case Study in Steel Industry

[55] Bjørn-Helge Mevik, Ron Wehrens(2007), The pls Package: Principal Component and Partial Least Squares Regression in R

APPENDIX

**A**

# DATA DICTIONARY

Lists of variables that can be collected from the data sources are shown in this section. The variables are collected from data warehouse, IBA and Setup sheet. Data warehouse variable can be found in Table A.1. Variables from IBA can be found in Table A.2. Variables from set up sheet can be found in Table A.3.

Table A.1: Variables from Data Warehouse

| **Data(Data Warehouse)** | **Type** | **Explanation** |
|---|---|---|
| DATE | Date | Date |
| TIME | Time | Time |
| BATCH_KEY | Numerical | Coil |
| MATERIAL | Numerical | Material ID |
| WTH | Numerical | Coil Width |
| PROCESS_TIME_ACTUAL | Numerical | Actual Prcessing time |
| PROCESS_TIME_STANDARD | Numerical | Standard Processing time |
| SOW | Numerical | Speed of work ratio speed actual/std_speed |
| STD_SPEED | Numerical | Standar speed |
| FIN_COIL_WGT_CUST_MAX | Numerical | Weight range the weight |
| SUBSTRATE_COIL_WGT | Numerical | Incoming weight |
| FIN_COIL_WGT_CALC | Numerical | Finished weight esitmated |
| EXIT_SCRAP_WGT | Numerical | Estimated scrap weight |
| DEGRADE_WEIGHT | Numerical | None prime material |
| FIN_COIL_WGT_DELIVERED | Numerical | Actual weight zero means it hasn't been delivered |
| DECISION | Categorical | Category the material. |
| IL_USAGE | Numerical | Category the material goes into corresponds to "Decision" |
| FIN_COIL_LENGTH | Numerical | Finished coil length |
| FIN_COIL_LENGTH_PRIME | Numerical | Finished coil prime length |
| EXIT_SCRAP_LENGTH | Numerical | The amount of the coil that has been cut |
| FIN_COIL_CROSS_SECTION | Numerical | Width of the coil multiply thickness |
| FIN_COIL_SURFACE_AREA | Numerical | Finished coil surface area |
| FIN_COIL_WIDTH_ACTUAL | Numerical | Finished coil actual width |
| FIN_COIL_GAUGE_ACTUAL | Numerical | Finish coil thickness |
| SUBSTRATE_GRADE_RECEIVED | Numerical | Actual material arrived |
| SUBSTRATE_WIDTH_ORDERED | Numerical | Width of the coil that ordered |
| SUBSTRATE_WIDTH_RECEIVED | Numerical | Width of the coil received |
| SUBSTRATE_GAUGE_ORDERED | Numerical | Thickness of the coil ordered |
| SUBSTRATE_GAUGE_RECEIVED | Numerical | Thickness of the coil received |
| FIN_GAUGE_ORDERED | Numerical | Thickness of the coil customer asked for |
| FIN_WIDTH_ORDERED | Numerical | Width of the coil customer asked for |
| FIN_WIDTH_TOL_ORDERED | Numerical | The amount of deviation that is allowed |
| FIN_COATING_WGT_ORDERED | Numerical | The amount of zinc coated asked by the customer |
| FIN_SURFACE_ORDERED | Numerical | MC smoothest, MA the least smooth |
| GGE-ORD | Numerical | Ordered gage(intended to make) |
| GGE-IN | Numerical | In coming coil gage |
| CW | Numerical | Coating weight(grams) |
| GRD | Numerical | Grade(property of coil, strength of the coil) |
| GRD-IN | Numerical | Grade in |
| SF | Catergorial | Surface finish |
| C | Catergorial | Crown:the curve of the coil,L:low curve N:normal |

| Data(Data Warehouse) | Type | Explanation |
|---|---|---|
| SP | Numerical | Line speed |
| AS | Numerical | Historical average speed |
| SS | Numerical | Standard line speed |
| SV | categorical | Line speed varies |
| LS | categorical | Line stop |
| ST-Z3X | Numerical | Strip temperature 23 Exit |
| ST-RTX | Numerical | Strip temperature RT Exit |
| ST-JCX | Numerical | Strip temperature Jet cool Exit |
| ST-SCX | Numerical | Strip temperature slow cool Exit |
| ST-TM | Numerical | Strip temperature temper mill |
| SBT-DIF | Numerical | Strip and bath temperature difference |
| BT-GV | Numerical | Bath temperature Galv |
| BT-GF | Numerical | Bath temperature GalvFan |
| BMH | Numerical | Bath metal Height |
| DPS | Numerical | Dew Point Snout |
| DRT | Numerical | Deflection Roll Temperature |
| GST | categorical | Cwgt Online Gauge Status |
| CWG | Numerical | Cwgt Online Gauge the sum of median top and median bottom |
| WSW | Numerical | Cwgt WsW result |
| CTA | categorical | Cwgt Alarm |
| X-MM | Numerical | Extension Matt Mill |
| X-TL | Numerical | Extention Tension Leveller |
| TN-B21 | Numerical | Tension Bridle 2/1 |
| TN-B22 | Numerical | Tension Bridle 2/2 |
| TN-B4 | Numerical | Tension Bridle 4 |
| TN-EX | Numerical | Tension Exit |
| GT-Z3 | Numerical | Gas Temperature Zone 3 |
| GT-Z4 | Numerical | Gas Temperature Zone 4 |
| GT-Z5 | Numerical | Gas Temperature Zone 5 |
| ZT-3A | Numerical | Zone Temperature 3A |
| ZT-6 | Numerical | Zone Temperature 06 |
| ZT-7 | Numerical | Zone Temperature 07 |
| ZT-8 | Numerical | Zone Temperature 08 |

Table A.2: Variables from IBA

| **Data (IBA)** | **Data Type** | **Explanation** |
|---|---|---|
| Date | Date | Date |
| Time | Time | Date |
| input.lineSpeedRaw | Numerical | Line Speed |
| input.lineSpeedRef | Numerical | Reference line speed |
| input.knifeHorzPos[TOP] | Numerical | Knife horizontal top position |
| input.knifeHorzPos[BOT] | Numerical | Knife horizontal position bottom |
| input.knifeSkewPos[LEFT] | Numerical | Knife Skew left position |
| input.knifeSkewPos[RIGHT] | Numerical | Knife Skew right position |
| input.knifeHeight | Numerical | Knife height |
| input.StripTensionRef | Numerical | Strip tension reference |
| input.stripTension | Numerical | Strip tension |
| :mean.knifeHeight | Numerical | Average Knife height |
| input2.bafflesClosed[LEFT} | Binary | If left baffles are closed |
| input2.bafflesClosed[RIGHT} | Binary | If right baffles are closed |
| input.knifeClean[TOP] | Binary | If top knife is cleaned |
| input.knifeClean[BOT] | Binary | If bottom knife is cleaned |
| input.knifeHeightNomAvg | Numerical | Knife height normalized average |
| input.corrRollPos[LEFT] | Numerical | Correcting roll position (left) |
| input.corrRollPos[RIGHT] | Numerical | Correcting roll position (right) |
| input.potTemp | Numerical | Pot temperature |
| input.stripTemp | Numerical | Strip temperature |
| input.potAlContent | Numerical | Pot aluminum content |
| input.controlPress[TOP] | Numerical | Control press from top |
| input.controlPress[BOT] | Numerical | Control press from bottom |

| Data (IBA) | Data Type | Explanation |
|---|---|---|
| input.corrRollIn[LEFT] | Binary | Not clear about the meaning |
| input.corrRollIn[RIGHT] | Binary | Not clear about the meaning |
| input.corrRollOut[LEFT] | Binary | Not clear about the meaning |
| input.corrRollOut[RIGHT] | Binary | Not clear about the meaning |
| input.rigDismantled | Binary | Not clear about the meaning |
| input.acEnabledAngle | Binary | Not clear about the meaning |
| input.acEnabledCorrRoll | Binary | Not clear about the meaning |
| input.acEnabledHeight | Binary | Not clear |
| input.acEnabledKnifePos | Binary | Not Clear |
| input.acEnabledPress | Binary | Not clear |
| input.coatMass[TOP] | Numerical | Amount of coating on top |
| input.coatMass[BOT] | Numerical | Amount of coating bottom |
| input.gaugePos[TOP] | Numerical | Gauge position top |
| input.gaugePos[BOT] | Numerical | Gauge position bottom |
| input.headerPressRaw[TOP] | Numerical | Header press from top |
| input.headerPressRaw[BOT] | Numerical | Header press from bottom |
| input.headerPress[TOP] | Numerical | Potentially repeating variables |
| input.headerPress[BOT] | Numerical | Potentially repeating variables |
| input.knifeAngle[TOP] | Binary | If knife top has an angle |
| input.knifeAngle[BOT] | Binary | If knife bottom has an angle |
| input.pressUp[TOP] | Binary | Press up from top |
| input.pressUp[BOT] | Binary | Press up from bottom |
| input.pressDown[TOP] | Binary | Press down from top |
| input.pressDown[BOT] | Binary | Press down from bottom |
| ZT-9 | Numerical | Zone Temperature 09 |
| ZT-10 | Numerical | Zone Temperature 10 |
| ZT-11 | Numerical | Zone Temperature 11 |
| ZT-12 | Numerical | Zone Temperature 12 |
| ZT-13 | Numerical | Zone Temperature 13 |
| ZT-14 | Numerical | Zone Temperature 14 |
| FP | Numerical | Furnace Pressure |
| TMR | Numerical | Temper Mill Load Reference |
| TMT | Numerical | Temper Mill Total |
| TMD | Numerical | Temper Mill Load Difference |
| TMB | Numerical | Temper Mill Load Bending |
| BTH-AL | Numerical | Bath Aluminium |
| RK | Numerical | Rockwell |
| WSW-TO | Numerical | Difference WSW Top up |
| WSW-BO | Numerical | Difference WSW Bottom up |
| WSW-TC | Numerical | Difference WSW Top center |
| WSW-BC | Numerical | Difference WSW Bottom center |
| WSW-TD | Numerical | Difference WSW Top down |
| WSW-BD | Numerical | Difference WSW Bottom up |

Table A.3: Variables from Setup sheet

| **Data (Setup sheet)** | **Data Type** | **Explanation** |
|---|---|---|
| Date | Date | Date |
| Time | Time | Time |
| Sink ID | categorical | Sink roll ID (A,C,G,F) |
| Stab ID | categorical | Stabilizing roll ID(A,D,G,H,E) |
| Leg ID | categorical | Leg ID(R252, R352, R152, S 1R3) |
| Sink Diameter | Numerical | Sink roll diameter |
| Sink End Float | Numerical | Sink roll end float |
| Stab Roll ID | Categorical | Stabilizing roll ID(A,D,G,H) |
| Stab Roll Diameter | Numerical | Stabilizing roll diameter |
| Stab End Float | Numerical | Stabilizing roll end float |
| Sleeves Coated | Binary | If Sleeves are coated (0,1) |
| Bushes New | Binary | If bushes are new (0,1) |
| Back stop | Numerical | Back stop distance |
| Sink Run out | Numerical | Wearing on sink roll |
| Stab Run out | Numerical | Wearing on Stabilizing roll |
| Bush Remaining left | Numerical | Left bush remaining |
| Bush Remaining right | Numerical | Right bush remaining |

APPENDIX

# B

# DATA EXPLORATION PLOT

Data exploration related plots are shown in this section. Each variable that are considered representative by PCA analysis is further analyzed by extracting 8 statistical features from them and do PCA again.

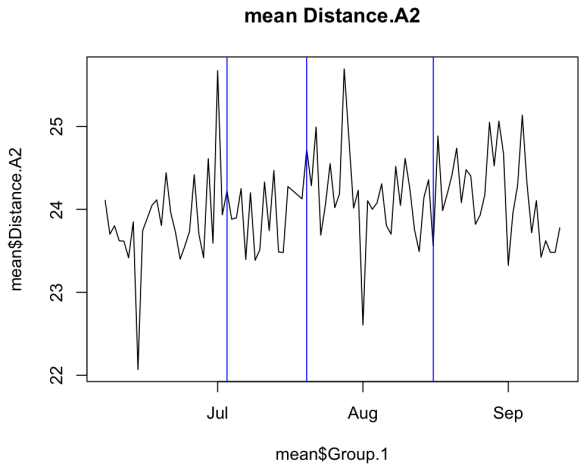


Figure B.1: Mean A2 plot

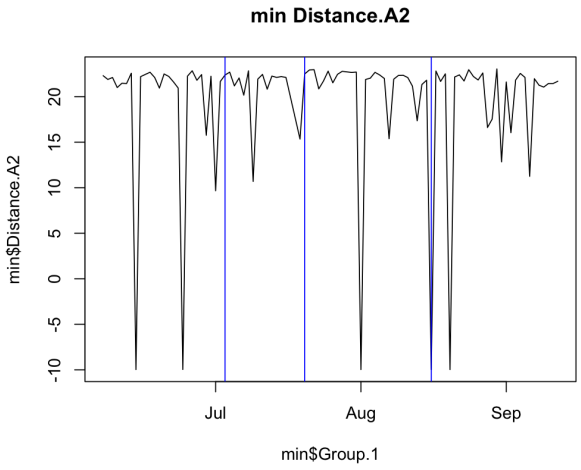


Figure B.2: Minimum A2 plot

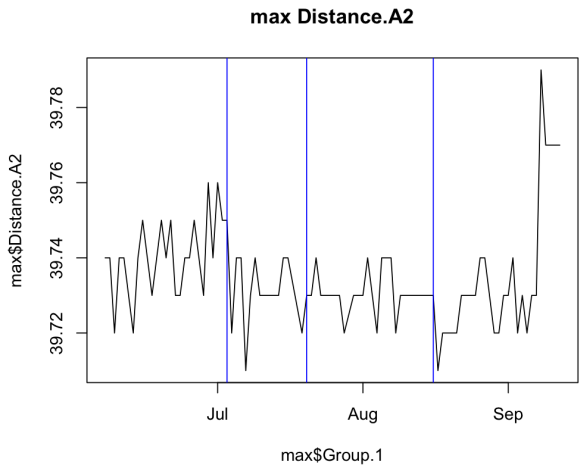


Figure B.3: Max A2 plot

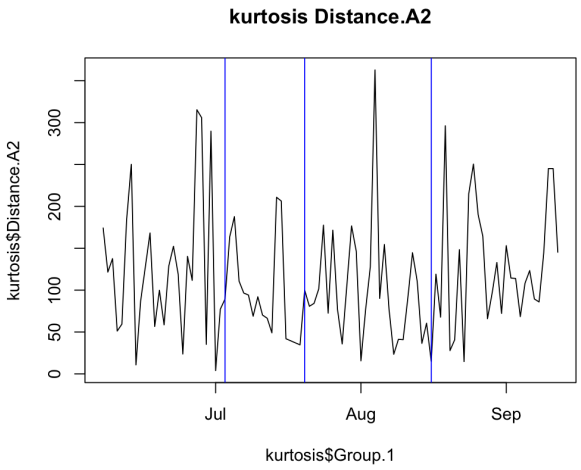


Figure B.4: Kurtosis A2 plot

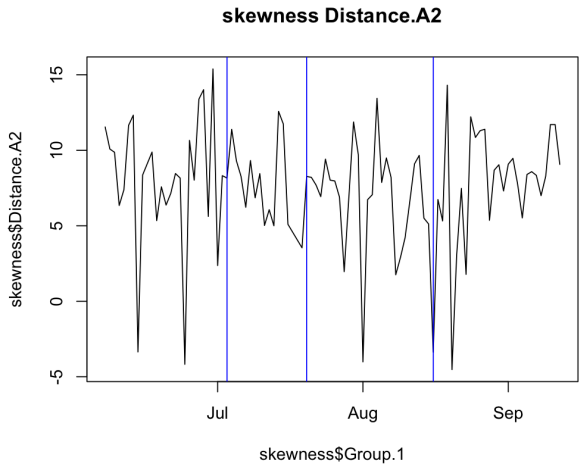


Figure B.5: Skewness A2 plot

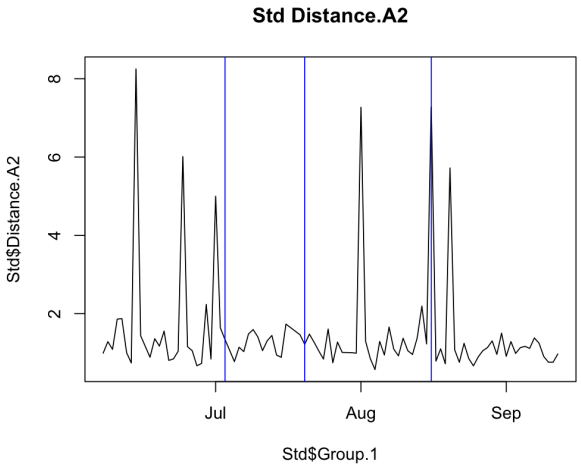


Figure B.6: Std A2 plot

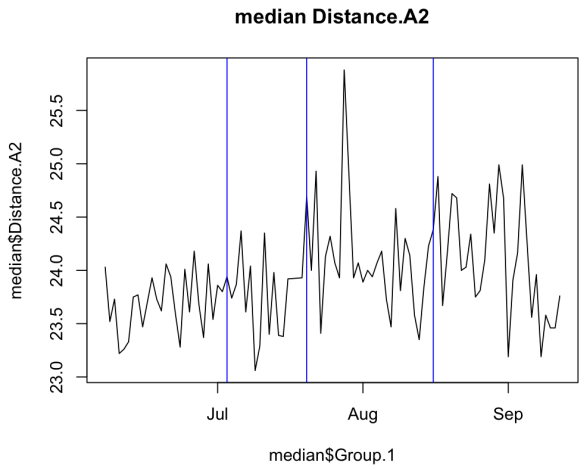


Figure B.7: Median A2 plot

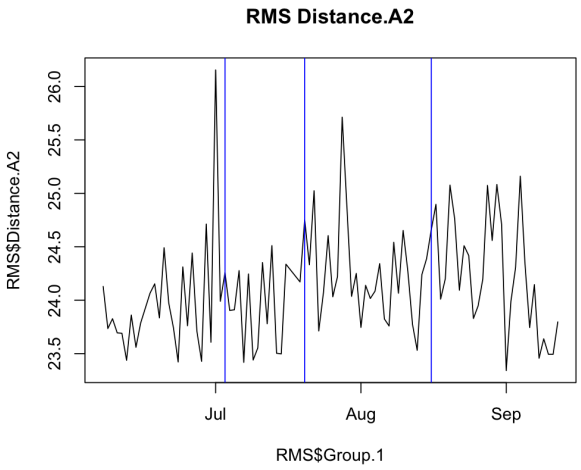


Figure B.8: RMS A2 plot

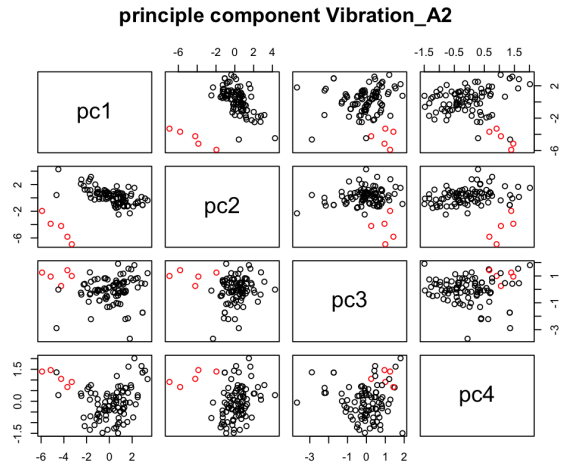


Figure B.9: Hcluster A2 plot

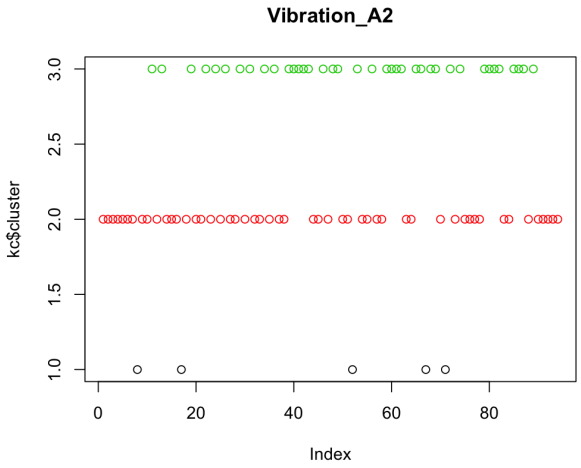


Figure B.10: K means A2 plot

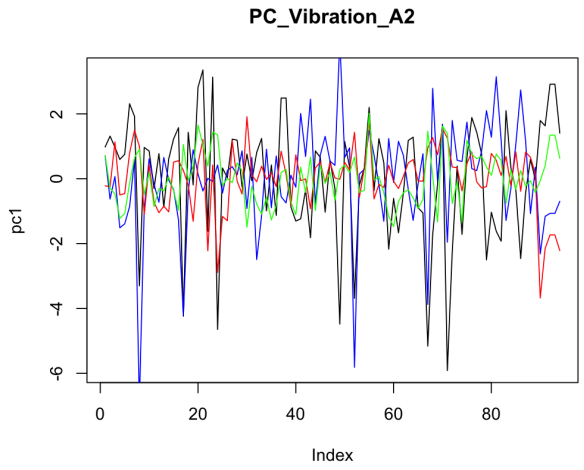


Figure B.11: Principle component A2 plot

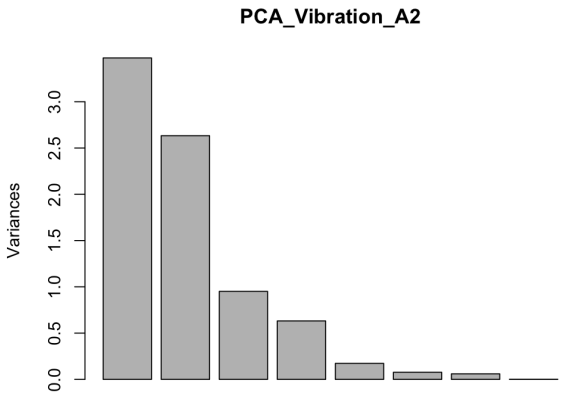


Figure B.12: PCA A2 plot

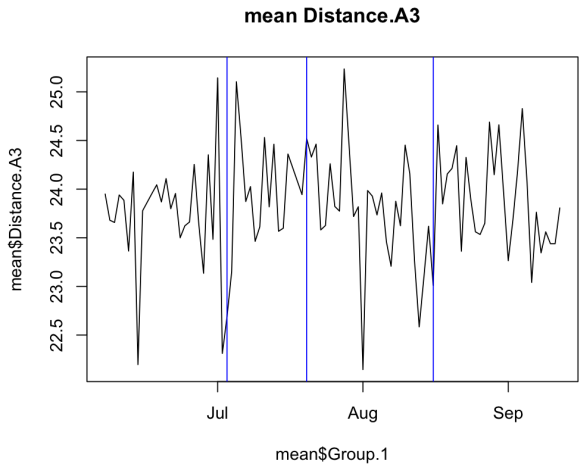


Figure B.13: Mean A3 plot

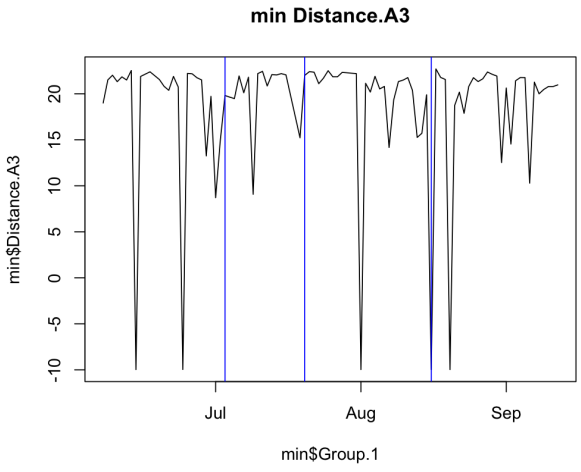


Figure B.14: Minimum A3 plot

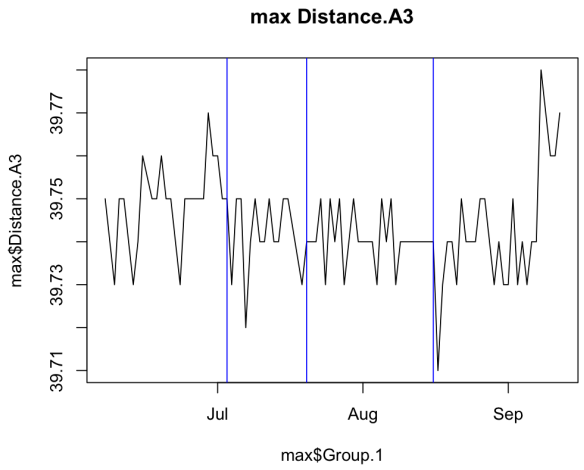


Figure B.15: Max A3 plot

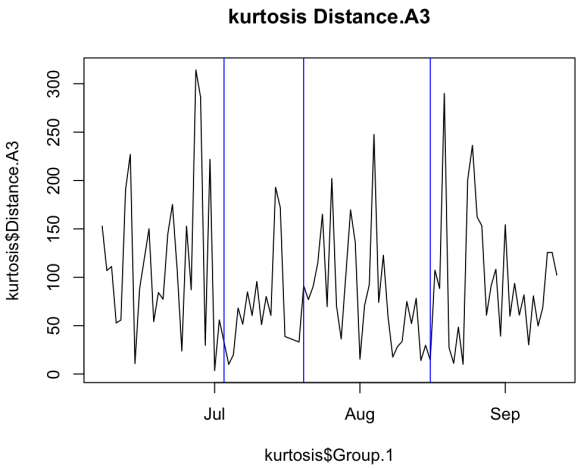


Figure B.16: Kurtosis A3 plot

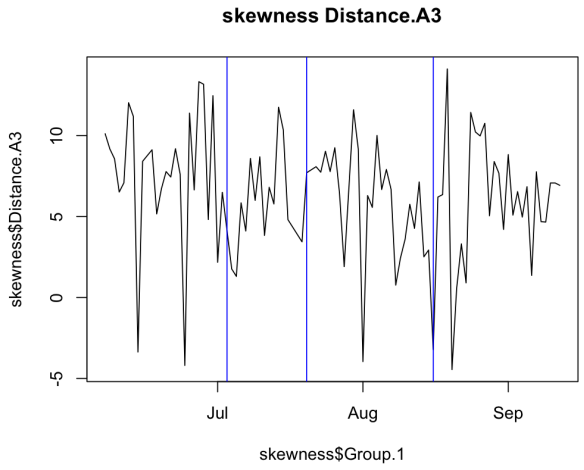


Figure B.17: Skewness A3 plot

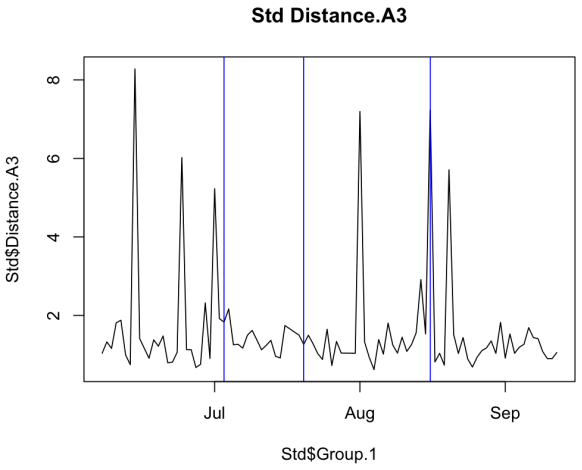


Figure B.18: Std A3 plot

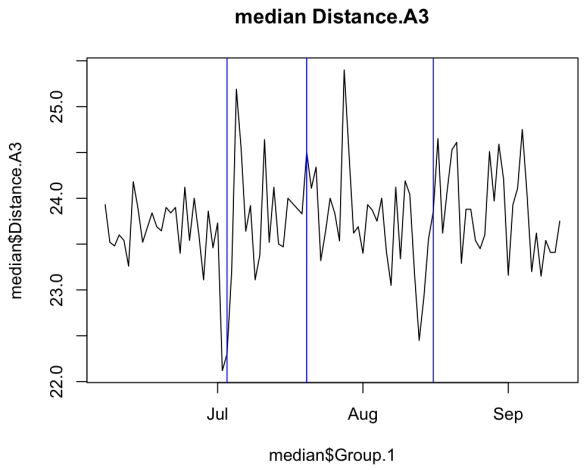


Figure B.19: Median A3 plot

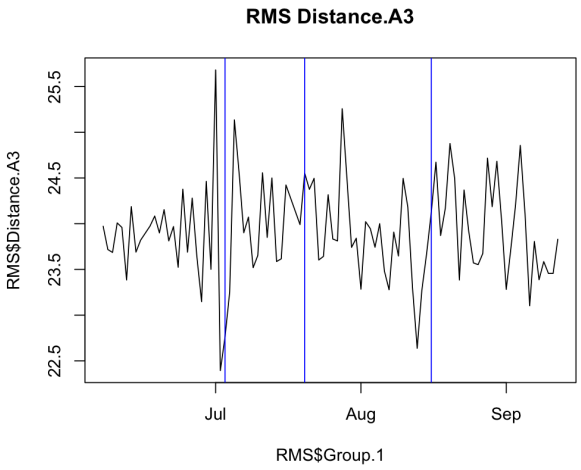


Figure B.20: RMS A3 plot

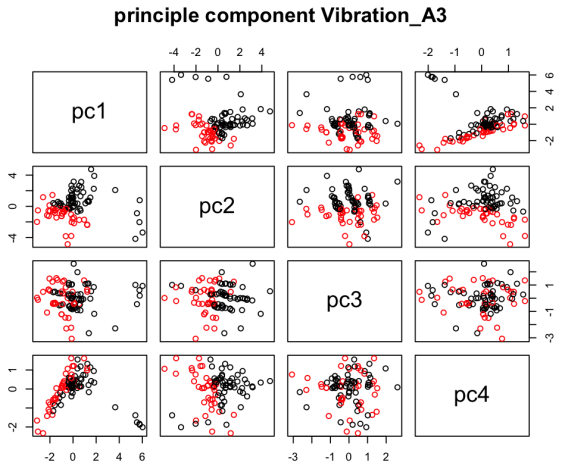


Figure B.21: Hcluster A3 plot

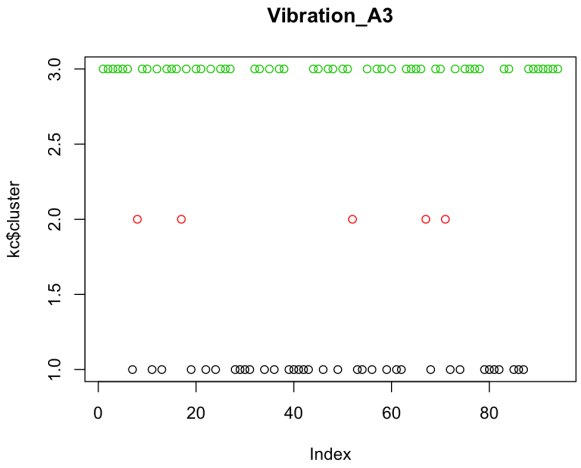


Figure B.22: K means A3 plot

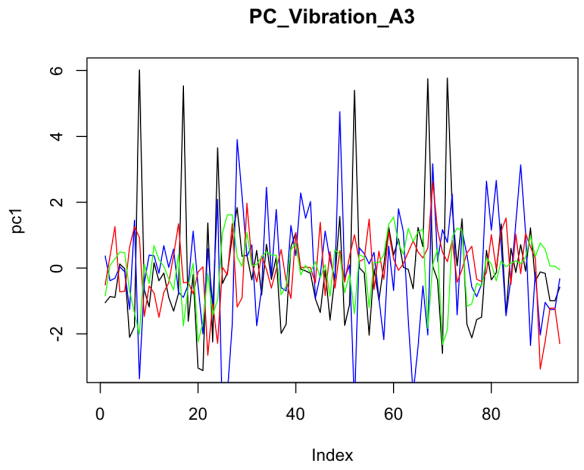


Figure B.23: Principle component A3 plot

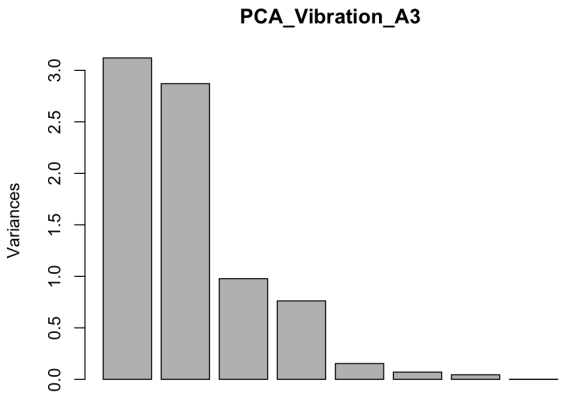


Figure B.24: PCA A3 plot

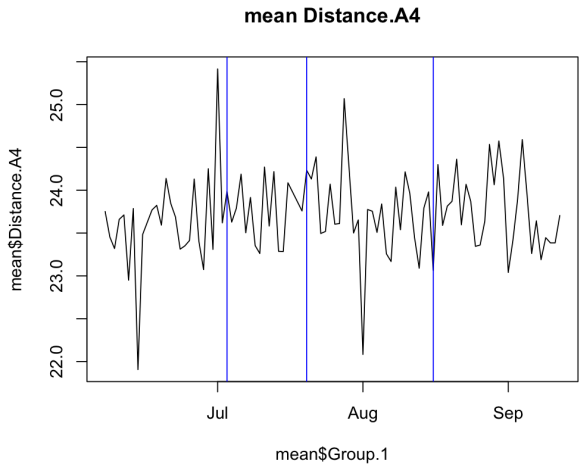


Figure B.25: Mean A4 plot

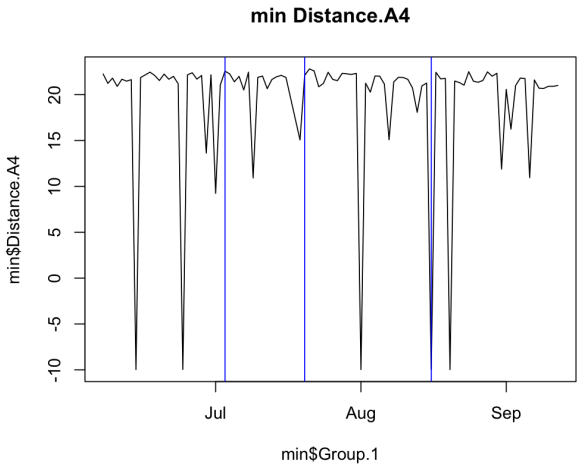


Figure B.26: Minimum A4 plot

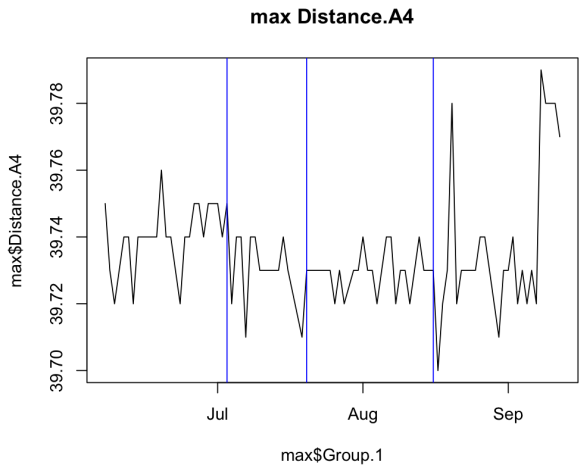


Figure B.27: Max A4 plot

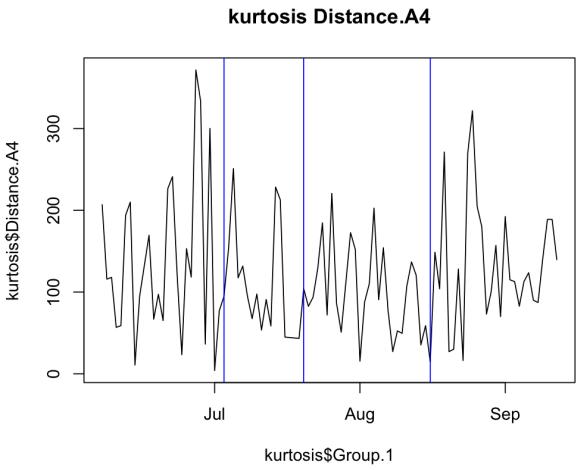


Figure B.28: Kurtosis A4 plot

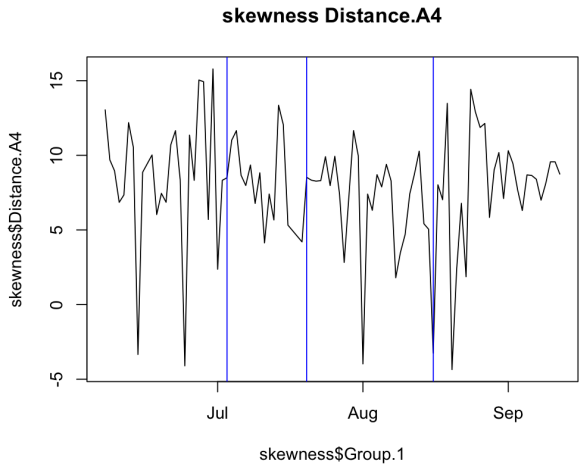


Figure B.29: Skewness A4 plot

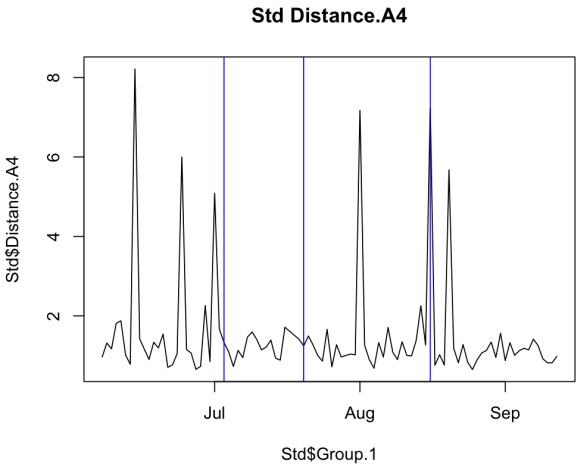


Figure B.30: Standard deviation A4 plot

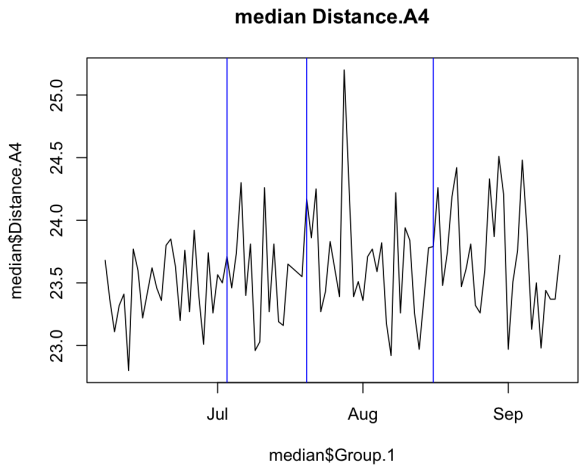


Figure B.31: Median A4 plot

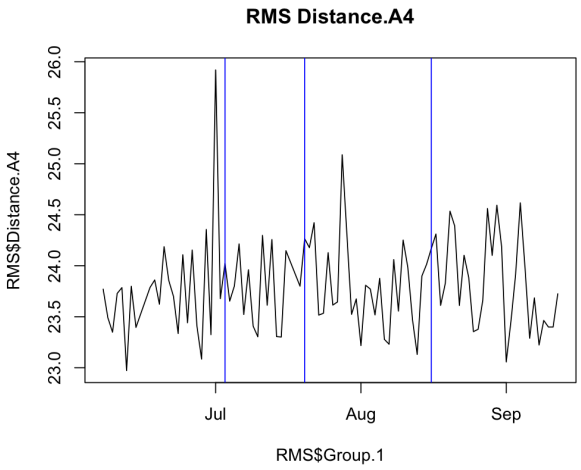


Figure B.32: RMS A4 plot

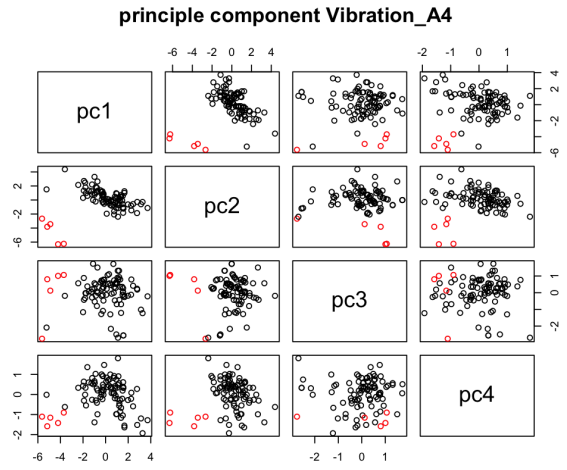


Figure B.33: Hcluster A4 plot

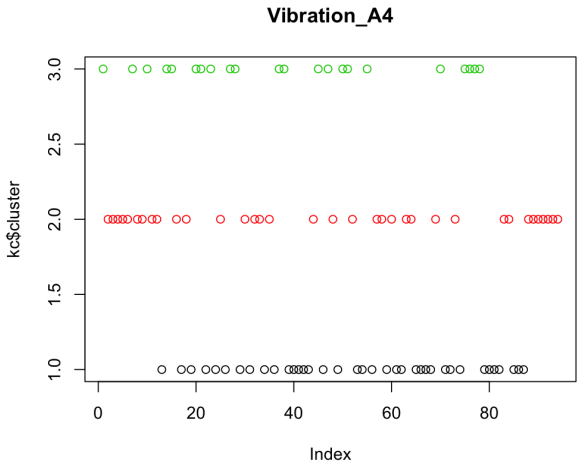


Figure B.34: K means A4 plot

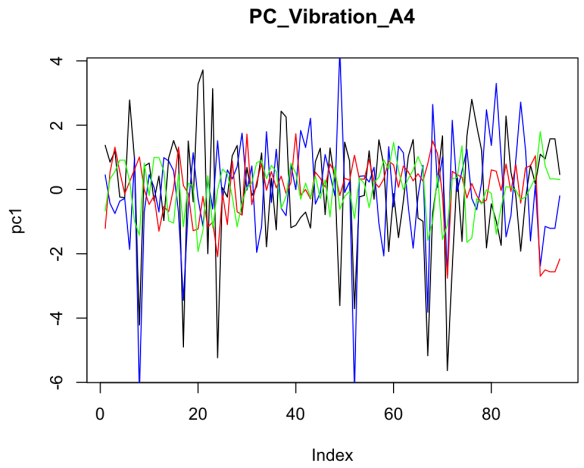


Figure B.35: Principle component A4 plot

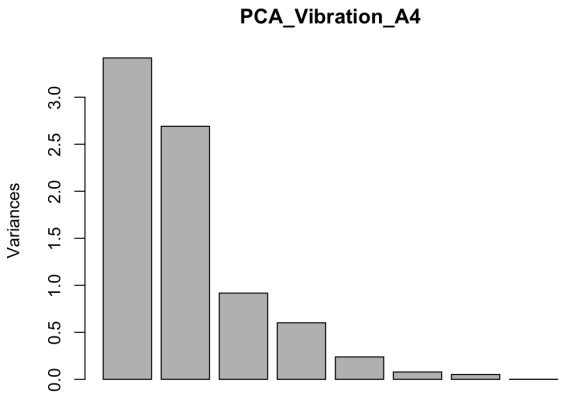


Figure B.36: PCA A4 plot

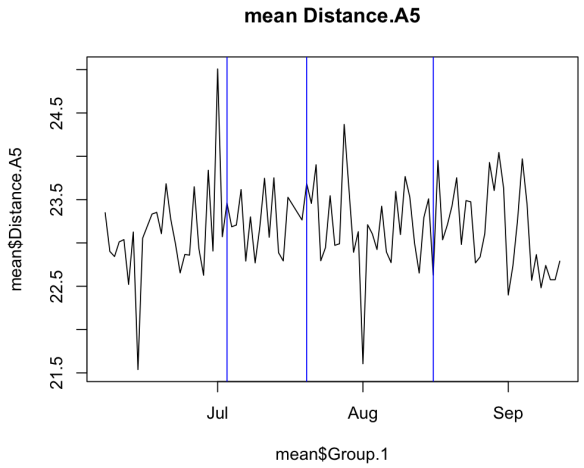


Figure B.37: Mean A5 plot

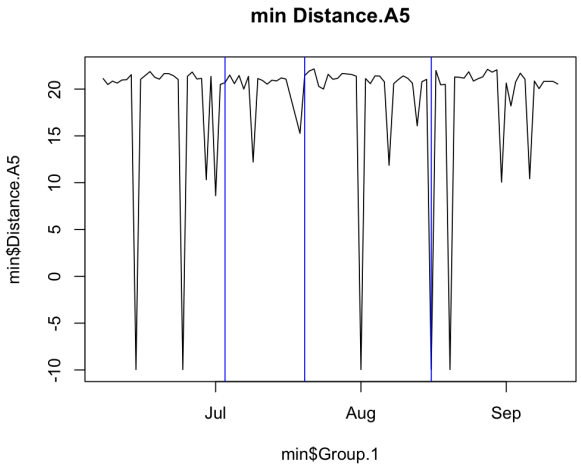


Figure B.38: Minimum A5 plot

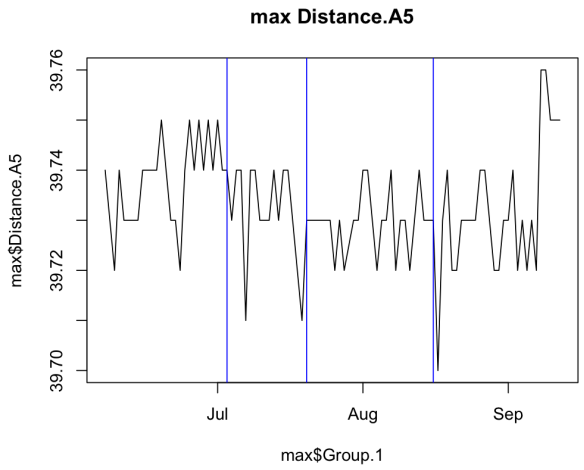


Figure B.39: Max A5 plot

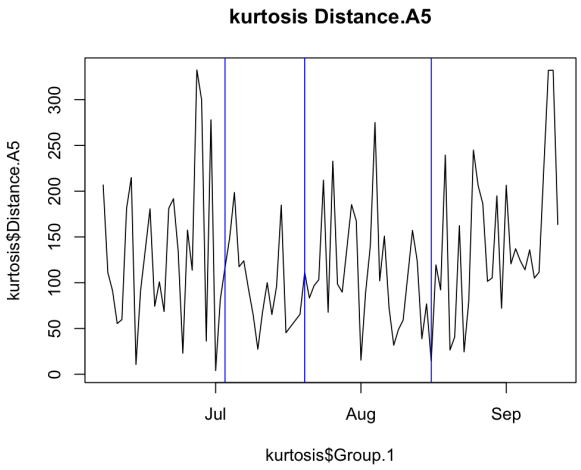


Figure B.40: Kurtosis A5 plot

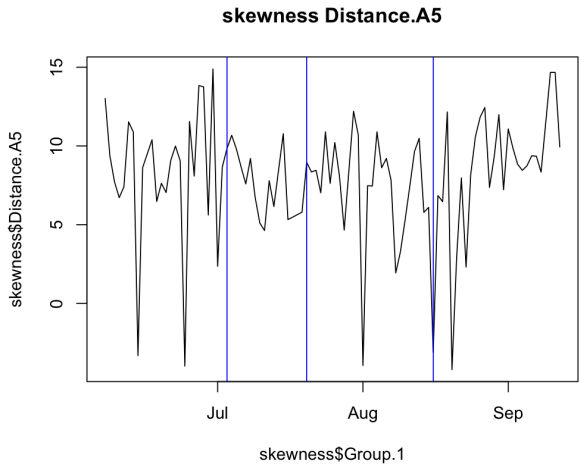


Figure B.41: Skewness A5 plot

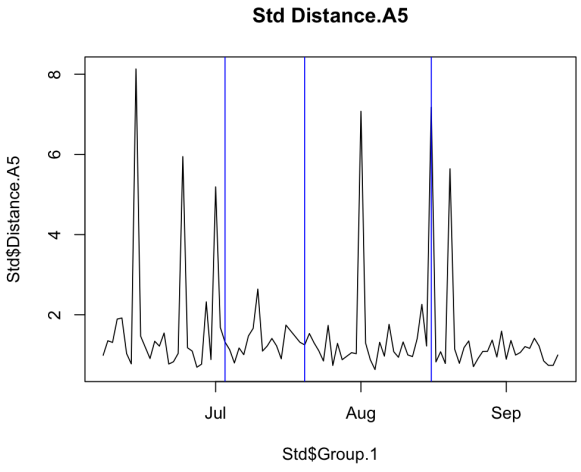


Figure B.42: Std A5 plot

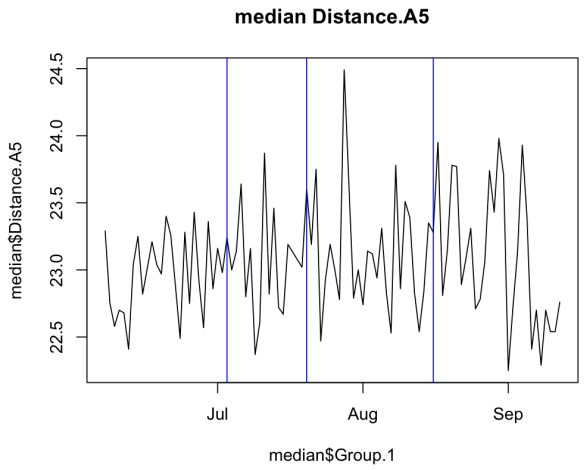


Figure B.43: Median A5 plot

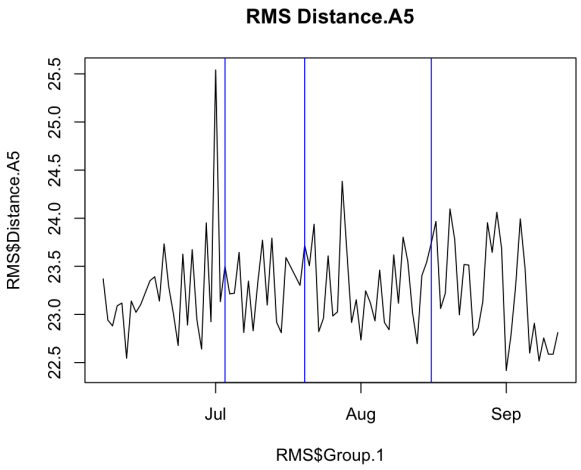


Figure B.44: RMS A5 plot

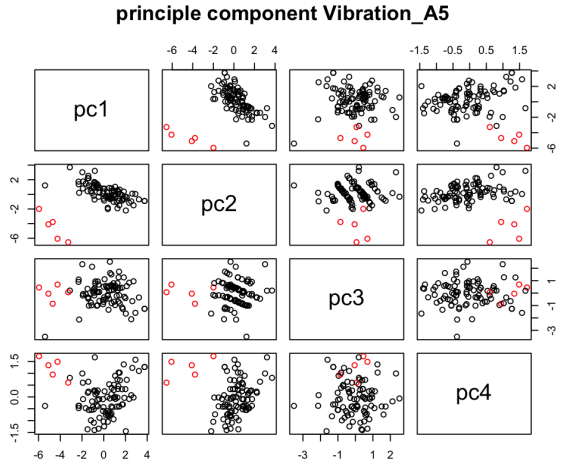


Figure B.45: Hcluster A5 plot

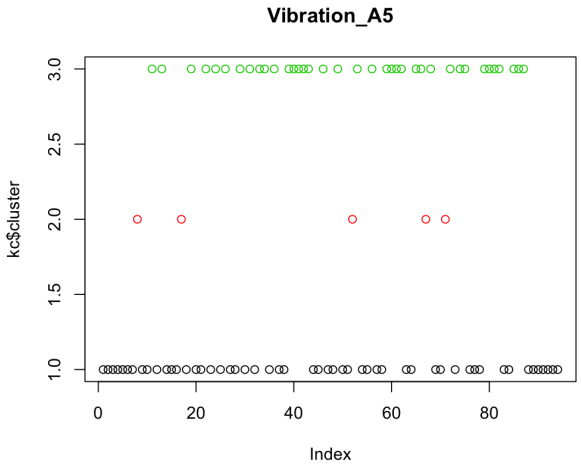


Figure B.46: K means A5 plot

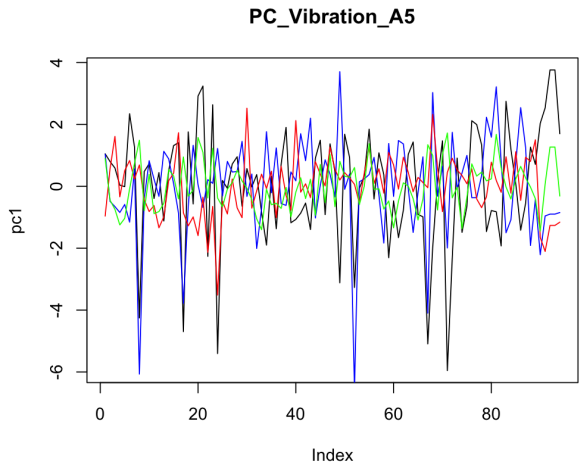


Figure B.47: Principle component A5 plot

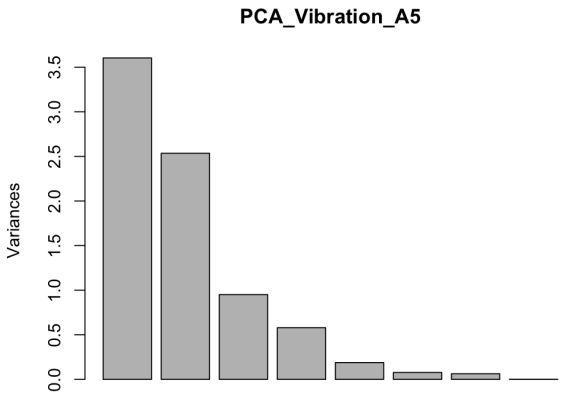


Figure B.48: PCA A5 plot

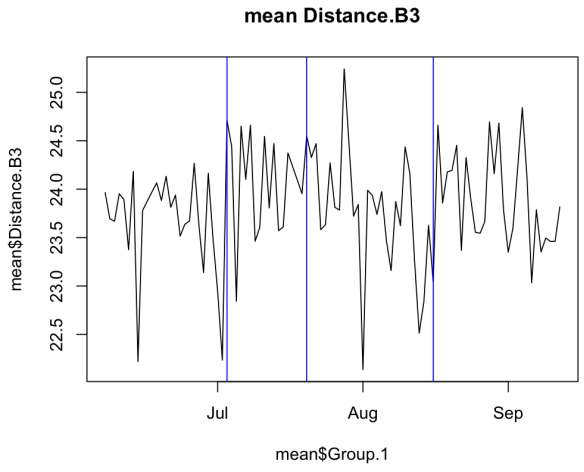


Figure B.49: Mean B3 plot

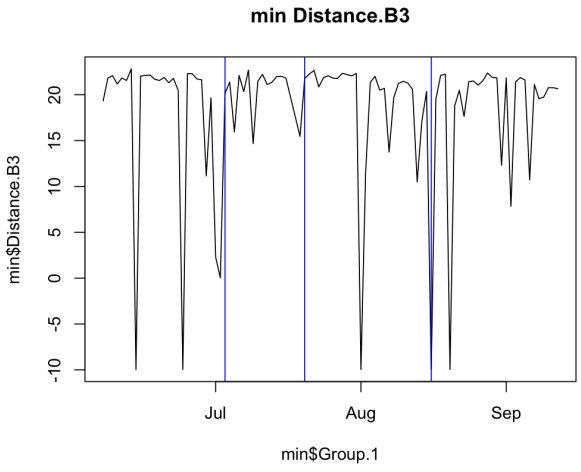


Figure B.50: Minimum B3 plot

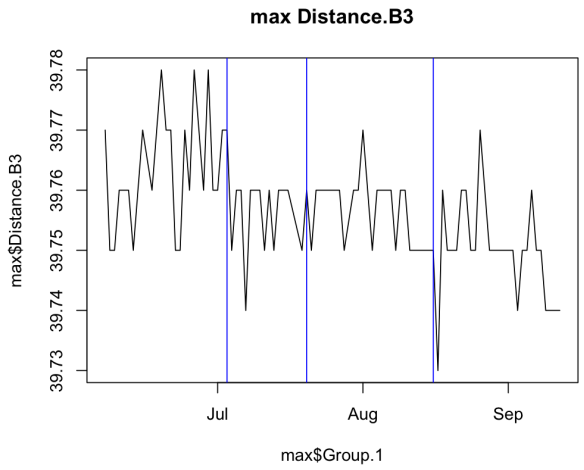


Figure B.51: Max B3 plot

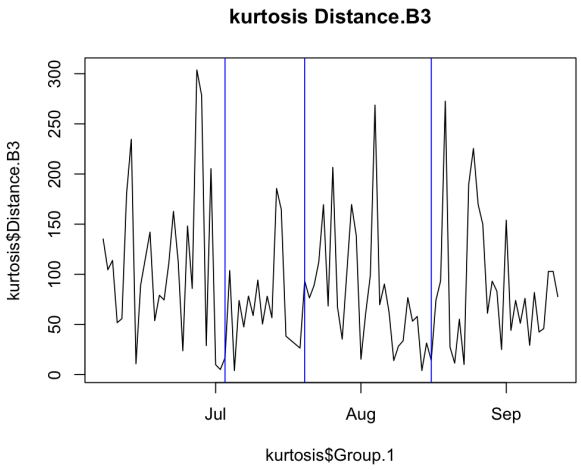


Figure B.52: Kurtosis B3 plot

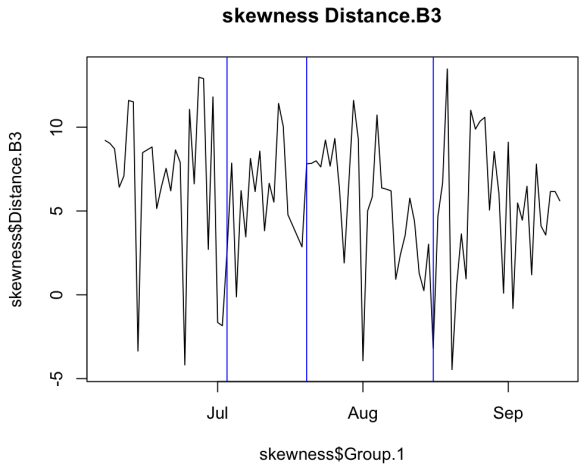


Figure B.53: Skewness B3 plot

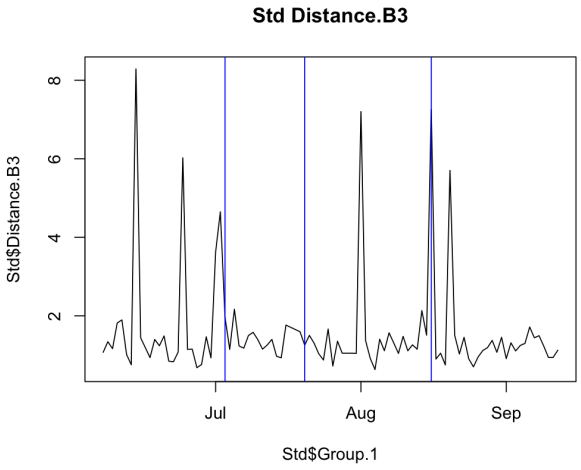


Figure B.54: Std B3 plot

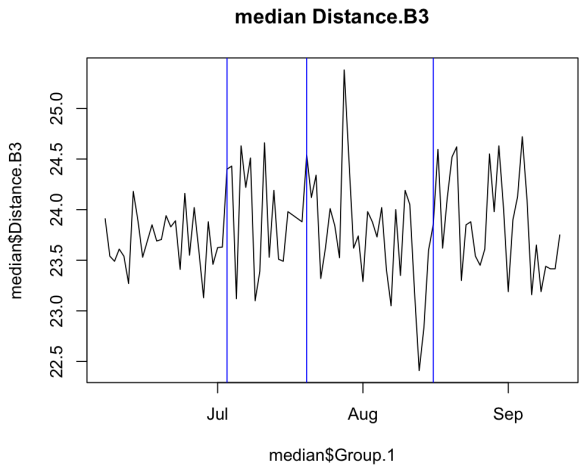


Figure B.55: Median B3 plot

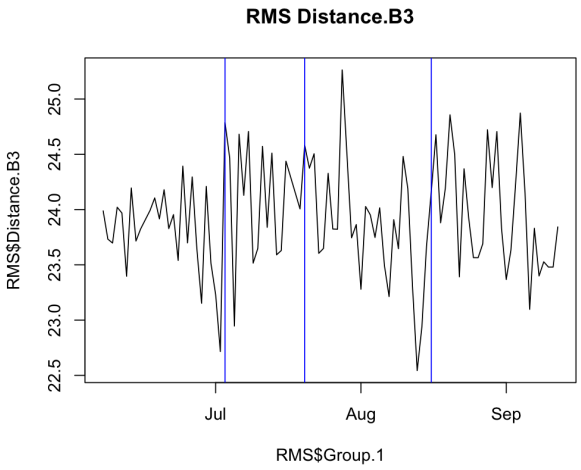


Figure B.56: RMS B3 plot

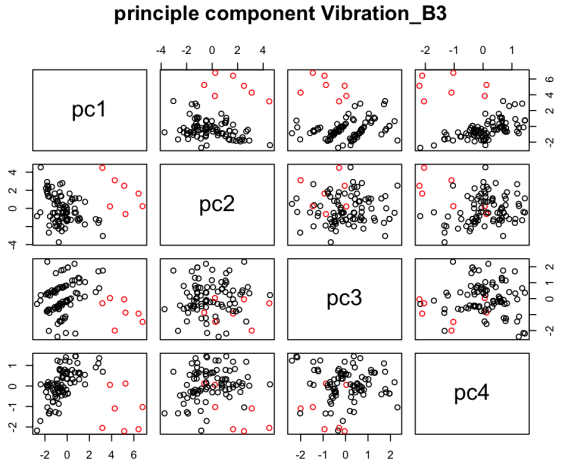


Figure B.57: Hcluster B3 plot

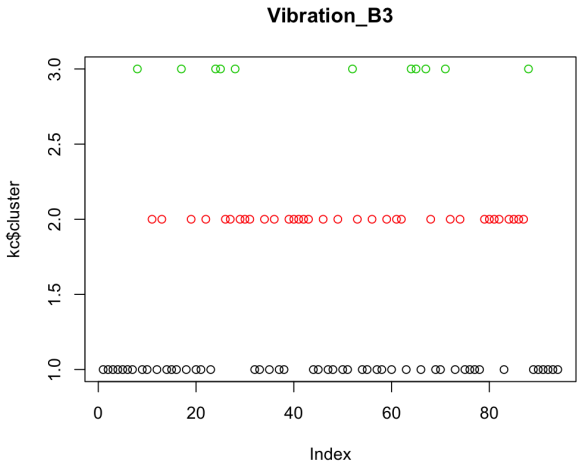


Figure B.58: K means B3 plot

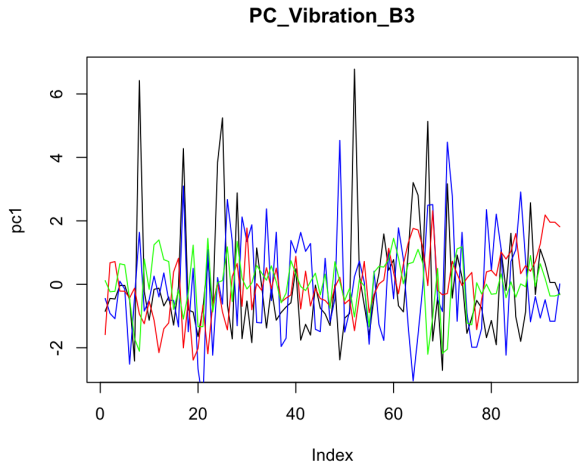


Figure B.59: Principle component B3 plot

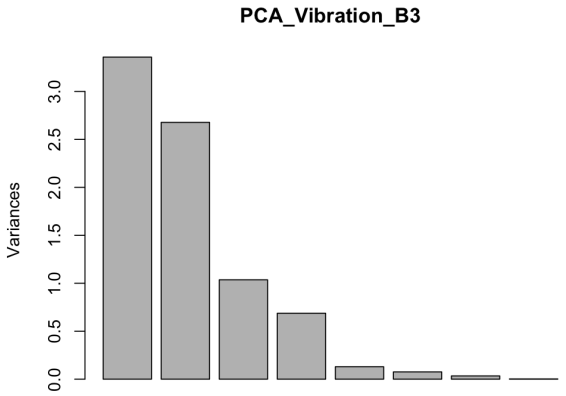


Figure B.60: PCA B3 plot

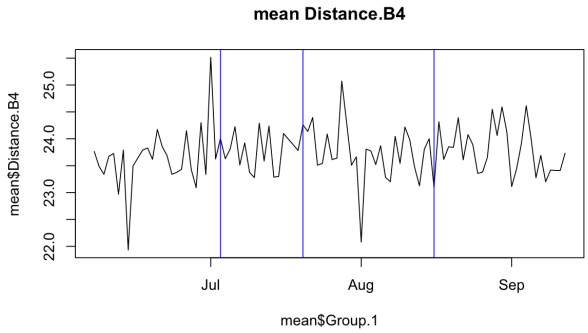


Figure B.61: Mean B4 plot

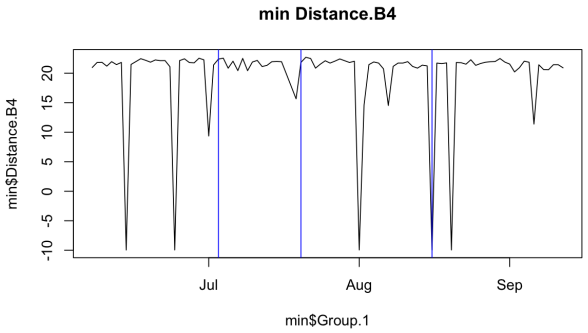


Figure B.62: Minimum B4 plot

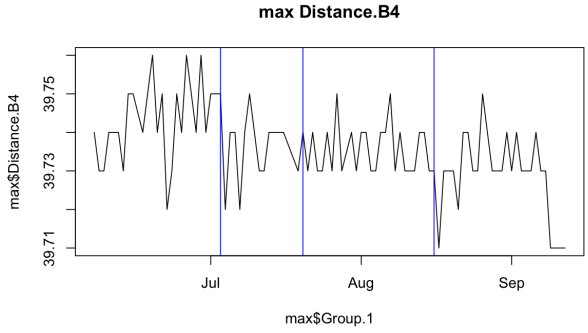


Figure B.63: Max B4 plot

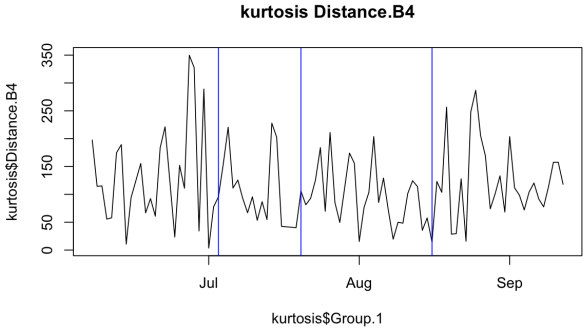


Figure B.64: Kurtosis B4 plot

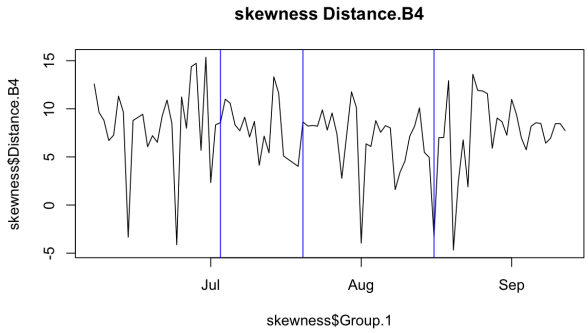


Figure B.65: Skewness B4 plot

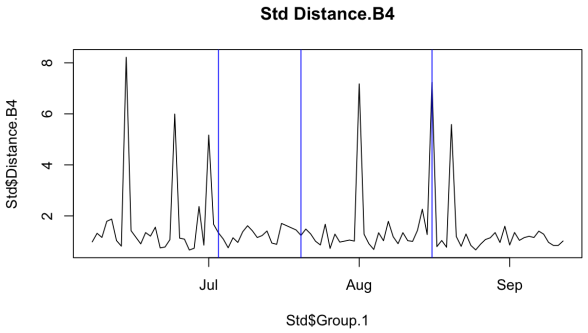


Figure B.66: Std B4 plot

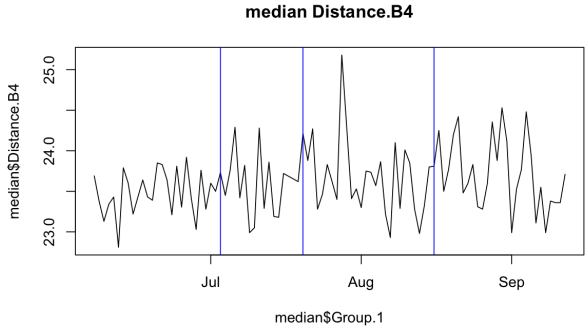


Figure B.67: Median B4 plot

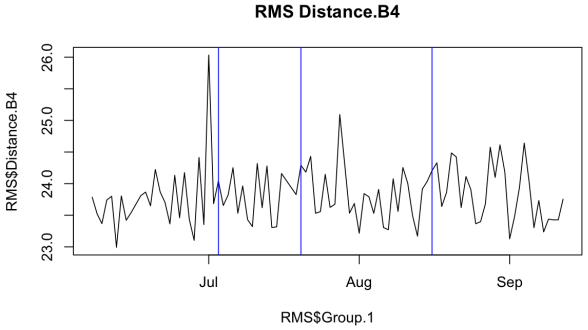


Figure B.68: RMS B4 plot

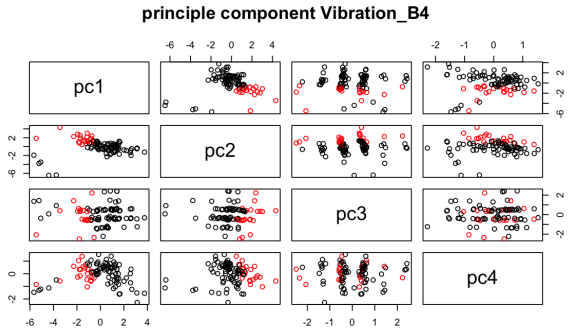


Figure B.69: Hcluster B4 plot

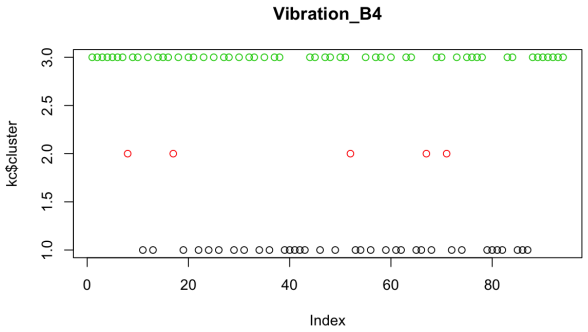


Figure B.70: K means B4 plot

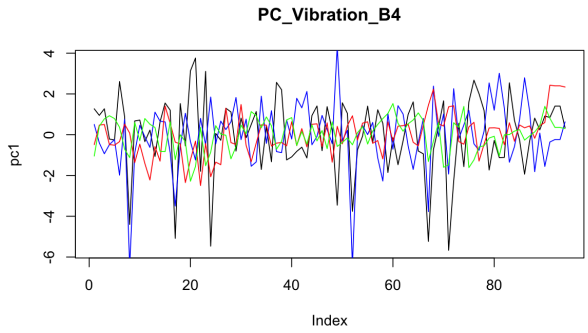


Figure B.71: Principle component B4 plot

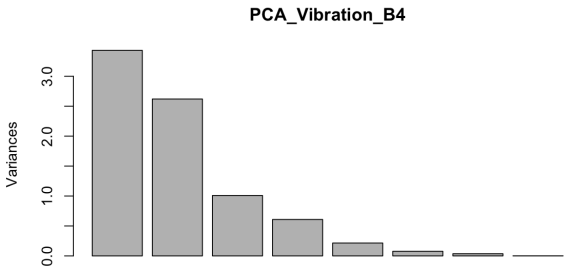


Figure B.72: PCA B4 plot

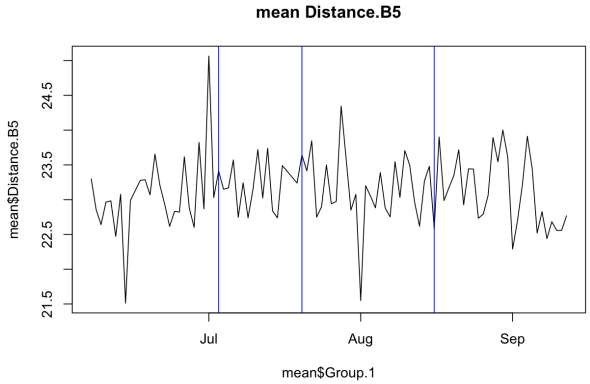


Figure B.73: Mean B5 plot

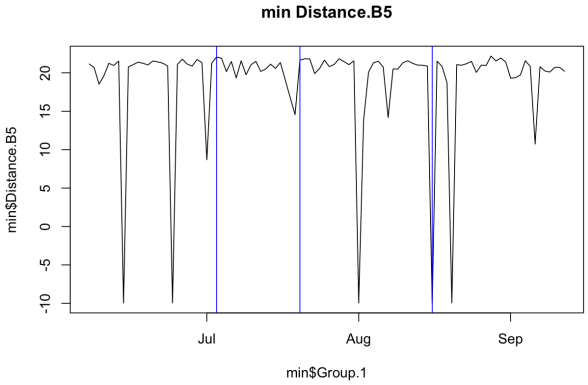


Figure B.74: Minimum B5 plot

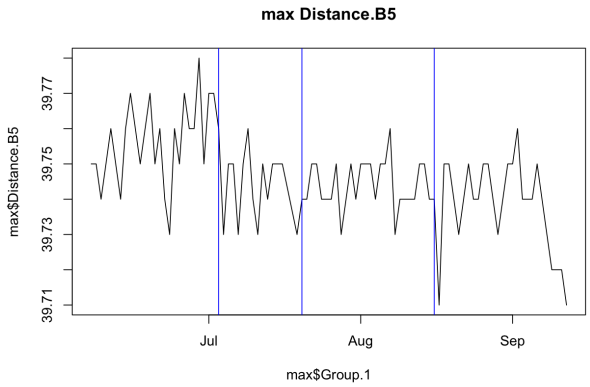


Figure B.75: Max B5 plot

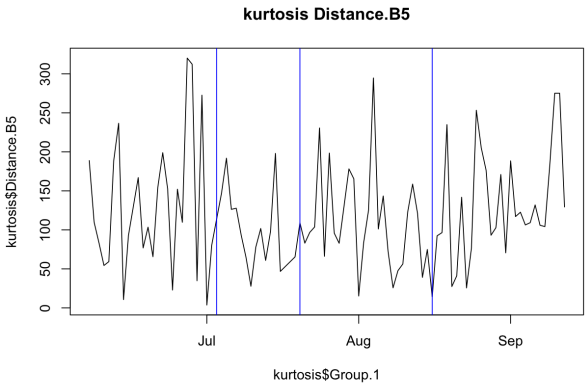


Figure B.76: Kurtosis B5 plot

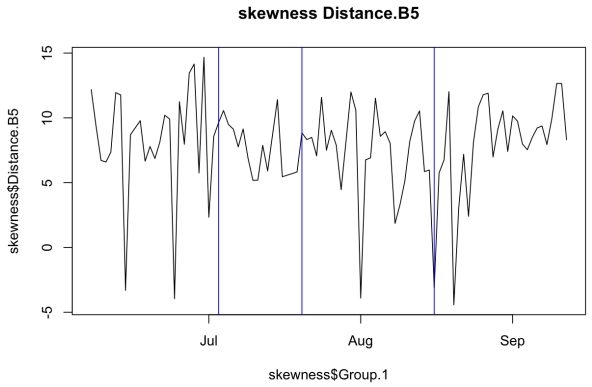


Figure B.77: Skewness B5 plot

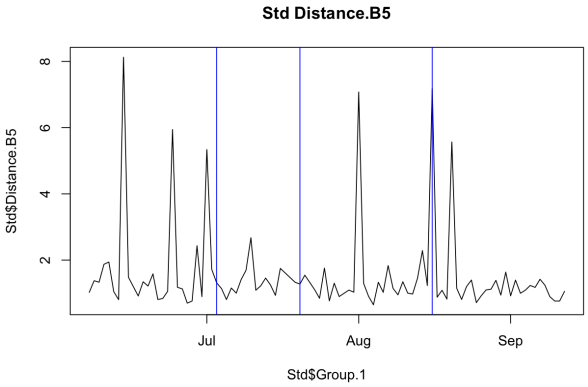


Figure B.78: Standard deviation B5 plot

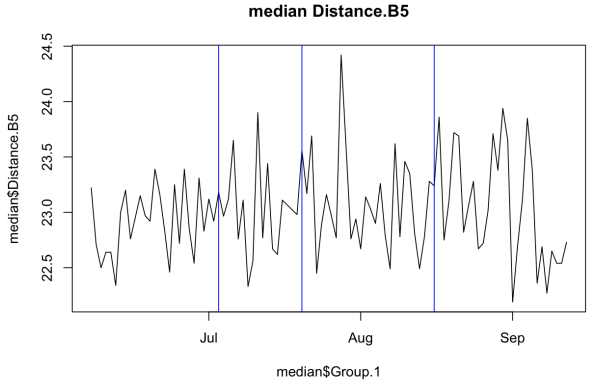


Figure B.79: Median B5 plot

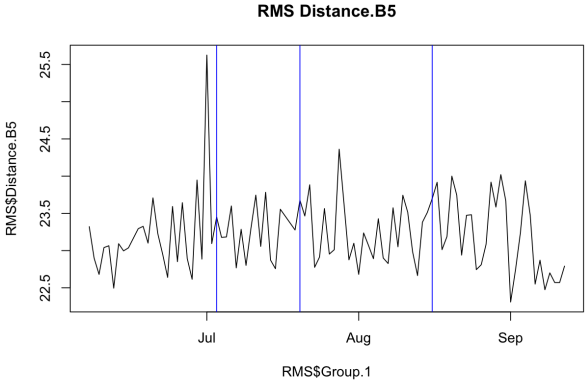


Figure B.80: RMS B5 plot

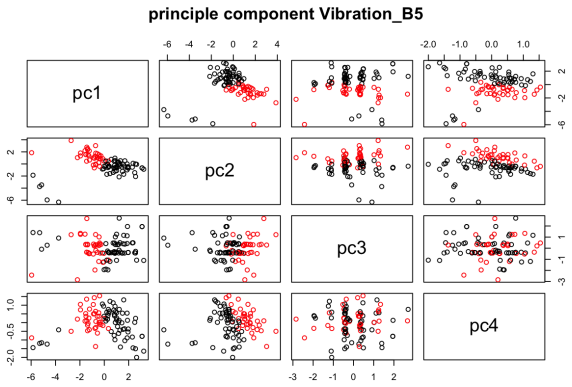


Figure B.81: Hcluster B5 plot

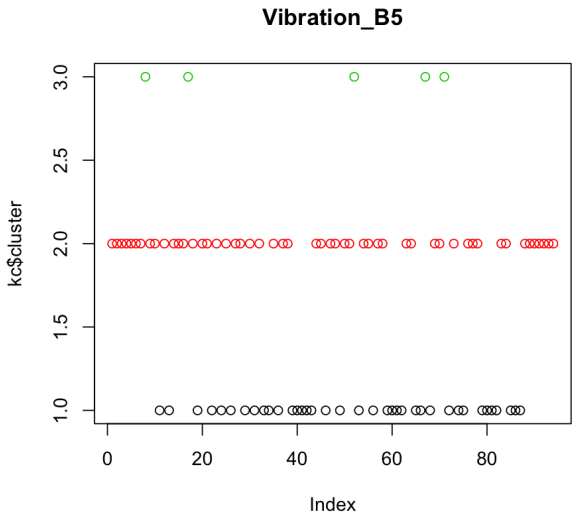


Figure B.82: K means B5 plot

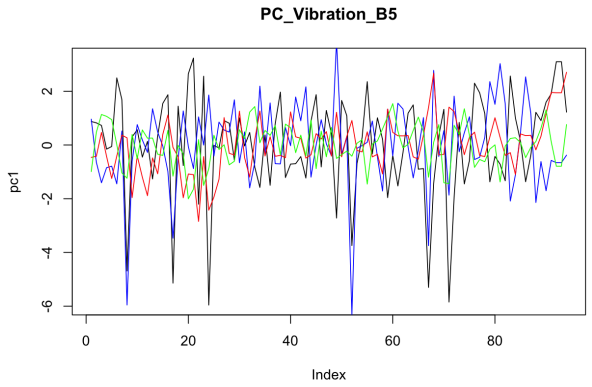


Figure B.83: Principle component B5 plot

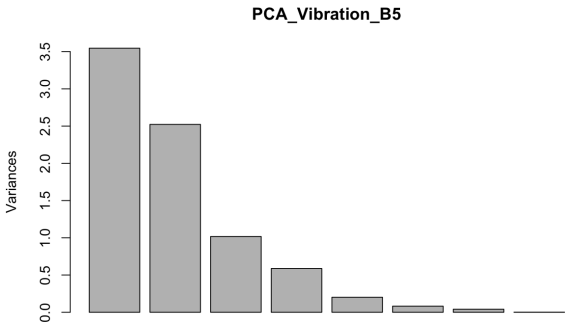


Figure B.84: PCA B5 plot

APPENDIX

# C

# CODE

## C.1 IBA Cleaning Code

```
library(reader)
setwd("~/Google␣Drive/Thesis/Data␣Collection/Data␣Collection/iba")
#count.fields("iba.txt",␣sep␣=␣"\t")#count␣the␣amount␣of␣observations␣in␣iba.txt
file.split("iba.txt",␣size␣=␣317455,␣same.dir␣=␣TRUE,␣suf␣=␣"part",␣win␣=␣TRUE)
#split␣the␣whole␣files␣into␣seperate␣files
iba_1<-data.frame((read.delim("iba_partaa.txt",␣header␣=␣FALSE,sep␣=␣"\t")))
#read␣the␣first␣split
iba_1.1␣=␣tail(iba_1,␣-1)␣#delete␣the␣first␣row␣of␣iba_1
names(iba_1.1)␣<-␣lapply(iba_1.1[1,␣],␣as.character)
#set␣the␣name␣of␣the␣columes␣as␣the␣first␣row
iba_1.1<-␣iba_1.1[-1,␣]
#delete␣the␣first␣row
iba_1_sub<-iba_1.1[,c(1,66,67,98,99,100,101,
162,164,165,652,2,3,30,31,66,67,82,83,84,85,
128,129,136,146,147,148,149,150,151,152,153,
154,155,156,157,158,162,163,164,165,168,169,
172,173,175,182,183,184,185,186,187,188,189,
194,195,196,197,198,199,200,201,202,203,204,
205,206,207,2,3,227,228,238,239,30,31,242,243,
244,245,246,247,248)]
which(iba_1_sub$time␣==␣"17.03.2019␣00:21:41.000000")
#check␣which␣row␣contains␣the␣data␣from␣17-03-2019␣since␣data␣before␣that␣were␣logged␣wrong
iba_1_sub=␣tail(iba_1_sub,␣-(which(iba_1_sub$time␣==␣"17.03.2019␣00:21:41.000000")+1))
#delete␣all␣the␣row␣before␣17-03-2019
out␣<-␣strsplit(as.character(iba_1_sub$time),'␣')
#seperate␣date␣and␣time␣into␣two␣columns␣and␣place␣them␣in␣the␣front
do.call(rbind,␣out)
iba_1<-data.frame(do.call(rbind,␣out),iba_1_sub)
iba_1<-␣subset(␣iba_1,␣select␣=␣-3)
#delete␣the␣third␣column␣of␣the␣data␣frame␣the␣original␣time␣column
names(iba_1)[names(iba_1)==␣"X1"]␣<-␣"Date"
names(iba_1)[names(iba_1)␣==␣"X2"]␣<-␣"Time"
#change␣the␣first␣two␣columns␣names␣into␣date␣and␣time
```

```
iba_1$Time<- substr(iba_1$Time, 0,5)
#omit the the time such that only mim is visable
iba_1$Date <- chartr(".", "/", iba_1$Date)
#change the date into differnt format
#iba_1 has been cleaned
#Cleaning iba_2
iba_2<-data.frame(read.table("iba_partab.txt", header=T, sep="\t"))
iba_2<-iba_2[ ,c(1,66,67,98,99,100,101,162,164,165,652,2,3
,30,31,66,67,82,83,84,85,128,129,
136,146,147,148,149,150,151,152,153,
154,155,156,157,158,162,163,164,165,168,169,
172,173,175,182,183,184,185,186,187,188,189,
194,195,196,197,198,199,200,201,
202,203,204,205,206,207,2,3,227,228,238,239,
30,31,242,243,244,245,246,247,248)]
names(iba_2)<-names(iba_1)[2:84]
out <- strsplit(as.character(iba_2$Time),'')
#seperate date and time into two columns and place them in the front
do.call(rbind, out)
iba_2<-data.frame(do.call(rbind, out),iba_2)
iba_2<- subset( iba_2, select = -3)
#delete the third column of the data frame the original time column
names(iba_2)[names(iba_2)== "X1"] <- "Date"
names(iba_2)[names(iba_2) == "X2"] <- "Time"
#change the first two columns names into date and time
iba_2$Time<- substr(iba_2$Time, 0,5)
#omit the the time such that only mim is visable
iba_2$Date <- chartr(".", "/", iba_2$Date)
#change the date into differnt format

#Cleaning iba_3
iba_3<-data.frame(read.table("iba_partac.txt", header=T, sep="\t"))
iba_3<-iba_3[ ,c(1,66,67,98,99,100,101,162,164,165,652,2,3,30,
31,66,67,82,83,84,85,128,129,136,146,147,148,149,150,151,152,
153,154,155,156,157,158,162,163,164,165,168,169,172,173,175,
182,183,184,185,186,187,188,189,194,195,196,197,198,199,200,
201,202,203,204,205,206,207,2,3,227,228,238,239,30,31,242,243,
244,245,246,247,248)]
names(iba_3)<-names(iba_1)[2:84]
out <- strsplit(as.character(iba_3$Time),' ')
#seperate date and time into two columns and place them in the front
do.call(rbind, out)
iba_3<-data.frame(do.call(rbind, out),iba_3)
iba_3<- subset( iba_3, select = -3)
#delete the third column of the data frame the original time column
names(iba_3)[names(iba_3)== "X1"] <- "Date"
names(iba_3)[names(iba_3) == "X2"] <- "Time"
#change the first two columns names into date and time
iba_3$Time<- substr(iba_3$Time, 0,5)
```

```
#omit␣the␣the␣time␣such␣that␣only␣mim␣is␣visable
iba_3$Date␣<-␣chartr(".",␣"/",␣iba_3$Date)
#change␣the␣date␣into␣differnt␣format

#Cleaning␣iba_4
iba_4<-data.frame(read.table("iba_partad.txt",␣header=T,␣sep="\t"))
iba_4<-iba_4[␣,␣c(1,66,67,98,99,100,101,162,164,165,652,2,3,30,31,66,
67,82,83,84,85,128,129,136,146,147,148,149,150,151,152,153,154,
155,156,157,158,162,163,164,165,168,169,172,173,175,182,183,184,
185,186,187,188,189,194,195,196,197,198,199,200,201,
202,203,204,205,206,207,2,3,227,228,238,239,30,31,242,
243,244,245,246,247,248)]
names(iba_4)<-names(iba_1)[2:84]
out␣<-␣strsplit(as.character(iba_4$Time),'␣')
#seperate␣date␣and␣time␣into␣two␣columns␣and␣place␣them␣in␣the␣front
do.call(rbind,␣out)
iba_4<-data.frame(do.call(rbind,␣out),iba_4)
iba_4<-␣subset(␣iba_4,␣select␣=␣-3)
#delete␣the␣third␣column␣of␣the␣data␣frame␣the␣original␣time␣column
names(iba_4)[names(iba_4)==␣"X1"]␣<-␣"Date"
names(iba_4)[names(iba_4)␣==␣"X2"]␣<-␣"Time"
#change␣the␣first␣two␣columns␣names␣into␣date␣and␣time
iba_4$Time<-␣substr(iba_4$Time,␣0,5)
#omit␣the␣the␣time␣such␣that␣only␣mim␣is␣visable
iba_4$Date␣<-␣chartr(".",␣"/",␣iba_4$Date)
#change␣the␣date␣into␣differnt␣format

#Cleaning␣iba_5
iba_5<-data.frame(read.table("iba_partae.txt",␣header=T,␣sep="\t"))
iba_5<-iba_5[␣,c(1,66,67,98,99,100,101,162,164,165,652,
2,3,30,31,66,67,82,83,84,85,128,129,136,146,147,148,
149,150,151,152,153,154,155,156,157,158,162,163,164,
165,168,169,172,173,175,182,183,184,185,186,187,188,
189,194,195,196,197,198,199,200,201,202,203,204,205,
206,207,2,3,227,228,238,239,30,31,242,243,244,245,246,247,248)]
names(iba_5)<-names(iba_1)[2:84]
out␣<-␣strsplit(as.character(iba_5$Time),'␣')
#seperate␣date␣and␣time␣into␣two␣columns␣and␣place␣them␣in␣the␣front
do.call(rbind,␣out)
iba_5<-data.frame(do.call(rbind,␣out),iba_5)
iba_5<-␣subset(␣iba_5,␣select␣=␣-3)
#delete␣the␣third␣column␣of␣the␣data␣frame␣the␣original␣time␣column
names(iba_5)[names(iba_5)==␣"X1"]␣<-␣"Date"
names(iba_5)[names(iba_5)␣==␣"X2"]␣<-␣"Time"
#change␣the␣first␣two␣columns␣names␣into␣date␣and␣time
iba_5$Time<-␣substr(iba_5$Time,␣0,5)
#omit␣the␣the␣time␣such␣that␣only␣mim␣is␣visable
iba_5$Date␣<-␣chartr(".",␣"/",␣iba_5$Date)
#change␣the␣date␣into␣differnt␣format
```

```
#Cleaning␣iba_6
iba_6<-data.frame(read.table("iba_partaf.txt",␣header=T,␣sep="\t"))
iba_6<-iba_6[␣,␣c(1,66,67,98,99,100,101,162,164,165,652,
2,3,30,31,66,67,82,83,84,85,128,129,136,146,147,148,149,150,
151,152,153,154,155,156,157,158,162,163,164,165,168,169,
172,173,175,182,183,184,185,186,187,188,189,194,195,196,
197,198,199,200,201,202,203,204,205,206,207,2,3,227,228,
238,239,30,31,242,243,244,245,246,247,248)]
names(iba_6)<-names(iba_1)[2:84]
out␣<-␣strsplit(as.character(iba_6$Time),'␣')
#seperate␣date␣and␣time␣into␣two␣columns␣and␣place␣them␣in␣the␣front
do.call(rbind,␣out)
iba_6<-data.frame(do.call(rbind,␣out),iba_6)
iba_6<-␣subset(␣iba_6,␣select␣=␣-3)
#delete␣the␣third␣column␣of␣the␣data␣frame␣the␣original␣time␣column
names(iba_6)[names(iba_6)==␣"X1"]␣<-␣"Date"
names(iba_6)[names(iba_6)␣==␣"X2"]␣<-␣"Time"
#change␣the␣first␣two␣columns␣names␣into␣date␣and␣time
iba_6$Time<-␣substr(iba_6$Time,␣0,5)
#omit␣the␣the␣time␣such␣that␣only␣mim␣is␣visable
iba_6$Date␣<-␣chartr(".",␣"/",␣iba_6$Date)
#change␣the␣date␣into␣differnt␣format


#iba_7
iba_7<-data.frame(read.table("iba_partag.txt",␣header=T,␣sep="\t"))
iba_7<-iba_7[␣,␣c(1,66,67,98,99,100,101,162,164,165,652,2,
3,30,31,66,67,82,83,84,85,128,129,136,146,147,148,149,150,
151,152,153,154,155,156,157,158,162,163,164,165,168,169,
172,173,175,182,183,184,185,186,187,188,189,194,195,196,
197,198,199,200,201,202,203,204,205,206,207,2,3,227,228,
238,239,30,31,242,243,244,245,246,247,248)]
names(iba_7)<-names(iba_1)[2:84]
out␣<-␣strsplit(as.character(iba_7$Time),'␣')
#seperate␣date␣and␣time␣into␣two␣columns␣and␣place␣them␣in␣the␣front
do.call(rbind,␣out)
iba_7<-data.frame(do.call(rbind,␣out),iba_7)
iba_7<-␣subset(␣iba_7,␣select␣=␣-3)
#delete␣the␣third␣column␣of␣the␣data␣frame␣the␣original␣time␣column
names(iba_7)[names(iba_7)==␣"X1"]␣<-␣"Date"
names(iba_7)[names(iba_7)␣==␣"X2"]␣<-␣"Time"
#change␣the␣first␣two␣columns␣names␣into␣date␣and␣time
iba_7$Time<-␣substr(iba_7$Time,␣0,5)
#omit␣the␣the␣time␣such␣that␣only␣mim␣is␣visable
iba_7$Date␣<-␣chartr(".",␣"/",␣iba_7$Date)
#change␣the␣date␣into␣differnt␣format
#iba_8
iba_8<-data.frame(read.table("iba_partah.txt",␣header=T,␣sep="\t"))
```

```
iba_8<-iba_8[␣,c(1,66,67,98,99,100,101,162,164,165,652,2,3,30,31,
66,67,82,83,84,85,128,129,136,146,147,148,149,150,151,152,153,
154,155,156,157,158,162,163,164,165,168,169,172,173,175,182,
183,184,185,186,187,188,189,194,195,196,197,198,199,200,201,
202,203,204,205,206,207,2,3,227,228,238,239,30,31,242,243,
244,245,246,247,248)]
names(iba_8)<-names(iba_1)[2:84]
out␣<-␣strsplit(as.character(iba_8$Time),'␣')
#seperate␣date␣and␣time␣into␣two␣columns␣and␣place␣them␣in␣the␣front
do.call(rbind,␣out)
iba_8<-data.frame(do.call(rbind,␣out),iba_8)
iba_8<-␣subset(␣iba_8,␣select␣=␣-3)
#delete␣the␣third␣column␣of␣the␣data␣frame␣the␣original␣time␣column
names(iba_8)[names(iba_8)==␣"X1"]␣<-␣"Date"
names(iba_8)[names(iba_8)␣==␣"X2"]␣<-␣"Time"
#change␣the␣first␣two␣columns␣names␣into␣date␣and␣time
iba_8$Time<-␣substr(iba_8$Time,␣0,5)
#omit␣the␣the␣time␣such␣that␣only␣mim␣is␣visable
iba_8$Date␣<-␣chartr(".",␣"/",␣iba_8$Date)
#change␣the␣date␣into␣differnt␣format
#iba_9
iba_9<-data.frame(read.table("iba_partai.txt",␣header=T,␣sep="\t"))
iba_9<-iba_9[␣,c(1,66,67,98,99,100,101,162,164,165,
652,2,3,30,31,66,67,82,83,84,85,128,129,136,146,147,
148,149,150,151,152,153,154,155,156,157,158,162,163,
164,165,168,169,172,173,175,182,183,184,185,186,187,
188,189,194,195,196,197,198,199,200,201,
202,203,204,205,206,207,2,3,227,228,238,239,
30,31,242,243,244,245,246,247,248)]
names(iba_9)<-names(iba_1)[2:84]
out␣<-␣strsplit(as.character(iba_9$Time),'␣')
#seperate␣date␣and␣time␣into␣two␣columns␣and␣place␣them␣in␣the␣front
do.call(rbind,␣out)
iba_9<-data.frame(do.call(rbind,␣out),iba_9)
iba_9<-␣subset(␣iba_9,␣select␣=␣-3)
#delete␣the␣third␣column␣of␣the␣data␣frame␣the␣original␣time␣column
names(iba_9)[names(iba_9)==␣"X1"]␣<-␣"Date"
names(iba_9)[names(iba_9)␣==␣"X2"]␣<-␣"Time"
#change␣the␣first␣two␣columns␣names␣into␣date␣and␣time
iba_9$Time<-␣substr(iba_9$Time,␣0,5)
#omit␣the␣the␣time␣such␣that␣only␣mim␣is␣visable
iba_9$Date␣<-␣chartr(".",␣"/",␣iba_9$Date)
#change␣the␣date␣into␣differnt␣format
#iba_10
iba_10<-data.frame(read.table("iba_partaj.txt",␣header=T,␣sep="\t"))
iba_10<-iba_10[␣,c(1,66,67,98,99,100,101,162,164,165,652,2,3,30,
31,66,67,82,83,84,85,128,129,136,146,147,148,149,150,151,152,153,
154,155,156,157,158,162,163,164,165,168,169,172,173,175,182,183,
184,185,186,187,188,189,194,195,196,197,198,199,200,201,202,203,
```

```
204,205,206,207,2,3,227,228,238,239,30,31,242,243,244,245,246,247,248)]
names(iba_10)<-names(iba_1)[2:84]
out␣<-␣strsplit(as.character(iba_10$Time),'␣')
#seperate␣date␣and␣time␣into␣two␣columns␣and␣place␣them␣in␣the␣front
do.call(rbind,␣out)
iba_10<-data.frame(do.call(rbind,␣out),iba_10)
iba_10<-␣subset(␣iba_10,␣select␣=␣-3)
#delete␣the␣third␣column␣of␣the␣data␣frame␣the␣original␣time␣column
names(iba_10)[names(iba_10)==␣"X1"]␣<-␣"Date"
names(iba_10)[names(iba_10)␣==␣"X2"]␣<-␣"Time"
#change␣the␣first␣two␣columns␣names␣into␣date␣and␣time
iba_10$Time<-␣substr(iba_10$Time,␣0,5)
#omit␣the␣the␣time␣such␣that␣only␣mim␣is␣visable
iba_10$Date␣<-␣chartr(".",␣"/",␣iba_10$Date)
#change␣the␣date␣into␣differnt␣format
#iba_11
iba_11<-data.frame(read.table("iba_partak.txt",␣header=T,␣sep="\t"))
iba_11<-iba_11[␣,␣c(1,66,67,98,99,100,101,162,164,165,652,
2,3,30,31,66,67,82,83,84,85,128,129,136,146,147,148,149,150,
151,152,153,154,155,156,157,158,162,163,164,165,168,169,
172,173,175,182,183,184,185,186,187,188,189,194,195,196,
197,198,199,200,201,202,203,204,205,206,207,2,3,227,228,
238,239,30,31,242,243,244,245,246,247,248)]
names(iba_11)<-names(iba_1)[2:84]
out␣<-␣strsplit(as.character(iba_11$Time),'␣')
#seperate␣date␣and␣time␣into␣two␣columns␣and␣place␣them␣in␣the␣front
do.call(rbind,␣out)
iba_11<-data.frame(do.call(rbind,␣out),iba_11)
iba_11<-␣subset(␣iba_11,␣select␣=␣-3)
#delete␣the␣third␣column␣of␣the␣data␣frame␣the␣original␣time␣column
names(iba_11)[names(iba_11)==␣"X1"]␣<-␣"Date"
names(iba_11)[names(iba_11)␣==␣"X2"]␣<-␣"Time"
#change␣the␣first␣two␣columns␣names␣into␣date␣and␣time
iba_11$Time<-␣substr(iba_11$Time,␣0,5)
#omit␣the␣the␣time␣such␣that␣only␣min␣is␣visable
iba_11$Date␣<-␣chartr(".",␣"/",␣iba_11\$Date)
#change␣the␣date␣into␣different␣format
################################################
#Combine␣all␣sub␣dataframe
IBA<-rbind(iba_1,iba_2,iba_3,iba_4,iba_5,iba_6,iba_7,iba_8,iba_9,iba_10,iba_11)
IBA<-subset(␣IBA,␣select␣=␣c(-17,-18,-38))
```

## C.2 EMASS Cleaning Code

```
#read␣in␣required␣packages
library("readxl")
library("readr")
```

```
library("dplyr")
library("plyr")
library(reader)
library("BBmisc")
library(reshape)
library(tidyr)
#create␣a␣list␣of␣the␣files␣from␣your␣target␣directory
setwd("/Volumes/Transcend/EMASS/Cycle␣1/2019.06.15")
file_list<-␣list.files(pattern='*.txt')
EMASS<-lapply(file_list,read.table,skip␣=␣51,␣fill=␣TRUE,sep␣=␣";",␣nrows␣=␣6000)
#read␣all␣txt␣files␣in␣a␣folder␣disregarding␣the␣first␣51␣rows,␣read␣the␣first␣6000␣rows
EMASS2<-bind_rows(EMASS)
EMASS_FILTERED<-data.frame(EMASS2[seq(1,nrow(EMASS2),by␣=␣100),])
#read␣1␣obervation␣per␣100␣observations
names(EMASS_FILTERED)<-c("Timestamp","Distance␣A1","Distance␣B1","Current␣A1","Current␣B1","Di
write.csv(EMASS_FILTERED,"/Volumes/Transcend/EMASS/2019.06.15.csv")
####These␣piece␣of␣code␣is␣repeated␣for
every␣folder␣from␣EMASS␣and␣a␣combined
csv␣file␣was␣exported␣with␣daily␣data.
```

## C.3 Data Visualization and Exploration Code

```
####IBA␣EXPLORATION######
#␣iba_1<-read.csv("iba_1.csv")
#␣iba_2<-read.csv("iba_2.csv")
#␣iba_3<-read.csv("iba_3.csv")
#␣iba_4<-read.csv("iba_4.csv")
#␣iba_5<-read.csv("iba_5.csv")
#␣iba_6<-read.csv("iba_6.csv")
#␣iba_7<-read.csv("iba_7.csv")
#␣iba_8<-read.csv("iba_8.csv")
#␣iba_9<-read.csv("iba_9.csv")
#␣iba_10<-read.csv("iba_10.csv")
#␣iba_11<-read.csv("iba_11.csv")
#␣IBA<-rbind(iba_1,iba_2,iba_3,iba_4,iba_5,iba_6,iba_7,iba_8,iba_9,iba_10,iba_11)
#␣IBA<-␣subset(␣IBA,␣select␣=␣c(-17,-18,-38))
#␣head(IBA)
#␣Reduced<-data.frame(IBA$Date,
IBA$Time,
IBA$input.lineSpeedRaw,
IBA$input.lineSpeedRef,
IBA$input.lineSpeedRef.1,
IBA$input.lineSpeed,
IBA$input.stripTension,
IBA$input.stripTension.1,
IBA$input.coatMass.TOP.,
IBA$input.coatMass.BOT.,
IBA$input.coatMass.TOP..1,
```

```
IBA$input.coatMass.BOT..1,
IBA$input.gaugePos.TOP..1,
IBA$input.gaugePos.BOT..1,
IBA$input.controlPress.TOP.,
IBA$input.controlPress.BOT.,
IBA$input.headerPressRaw.BOT,
IBA$input.headerPressRaw.TOP,
IBA$input.headerPress.TOP.,
IBA$input.headerPress.BOT,
IBA$input.gaugePos.TOP.,
IBA$input.knifeHeight,
IBA$input.potTemp,
IBA$input.stripTemp,
IBA$X.mean.knifeHeight,
IBA$input.knifeHeight.1,
IBA$input.knifeHeightNomAvg,
IBA$input.corrRollPos.RIGHT.,
IBA$input.potAlContent)
#␣head(Reduced)
#␣Reduced[,3]<-as.double(Reduced[,3])
#␣Reduced[,4]<-as.double(Reduced[,4])
#␣Reduced[,5]<-as.double(Reduced[,5])
#␣Reduced[,6]<-as.double(Reduced[,6])
#␣Reduced[,7]<-as.double(Reduced[,7])
#␣Reduced[,8]<-as.double(Reduced[,8])
#␣Reduced[,9]<-as.double(Reduced[,9])
#␣Reduced[,10]<-as.double(Reduced[,10])
#␣Reduced[,11]<-as.double(Reduced[,11])
#␣Reduced[,12]<-as.double(Reduced[,12])
#␣Reduced[,13]<-as.double(Reduced[,13])
#␣Reduced[,14]<-as.double(Reduced[,14])
#␣Reduced[,15]<-as.double(Reduced[,15])
#␣Reduced[,16]<-as.double(Reduced[,16])
#␣Reduced[,17]<-as.double(Reduced[,17])
#␣Reduced[,18]<-as.double(Reduced[,18])
#␣Reduced[,19]<-as.double(Reduced[,19])
#␣Reduced[,20]<-as.double(Reduced[,20])
#␣Reduced[,21]<-as.double(Reduced[,21])
#␣Reduced[,22]<-as.double(Reduced[,22])
#␣Reduced[,23]<-as.double(Reduced[,23])
#␣Reduced[,24]<-as.double(Reduced[,24])
#␣Reduced[,25]<-as.double(Reduced[,25])
#␣Reduced[,26]<-as.double(Reduced[,26])
#␣Reduced[,27]<-as.double(Reduced[,27])
#␣Reduced[,28]<-as.double(Reduced[,28])
#␣Reduced[,29]<-as.double(Reduced[,29])
#␣#Reduced$IBA.Date<-substring(Reduced$IBA.Date,3,7)
#␣#Reduced$IBA.Date
#␣#Reduced$IBA.Date
```

```
#␣Reduced<-Reduced[,-c(5,8,24)]
#␣class(Reduced$IBA.Date)
#␣unique(Reduced$IBA.Date)
#␣write.csv(Reduced,"IBA_REDUCED.csv")
#write␣selected␣variables␣into␣a␣new␣csv␣file␣called␣'Reduced'
library(e1071)
library("readxl")
library("readr")
library("dplyr")
library("plyr")
library(reader)
library("BBmisc")
library(reshape)
library(tidyr)
library(plotly)
library("scatterplot3d")
library("rgl")
library("RColorBrewer")
setwd("~/Google␣Drive/Thesis/Data␣Collection/iba")
Reduced<-read.csv("IBA_REDUCED.csv")
head(Reduced)
Reduced$IBA.Date<-as.Date(Reduced$IBA.Date)
Reduced[,3]<-as.double(Reduced[,3])
Reduced[,4]<-as.double(Reduced[,4])
Reduced[,5]<-as.double(Reduced[,5])
Reduced[,6]<-as.double(Reduced[,6])
Reduced[,7]<-as.double(Reduced[,7])
Reduced[,8]<-as.double(Reduced[,8])
Reduced[,9]<-as.double(Reduced[,9])
Reduced[,10]<-as.double(Reduced[,10])
Reduced[,11]<-as.double(Reduced[,11])
Reduced[,12]<-as.double(Reduced[,12])
Reduced[,13]<-as.double(Reduced[,13])
Reduced[,14]<-as.double(Reduced[,14])
Reduced[,15]<-as.double(Reduced[,15])
Reduced[,16]<-as.double(Reduced[,16])
Reduced[,17]<-as.double(Reduced[,17])
Reduced[,18]<-as.double(Reduced[,18])
Reduced[,19]<-as.double(Reduced[,19])
Reduced[,20]<-as.double(Reduced[,20])
Reduced[,21]<-as.double(Reduced[,21])
Reduced[,22]<-as.double(Reduced[,22])
Reduced[,23]<-as.double(Reduced[,23])
Reduced[,24]<-as.double(Reduced[,24])
Reduced[,25]<-as.double(Reduced[,25])
Reduced[,26]<-as.double(Reduced[,26])
Reduced[,27]<-as.double(Reduced[,27])
################Feature␣extraction##############
RMS<-function(X){
```

```
␣␣rms<-sqrt(mean(X^2))
␣␣return(rms)
}
mean_TOTAL<-aggregate(Reduced[,3:27],␣by=list(Reduced$IBA.Date),
FUN=mean,␣na.rm=TRUE)
mean_TOTAL<-␣mean_TOTAL[order(mean_TOTAL$Group.1),]
mean_TOTAL$Group.1
###
max_TOTAL<-aggregate(Reduced[,3:27],␣by=list(Reduced$IBA.Date),
FUN=max,␣na.rm=TRUE)
max_TOTAL<-␣max_TOTAL[order(max_TOTAL$Group.1),]
max_TOTAL$Group.1
###
min_TOTAL<-␣aggregate(Reduced[,3:27],␣by=list(Reduced$IBA.Date),
FUN=min,␣na.rm=TRUE)
min_TOTAL<-␣min_TOTAL[order(min_TOTAL$Group.1),]
###
Std_TOTAL<-␣aggregate(Reduced[,3:27],␣by=list(Reduced$IBA.Date),
FUN=sd,␣na.rm=TRUE)
Std_TOTAL<-␣Std_TOTAL[order(Std_TOTAL$Group.1),]
###
median_TOTAL<-␣aggregate(Reduced[,3:27],␣by=list(Reduced$IBA.Date),
FUN=median,␣na.rm=TRUE)
median_TOTAL<-median_TOTAL[order(median_TOTAL$Group.1),]
###
kurtosis_TOTAL<-aggregate(Reduced[,3:27],␣by=list(Reduced$IBA.Date),
FUN=kurtosis,␣na.rm=TRUE)
kurtosis_TOTAL<-kurtosis_TOTAL[order(kurtosis_TOTAL$Group.1),]
###
skewness_TOTAL<-aggregate(Reduced[,3:27],␣by=list(Reduced$IBA.Date),
FUN=skewness,␣na.rm=TRUE)
skewness_TOTAL<-skewness_TOTAL[order(skewness_TOTAL$Group.1),]
###
RMS_TOTAL<-aggregate(Reduced[,3:27],␣by=list(Reduced$IBA.Date),
FUN=RMS)
RMS_TOTAL<-RMS_TOTAL[order(RMS_TOTAL$Group.1),]
##
#########################potTemp##############
plot(mean_TOTAL$Group.1,mean_TOTAL$IBA.input.potTemp,type␣=’l’,
main␣=␣"mean␣potTemp")
abline(v␣=c(18031,18080,18097),col␣=␣’blue’␣)
plot(median_TOTAL$Group.1,median_TOTAL$IBA.input.potTemp,type␣=␣’l’,
main␣=␣"median␣potTemp")
abline(v␣=c(18031,18080,18097),col␣=␣’blue’␣)
plot(min_TOTAL$Group.1,min_TOTAL$IBA.input.potTemp,type␣=␣’l’,
main␣=␣"min␣potTemp")
abline(v␣=c(18031,18080,18097),col␣=␣’blue’␣)
plot(max_TOTAL$Group.1,max_TOTAL$IBA.input.potTemp,type␣=␣’l’,
main␣=␣"max␣potTemp")
```

```
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(kurtosis_TOTAL$Group.1,kurtosis_TOTAL$IBA.input.potTemp,type␣=␣'l',
main␣=␣"kurtosis␣potTemp")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(skewness_TOTAL$Group.1,skewness_TOTAL$IBA.input.potTemp,type␣=␣'l',
main␣=␣"skewness␣potTemp")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(Std_TOTAL$Group.1,Std_TOTAL$IBA.input.potTemp,type␣=␣'l',
main␣=␣"Std␣potTemp")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(RMS_TOTAL$Group.1,RMS_TOTAL$IBA.input.potTemp,type␣=␣'l',
main␣=␣"RMS␣potTemp")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)

potTemp.<-data.frame(cbind(mean_TOTAL$IBA.input.potTemp,
min_TOTAL$IBA.input.potTemp,RMS_TOTAL$IBA.input.potTemp,
Std_TOTAL$IBA.input.potTemp))
names(potTemp.)<-c("mean","min","RMS","Std")
potTemp.<-na.omit(potTemp.)
########pca
PCA_potTemp.<-␣prcomp(potTemp.,scale=TRUE␣)
summary(PCA_potTemp.,main␣=␣"potTemp")
plot(PCA_potTemp.)
cor(potTemp.,PCA_potTemp.$x[,c(1,2,3,4)])
pc1<-PCA_potTemp.$x[,1]
pc2<-PCA_potTemp.$x[,2]
pc3<-PCA_potTemp.$x[,3]
pc4<-PCA_potTemp.$x[,4]
PC<-data.frame(pc1,pc2,pc3,pc4)
#####Hcluster############
distance<-dist(PC[,c(1,2)])
cluster<-hclust(distance)
plot(cluster,␣main␣=␣"potTemp")
rect.hclust(cluster,k=2,border␣=␣'red')
check<-cutree(cluster,2)
which(check==2)
plot(PC[,1:2],col␣=␣check,␣main␣=␣"potTemp")
plot(pc1,type␣=␣'l',␣main␣=␣"potTemp")
lines(pc2,col␣=␣'blue')
lines(pc3,col␣=␣"red")
lines(pc4,col␣=␣"green")
##K␣means␣cluster
kc<-kmeans(PC[,1:2],3)
which(kc$cluster==3)
plot(kc$cluster,col␣=␣kc$cluster,type␣='l',␣main␣=␣"potTemp")
#########################Al_Content############
plot(mean_TOTAL$Group.1,mean_TOTAL$IBA.input.potAlContent,type␣='l',
main␣=␣"mean␣Al_Content")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
```

```
plot(median_TOTAL$Group.1,median_TOTAL$IBA.input.potAlContent,type␣=␣'l',
main␣=␣"median␣Al_Content")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(min_TOTAL$Group.1,min_TOTAL$IBA.input.potAlContent,type␣=␣'l',
main␣=␣"min␣Al_Content")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(max_TOTAL$Group.1,max_TOTAL$IBA.input.potAlContent,type␣=␣'l',
main␣=␣"max␣Al_Content")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(kurtosis_TOTAL$Group.1,kurtosis_TOTAL$IBA.input.potAlContent,type␣=␣'l',
main␣=␣"kurtosis␣Al_Content")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(skewness_TOTAL$Group.1,skewness_TOTAL$IBA.input.potAlContent,type␣=␣'l',
main␣=␣"skewness␣Al_Content")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(Std_TOTAL$Group.1,Std_TOTAL$IBA.input.potAlContent,type␣=␣'l',
main␣=␣"Std␣Al_Content")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(RMS_TOTAL$Group.1,RMS_TOTAL$IBA.input.potAlContent,type␣=␣'l',
main␣=␣"RMS␣Al_Content")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)

potAlContent.<-data.frame(cbind(mean_TOTAL$IBA.input.potAlContent,
median_TOTAL$IBA.input.potAlContent,min_TOTAL$IBA.input.potAlContent,
max_TOTAL$IBA.input.potAlContent,RMS_TOTAL$IBA.input.potAlContent,
Std_TOTAL$IBA.input.potAlContent))
names(potAlContent.)<-c("mean","median","min","max","RMS","Std")
potAlContent.<-na.omit(potAlContent.)
########pca
PCA_potAlContent.<-␣prcomp(potAlContent.,scale␣=␣TRUE)
summary(PCA_potAlContent.)
plot(PCA_potAlContent.,␣main␣=␣"Al_Content")
cor(potAlContent.,PCA_potAlContent.$x[,c(1,2,3,4)])
pc1<-PCA_potAlContent.$x[,1]
pc2<-PCA_potAlContent.$x[,2]
pc3<-PCA_potAlContent.$x[,3]
pc4<-PCA_potAlContent.$x[,4]
PC<-data.frame(pc1,pc2,pc3,pc4)
#####Hcluster############
distance<-dist(PC[,c(1,2)])
cluster<-hclust(distance)
plot(cluster,␣main␣=␣"Al_Content")
rect.hclust(cluster,k=2,border␣=␣'red')
check<-cutree(cluster,2)
check
which(check␣==␣2)
plot(PC[,1:2],col␣=␣check,␣main␣=␣"Al_Content")
plot(pc1,type␣=␣'l',␣main␣=␣"Al_Content")
lines(pc2,col␣=␣'blue')
```

```
##K␣means␣cluster
kc<-kmeans(PC[,1:2],2)
which(kc$cluster==2)
plot(kc$cluster,col␣=␣kc$cluster,type␣=␣'l',main␣=␣"Al_Content")
########################corrRollPOS############
RMS<-function(X){
␣␣rms<-sqrt(mean(X^2))
␣␣return(rms)
}
plot(mean_TOTAL$Group.1,mean_TOTAL$IBA.input.corrRollPos.RIGHT.,
type␣='l',main␣=␣"mean␣corrRollPOS")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(median_TOTAL$Group.1,median_TOTAL$IBA.input.corrRollPos,
type␣=␣'l',main␣=␣"median␣corrRollPOS")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(min_TOTAL$Group.1,min_TOTAL$IBA.input.corrRollPos,
type␣=␣'l',main␣=␣"min␣corrRollPOS")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(max_TOTAL$Group.1,max_TOTAL$IBA.input.corrRollPos,
type␣=␣'l',main␣=␣"max␣corrRollPOS")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(kurtosis_TOTAL$Group.1,kurtosis_TOTAL$IBA.input.corrRollPos,
type␣=␣'l',main␣=␣"kurtosis␣corrRollPOS")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(skewness_TOTAL$Group.1,skewness_TOTAL$IBA.input.corrRollPos,
type␣=␣'l',main␣=␣"skewness␣corrRollPOS")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(Std_TOTAL$Group.1,Std_TOTAL$IBA.input.corrRollPos,
type␣=␣'l',main␣=␣"Std␣corrRollPOS")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(RMS_TOTAL$Group.1,RMS_TOTAL$IBA.input.corrRollPos,
type␣=␣'l',main␣=␣"RMS␣corrRollPOS")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)

corrRollPos.<-data.frame(cbind(mean_TOTAL$IBA.input.corrRollPos,
median_TOTAL$IBA.input.corrRollPos,min_TOTAL$IBA.input.corrRollPos,
max_TOTAL$IBA.input.corrRollPos,RMS_TOTAL$IBA.input.corrRollPos,
Std_TOTAL$IBA.input.corrRollPos))
names(corrRollPos.)<-c("mean","median","min","max","RMS","Std")
corrRollPos.<-na.omit(corrRollPos.)

########pca
PCA_corrRollPos.<-␣prcomp(corrRollPos.,scale␣=␣TRUE)
summary(PCA_corrRollPos.)
plot(PCA_corrRollPos.,␣main␣=␣"corrRollPOS")
cor(corrRollPos.,PCA_corrRollPos.$x[,c(1,2,3,4)])
pc1<-PCA_corrRollPos.$x[,1]
pc2<-PCA_corrRollPos.$x[,2]
```

```
pc3<-PCA_corrRollPos.$x[,3]
pc4<-PCA_corrRollPos.$x[,4]
PC<-data.frame(pc1,pc2,pc3,pc4)
#####Hcluster############
distance<-dist(PC[,c(1,2)])
cluster<-hclust(distance)
plot(cluster,␣main␣=␣"corrRollPOS")
rect.hclust(cluster,k=2,border␣=␣'red')
check<-cutree(cluster,2)
check
which(check␣==␣2)
plot(PC[,1:2],col␣=␣check,␣main␣=␣"corrRollPOS")
plot(pc1,type␣=␣'l',main␣=␣"corrRollPOS")
lines(pc2,col␣=␣'blue')
lines(pc3,col␣=␣"red")
lines(pc4,col␣=␣"green")
##K␣means␣cluster
kc<-kmeans(PC[,1:2],3)
which(kc$cluster==3)
plot(kc$cluster,col␣=␣kc$cluster,type␣=␣'l',␣main␣=␣"corrRollPOS")

#"IBA.input.headerPress.TOP."#####################
plot(mean_TOTAL$Group.1,mean_TOTAL$IBA.input.headerPress.TOP.,
type␣='l',main␣=␣"mean␣headerPress.TOP.")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(median_TOTAL$Group.1,median_TOTAL$IBA.input.headerPress.TOP.,
type␣=␣'l',main␣=␣"median␣headerPress.TOP.")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(min_TOTAL$Group.1,min_TOTAL$IBA.input.headerPress.TOP.,
type␣=␣'l',main␣=␣"min␣headerPress.TOP.")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(max_TOTAL$Group.1,max_TOTAL$IBA.input.headerPress.TOP.,
type␣=␣'l',main␣=␣"max␣headerPress.TOP.")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(kurtosis_TOTAL$Group.1,kurtosis_TOTAL$IBA.input.headerPress.TOP.,
type␣=␣'l',main␣=␣"kurtosis␣headerPress.TOP.")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(skewness_TOTAL$Group.1,skewness_TOTAL$IBA.input.headerPress.TOP.,
type␣=␣'l',main␣=␣"skewness␣headerPress.TOP.")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(Std_TOTAL$Group.1,Std_TOTAL$IBA.input.headerPress.TOP.,
type␣=␣'l',main␣=␣"Std␣headerPress.TOP.")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(RMS_TOTAL$Group.1,RMS_TOTAL$IBA.input.headerPress.TOP.,
type␣=␣'l',main␣=␣"RMS␣headerPress.TOP.")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)

headerPress.TOP..<-data.frame(cbind(mean_TOTAL$IBA.input.headerPress.TOP.,
median_TOTAL$IBA.input.headerPress.TOP.,min_TOTAL$IBA.input.headerPress.TOP.,
```

```
max_TOTAL$IBA.input.headerPress.TOP.,RMS_TOTAL$IBA.input.headerPress.TOP.,
Std_TOTAL$IBA.input.headerPress.TOP.))
names(headerPress.TOP..)<-c("mean","median","min","max","RMS","Std")
headerPress.TOP..<-na.omit(headerPress.TOP..)
########pca
PCA_headerPress.TOP..<-␣prcomp(headerPress.TOP..,scale␣=␣TRUE)
summary(PCA_headerPress.TOP..)
plot(PCA_headerPress.TOP..,main␣=␣"headerPress.TOP.")
cor(headerPress.TOP..,PCA_headerPress.TOP..$x[,c(1,2,3,4)])
pc1<-PCA_headerPress.TOP..$x[,1]
pc2<-PCA_headerPress.TOP..$x[,2]
pc3<-PCA_headerPress.TOP..$x[,3]
pc4<-PCA_headerPress.TOP..$x[,4]
PC<-data.frame(pc1,pc2,pc3,pc4)
#####Hcluster############
distance<-dist(PC[,c(1,2,3)])
cluster<-hclust(distance)
plot(cluster,␣main␣=␣"headerPress.TOP.")
rect.hclust(cluster,k=3,border␣=␣'red')
check<-cutree(cluster,2)
check
which(check␣==␣2)
plot(PC[,1:3],col␣=␣check,␣main␣=␣"headerPress.TOP.")
plot(pc1,type␣=␣'l',␣main␣=␣"headerPress.TOP.")
lines(pc2,col␣=␣'blue')
lines(pc3,col␣=␣"red")
lines(pc4,col␣=␣"green")
##K␣means␣cluster
kc<-kmeans(PC[,1:3],3)
kc
which(kc$cluster==3)
plot(kc$cluster,col␣=␣kc$cluster)
#"IBA.input.headerPress.BOT"###############

plot(mean_TOTAL$Group.1,mean_TOTAL$IBA.input.headerPress.BOT,
type␣='l',main␣=␣"mean␣headerPress.BOT")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(median_TOTAL$Group.1,median_TOTAL$IBA.input.headerPress.BOT,
type␣=␣'l',main␣=␣"median␣headerPress.BOT")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(min_TOTAL$Group.1,min_TOTAL$IBA.input.headerPress.BOT,
type␣=␣'l',main␣=␣"min␣headerPress.BOT")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(max_TOTAL$Group.1,max_TOTAL$IBA.input.headerPress.BOT,
type␣=␣'l',main␣=␣"max␣headerPress.BOT")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(kurtosis_TOTAL$Group.1,kurtosis_TOTAL$IBA.input.headerPress.BOT,
type␣=␣'l',main␣=␣"kurtosis␣headerPress.BOT")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
```

```
plot(skewness_TOTAL$Group.1,skewness_TOTAL$IBA.input.headerPress.BOT,
type␣=␣'l',main␣=␣"skewness␣headerPress.BOT")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(Std_TOTAL$Group.1,Std_TOTAL$IBA.input.headerPress.BOT,
type␣=␣'l',main␣=␣"Std␣headerPress.BOT")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)
plot(RMS_TOTAL$Group.1,RMS_TOTAL$IBA.input.headerPress.BOT,
type␣=␣'l',main␣=␣"RMS␣headerPress.BOT")
abline(v␣=c(18031,18080,18097),col␣=␣'blue'␣)

headerPress.BOT.<-data.frame(cbind(mean_TOTAL$IBA.input.headerPress.BOT,
median_TOTAL$IBA.input.headerPress.BOT,min_TOTAL$IBA.input.headerPress.BOT,
max_TOTAL$IBA.input.headerPress.BOT,RMS_TOTAL$IBA.input.headerPress.BOT,
Std_TOTAL$IBA.input.headerPress.BOT))
names(headerPress.BOT.)<-c("mean","median","min","max","RMS","Std")
headerPress.BOT.<-na.omit(headerPress.BOT.)
########pca
PCA_headerPress.BOT.<-␣prcomp(headerPress.BOT.,scale␣=␣TRUE)
summary(PCA_headerPress.BOT.)
plot(PCA_headerPress.BOT.,main␣=␣"headerPress.BOT")
cor(headerPress.BOT.,PCA_headerPress.BOT.$x[,c(1,2,3,4)])
pc1<-PCA_headerPress.BOT.$x[,1]
pc2<-PCA_headerPress.BOT.$x[,2]
pc3<-PCA_headerPress.BOT.$x[,3]
pc4<-PCA_headerPress.BOT.$x[,4]
PC<-data.frame(pc1,pc2,pc3,pc4)
#####Hcluster############
distance<-dist(PC[,c(1,2,3)])
cluster<-hclust(distance)
plot(cluster,␣main␣=␣"headerPress.BOT")
rect.hclust(cluster,k=3,border␣=␣'red')
check<-cutree(cluster,2)
check
which(check␣==␣2)
plot(PC[,1:3],col␣=␣check,␣main␣=␣"headerPress.BOT")
plot(pc1,type␣=␣'l',␣main␣=␣"headerPress.BOT")
lines(pc2,col␣=␣'blue')
lines(pc3,col␣=␣"red")
lines(pc4,col␣=␣"green")
##K␣means␣cluster
kc<-kmeans(PC[,1:3],3)
kc
which(kc$cluster==3)
plot(kc$cluster,col␣=␣kc$cluster,␣main␣=␣"headerPress.BOT")

##EXPLORE␣ON␣EMASS############

setwd("/Volumes/Transcend/EMASS/Preprocess␣and␣visulization␣")
Cycle1<-read.csv("Cycle1.csv")
```

```
Cycle2<-read.csv("Cycle2.csv")
Cycle3<-read.csv("Cycle3.csv")
Cycle4<-read.csv("Cycle4.csv")
TwoDays<-read.csv("TwoDays.csv")
Cycle1<-Cycle1[,-1]
Cycle2<-Cycle2[,-1]
Cycle3<-Cycle3[,-1]
Cycle4<-Cycle4[,-1]
TwoDays<-TwoDays[,-1]
head(TwoDays)
head(Cycle1)
Train<-bind_rows(Cycle1,Cycle2,TwoDays,Cycle3,Cycle4)
Train$X<-as.Date(Train$X)
head(Train)
unique(Train$X)
######␣Feature␣extraction#################
mean<-aggregate(Train[,3:35],␣by=list(Train$X),
FUN=mean,␣na.rm=TRUE)
mean$Group.1<-as.Date(mean$Group.1,origin␣=␣"1970-01-01")
min<-␣aggregate(Train[,3:35],␣by=list(Train$X),
FUN=min,␣na.rm=TRUE)
min$Group.1<-as.Date(min$Group.1,origin␣=␣"1970-01-01")
max<-␣aggregate(Train[,3:35],␣by=list(Train$X),
FUN=max,␣na.rm=TRUE)
max$Group.1<-as.Date(max$Group.1,origin␣=␣"1970-01-01")
Std<-␣aggregate(Train[,3:35],␣by=list(Train$X),
FUN=sd,␣na.rm=TRUE)
Std$Group.1<-as.Date(Std$Group.1,origin␣=␣"1970-01-01")
median<-␣aggregate(Train[,3:35],␣by=list(Train$X),
FUN=median,␣na.rm=TRUE)
median$Group.1<-as.Date(median$Group.1,origin␣=␣"1970-01-01")
kurtosis<-aggregate(Train[,3:35],␣by=list(as.double(unlist(Train$X))),
FUN=kurtosis,␣na.rm=TRUE)
kurtosis$Group.1<-as.Date(kurtosis$Group.1,origin␣=␣"1970-01-01")
skewness<-aggregate(Train[,3:35],␣by=list(as.double(unlist(Train$X))),
FUN=skewness,␣na.rm=TRUE)
skewness$Group.1<-as.Date(skewness$Group.1,origin␣=␣"1970-01-01")
RMS<-aggregate(Train[,3:35],␣by=list(as.double(unlist(Train$X))),
FUN=RMS)
RMS$Group.1<-as.Date(RMS$Group.1,origin␣=␣"1970-01-01")
#########␣␣PAC␣ON␣EMASS␣␣␣##########
Train<-Train[,-35]
Train␣<-␣na.omit(Train)
myPr<-␣prcomp(Train[,3:34],scale␣=␣TRUE)
summary(myPr)
plot(myPr,␣main␣=␣"PCA␣on␣Emass␣variables")
cor(Train[,3:34],myPr$x[,1:13])
######
myPr.var<-myPr$sdev^2
```

```
myPr.var
myPr.var.per<-round(myPr.var/sum(myPr.var)*100,1)
myPr.var.per
barplot(myPr.var.per,main␣=␣"PCA",xlab␣=␣"Principal␣Component",ylab␣=␣"Percent␣Variation")
########PCA␣ON␣ONLY␣avg␣DISTANCE
head(Day_avg1)
pca<-prcomp(Day_avg1[,c(40:47)],scale␣=␣TRUE)
pca
summary(pca)
plot(pca,␣type␣=␣'l')
######Distance␣A5-A5,␣DistanceB2-B5,Current␣B4,␣Samole_Block_Index############
#Distance␣A2
plot(mean$Group.1,mean$Distance.A2,type␣='l',main␣=␣"mean␣Distance.A2")
abline(v␣=c(18080,18097,18124),col␣=␣'blue')
plot(median$Group.1,median$Distance.A2,type␣=␣'l',main␣=␣"median␣Distance.A2")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(min$Group.1,min$Distance.A2,type␣=␣'l',main␣=␣"min␣Distance.A2")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(max$Group.1,max$Distance.A2,type␣=␣'l',main␣=␣"max␣Distance.A2")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(kurtosis$Group.1,kurtosis$Distance.A2,type␣=␣'l',main␣=␣"kurtosis␣Distance.A2")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(skewness$Group.1,skewness$Distance.A2,type␣=␣'l',main␣=␣"skewness␣Distance.A2")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(Std$Group.1,Std$Distance.A2,type␣=␣'l',main␣=␣"Std␣Distance.A2")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(RMS$Group.1,RMS$Distance.A2,type␣=␣'l',main␣=␣"RMS␣Distance.A2")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
Vibration_A2<-data.frame(cbind(mean$Distance.A2,
median$Distance.A2,min$Distance.A2,
max$Distance.A2,RMS$Distance.A2,
skewness$Distance.A2,kurtosis$Distance.A2,
Std$Distance.A2))
names(Vibration_A2)<-c("mean","median",
"min","max","RMS","skewness","kurtosis","Std")
Vibration_A2<-na.omit(Vibration_A2)
########pca
PCA_Vibration_A2<-␣prcomp(Vibration_A2,scale␣=␣TRUE)
summary(PCA_Vibration_A2)
plot(PCA_Vibration_A2,main="PCA_Vibration_A2")
pc1<-PCA_Vibration_A2$x[,1]
pc2<-PCA_Vibration_A2$x[,2]
pc3<-PCA_Vibration_A2$x[,3]
pc4<-PCA_Vibration_A2$x[,4]
PC<-data.frame(pc1,pc2,pc3,pc4)
#####Hcluster############
distance<-dist(PC[,c(1,2,3)])
cluster<-hclust(distance)
plot(cluster,main␣=␣"Hcluster␣Vibration_A2␣")
```

```
rect.hclust(cluster,k=2,border␣=␣'red')
check<-cutree(cluster,2)
which(check␣==␣2)
plot(PC[,1:4],col␣=␣check,main␣="principle␣component␣Vibration_A2")
plot(pc1,type␣=␣'l',main="PC_Vibration_A2")
lines(pc2,col␣=␣'blue')
lines(pc3,col␣=␣"red")
lines(pc4,col␣=␣"green")
##K␣means␣cluster
kc<-kmeans(PC[,1:4],3)
kc
plot(kc$cluster,col␣=␣kc$cluster,main="Vibration_A2")
##############################
#Distance␣A5
plot(mean$Group.1,mean$Distance.A5,type␣='l',main␣=␣"mean␣Distance.A5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue')
plot(median$Group.1,median$Distance.A5,type␣=␣'l',main␣=␣"median␣Distance.A5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(min$Group.1,min$Distance.A5,type␣=␣'l',main␣=␣"min␣Distance.A5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(max$Group.1,max$Distance.A5,type␣=␣'l',main␣=␣"max␣Distance.A5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(kurtosis$Group.1,kurtosis$Distance.A5,type␣=␣'l',main␣=␣"kurtosis␣Distance.A5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(skewness$Group.1,skewness$Distance.A5,type␣=␣'l',main␣=␣"skewness␣Distance.A5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(Std$Group.1,Std$Distance.A5,type␣=␣'l',main␣=␣"Std␣Distance.A5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(RMS$Group.1,RMS$Distance.A5,type␣=␣'l',main␣=␣"RMS␣Distance.A5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
Vibration_A5<-data.frame(cbind(mean$Distance.A5,
median$Distance.A5,min$Distance.A5,
max$Distance.A5,RMS$Distance.A5,
skewness$Distance.A5,kurtosis$Distance.A5,
Std$Distance.A5))
names(Vibration_A5)<-c("mean","median",
"min","max","RMS","skewness","kurtosis","Std")
Vibration_A5<-na.omit(Vibration_A5)
########pca
PCA_Vibration_A5<-␣prcomp(Vibration_A5,scale␣=␣TRUE)
summary(PCA_Vibration_A5)
plot(PCA_Vibration_A5,main␣=␣"PCA_Vibration_A5")
pc1<-PCA_Vibration_A5$x[,1]
pc2<-PCA_Vibration_A5$x[,2]
pc3<-PCA_Vibration_A5$x[,3]
pc4<-PCA_Vibration_A5$x[,4]
PC<-data.frame(pc1,pc2,pc3,pc4)
#####Hcluster############
distance<-dist(PC[,c(1,2,3)])
```

```
cluster<-hclust(distance)
plot(cluster,␣main␣=␣"Hcluster␣Vibration_A5")
rect.hclust(cluster,k=2,border='red')
check<-cutree(cluster,2)
which(check␣==␣1)
plot(PC[,1:4],col␣=␣check,main="principle␣component␣Vibration_A5")
plot(pc1,type␣=␣'l',main="PC_Vibration_A5")
lines(pc2,col␣=␣'blue')
lines(pc3,col␣=␣"red")
lines(pc4,col␣=␣"green")
##K␣means␣cluster
kc<-kmeans(PC[,1:4],3)
kc
plot(kc$cluster,col=kc$cluster,main="Vibration_A5")
which(kc$cluster==2)
###################################
#Distance␣B5
plot(mean$Group.1,mean$Distance.B5,type␣='l',
main␣=␣"mean␣Distance.B5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue')
plot(median$Group.1,median$Distance.B5,type␣=␣'l',
main␣=␣"median␣Distance.B5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(min$Group.1,min$Distance.B5,type␣=␣'l',
main␣=␣"min␣Distance.B5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(max$Group.1,max$Distance.B5,type␣=␣'l',
main␣=␣"max␣Distance.B5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(kurtosis$Group.1,kurtosis$Distance.B5,type␣=␣'l',
main␣=␣"kurtosis␣Distance.B5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(skewness$Group.1,skewness$Distance.B5,type␣=␣'l',
main␣=␣"skewness␣Distance.B5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(Std$Group.1,Std$Distance.B5,type␣=␣'l',
main␣=␣"Std␣Distance.B5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(RMS$Group.1,RMS$Distance.B5,type␣=␣'l',
main␣=␣"RMS␣Distance.B5")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
Vibration_B5<-data.frame(cbind(mean$Distance.B5,
median$Distance.B5,min$Distance.B5,max$Distance.B5,
RMS$Distance.B5,skewness$Distance.B5,
kurtosis$Distance.B5,Std$Distance.B5))
names(Vibration_B5)<-c("mean","median",
"min","max","RMS","skewness","kurtosis","Std")
Vibration_B5<-na.omit(Vibration_B5)
########pca
```

```
PCA_Vibration_B5<-␣prcomp(Vibration_B5,scale␣=␣TRUE)
summary(PCA_Vibration_B5)
plot(PCA_Vibration_B5,main␣=␣"PCA_Vibration_B5")
pc1<-PCA_Vibration_B5$x[,1]
pc2<-PCA_Vibration_B5$x[,2]
pc3<-PCA_Vibration_B5$x[,3]
pc4<-PCA_Vibration_B5$x[,4]
PC<-data.frame(pc1,pc2,pc3,pc4)
#####Hcluster############
distance<-dist(PC[,c(1,2,3)])
cluster<-hclust(distance)
plot(cluster,main␣=␣"Hcluster␣Vibration_B5")

check<-cutree(cluster,2)
which(check␣==␣2)
plot(PC[,1:4],col=check,main="principle␣component␣Vibration_B5")
plot(pc1,type␣=␣'l',main␣=␣"PC_Vibration_B5")
lines(pc2,col␣=␣'blue')
lines(pc3,col␣=␣"red")
lines(pc4,col␣=␣"green")
##K␣means␣cluster
kc<-kmeans(PC[,1:4],3)
kc
plot(kc$cluster,col␣=␣kc$cluster,main␣=␣"Vibration_B5")
which(kc$cluster==2)
plots.dir.path␣<-␣list.files(tempdir(),␣pattern="rs-graphics",␣full.names␣=␣TRUE)
plots.png.paths␣<-␣list.files(plots.dir.path,␣pattern=".png",␣full.names␣=␣TRUE)
file.copy(from=plots.png.paths,␣to="~/Google␣Drive/VIB_B5")
#Current␣B4
plot(mean$Group.1,mean$Current.B4,type␣='l',
main␣=␣"mean␣Current.B4")
abline(v␣=c(18080,18097,18124),col␣=␣'blue')
plot(median$Group.1,median$Current.B4,type␣=␣'l',
main␣=␣"median␣Current.B4")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(min$Group.1,min$Current.B4,type␣=␣'l',
main␣=␣"min␣Current.B4")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(max$Group.1,max$Current.B4,type␣=␣'l',
main␣=␣"max␣Current.B4")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(kurtosis$Group.1,kurtosis$Current.B4,type␣=␣'l',
main␣=␣"kurtosis␣Current.B4")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(skewness$Group.1,skewness$Current.B4,type␣=␣'l',
main␣=␣"skewness␣Current.B4")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(Std$Group.1,Std$Current.B4,type␣=␣'l',
main␣=␣"Std␣Current.B4")
```

```
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(RMS$Group.1,RMS$Current.B4,type␣=␣'l',
main␣=␣"RMS␣Current.B4")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
Current.B4<-data.frame(cbind(mean$Current.B4,
median$Current.B4,min$Current.B4,max$Current.B4,
RMS$Current.B4,skewness$Current.B4,kurtosis$Current.B4,
Std$Current.B4))
names(Current.B4)<-c("mean","median",
"min","max","RMS","skewness","kurtosis","Std")
Current.B4<-na.omit(Current.B4)
########pca
PCA_Current.B4<-␣prcomp(Current.B4,scale␣=␣TRUE)
summary(PCA_Current.B4)
plot(PCA_Current.B4,main␣=␣"PCA_Current.B4")
pc1<-PCA_Current.B4$x[,1]
pc2<-PCA_Current.B4$x[,2]
pc3<-PCA_Current.B4$x[,3]
pc4<-PCA_Current.B4$x[,4]
PC<-data.frame(pc1,pc2,pc3,pc4)
#####Hcluster############
distance<-dist(PC[,c(1,2,3)])
cluster<-hclust(distance)
plot(cluster,main␣=␣"Hcluster␣Current_B4")
check<-cutree(cluster,2)
which(check␣==␣2)
plot(PC[,1:4],col␣=␣check,main␣=␣"principle␣component␣Current.B4")
plot(pc1,type␣=␣'l',main␣=␣"PC_Current.B4")
lines(pc2,col␣=␣'blue')
lines(pc3,col␣=␣"red")
lines(pc4,col␣=␣"green")
##K␣means␣cluster
kc<-kmeans(PC[,1:4],3)
kc
plot(kc$cluster,col␣=␣kc$cluster,main␣=␣"Current.B4")
which(kc$cluster==2)
plots.dir.path␣<-␣list.files(tempdir(),␣pattern="rs-graphics",␣full.names␣=␣TRUE)
plots.png.paths␣<-␣list.files(plots.dir.path,␣pattern=".png",␣full.names␣=␣TRUE)
file.copy(from=plots.png.paths,␣to="~/Google␣Drive/CUR_B4")

##################Sample_Block_Index########
plot(mean$Group.1,mean$Sample_Block_Index,type␣='l',
main␣=␣"mean␣Sample_Block_Index")
abline(v␣=c(18080,18097,18124),col␣=␣'blue')
plot(median$Group.1,median$Sample_Block_Index,
type␣=␣'l',main␣=␣"median␣Sample_Block_Index")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(min$Group.1,min$Sample_Block_Index,type␣=␣'l',
main␣=␣"min␣Sample_Block_Index")
```

```
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(max$Group.1,max$Sample_Block_Index,type␣=␣'l',
main␣=␣"max␣Sample_Block_Index")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(kurtosis$Group.1,kurtosis$Sample_Block_Index,
type␣=␣'l',main␣=␣"kurtosis␣Sample_Block_Index")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(skewness$Group.1,skewness$Sample_Block_Index,
type␣=␣'l',main␣=␣"skewness␣Sample_Block_Index")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(Std$Group.1,Std$Sample_Block_Index,type␣=␣'l',
main␣=␣"Std␣Sample_Block_Index")
abline(v␣=c(18080,18097,18124),col␣=␣'blue'␣)
plot(RMS$Group.1,RMS$Sample_Block_Index,type␣=␣'l',
main␣=␣"RMS␣Sample_Block_Index")
abline(v␣=c(18080,18097,18124),col='blue'␣)
Sample_Block_Index<-data.frame(cbind(mean$Sample_Block_Index,median$Sample_Block_Index,
min$Sample_Block_Index,max$Sample_Block_Index,
RMS$Sample_Block_Index,skewness$Sample_Block_Index,
kurtosis$Sample_Block_Index,Std$Sample_Block_Index))
names(Sample_Block_Index)<-c("mean","median",
"min","max","RMS","skewness","kurtosis","Std")
Sample_Block_Index<-na.omit(Sample_Block_Index)
########pca
PCA_Sample_Block_Index<-␣prcomp(Sample_Block_Index[,-3],
scale␣=␣TRUE)
summary(PCA_Sample_Block_Index)
plot(PCA_Sample_Block_Index,
main␣=␣"PCA_Sample_Block_Index")
pc1<-PCA_Sample_Block_Index$x[,1]
pc2<-PCA_Sample_Block_Index$x[,2]
pc3<-PCA_Sample_Block_Index$x[,3]
pc4<-PCA_Sample_Block_Index$x[,4]
PC<-data.frame(pc1,pc2,pc3,pc4)
#####Hcluster############
distance<-dist(PC[,c(1,2)])
cluster<-hclust(distance)
plot(cluster)
rect.hclust(cluster,k=2,border='red')
check<-cutree(cluster,2)
which(check␣==␣2)
plot(PC[,1:4],col␣=␣check,main="principle␣component␣Sample_Block_Index")
plot(pc1,type␣=␣'l',main="PC_Sample_Block_Index")
lines(pc2,col␣='blue')
lines(pc3,col␣="red")
lines(pc4,col␣="green")
##K␣means␣cluster
kc<-kmeans(PC[,1:2],3)
kc
```

```
plot(kc$cluster,col␣=␣kc$cluster,main␣="Sample_Block_Index")
which(kc$cluster==2)

########pca
PCA_distance_A1<-␣prcomp(Vibration_A1,scale␣=␣TRUE)
summary(PCA_distance_A1)
plot(PCA_distance_A1,type␣=␣'l')
cor(Vibration_A1,PCA_distance_A1$x[,c(1,2,3)])
############statistic
pc1<-PCA_distance_A1$x[,1]
pc2<-PCA_distance_A1$x[,2]
pc3<-PCA_distance_A1$x[,3]
pc4<-PCA_distance_A1$x[,4]
PC<-data.frame(pc1,pc2,pc3,pc4)
PC<-scale(PC)
#################hcluster
distance<-dist(PC[,c(1,2,3,4)])
cluster<-hclust(distance)
plot(cluster)
##K␣means␣cluster
kc<-kmeans(PC[,1:4],3)
kc
plot(kc$cluster,col=kc$cluster,type␣=␣'l')
abline(v␣=c(18080,18097),col␣=␣'blue'␣)
plot3d(pc1,
pc2,
pc3,
xlab␣=␣"pc1",
ylab␣=␣"pc2",
zlab␣=␣"pc3",
col␣=kc$cluster,
size␣=␣8)
#export␣all␣plots␣into␣a␣directory#
plots.dir.path␣<-␣list.files(tempdir(),␣pattern="rs-graphics",␣full.names␣=␣TRUE)
plots.png.paths␣<-␣list.files(plots.dir.path,␣pattern=".png",␣full.names␣=␣TRUE)
file.copy(from=plots.png.paths,␣to="~/Google␣Drive/VIB_A5")
```

## C.4 Modeling code

```
library(e1071)
library("readxl")
library("readr")
library("dplyr")
library("plyr")
library(reader)
library("BBmisc")
library(reshape)
library(tidyr)
```

```
library(plotly)
library("scatterplot3d")
library("rgl")
library("RColorBrewer")
library("neuralnet")
library(pls)
library(hydroGOF)
library(randomForest)
library(caTools)

##############################################set␣working␣directory###################
setwd("~/Google␣Drive/Thesis/Data␣Collection/iba")
iba_1<-read.csv("iba_1.csv")
iba_2<-read.csv("iba_2.csv")
iba_3<-read.csv("iba_3.csv")
iba_4<-read.csv("iba_4.csv")
iba_5<-read.csv("iba_5.csv")
iba_6<-read.csv("iba_6.csv")
iba_7<-read.csv("iba_7.csv")
iba_8<-read.csv("iba_8.csv")
iba_9<-read.csv("iba_9.csv")
iba_10<-read.csv("iba_10.csv")
iba_11<-read.csv("iba_11.csv")
IBA<-rbind(iba_1,iba_2,iba_3,iba_4,iba_5,iba_6,iba_7,iba_8,iba_9,iba_10,iba_11)

#####␣Read␣splitted␣iba␣data␣files#############################################################
Tension␣<-data.frame(IBA$Date,IBA$input.stripTension,IBA$input.StripTensionRef)
##Read␣strip␣tension␣and␣trip␣tension␣reference␣from␣iba␣data␣files###
Tension<-na.omit(Tension)
##Delete␣all␣lines␣that␣contains␣missing␣value
Reduced<-read.csv("IBA_REDUCED.csv")
##Read␣pre-cleaned␣data␣from␣IBA␣####
Tension$IBA.Date<-as.Date(Reduced$IBA.Date)
##Extract␣date␣from␣the␣reduced␣file␣####pay␣special␣attention␣to␣the␣data␣format!!####
Tension$IBA.input.stripTension<-as.numeric(as.character(Tension$IBA.input.stripTension))
Tension$IBA.input.StripTensionRef<-as.numeric(as.character(Tension$IBA.input.StripTensionRef))

########################setting␣directory␣for␣another␣data␣directory#####################
setwd("~/Google␣Drive/Thesis/Data␣Collection/Data␣Warehouse")

#Read␣the␣other␣files␣that␣contains␣length␣and␣surface␣data##############################
DWH2<-read_excel("NUM_TURN.xlsx")
DWH2$DATE<-as.Date(DWH2$DATE)
head(DWH2)

###aggregate␣the␣length␣by␣date#########################################################
sum<-aggregate(DWH2[,3:5],␣by=list(DWH2$DATE),␣FUN=sum,␣na.rm=TRUE)
temp_mean<-aggregate(DWH2[,8:9],␣by=list(DWH2$DATE),␣FUN=mean,␣na.rm=TRUE)
colnames(sum)<-c(\Date","FIN_COIL_LENGTH_PRIME","EXIT_SCRAP_LENGTH","FIN_COIL_SURFACE_AREA")
```

```
colnames(temp_mean)<-c("Date","BATH_GV","BATH_GF")
class(sum$Date)
class(temp_mean$Date)
sum<-cbind(sum,temp_mean$BATH_GV,temp_mean$BATH_GF)

####Expolre␣each␣cycle␣############################################################
DWH_cycle1<-subset(sum,sum$Date>="2019-05-15"␣&␣sum$Date<="2019-06-06")
DWH_cycle2<-subset(sum,sum$Date>="2019-06-07"␣&␣sum$Date<="2019-07-04")
DWH_cycle3<-subset(sum,sum$Date>="2019-07-05"␣&␣sum$Date<="2019-07-18")
DWH_cycle4<-subset(sum,sum$Date>="2019-07-18"␣&␣sum$Date<="2019-07-20")
DWH_cycle5<-subset(sum,sum$Date>="2019-07-20"␣&␣sum$Date<="2019-08-16")
DWH_cycle6<-subset(sum,sum$Date>="2019-08-16"␣&␣sum$Date<="2019-09-12")
#DWH_cycle7<-subset(sum,sum$Date>="2019-09-12"␣&␣sum$Date<="2019-10-04")

####Separate␣data␣per␣cycle␣and␣put␣them␣into␣different␣data␣frame#####################
cycle1<-c(sum(DWH_cycle1$FIN_COIL_LENGTH_PRIME),sum(DWH_cycle1$EXIT_SCRAP_LENGTH)
,sum(DWH_cycle1$FIN_COIL_SURFACE_AREA),mean(DWH_cycle1$`temp_mean$BATH_GV`),
mean(DWH_cycle1$`temp_mean$BATH_GF`))
cycle2<-c(sum(DWH_cycle2$FIN_COIL_LENGTH_PRIME),
sum(DWH_cycle2$EXIT_SCRAP_LENGTH),sum(DWH_cycle2$FIN_COIL_SURFACE_AREA),
mean(DWH_cycle2$`temp_mean$BATH_GV`),
mean(DWH_cycle2$`temp_mean$BATH_GF`))
cycle3<-c(sum(DWH_cycle3$FIN_COIL_LENGTH_PRIME),sum(DWH_cycle3$EXIT_SCRAP_LENGTH),
sum(DWH_cycle3$FIN_COIL_SURFACE_AREA),mean(DWH_cycle3$`temp_mean$BATH_GV`),
mean(DWH_cycle3$`temp_mean$BATH_GF`))
cycle4<-c(sum(DWH_cycle4$FIN_COIL_LENGTH_PRIME),sum(DWH_cycle4$EXIT_SCRAP_LENGTH),
sum(DWH_cycle4$FIN_COIL_SURFACE_AREA),mean(DWH_cycle4$`temp_mean$BATH_GV`),
mean(DWH_cycle4$`temp_mean$BATH_GF`))
cycle5<-c(sum(DWH_cycle5$FIN_COIL_LENGTH_PRIME),sum(DWH_cycle5$EXIT_SCRAP_LENGTH),
sum(DWH_cycle5$FIN_COIL_SURFACE_AREA),mean(DWH_cycle5$`temp_mean$BATH_GV`),
mean(DWH_cycle5$`temp_mean$BATH_GF`))
cycle6<-c(sum(DWH_cycle6$FIN_COIL_LENGTH_PRIME),sum(DWH_cycle6$EXIT_SCRAP_LENGTH),
sum(DWH_cycle6$FIN_COIL_SURFACE_AREA),mean(DWH_cycle6$`temp_mean$BATH_GV`),
mean(DWH_cycle6$`temp_mean$BATH_GF`))
#cycle7<-c(sum(DWH_cycle7$FIN_COIL_LENGTH_PRIME),sum(DWH_cycle7$EXIT_SCRAP_LENGTH),
sum(DWH_cycle7$FIN_COIL_SURFACE_AREA))
Cycle<-data.frame(rbind(cycle1,cycle2,cycle3,cycle4,cycle5,cycle6))
names(Cycle)<-c("Total␣Length","Scrape␣Length","Total␣Surface","Temp␣GV","Temp␣GF")

###############saperate␣Tension␣into␣saperate␣cycles#################################
Tension<-na.omit(Tension)
summary(Tension$IBA.input.stripTension)
Ten_cycle1<-subset(Tension,Tension$IBA.Date>="2019-05-15"␣&␣Tension$IBA.Date<="2019-06-06")
Ten_cycle2<-subset(Tension,Tension$IBA.Date>="2019-06-07"␣&␣Tension$IBA.Date<="2019-07-04")
Ten_cycle3<-subset(Tension,Tension$IBA.Date>="2019-07-05"␣&␣Tension$IBA.Date<="2019-07-18")
Ten_cycle4<-subset(Tension,Tension$IBA.Date>="2019-07-18"␣&␣Tension$IBA.Date<="2019-07-20")
Ten_cycle5<-subset(Tension,Tension$IBA.Date>="2019-07-20"␣&␣Tension$IBA.Date<="2019-08-16")
Ten_cycle6<-subset(Tension,Tension$IBA.Date>="2019-08-16"␣&␣Tension$IBA.Date<="2019-09-12")
```

```
######################Extract␣statistical␣features␣from␣tension############################
mean_T<-c(mean(Ten_cycle1$IBA.input.stripTension),mean(Ten_cycle2$IBA.input.stripTension),
mean(Ten_cycle3$IBA.input.stripTension),mean(Ten_cycle4$IBA.input.stripTension),
mean(Ten_cycle5$IBA.input.stripTension),mean(Ten_cycle6$IBA.input.stripTension))

min_T<-c(min(Ten_cycle1$IBA.input.stripTension),min(Ten_cycle2$IBA.input.stripTension),
min(Ten_cycle3$IBA.input.stripTension),min(Ten_cycle4$IBA.input.stripTension),
min(Ten_cycle5$IBA.input.stripTension),min(Ten_cycle6$IBA.input.stripTension))

max_T<-c(max(Ten_cycle1$IBA.input.stripTension),max(Ten_cycle2$IBA.input.stripTension),
max(Ten_cycle3$IBA.input.stripTension),max(Ten_cycle4$IBA.input.stripTension),
max(Ten_cycle5$IBA.input.stripTension),max(Ten_cycle6$IBA.input.stripTension))

median_T<-c(median(Ten_cycle1$IBA.input.stripTension),
median(Ten_cycle2$IBA.input.stripTension),
median(Ten_cycle3$IBA.input.stripTension),␣median(Ten_cycle4$IBA.input.stripTension),
median(Ten_cycle5$IBA.input.stripTension),median(Ten_cycle6$IBA.input.stripTension))

skewness_T<-c(skewness(Ten_cycle1$IBA.input.stripTension),
skewness(Ten_cycle2$IBA.input.stripTension),
skewness(Ten_cycle3$IBA.input.stripTension),skewness(Ten_cycle4$IBA.input.stripTension),
skewness(Ten_cycle5$IBA.input.stripTension),skewness(Ten_cycle6$IBA.input.stripTension))

kurtosis_T<-c(kurtosis(Ten_cycle1$IBA.input.stripTension),
kurtosis(Ten_cycle2$IBA.input.stripTension),kurtosis(Ten_cycle3$IBA.input.stripTension),
kurtosis(Ten_cycle4$IBA.input.stripTension),kurtosis(Ten_cycle5$IBA.input.stripTension),
kurtosis(Ten_cycle6$IBA.input.stripTension))

sd_T<-c(sd(Ten_cycle1$IBA.input.stripTension),
sd(Ten_cycle2$IBA.input.stripTension),
sd(Ten_cycle3$IBA.input.stripTension),sd(Ten_cycle4$IBA.input.stripTension),
sd(Ten_cycle5$IBA.input.stripTension),sd(Ten_cycle6$IBA.input.stripTension))
RMS<-function(X){
␣␣rms<-sqrt(mean(X^2))
␣␣return(rms)
}
rms_T<-c(RMS(Ten_cycle1$IBA.input.stripTension),
RMS(Ten_cycle2$IBA.input.stripTension),
RMS(Ten_cycle3$IBA.input.stripTension),RMS(Ten_cycle4$IBA.input.stripTension),
RMS(Ten_cycle5$IBA.input.stripTension),RMS(Ten_cycle6$IBA.input.stripTension))
summary(Ten_cycle6$IBA.input.stripTension)
Cycle<-cbind(Cycle,mean_T,min_T,max_T,median_T,skewness_T,kurtosis_T,sd_T,rms_T)

##################Add␣label,␣days␣and␣roll␣diameter################
RollD<-c(593,600.32,596.00,578,593,578)
Days<-c(23,29,15,2,28,28)
Cycle<-cbind(Cycle,RollD,Days,label)
head(Cycle)
```

```
##############################PLSR Model####################
wearTrain <- I(Predictor[c(1,2,5,3,4),])
wearTest <- I(Predictor[6,])
colnames(wearTrain)
###This fits a model with 10 components, and includes
leave-one-out (LOO) cross-validated predictions######
wearMOD <- plsr(label~., data =wearTrain,scale = TRUE, validation = "LOO")
summary(wearMOD)
plot(RMSEP(wearMOD), legendpos = "topright")
predict(wearMOD, ncomp = 1, newdata = wearTest)

####plot correlation plot###################################
plot(wearMOD, plottype = "correlation", ncomp=1,
legendpos = "bottomleft",labels = "names", xlab = "nm",cex = 1,main = "Correlation

############plot modeling performance using different cycle as test set
Valid0_p<-c(14.42037,18.97449,8.192258,7.361097,15.66091,15.33483)
Valid1_p<-c(14.41463,18.75007,8.059945,7.444879,15.84379,15.48952)
Valid2_p<-c(14.50607,18.94005,8.075307,-2.608508,15.8689,14.14748)
Valid3_p<-c(14.2385,18.546,8.25564,0.8069429,15.90029,14.27897)
Real<-c(12,19,12,1,15,17)
RMSE<-c(rmse(Valid0_p,Real),rmse(Valid1_p,Real),rmse(Valid2_p,Real),rmse(Valid3_p,Real))
par(las=2)
p<-barplot(RMSE,names.arg=c("Features","Add Days","Add RollD and Days",
"Add BathTemp and Days"),cex.names = 0.7,main = "PLSR RMSE on different variables",
col = "cadetblue")
label<-round(RMSE,digits = 2)
text(p, label, labels=label, xpd=TRUE,pos = 3,cex = 0.8)


############Artificial neural network############################
Predictor<-Cycle
Predictor[,1:15]<-scale(Predictor[,1:15])
wearTrain <- Predictor[c(5,3,2,4,1),]
colnames(wearTrain)<-c("Total_Length", "Scrape_Length", "Total_Surface",
"Temp_GV","Temp_GF","mean_T", "min_T", "max_T",
"median_T", "skewness_T", "kurtosis_T", "sd_T",
"rms_T","RollD", "Days", "label")
wearTest <- Predictor[6,]
colnames(wearTest)<-c("Total_Length", "Scrape_Length", "Total_Surface",
"Temp_GV","Temp_GF","mean_T", "min_T", "max_T",
"median_T", "skewness_T", "kurtosis_T", "sd_T",
"rms_T","RollD", "Days", "label")
colnames(wearTrain)
nn=neuralnet(label~Total_Length+Scrape_Length+Total_Surface+Temp_GV+Temp_GF+mean_T
+min_T+max_T+median_T+skewness_T+kurtosis_T+sd_T+rms_T+RollD+Days,
data = wearTrain,hidden=c(10),linear.output = T,startweights = initialweight2)
plot(nn,cex = 0.5)
```

```
############Prediction␣using␣neural␣network####################
Predict=compute(nn,wearTest)
Predict$net.result
############record␣weights␣whenever␣there␣is␣a␣good␣result␣and␣set␣it␣as␣initial␣weights######
weights<-data.frame(nn$result.matrix)
initialweight2<-weights[4:nrow(weights),1]

###################plotting␣and␣comparison␣ANN##############
Real<-c(12,19,12,1,15,17)
Pred_1<-c(12.03502,18.90393,11.68794,1.035856,14.81206,17.13211)

Pred_2<-c(11.9911,18.38535,11.97321,␣0.7945518,15.29512,17.33582)

Pred_3<-c(12.98757,␣18.8983,7.818993,1.264413,13.57408,14.31401)

Pred_4<-c(11.9183,18.85573,10.39384,1.133062,15.38007,17.03653)

Pred_5<-c(12.10058,19.1987,11.88019,1.050418,14.77899,16.68621)

Pred_6<-c(12.19018,19.29073,11.83118,0.901541,14.11694,16.97701)
plot(Real,lty=1,␣lwd=2,main␣=␣"Model␣selection␣ANN",xlab␣=␣"Cycles",ylab␣=␣"Prediction")
lines(Pred_1,col␣=␣"red",type␣=␣'l',lty=1,␣lwd=2)
lines(Pred_2,col␣=␣"cadetblue4",lty=2,␣lwd=2)
lines(Pred_3,col␣=␣"blue",lty=1,␣lwd=2)
lines(Pred_4,col␣=␣"coral3",lty=1,␣lwd=2)
lines(Pred_5,col␣=␣"cyan4",lty=1,␣lwd=2)
lines(Pred_6,col␣=␣"darkseagreen4",lty=1,␣lwd=2)
legend("bottomleft",␣legend=c("Real",␣"Cycle1␣test␣",␣"Cycle2␣test"
,␣"Cycle3␣test",␣"Cycle4␣test",␣"Cycle5␣test",␣"Cycle6␣test"),
col=c("black","red","cadetblue4",␣"blue","coral3","cyan4",
"darkseagreen4"),␣lty=1,␣cex=0.7)

###################plot␣RMSE#################################
RMSE<-c(rmse(Pred_1,Real),rmse(Pred_2,Real),rmse(Pred_3,Real),
rmse(Pred_4,Real),rmse(Pred_5,Real),rmse(Pred_6,Real))
par(las=2)
p<-barplot(RMSE,names.arg=c("Cycle1␣test␣",␣"Cycle2␣test",␣"Cycle3␣test",
"Cycle4␣test",␣"Cycle5␣test",␣"Cycle6␣test"),cex.names␣=␣0.7,
main␣=␣"RMSE␣different␣test␣sample␣used␣to␣tune␣initial␣weight",col␣=␣rainbow(6))
label<-round(RMSE,digits␣=␣2)
text(p,␣label,␣labels=label,␣xpd=TRUE,pos␣=␣3,cex␣=␣0.8)

#################################
plot(sqrt((Pred_1-Real)^2),col␣=␣"red",type="l",␣ylim=c(-1,1),lty=1,
lwd=2,main␣=␣"Model␣selection␣ANN",xlab␣=␣"Cycles",
ylab␣=␣"Root␣Mean␣squre␣error")
abline(h␣=␣0,lty=2,␣lwd=2)
lines(sqrt((Pred_2-Real)^2),col␣=␣"cadetblue4",lty=2,␣lwd=1)
lines(sqrt((Pred_3-Real)^2),col␣=␣"blue",lty=1,␣lwd=2)
```

```
lines(sqrt((Pred_4-Real)^2),col␣=␣"coral3",lty=2,␣lwd=1)
lines(sqrt((Pred_5-Real)^2),col␣=␣"cyan4",lty=2,␣lwd=1)
lines(sqrt((Pred_6-Real)^2),col␣=␣"darkseagreen4",lty=2,␣lwd=1)
legend("topleft",␣legend=c("Cycle1␣test␣",
"Cycle2␣test",␣"Cycle3␣test",
"Cycle4␣test",␣"Cycle5␣test",
"Cycle6␣test"),col=c("red","cadetblue4",
"blue","coral3","cyan4","darkseagreen4"),␣lty=1,␣cex=0.7)

#################Random␣Forest##########################
wearTrain␣<-␣Predictor[c(6,3,4,2,5),]
colnames(wearTrain)<-c("Total_Length",
"Scrape_Length","Total_Surface",
"Temp_GV","Temp_GF","mean_T",
"min_T","max_T","median_T",
"skewness_T","kurtosis_T","sd_T",
"rms_T","RollD","Days","label")
wearTest␣<-␣Predictor[1,]
colnames(wearTest)<-c("Total_Length",
"Scrape_Length","Total_Surface",
"Temp_GV","Temp_GF","mean_T",
"min_T","max_T","median_T","skewness_T",
"kurtosis_T","sd_T","rms_T",
"RollD",␣"Days",␣"label")

#######build␣random␣forest␣on␣'training␣set'#########################
#By␣default,␣the␣number␣of␣decision␣trees␣in␣the␣forest␣is␣500
#and␣the␣number␣of␣features␣used␣as␣potential␣candidates␣for␣each␣split␣is␣3.
rf␣<-␣randomForest(label~.,data=wearTrain,importance␣=␣TRUE)

#########predict␣on␣test␣set#################################
pred␣=␣predict(rf,␣newdata=wearTest)
pred

#######get␣RMSE␣value#####################################
pred<-c(14.38167,14.21133␣,13.45033,13.5035,14.86917,13.51733␣)
real<-c(12,19,12,1,15,17)
rmse(pred,real)

##########################comparison␣of␣three␣models
Real<-c(12,19,12,1,15,17)
ANN<-c(12.03502,18.90393,11.68794,1.035856,14.81206,17.13211)
PLSR<-c(14.2385,18.546,8.25564,0.8069429,15.90029,14.27897)
RF<-c(14.38167,14.21133␣,13.45033,13.5035,14.86917,13.51733)
plot(sqrt((ANN-Real)^2),col␣=␣"red",type="l",␣lty=1,␣lwd=2,
ylim=c(-10,20),main␣=␣"Model␣prediction␣power␣comparison",xlab␣=␣"Cycles",
ylab␣=␣"Root␣Mean␣squre␣error␣of␣prediction")
abline(h␣=␣0,lty=2,␣lwd=2)
lines(sqrt((PLSR-Real)^2),col␣=␣"blue",lty=1,␣lwd=2)
```

```
lines(sqrt((RF-Real)^2),col␣=␣"cyan4",lty=1,␣lwd=2)
legend("topleft",␣legend=c("ANN␣",␣"PLSR",␣"RF"),
␣␣␣␣␣␣␣col=c("red","blue",␣"cyan4"),␣lty=1,lwd␣=␣2,␣cex=0.7)

#####
plot(Real,lty=1,␣lwd=2,xlab␣=␣"Cycles",ylab␣=␣"Prediction",main␣=␣"Model␣comparison")
lines(ANN,col␣=␣"red",lty=2,␣lwd=2)
lines(PLSR,col␣=␣"cadetblue4",lty=2,␣lwd=2)
lines(RF,␣col␣=␣"coral3",lty=2,␣lwd=2)
legend("bottomleft",␣legend=c("ANN␣",␣"PLSR",␣"RF"),
␣␣␣␣␣␣␣col=c("red","cadetblue4",␣"coral3"),␣lty=1,lwd␣=␣2,␣cex=0.7)

######
plot(x=Real,␣y=ANN,lty=1,␣lwd=2,
xlim=c(0,20),ylim=c(0,20),
xlab␣=␣"Real␣label",
ylab␣=␣"Predicted␣label␣by␣ANN")
abline(a=0,␣b=1)
plot(x␣=␣Real,xlim=c(0,20),ylim=c(0,20)
,y␣=␣PLSR,lty=1,␣lwd=2,col␣=␣'red',
xlab␣=␣"Real␣label",
ylab␣=␣"Predicted␣label␣by␣PLSR")
abline(a=0,␣b=1)
```

## C.5 Tool Development Code

```
library(shiny)
library("readxl")
library("readr")
library("dplyr")
library("plyr")
library(pls)
library(hydroGOF)
library(DT)
library(moments)
ui␣<-␣fluidPage(
␣␣titlePanel("Bush␣Wear␣prediction"),

␣␣dateInput(inputId␣=␣"date",␣label␣=␣"date",
␣␣value␣=␣"2019-11-30",␣format␣=␣"yyyy-mm-dd"),
␣␣numericInput(inputId␣=␣"num",␣label␣=␣"Sink␣Roll␣Diameter",
␣␣value␣=␣600,min␣=␣500,␣max␣=␣700,␣step␣=␣1),
␣␣#numericInput(inputId␣=␣"comp",␣label␣=␣"Num␣Component",
␣␣value␣=␣2,min␣=␣1,␣max␣=␣3,␣step␣=␣1),
␣␣numericInput(inputId␣=␣"bush",␣label␣=␣"Bush␣diameter",value␣=␣30,
␣␣min␣=␣30,␣max␣=␣40,␣step␣=␣1),
␣␣mainPanel(
␣␣␣␣DT::dataTableOutput("mytable"),
␣␣␣␣plotOutput("plot"),
␣␣␣␣htmlOutput("width")
␣␣)
)

server␣<-␣function(input,␣output)␣{
␣␣########################################outputTABLE##########
␣␣output\$mytable<-DT::renderDataTable({
␣␣␣␣CycleDate<-c("2019-05-14","2019-06-06","2019-07-04",
␣␣␣␣"2019-07-18","2019-07-20","2019-08-16","2019-09-12",
␣␣␣␣"2019-10-04","2019-10-16",␣"2019-11-18","2019-11-30")

␣␣␣␣inputDate<-c(input\$date)
␣␣␣␣count␣<-␣0
␣␣␣␣for␣(val␣in␣CycleDate)␣{
␣␣␣␣␣␣if(val␣>=␣inputDate)␣break
␣␣␣␣␣␣else␣count␣=␣count␣+␣1
␣␣␣␣}
␣␣␣␣startDate␣=␣as.Date(CycleDate[count])+1
␣␣␣␣print(startDate)
␣␣␣␣sum_sub<-subset(Len_temp,Len_temp\$Date>=startDate&Len_temp\$Date<=inputDate)
␣␣␣␣TDate<-Tension\$Date
␣␣␣␣Tension_sub<-subset(Tension,TDate>=startDate&TDate<=inputDate)
␣␣␣␣Len<-c(sum(sum_sub\$FIN_COIL_LENGTH_PRIME),sum(sum_sub\$EXIT_SCRAP_LENGTH),
␣␣␣␣sum(sum_sub\$FIN_COIL_SURFACE_AREA))
```

```
    temp_mean<-c(mean(sum_sub\$BTH_T_GV),mean(sum_sub\$BTH_T_GF))

    T_fea<-c(mean(Tension_sub\$Tension),min(Tension_sub\$Tension),
    max(Tension_sub\$Tension),median(Tension_sub\$Tension),
    skewness(Tension_sub\$Tension),kurtosis(Tension_sub\$Tension),
    sd(Tension_sub\$Tension),RMS(Tension_sub\$Tension))
    Days<-c(as.numeric(difftime(inputDate ,startDate ,units = c("days")))+1)
    inputRollD<-c(input\$num)
    RollD<-inputRollD
    Validation<-c(Len,temp_mean,T_fea,RollD,Days,1)
    validation<-data.frame(name=Validation[1])
    for (i in 2:15) {
      #name = names(Predictor)[i]
      a<-data.frame(name=Validation[i])
      validation<-cbind(validation,a)
    }

    names(validation)<-names(Predictor[1:15])
    validation
  })
  ######################################output Plot ################################3
  output\$plot <- renderPlot({
    CycleDate<-c("2019-05-14","2019-06-06","2019-07-04",
    "2019-07-18","2019-07-20","2019-08-16","2019-09-12",
    "2019-10-04","2019-10-16", "2019-11-18","2019-11-30")

    inputDate<-c(input\$date)
    count <- 0
    for (val in CycleDate) {
      if(val >= inputDate) break
      else count = count + 1
    }
    startDate = as.Date(CycleDate[count])+1
    print(startDate)
    sum_sub<-subset(Len_temp,Len_temp\$Date>=startDate&Len_temp\$Date<=inputDate)
    TDate<-Tension\$Date
    Tension_sub<-subset(Tension,TDate>=startDate&TDate<=inputDate)
    Len<-c(sum(sum_sub\$FIN_COIL_LENGTH_PRIME),
    sum(sum_sub\$EXIT_SCRAP_LENGTH),sum(sum_sub\$FIN_COIL_SURFACE_AREA))
    temp_mean<-c(mean(sum_sub\$BTH_T_GV),mean(sum_sub\$BTH_T_GF))

    T_fea<-c(mean(Tension_sub\$Tension),min(Tension_sub$Tension),max(Tension_sub$Tension),
    median(Tension_sub\$Tension),skewness(Tension_sub$Tension),
    kurtosis(Tension_sub\$Tension),sd(Tension_sub$Tension),
    RMS(Tension_sub$Tension))
    Days<-c(as.numeric(difftime(inputDate, startDate ,units = c("days")))+1)
    inputRollD<-c(input$num)
    RollD<-inputRollD
    Validation<-c(Len,temp_mean,T_fea,RollD,Days,1)
```

```
␣␣␣␣validation<-data.frame(name=Validation[1])
␣␣␣␣for␣(i␣in␣2:15)␣{
␣␣␣␣␣␣#name␣=␣names(Predictor)[i]
␣␣␣␣␣␣a<-data.frame(name=Validation[i])
␣␣␣␣␣␣validation<-cbind(validation,a)
␣␣␣␣}

␣␣␣␣names(validation)<-names(Predictor[1:15])
␣␣␣␣validation
␣␣␣␣wearTrain␣<-␣I(Predictor)
␣␣␣␣validation1<-I(validation)
␣␣␣␣wearMOD␣<-␣plsr(label~.,␣data␣=wearTrain,scale␣=␣TRUE,␣validation␣=␣"LOO")
␣␣␣␣summary(wearMOD)
␣␣␣␣#plot(RMSEP(wearMOD),␣legendpos␣=␣"topright")
␣␣␣␣class(RMSEP(wearMOD))
␣␣␣␣cverr␣<-␣RMSEP(wearMOD)$val[1,,]
␣␣␣␣imin␣<-␣which.min(cverr)-1
␣␣␣␣result<-predict(wearMOD,␣ncomp␣=␣imin,␣newdata␣=validation1␣)
␣␣␣␣##############################################################################
␣␣␣␣#count␣=␣2
␣␣␣␣my_vector␣<-␣vector("numeric")
␣␣␣␣for(count␣in␣1:round(Days)){
␣␣␣␣␣␣inputDate<-startDate+count

␣␣␣␣␣␣sum_sub<-subset(Len_temp,Len_temp\$Date>=startDate\&Len_temp$Date<=inputDate)
␣␣␣␣␣␣TDate<-Tension$Date
␣␣␣␣␣␣Tension_sub<-subset(Tension,TDate>=startDate&TDate<=inputDate)
␣␣␣␣␣␣Len<-c(sum(sum_sub\$FIN_COIL_LENGTH_PRIME),
␣␣␣␣␣␣sum(sum_sub$EXIT_SCRAP_LENGTH),sum(sum_sub\$FIN_COIL_SURFACE_AREA))
␣␣␣␣␣␣temp_mean<-c(mean(sum_sub\$BTH_T_GV),mean(sum_sub\$BTH_T_GF))

␣␣␣␣␣␣T_fea<-c(mean(Tension_sub\$Tension),min(Tension_sub\$Tension),
␣␣␣␣␣␣max(Tension_sub$Tension),median(Tension_sub$Tension),
␣␣␣␣␣␣skewness(Tension_sub$Tension),kurtosis(Tension_sub$Tension),
␣␣␣␣␣␣sd(Tension_sub$Tension),RMS(Tension_sub$Tension))
␣␣␣␣␣␣Days<-c(as.numeric(difftime(inputDate,␣startDate,␣units␣=␣c("days")))+1)
␣␣␣␣␣␣inputRollD<-c(input$num)
␣␣␣␣␣␣RollD<-inputRollD
␣␣␣␣␣␣Validation<-c(Len,temp_mean,T_fea,RollD,Days,1)
␣␣␣␣␣␣validation<-data.frame(name=Validation[1])
␣␣␣␣␣␣for␣(i␣in␣2:15)␣{
␣␣␣␣␣␣␣␣#name␣=␣names(Predictor)[i]
␣␣␣␣␣␣␣␣a<-data.frame(name=Validation[i])
␣␣␣␣␣␣␣␣validation<-cbind(validation,a)
␣␣␣␣␣␣}

␣␣␣␣␣␣names(validation)<-names(Predictor[1:15])
␣␣␣␣␣␣validation
␣␣␣␣␣␣wearTrain␣<-␣I(Predictor)
```

```
      validation1<-I(validation)
      wearMOD <- plsr(label~.,data =wearTrain,scale = TRUE,validation = "LOO")
      summary(wearMOD)
      class(RMSEP(wearMOD))
      cverr <- RMSEP(wearMOD)$val[1,,]
      imin <- which.min(cverr)-1
      result<-predict(wearMOD,ncomp = imin,newdata =validation1 )
      my_vector[count] <- result
      #ncomp[count]<- imin
    }
    plot(input$bush-my_vector,main = "Predicted wearing pattern",
    xlab = "Days",ylab = "Remaining width",col = "red",lty=2,lwd=2)

  })
  ##############################OutPutText###############################################
  output\$width<-renderText({
    CycleDate<-c("2019-05-14","2019-06-06","2019-07-04",
    "2019-07-18","2019-07-20","2019-08-16",
    "2019-09-12","2019-10-04","2019-10-16",
    "2019-11-17","2019-11-30")

    inputDate<-c(input\$date)
    count <- 0
    for (val in CycleDate) {
      if(val >= inputDate) break
      else count = count + 1
    }
    startDate = as.Date(CycleDate[count])+1
    print(startDate)
    sum_sub<-subset(Len_temp,Len_temp\$Date>=startDate&Len_temp\$Date<=inputDate)
    TDate<-Tension\$Date
    Tension_sub<-subset(Tension,TDate>=startDate&TDate<=inputDate)
    Len<-c(sum(sum_sub\$FIN_COIL_LENGTH_PRIME),
    sum(sum_sub\$EXIT_SCRAP_LENGTH),sum(sum_sub\$FIN_COIL_SURFACE_AREA))
    temp_mean<-c(mean(sum_sub\$BTH_T_GV),mean(sum_sub\$BTH_T_GF))

    T_fea<-c(mean(Tension_sub\$Tension),min(Tension_sub\$Tension),
    max(Tension_sub\$Tension),median(Tension_sub\$Tension),
    skewness(Tension_sub\$Tension),kurtosis(Tension_sub\$Tension),
    sd(Tension_sub\$Tension),RMS(Tension_sub\$Tension))
    Days<-c(as.numeric(difftime(inputDate ,startDate ,units = c("days")))+1)
    inputRollD<-c(input\$num)
    RollD<-inputRollD
    Validation<-c(Len,temp_mean,T_fea,RollD,Days,1)
    validation<-data.frame(name=Validation[1])
    for (i in 2:15) {
      a<-data.frame(name=Validation[i])
      validation<-cbind(validation,a)
    }
```

```
␣␣␣␣names(validation)<-names(Predictor[1:15])
␣␣␣␣validation
␣␣␣␣wearTrain␣<-␣I(Predictor)
␣␣␣␣validation1<-I(validation)
␣␣␣␣wearMOD␣<-␣plsr(label~.,␣data␣=wearTrain,scale␣=␣TRUE,␣validation␣=␣"LOO")
␣␣␣␣summary(wearMOD)
␣␣␣␣class(RMSEP(wearMOD))
␣␣␣␣cverr␣<-␣RMSEP(wearMOD)\$val[1,,]
␣␣␣␣imin␣<-␣which.min(cverr)-1
␣␣␣␣result<-predict(wearMOD,␣ncomp␣=␣imin,␣newdata␣=validation1␣)
␣␣␣␣str1<-paste("Bush␣wear␣width:",␣result)
␣␣␣␣str2<-paste("Remaining␣bush␣width:",␣input\$bush-result)
␣␣␣␣HTML(paste(str1,␣str2,␣sep␣=␣'<br/>'))
␣␣})

}

shinyApp(server␣=␣server,␣ui␣=␣ui)
```

APPENDIX

# D

# TENSION EXPLORATION

As an additional request from the company, the difference between Tension and Reference Tension are plotted. As the Maximum Tension and Minimum Tension are contributing the most to the prediction, we only plotted these two features and the difference they are to the reference.

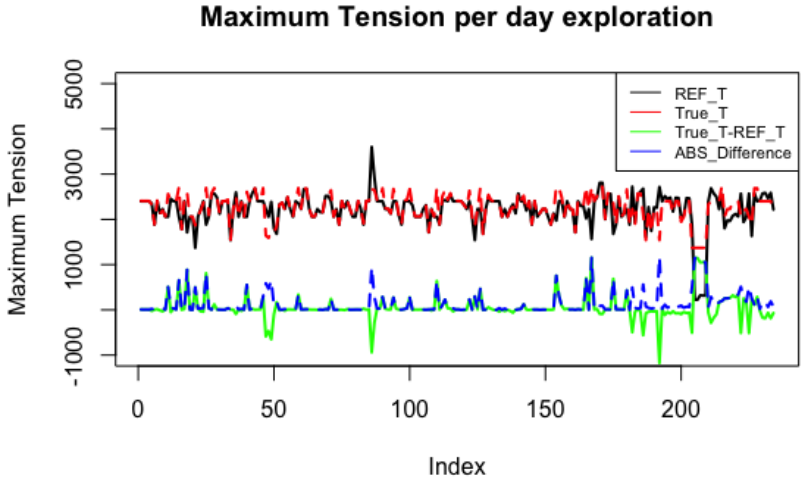


Figure D.1: Maximum Tension VS Reference maximum tension

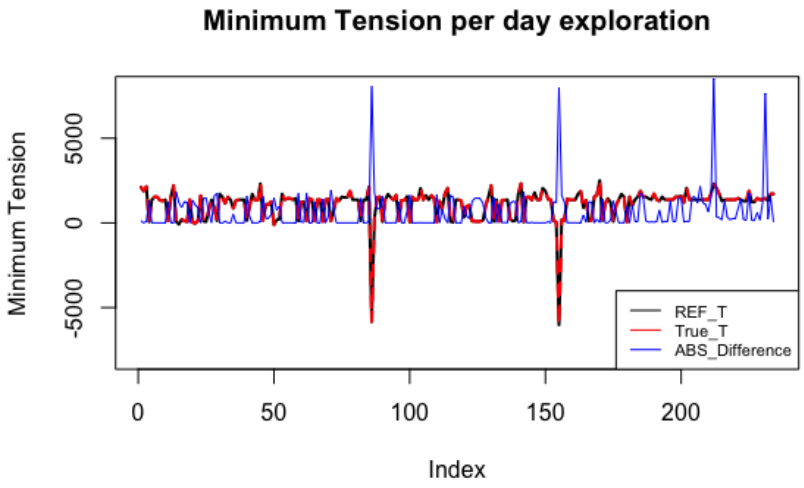


Figure D.2: Minimum Tension VS Reference minimum tension