Employee Absenteeism

Sanjib Kumar Mishra

Contents

Chapter 1

Introduction

1.1 Problem Statement

XYZ is a courier company. As we appreciate that human capital plays an important role in collection, transportation and delivery. The company is passing through genuine issue of Absenteeism. The company has shared it dataset and requested to have an answer on the following areas:

1. What changes company should bring to reduce the number of absenteeism?

2. How much losses every month can we project in 2011 if same trend of absenteeism continues?

1.2 Data

Dataset Details:

Dataset Characteristics: Timeseries Multivariant

Number of Attributes: 21

Missing Values : Yes

Attribute Information:

1. Individual identification (ID)

2. Reason for absence (ICD).

Absences attested by the International Code of Diseases (ICD) stratified into 21 categories (I to XXI) as follows:

I Certain infectious and parasitic diseases

II Neoplasms

III Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism

IV Endocrine, nutritional and metabolic diseases

V Mental and behavioural disorders

VI Diseases of the nervous system

VII Diseases of the eye and adnexa

VIII Diseases of the ear and mastoid process

IX Diseases of the circulatory system

X Diseases of the respiratory system

XI Diseases of the digestive system

XII Diseases of the skin and subcutaneous tissue

XIII Diseases of the musculoskeletal system and connective tissue

XIV Diseases of the genitourinary system

XV Pregnancy, childbirth and the puerperium

XVI Certain conditions originating in the perinatal period

XVII Congenital malformations, deformations and chromosomal abnormalities

XVIII Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified

XIX Injury, poisoning and certain other consequences of external causes

XX External causes of morbidity and mortality

XXI Factors influencing health status and contact with health services.

And 7 categories without (CID)

(22)patient follow-up

(23) medical consultation

(24) blood donation

(25) laboratory examination

(26) unjustified absence

(27) physiotherapy

(28) dental consultation

3. Month of absence

4. Day of the week (Monday (2), Tuesday (3), Wednesday (4), Thursday (5), Friday (6))

5. Seasons (summer (1), autumn (2), winter (3), spring (4))

6. Transportation expense

7. Distance from Residence to Work (kilometers)

8. Service time

9. Age

10. Work load Average/day

- Dataset is having 740 observations and 21 variables
- Sample dataset with 10 rows:

Table 1: Employee Absenteeism Sample Data(Columns: 1-6)

ID	Reason for absence	Month of absence	Day of the week	Seasons	Transportation expense
11	26	7	3	1	289
36	0	7	3	1	118
3	23	7	4	1	179
7	7	7	5	1	279
11	23	7	5	1	289
3	23	7	6	1	179
10	22	7	6	1	
20	23	7	6	1	260
14	19	7	2	1	155

1	22	7	2	1	235

Table 2: Employee Absenteeism Sample Data(Columns: 7-13)

Distance from Residence to Work	Service time	Age	Work load Average/day	Hit target	Disciplinary failure	Education
36	13	33	239,554	97	0	1
13	18	50	239,554	97	1	1
51	18	38	239,554	97	0	1
5	14	39	239,554	97	0	1
36	13	33	239,554	97	0	1
51	18	38	239,554	97	0	1
52	3	28	239,554	97	0	1
50	11	36	239,554	97	0	1
12	14	34	239,554	97	0	1
11	14	37	239,554	97	0	3

Table 3: Employee Absenteeism Sample Data(Columns: 14-21)

Son	Social drinker	Social smoker	Pet	Weight	Height	Body mass index	Absenteeism time in hours
2	1	0	1	90	172	30	4
1	1	0	0	98	178	31	0
0	1	0	0	89	170	31	2
2	1	1	0	68	168	24	4
2	1	0	1	90	172	30	2
0	1	0	0	89	170	31	
1	1	0	4	80	172	27	8
4	1	0	0	65	168	23	4
2	1	0	0	95	196	25	40
1	0	0	1	88	172	29	8

Chapter 2

Methodology

2.1 Pre Processing

Data preprocessing is a technique that involves transforming raw data into an understandable format. Real-world data is often **incomplete**, **inconsistent**, and/or **lacking** in certain **behaviors or trends**, and is likely to contain many **errors**. Data preprocessing is a proven method of resolving such issues. Data preprocessing **prepares raw data** for **further processing**.

2.1.1 Missing Value Analysis

In my Absenteeism at work dataset I found missing values in every variable and the missing percentage is less than 30% of the data in every variable. So, I used three method (Mean, Median & KNN) to fill the value in missing place and out of these three method I found KNN Imputation works good in this dataset.So, I used KNNImputation method for missing value analysis and use K value as 5.

R code for Missing value analysis:

```
missing_percentage=data.frame(colSums(is.na(employee))/nrow(employee))*100

names(missing_percentage)[1]="missing_percentage"

View(missing_percentage)

employee=knnImputation(employee,k=5)
```

Missing percentage:

	missing_percentage
reason_for_absence	2.586207
month_of_absence	2.729885
day_of_the_week	2.586207
seasons	2.586207
transportation_expense	3.448276
distance_from_residence_to_work	3.017241
service_time	3.017241
age	2.873563
work_load_average_per_day	3.735632
hit_target	3.448276
disciplinary_failure	3.304598
education	4.022989
son	3.448276
social_drinker	3.017241
social_smoker	3.160920
pet	2.873563
weight	2.729885
height	4.454023
body_mass_index	6.465517
absenteeism_time_in_hours	2.586207

2.1.2 Outlier Analysis

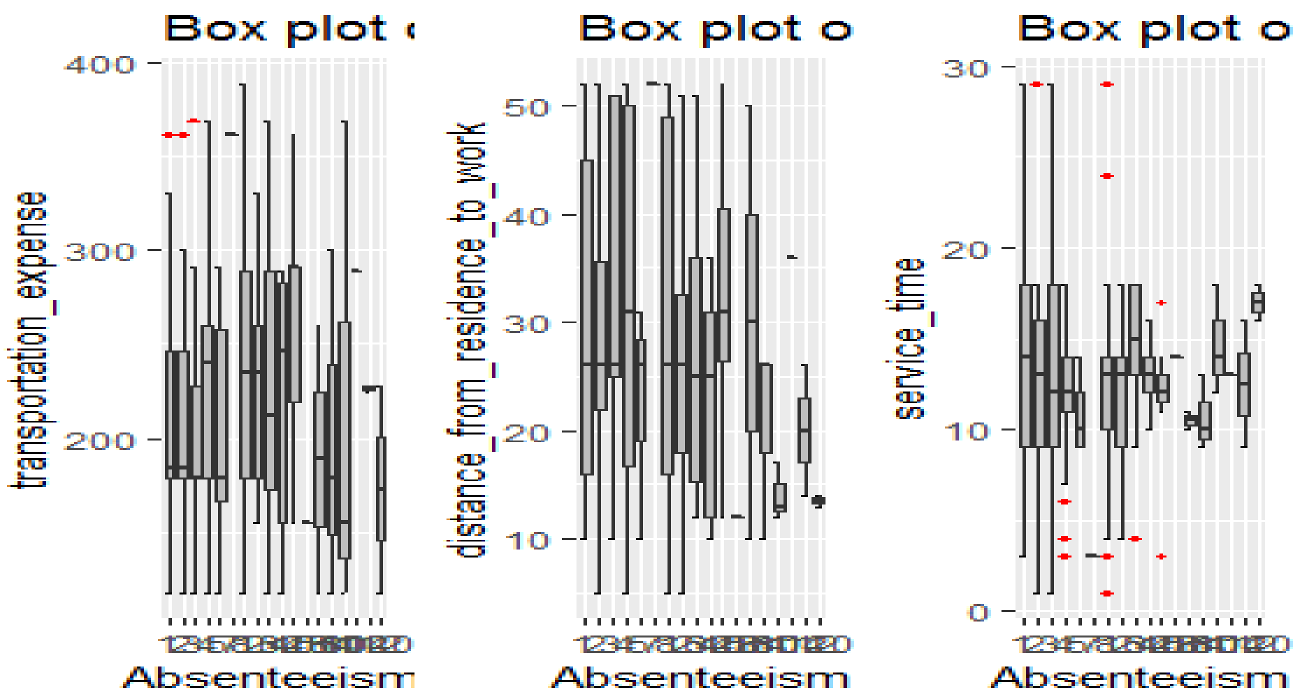
I observed from the dataset that most of the variables are skewed like; transportation expense, height, age, service time, etc. The skew in these variables can be most likely explained by the presence of outliers and extreme values in the data.

So first I replace the outliers with NA and then apply the KNNImputation method to fill these place with some proper value.

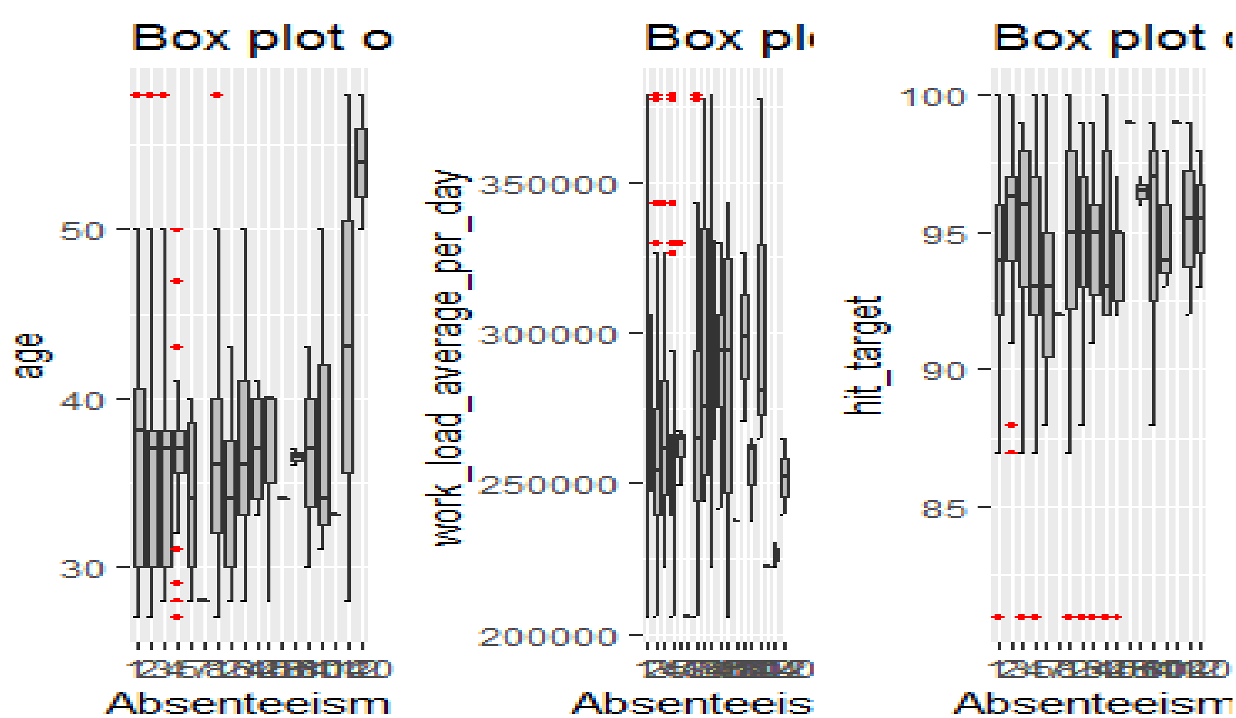
R code for outlier analysis:

```
➢ for(i in cnames)
➢ {val = employee[,i][employee[,i]%in%
          boxplot.stats(employee[,i])$out]
➢ employee[,i][employee[,i]%in%val]=NA}
➢ employee = knnImputation(employee, k = 3)
```

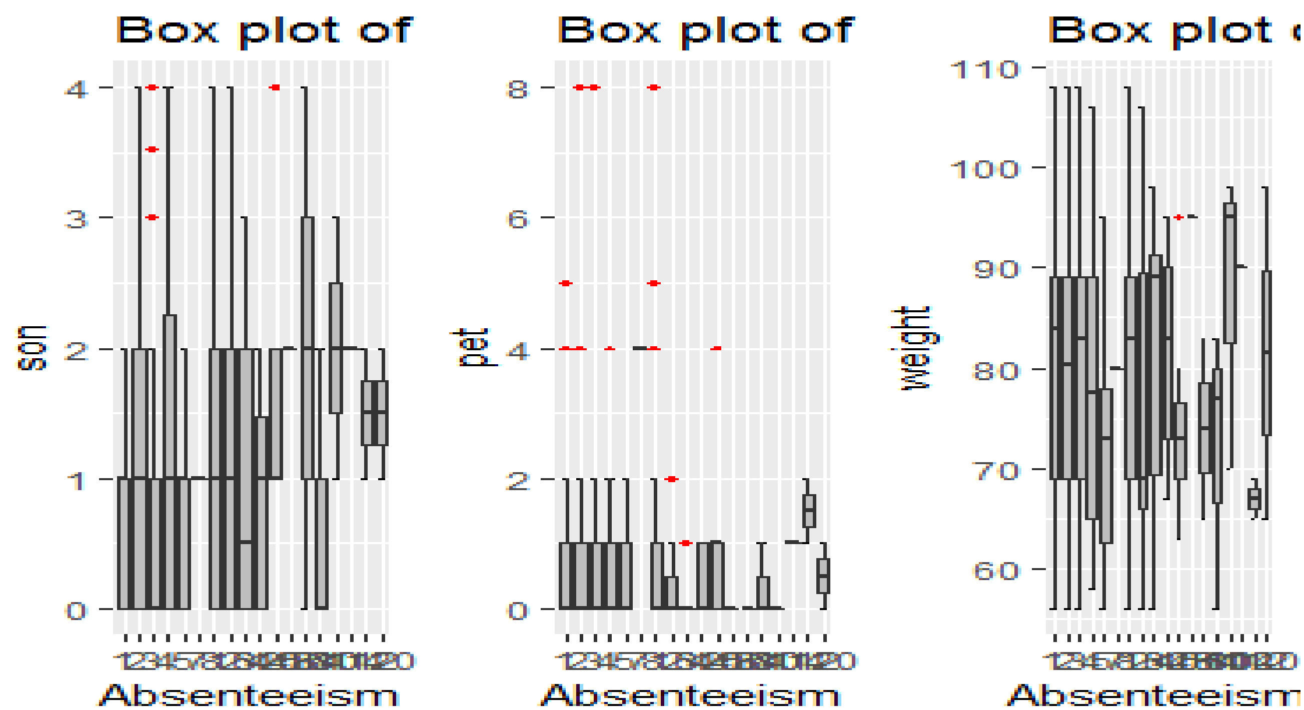
Box plot for outliers



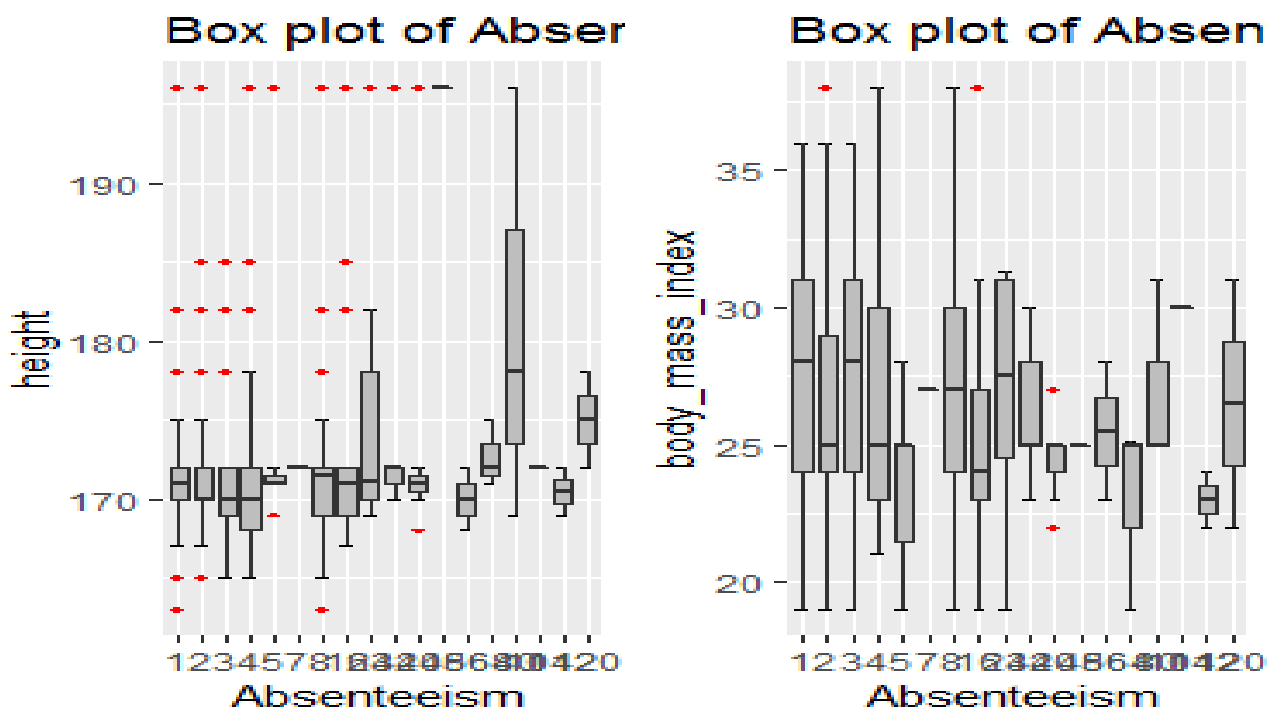
(Fig 1)



(Fig 2)


Box plot of
son
Absenteeism
Box plot of
pet
Absenteeism
weight
Absenteeism

(Fig 3)

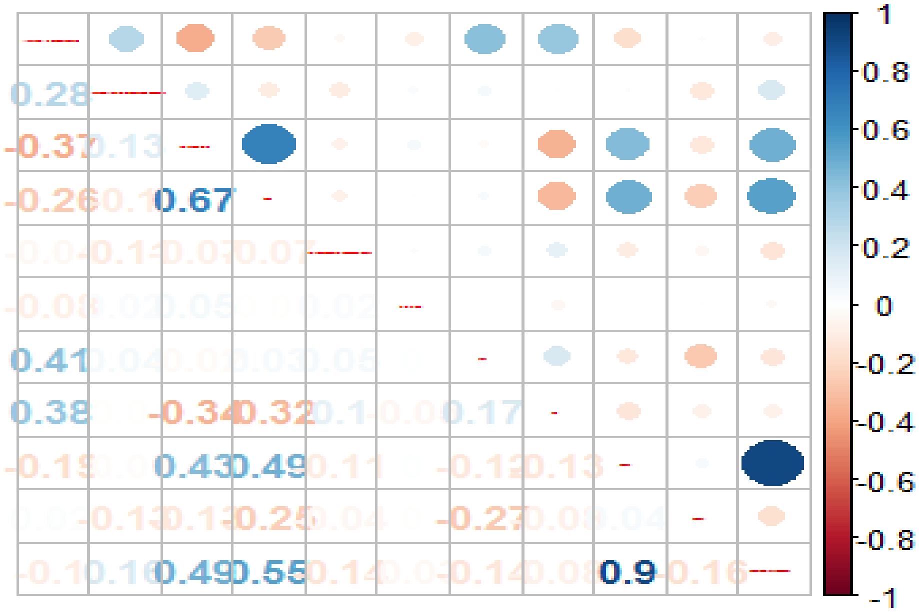

Box plot of Abser
height
Absenteeism
Box plot of Absent
body_mass_index
Absenteeism

(Fig 4)

2.1.3 Correlation Check

- In feature selection method I checked the association between two variables.
- For continuous variables I used correlation analysis which tells the direction and strength of the linear relationship between two quantitative variables.
- I found service time and age are highly correlated with each other and height and body mass index is highly correlated.
- So I remove height and age from the dataset.
- For categorical variable I used chi-square test of independence to compare two variables in a contingency table to see if they are related.
- I reject two variable social smoker and education as the p-value is greater than 0.05.

<u>Correlation analysis plot:</u>



Chi-square test results:

```
[1] "reason_for_absence"

        Pearson's Chi-squared test

data:  table(factor_data$absenteeism_time_in_hours, factor_data[, i])
X-squared = 1045.4, df = 442, p-value < 2.2e-16

[1] "month_of_absence"

        Pearson's Chi-squared test

data:  table(factor_data$absenteeism_time_in_hours, factor_data[, i])
X-squared = 300.22, df = 187, p-value = 2.835e-07

[1] "day_of_the_week"

        Pearson's Chi-squared test

data:  table(factor_data$absenteeism_time_in_hours, factor_data[, i])
X-squared = 105.47, df = 68, p-value = 0.00243

[1] "seasons"

        Pearson's Chi-squared test

data:  table(factor_data$absenteeism_time_in_hours, factor_data[, i])
X-squared = 105.13, df = 51, p-value = 1.258e-05

[1] "disciplinary_failure"

        Chi-squared test for given probabilities

data:  table(factor_data$absenteeism_time_in_hours, factor_data[, i])
X-squared = 1708, df = 17, p-value < 2.2e-16
```

```
[1] "education"

        Pearson's Chi-squared test

data:  table(factor_data$absenteeism_time_in_hours, factor_data[, i])
X-squared = 28.56, df = 51, p-value = 0.9954

[1] "social_drinker"

        Pearson's Chi-squared test

data:  table(factor_data$absenteeism_time_in_hours, factor_data[, i])
X-squared = 49.342, df = 17, p-value = 5.341e-05

[1] "social_smoker"

        Pearson's Chi-squared test

data:  table(factor_data$absenteeism_time_in_hours, factor_data[, i])
X-squared = 14.145, df = 17, p-value = 0.6568
```

2.2 Modeling

2.2.1 Model Selection

I am selecting the classification model for Absenteeism at work dataset to predict
I) What changes company should bring to reduce the number of absenteeism?

ii) How much losses every month can we project in 2011 if same trend of absenteeism continues?

For classification I applied Naive Bayes and Random Forest but Random Forest Classifier model gives the higher accuracy than Naïve Bayes. So, I choose Random Forest for the dataset.

2.2.2 Model Evaluation

I applied Random Forest model and tried to evaluate the accuracy and false negative rate for the dataset by using "ntree" as 100,200,300,500.I got the highest accuracy by using ntree as 500.

R Code and the output:

- **rf=randomForest(absenteeism_time_in_hours~.,data=train,ntree=500)**
- **predictions=predict(rf,test)**
- **confmatrix_rf=table(test$absenteeism_time_in_hours,predictions)**
- **confusionMatrix(confmatrix_rf)**

Output:

Confusion Matrix and Statistics

```
                predictions
                 greater than 4 less than 4
  greater than 4            76          28
  less than 4               14          91

               Accuracy : 0.799
                 95% CI : (0.7382, 0.8512)
    No Information Rate : 0.5694
    P-Value [Acc > NIR] : 2.152e-12

                  Kappa : 0.5978
 Mcnemar's Test P-Value : 0.04486

            Sensitivity : 0.8444
            Specificity : 0.7647
```

```
         Pos Pred Value : 0.7308
         Neg Pred Value : 0.8667
             Prevalence : 0.4306
         Detection Rate : 0.3636
   Detection Prevalence : 0.4976
      Balanced Accuracy : 0.8046

       'Positive' Class : greater than 4
```

3. Conclusion:

By using randomforest I got several rules out of these I mentioned two rules below ;

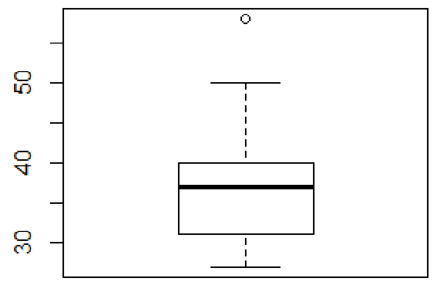
```
> readableRules[1:2,]
[1] "reason_for_absence %in% c('2','3','5','6','7','8','9','12','13','14','15','16','17','19','21','23','25','27','28
') & month_of_absence %in% c('1','2','10','12') & seasons %in% c('2','3') & age<=27.5 & son<=2.5"
[2] "reason_for_absence %in% c('2','3','5','6','7','8','9','12','13','14','15','16','17','19','21','23','25','27','28
') & month_of_absence %in% c('2','10','12') & seasons %in% c('2','3') & age>27.5 & son<=2.5"
> rulemetric[1:2,]
     len freq    err
[1,] "5" "0.004" "0.5"
[2,] "5" "0.094" "0.152"
     condition

[1,] "X[,1] %in% c('2','3','5','6','7','8','9','12','13','14','15','16','17','19','21','23','25','27','28') & X[,2] %
in% c('1','2','10','12') & X[,4] %in% c('2','3') & X[,7]<=27.5 & X[,11]<=2.5"
[2,] "X[,1] %in% c('2','3','5','6','7','8','9','12','13','14','15','16','17','19','21','23','25','27','28') & X[,2] %
in% c('2','10','12') & X[,4] %in% c('2','3') & X[,7]>27.5 & X[,11]<=2.5"
     pred
[1,] "greater than 4"
[2,] "less than 4"
```

- If the company will take care the health issue of the employee of aged less than or equal to 27.5 in the month of January, February, October and December then the company can able to reduce the absenteeism.
- If same trend of absenteeism continues then in 2011 more employee will be absent more than 5 hours.

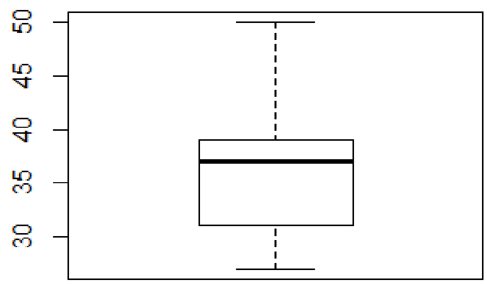
Appendix 1– Extra Figures

<u>Outliers on Age</u>

With Outliers

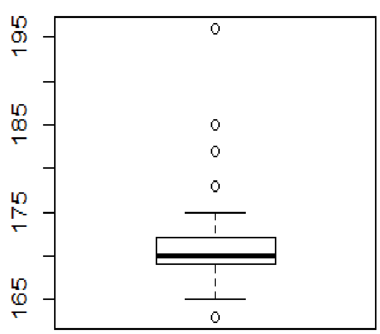


Without Outliers

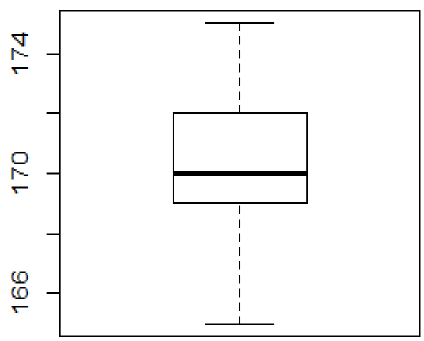


<u>Outliers on Height</u>

With Outliers

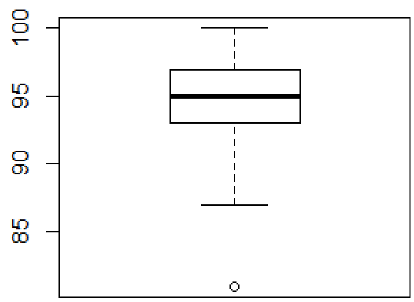


Without Outliers

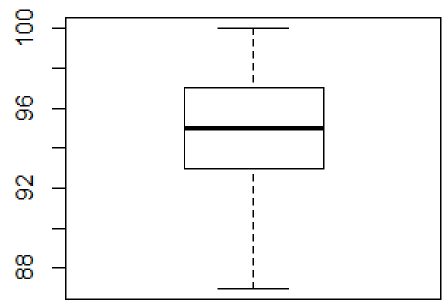


Outliers on Hit_Target

With Outliers

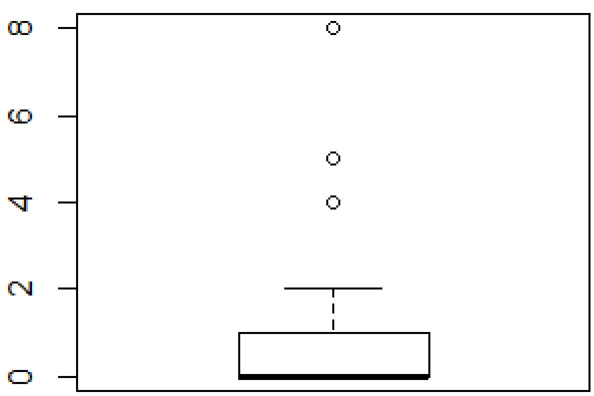


Without Outliers

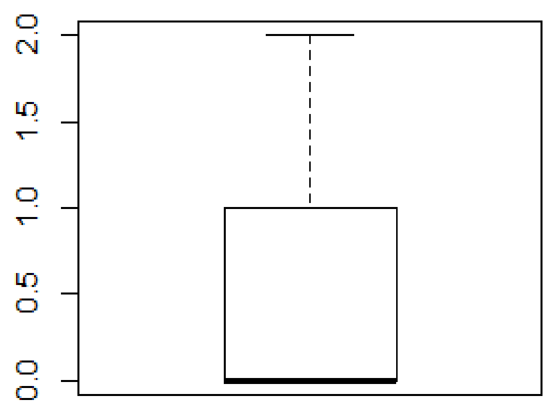


Outliers on Pet

With Outliers

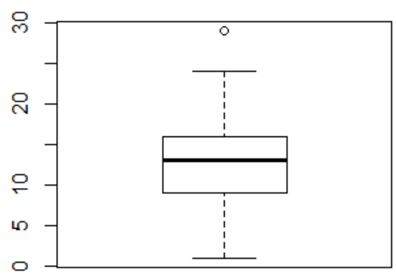


Without Outliers

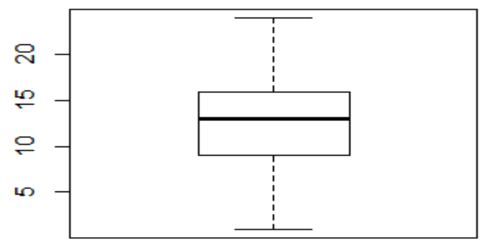


Outliers on Service Time

With Outliers

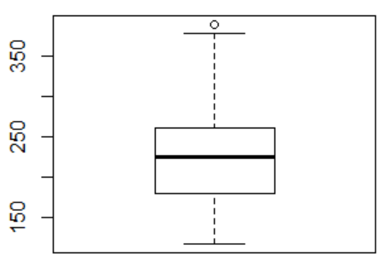


Without Outliers

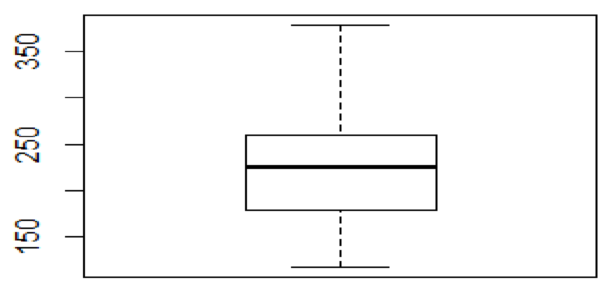


Outliers on Transportation expense

With Outliers

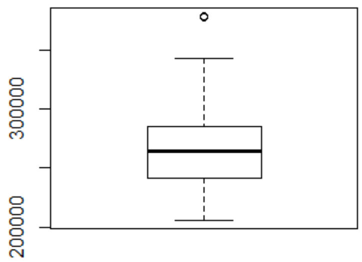


Without Outliers

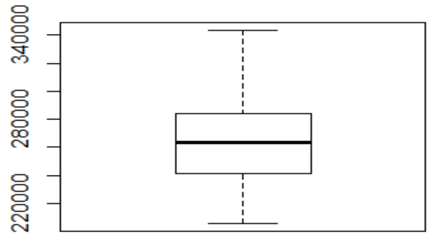


Outliers on Workload average per day

With Outliers



Without Outliers



Appendix 2 – R Code

```
rm(list=ls())
setwd("G:/data science project")
employee= read_xls("Absenteeism_at_work_Project.xls", col_names=T)
employee=as.data.frame(employee)

colnames(employee)=tolower(gsub(' ','_',colnames(employee)))
colnames(employee)
names(employee)[10]="work_load_average_per_day"
table(employee$reason_for_absence)
employee=employee[!employee$reason_for_absence==0,]
table(employee$absenteeism_time_in_hours)
employee=employee[!employee$absenteeism_time_in_hours==0,]
employee=employee[,-1]
for (i in c(1,2,3,4,11,12,14,15,20))
{
  employee[,i]=as.factor(employee[,i])
}

#library("DMwR")
missing_percentage=data.frame(colSums(is.na(employee))/nrow(employee))*100
names(missing_percentage)[1]="missing_percentage"
View(missing_percentage)
employee=knnImputation(employee,k=5)


#library(ggplot2)
numeric_index=sapply(employee,is.numeric)
numeric_data=employee[,numeric_index]
cnames=colnames(numeric_data)
cnames


for(i in 1:length(cnames))
{assign(paste0("gn", i),ggplot(aes_string(y = (cnames[i]),x =
"absenteeism_time_in_hours")
                              ,data = subset(employee))+ stat_boxplot(geom = "errorbar", width =
0.5)+
          geom_boxplot(outlier.colour="RED",
                       fill ="grey", outlier.shape = 18, outlier.size = 1, notch =
FALSE)+theme(legend.position = 'bottom')
        +
          labs(y = cnames[i],x = 'Absenteeism')+
          ggtitle(paste("Box plot of Absenteeism for", cnames[i]))))}
```

```
#library(gridExtra)
gridExtra::grid.arrange(gn1,gn2,gn3,ncol = 3)
gridExtra::grid.arrange(gn4,gn5,gn6,ncol = 3)
gridExtra::grid.arrange(gn7,gn8,gn9,ncol = 3)
gridExtra::grid.arrange(gn10,gn11,ncol = 2)


for(i in cnames)
{val = employee[,i][employee[,i]%in%
                    boxplot.stats(employee[,i])$out]
employee[,i][employee[,i]%in%val]=NA}
employee = knnImputation(employee, k = 3)

correlation=cor(employee[,numeric_index])
correlation
corrplot.mixed(correlation,tl.offset=0.01,tl.cex=0.01)
findCorrelation(correlation,cutoff = 0.6)
cnames




factor_index=sapply(employee,is.factor)
factor_data=employee[,factor_index]
for(i in 1:8)
{print(names(factor_data)[i])
  print(chisq.test(table(factor_data$absenteeism_time_in_hours,factor_data[,i])))}
colnames(employee)
new_data=employee[,-c(7,12,15,17,18,16)]
colnames(new_data)
new_data$absenteeism_time_in_hours=as.numeric(new_data$absenteeism_time_in_h
ours)
new_data$absenteeism_time_in_hours=ifelse(new_data$absenteeism_time_in_hours<
4,"less than 4","greater than 4")
new_data$absenteeism_time_in_hours=as.factor(new_data$absenteeism_time_in_hou
rs)
View(new_data)
index=sample(nrow(new_data),0.7*nrow(new_data))
train=new_data[index,]
test=new_data[-index,]
rf=randomForest(absenteeism_time_in_hours~.,data = train,ntree=500)
predictions=predict(rf,test)
confmatrix_rf=table(test$absenteeism_time_in_hours,predictions)
confusionMatrix(confmatrix_rf)
colnames(train)
```

```
install.packages("inTrees")
library(inTrees)
treelist=RF2List(rf)
exec=extractRules(treelist,train[,-14])
exec[1:2,]
readableRules=presentRules(exec,colnames(train))
readableRules[1:2,]
rulemetric=getRuleMetric(exec,train[,-14],train$absenteeism_time_in_hours)
rulemetric[1:2,]
```

Appendix 3 – Python Code

```
import os
import pandas as pd
import numpy as np
import matplotlib as mlt
import matplotlib.pyplot as plt
import seaborn as sn

os.chdir("G:/data science project")

employee=pd.read_excel("Absenteeism_at_work_Project.xls")

employee=employee[employee['Reason for absence']!=0]

employee=employee[employee['Absenteeism time in hours']!=0]

employee['Body mass index']=employee['Body mass index'].fillna(employee['Body mass index'].median())

employee['Absenteeism time in hours']=employee['Absenteeism time in hours'].fillna(employee['Absenteeism time in hours'].mean())

employee['Reason for absence']=employee['Reason for absence'].fillna(employee['Reason for absence'].median())

employee['Month of absence']=employee['Month of absence'].fillna(employee['Month of absence'].median())

employee['Education']=employee['Education'].fillna(employee['Education'].median())

employee['Disciplinary failure']=employee['Disciplinary failure'].fillna(employee['Disciplinary failure'].median())

employee['Social drinker']=employee['Social drinker'].fillna(employee['Social drinker'].median())


employee['Social smoker']=employee['Social smoker'].fillna(employee['Social smoker'].median())

employee['Pet']=employee['Pet'].fillna(employee['Pet'].mean())

employee['Weight']=employee['Weight'].fillna(employee['Weight'].mean())

employee['Height']=employee['Height'].fillna(employee['Height'].mean())

employee['Transportation expense']=employee['Transportation expense'].fillna(employee['Transportation expense'].mean())

employee['Distance from Residence to Work']=employee['Distance from Residence to Work'].fillna(employee['Distance from Residence to Work'].mean())

employee['Service time']=employee['Service time'].fillna(employee['Service time'].mean())
```

```
employee['Age']=employee['Age'].fillna(employee['Age'].mean())

employee['Son']=employee['Son'].fillna(employee['Son'].mean())

employee['Hit target']=employee['Hit target'].fillna(employee['Hit
target'].mean())

employee=employee.rename(columns={"Work load Average/day ":"Work load Average
per day"})

employee['Work load Average per day']=employee['Work load Average per
day'].fillna(employee['Work load Average per day'].mean())

missing_percent=((employee.isnull().sum()*100)/len(employee))

cnames=['ID','Transportation expense','Distance from Residence to
Work','Service time','Age','Work load Average per day',
        'Hit target','Son','Pet','Weight','Height','Body mass index']

f,ax=plt.subplots(figsize=(7,5))
corr=employee_corr.corr()
sn.heatmap(corr,mask=np.zeros_like(corr,dtype=np.bool),cmap=sn.diverging_pale
tte(220,10,as_cmap=True),square=True,ax=ax)

<matplotlib.axes._subplots.AxesSubplot at 0x3d173f0>

for i in cnames:
    q75,q25=np.percentile(employee.loc[:,i],[75,25])
    iqr=q75-q25
    min=q25-(iqr*1.5)
    max=q75+(iqr*1.5)
    employee=employee.drop(employee[employee.loc[:,i]<min].index)
    employee=employee.drop(employee[employee.loc[:,i]>max].index)

employee['Absenteeism time in hours']=['Less than or equal to 4' if val
in(range(1,4)) else 'Greater than 4'
                                        for val in employee['Absenteeism time
in hours']]

employee=employee.drop(['Disciplinary failure','ID','Height'],axis=1)

employee=employee.drop(['Service time','Weight'],axis=1)

employee=employee.drop(['Education'],axis=1)

from sklearn.cross_validation import train_test_split
from sklearn.ensemble import RandomForestClassifier

x=employee.values[:,0:14]
y=employee.values[:,14]

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```python
model=RandomForestClassifier(n_estimators=500).fit(x_train,y_train)

predictions=model.predict(x_test)

from sklearn.metrics import confusion_matrix

cm=pd.crosstab(y_test,predictions)


TN=cm.iloc[0,0]
FN=cm.iloc[1,0]
TP=cm.iloc[1,1]
FP=cm.iloc[0,1]
 ((TP+TN)*100)/(TP+TN+FP+FN)

 (FN*100)/(FN+TP)
```