EMPLOYEE ABSENTEEISM

Sayantan Adak

17 JANUARY 2019

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

There are 21 variables in our data in which 20 are independent variables and 1 (Absenteeism time in hours) is dependent variable. Since our target variable is continuous in nature, this is a regression problem.

Variables 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 behavioral 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) patient follow-up (22), medical consultation (23), blood donation (24), laboratory examination (25), unjustified absence (26), physiotherapy (27), dental consultation (28).

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

11. Hit target

12. Disciplinary failure (yes=1; no=0)

13. Education (high school (1), graduate (2), postgraduate (3), master and doctor (4))

14. Son (number of children)

15. Social drinker (yes=1; no=0)

16. Social smoker (yes=1; no=0)

17. Pet (number of pet)

18. Weight

19. Height

20. Body mass index

21. Absenteeism time in hours (target)

1.3 Data Exploration

Exploratory Data Analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics. In the given data set there are 21 variables and data types of all variables are either float64 or int64. There are 740 observations and 21 columns in our data set. Missing value is also present in our data.

Structure of the data:

```
ID                              : num  11 36 3 7 11 3 10 20 14 1 ...
Reason.for.absence              : num  26 0 23 7 23 23 22 23 19 22 ...
Month.of.absence                : num  7 7 7 7 7 7 7 7 7 7 ...
Day.of.the.week                 : num  3 3 4 5 5 6 6 6 2 2 ...
Seasons                         : num  1 1 1 1 1 1 1 1 1 1 ...
Transportation.expense          : num  289 118 179 279 289 179 NA 260 155
235 ...
Distance.from.Residence.to.Work: num  36 13 51 5 36 51 52 50 12 11 ...
Service.time                    : num  13 18 18 14 13 18 3 11 14 14 ...
Age                             : num  33 50 38 39 33 38 28 36 34 37 ...
Work.load.Average.day.          : num  239554 239554 239554 239554 239554
...
Hit.target                      : num  97 97 97 97 97 97 97 97 97 97 ...
Disciplinary.failure            : num  0 1 0 0 0 0 0 0 0 0 ...
Education                       : num  1 1 1 1 1 1 1 1 1 3 ...
Son                             : num  2 1 0 2 2 0 1 4 2 1 ...
Social.drinker                  : num  1 1 1 1 1 1 1 1 1 0 ...
Social.smoker                   : num  0 0 0 1 0 0 0 0 0 0 ...
Pet                             : num  1 0 0 0 1 0 4 0 0 1 ...
Weight                          : num  90 98 89 68 90 89 80 65 95 88 ...
Height                          : num  172 178 170 168 172 170 172 168 196
172 ...
Body.mass.index                 : num  30 31 31 24 30 31 27 23 25 29 ...
Absenteeism.time.in.hours       : num  4 0 2 4 2 NA 8 4 40 8 ...
```

List of columns and their number of unique values -

```
ID                                 36
Reason for absence                 28
Month of absence                   13
Day of the week                     5
Seasons                             4
Transportation expense             24
Distance from Residence to Work    25
Service time                       18
Age                                22
Work load Average/day              38
Hit target                         13
Disciplinary failure                2
```

```
Education                          5
Son                                5
Social drinker                     2
Social smoker                      2
Pet                                6
Weight                            26
Height                            14
Body mass index                   17
Absenteeism time in hours         19
```

Chapter 2

Methodology

2.1 Pre Processing

Data pre-processing is a crucial stage for any predictive model. The quality of the input decides the quality of the output.

Train Data: The predictive model is always built on train data set. An intuitive way to identify the train data is, that it always has the ‘response variable’ included.

Test Data: Once the model is built, it’s accuracy is ‘tested’ on test data. This data always contains less number of observations than train data set. Also, it does not include ‘response variable’.

Need for Data Cleaning or Data Preparation:

- Dataset might contain discrepancies in the names or codes.
- Dataset might contain outliers or errors.
- Dataset lacks your attributes of interest for analysis.
- All in all the dataset is not qualitative but is just quantitative.

From the experiences of many data scientists it is said that the data exploration, cleaning and preparation can take upto 70% time of the total project.

2.1.1 Distribution of variables

From the data exploration step it is seen that all the variables are numerical in nature in the raw data-set. So to understand the data distribution visually, we have plotted the histogram of each variable.

Few inference can be drawn looking at the histogram:

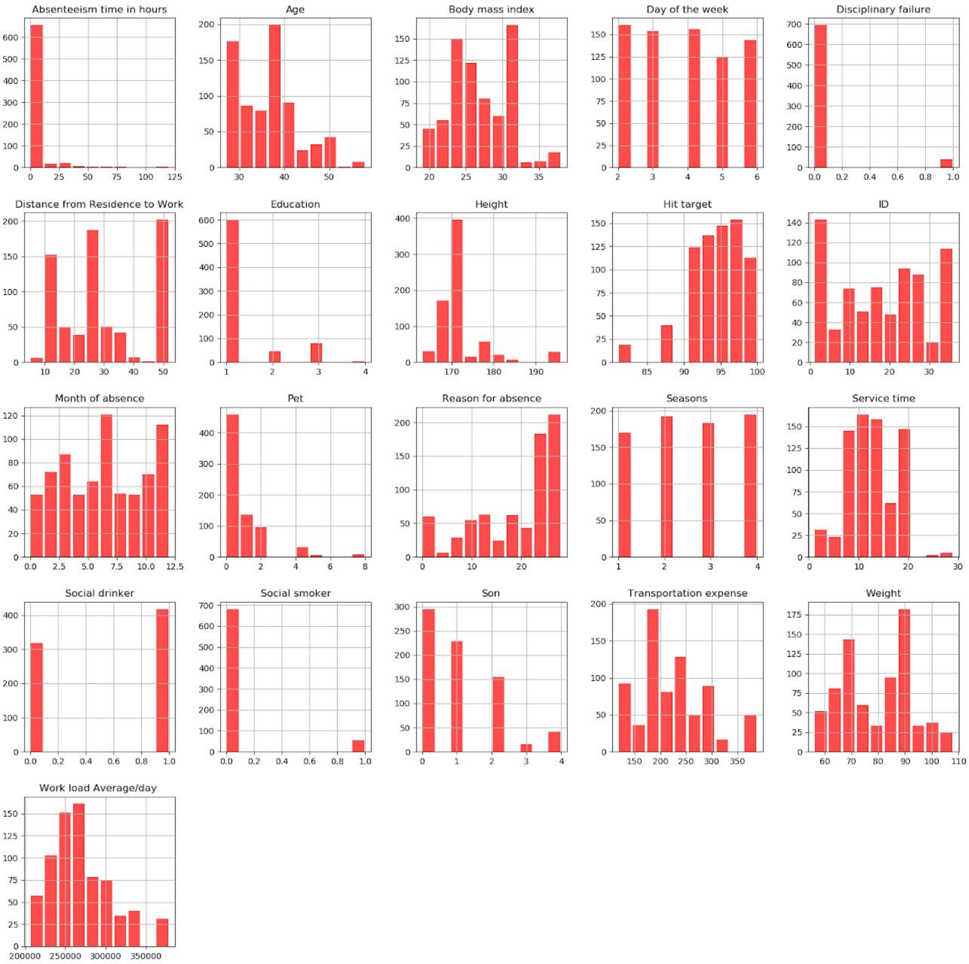


Fig. Distribution of variables

Few inference can be drawn looking at the histogram:

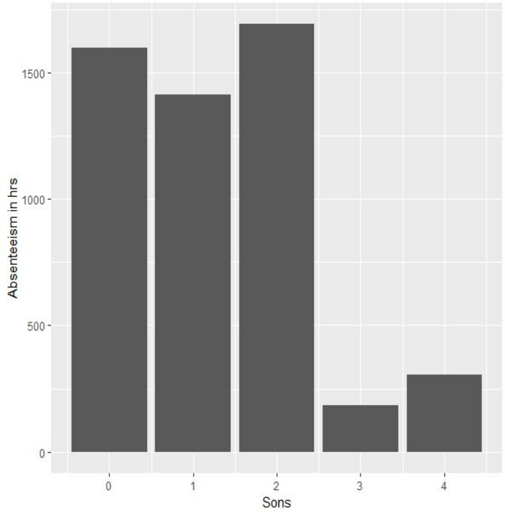
- **Season** has four categories of almost equal distribution.
- Most of the observations in the **Education** are from 1(high school) level.
- **Disciplinary failure, Social smoker or drinker** has 2 categories as mentioned in the data given.
- **Transportation expense, Service time, Weight** look naturally distributed.

As the **No. of Pet , No. of Son** have too few discrete level we can treat them as categorical variable for the sake of simplicity.

2.1.2 Multivariate analysis

So far we have fair understanding of the distribution of the data. Now we will plot Absenteeism time in hours on basis of each category.

If we look closely at the distribution of **Pet** and **Son,** we can observe the data is highly left skewed.



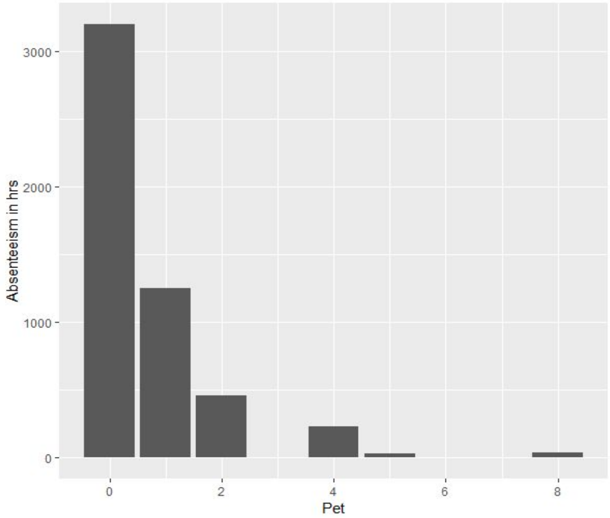
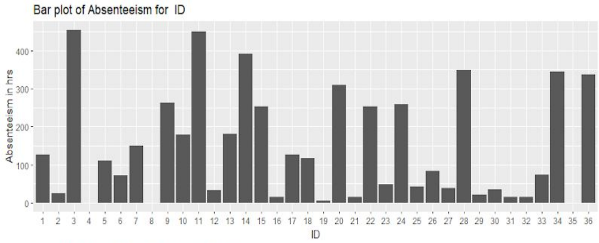


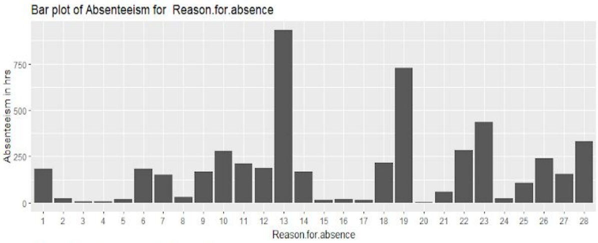
Fig. Distribution of Son and Pet

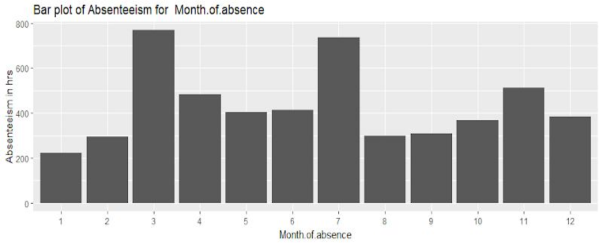
For the sake of simplicity we have categorized the **Son** of two categories: category 1 if number of sons >= 2 else category 0

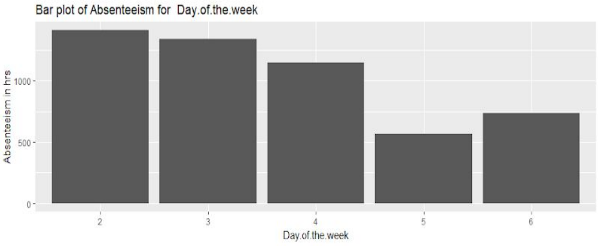
The pet is categorized of two different categories: “without pet” if number of pet is 0 else “With pet”.

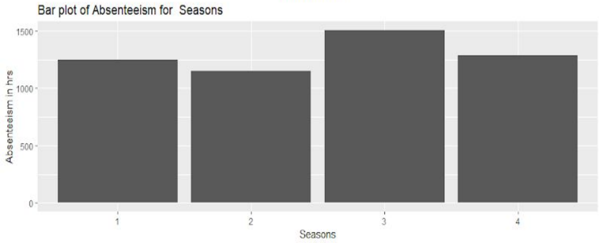
Now take a look at the bar plot of **Absenteeism time** against each categorical variables.

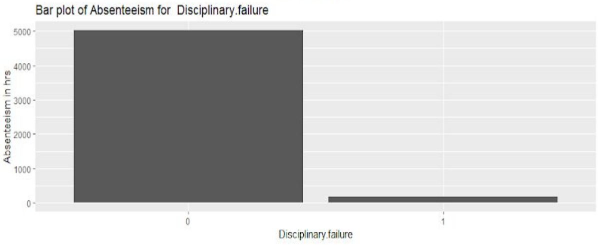


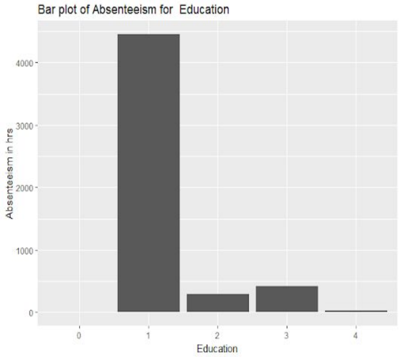


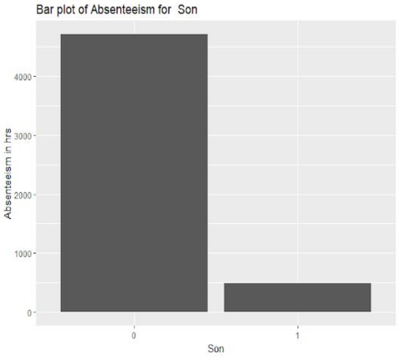


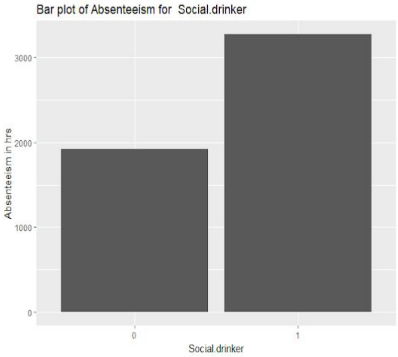


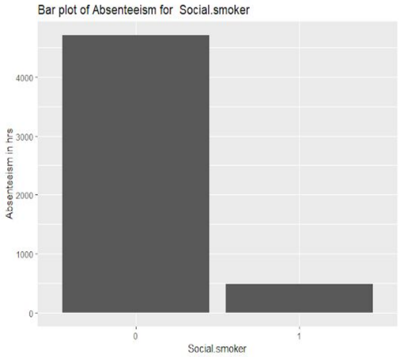


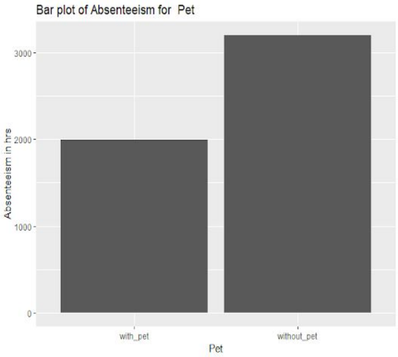












Few inferences can be drawn from the plot:

- ID no 3 and 11 had high amount of absenteeism time.
- Reason no. 13(Diseases of the musculoskeletal system and connective tissue) and 19(Injury, poisoning and certain other consequences of external causes). Company should take care of it.
- Employee Absenteeism is mostly observed in March(3) and July(7) month.
- Employee absenteeism is high on Monday(2)
- Disciplinary failure heavily impacted on absenteeism.
- People with lower education tends to absent more.
- People who has more than 2 children absents less.
- People without pet tends to absent more.

2.1.3 Missing Value Analysis

The data consists of 135 missing values. Take a look at variable wise missing value:

```
ID                                                              0
Reason.for.absence                                              3
Month.of.absence                                                1
Day.of.the.week                                                 0
Seasons                                                         0
Transportation.expense                                          7
Distance.from.Residence.to.Work                                 3
Service.time                                                    3
Age                                                             3
Work.load.Average.day.                                         10
Hit.target                                                      6
Disciplinary.failure                                            6
Education                                                      10
Son                                                             6
Social.drinker                                                  3
Social.smoker                                                   4
Pet                                                             2
Weight                                                          1
Height                                                         14
Body.mass.index                                                31
Absenteeism.time.in.hours                                      22
```

Last three observations in **Reason for absence** and **Month of absence** are 0 which does not make any sense. So we removed the last three observations in **Reason for absence** and replace with missing values and for **Month** variable we can see the data is a time series data, so the last 0 will be either equal to the previous value or greater than it. So we have replaced 0 with 7 in **Month** variable.

```
ID        Reason.for.absence   Month.of.absence
11                14                 7
 1                11                 7
 4                 0                 0
 8                 0                 0
35                 0                 0
```

Now we have imputed missing values with **knnImputation** method.

As all the values in the raw data were integer in values, we have converted all the variables to integer after **knnImputation** for simplicity.

2.1.4 Conversion of datatype

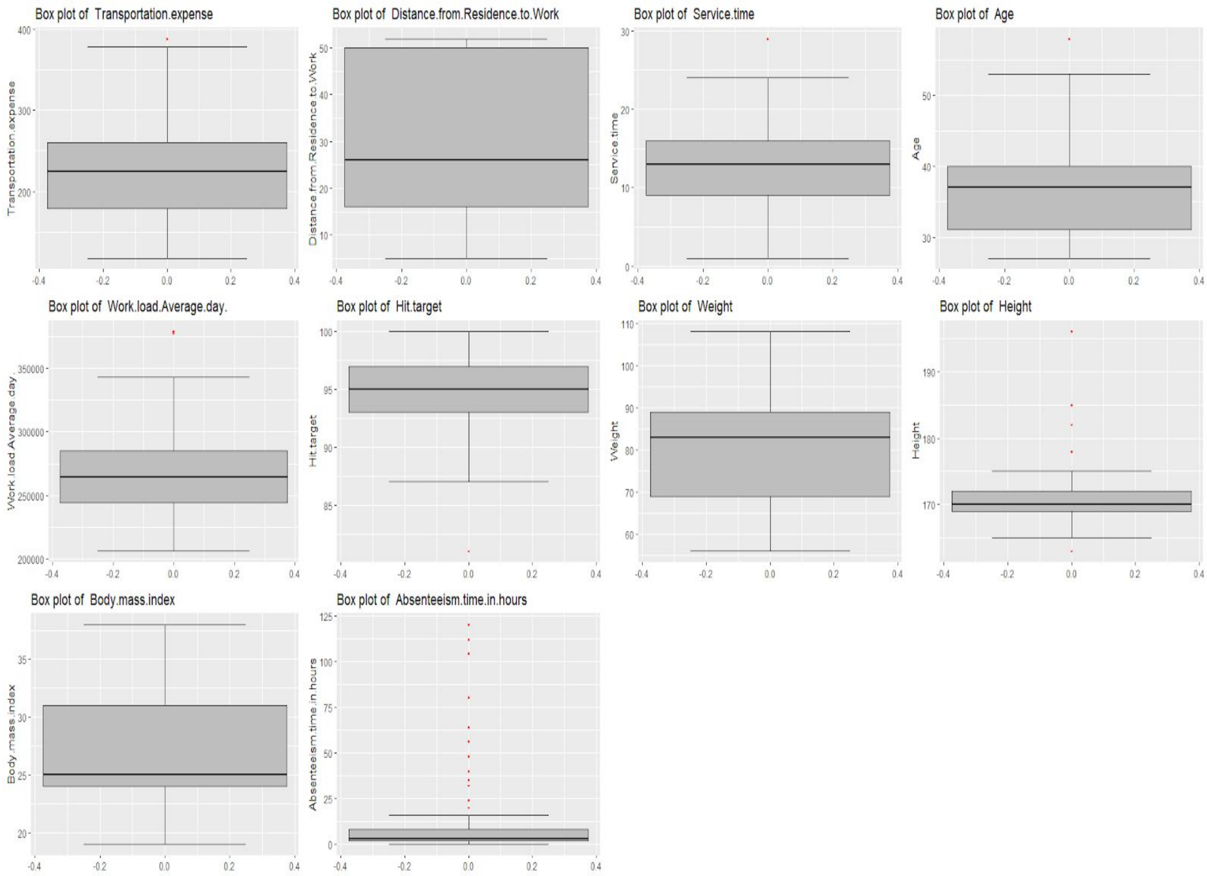
After **knnImputation** we have changed the data-type of required variables to categorical. The variables that are identified as a category are:

Reason of absence, Month, Day of the week, Seasons, Disciplinary failure, Education, Son, Pet, Social Drinker, Social smoker

2.1.5 Outlier analysis

We have observed outliers in few variables. If we don’t treat them properly then it will affect our overall predictions.

We have used boxplot method to visualize the outliers and will treat them as missing values.

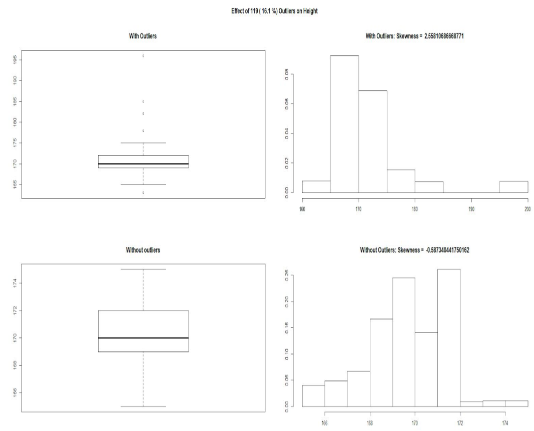


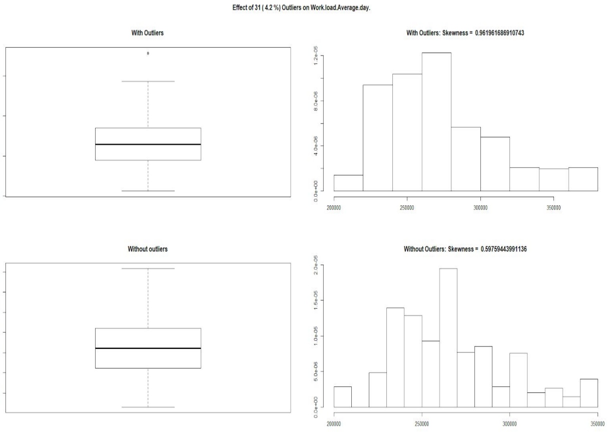
The variables **Transportation expense, Service time, Age, Work load, Hit target, Height** and **Absenteeism in hours** consists of outliers.

There seems to be many outlier in our target variable **Absenteeism_time_in_hours**. There are value of 120, 100, 80, 60 which is not possible.

Reason: The data set is a daily data set with no of absent hour per day. A day has max 24 hours, so all these values seems redundant and we need to eliminate these out. Logically the absenteeism hours should be less than the service time of that employee.

Now have a look at some of the distribution of variables with and without outliers.





We can see the skewness in the distribution has been reduced after removing outliers.

2.1.6 Feature Scaling

In the raw data there are different feature of different scale in their magnitude. If we feed them directly into the machine learning model without scaling, object function will not work out properly and the result will be biased. That’s why we have gone for feature scaling as a data pre-processing technique to reduce variation either within or between variables. We have normalized all the numeric predictors to a fixed range using the formula:

Valuenew = (Value - minValue)/(maxValue - minValue)

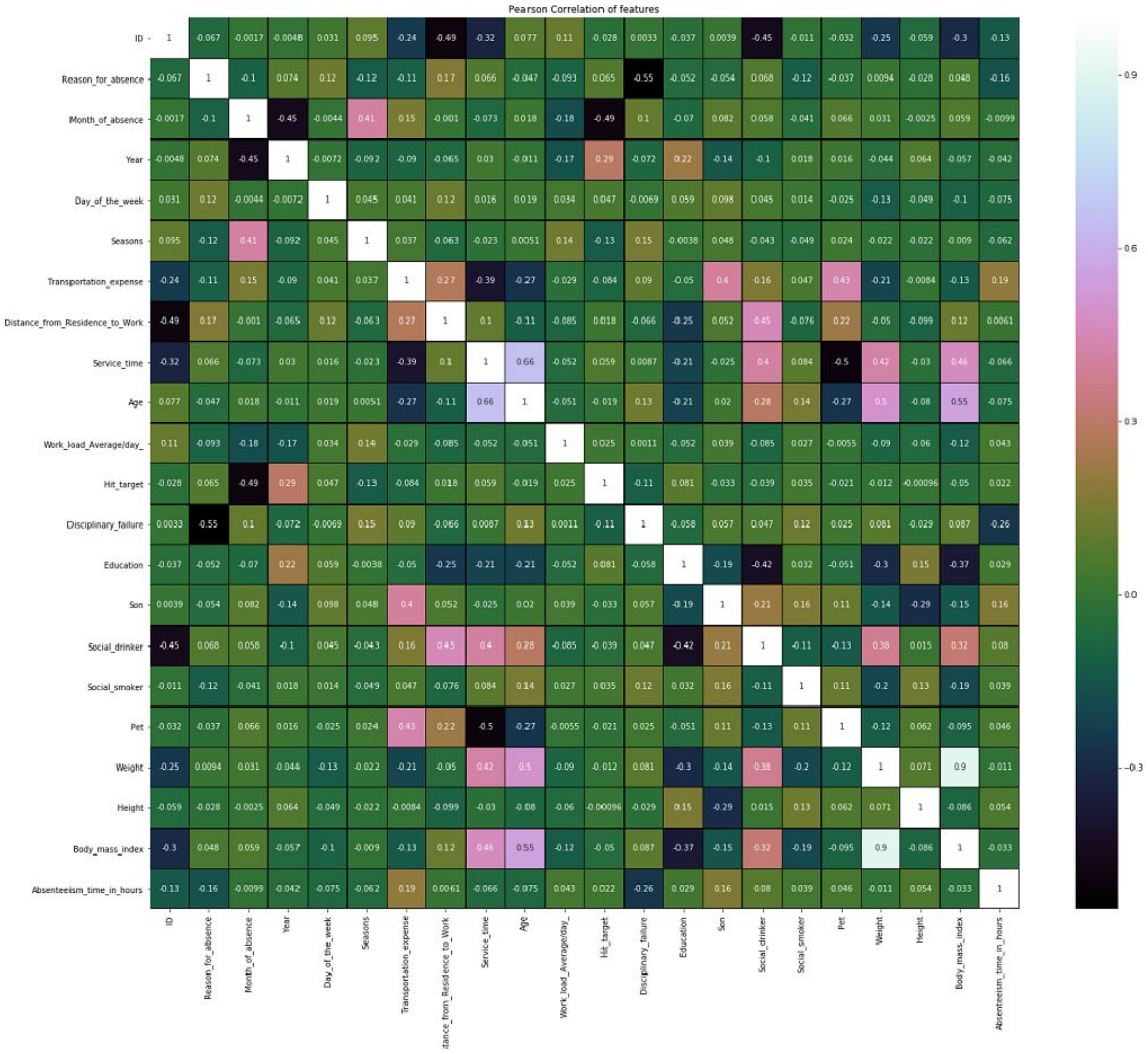
This changes the range of each numerical predictor variables to 0 to 1.

2.2 Feature Engineering

For reducing complexity of the model and getting better prediction we may need to exclude few variables that are not important for prediction.

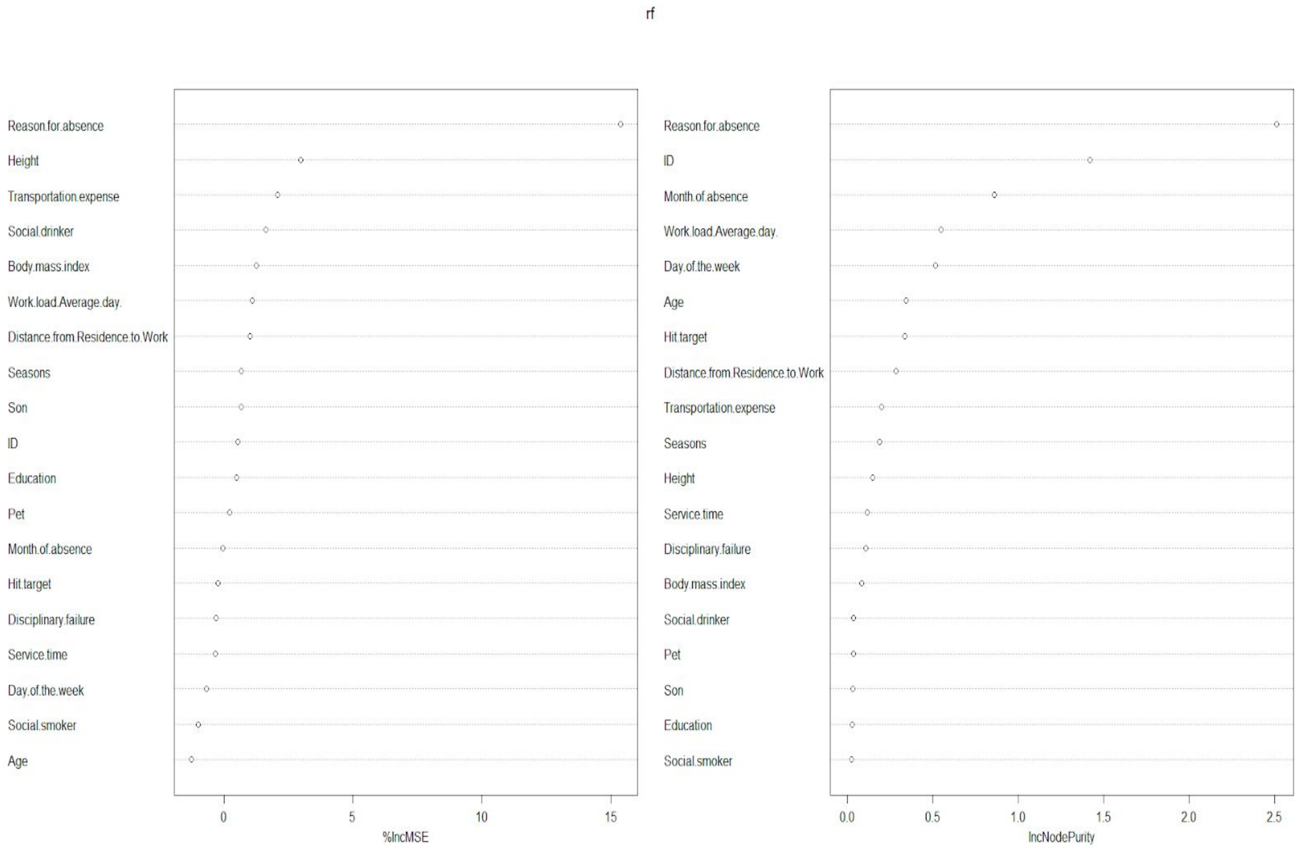
We have used **Correlation plot** and **RandomForest** machine learning technique for selecting important features for prediction.

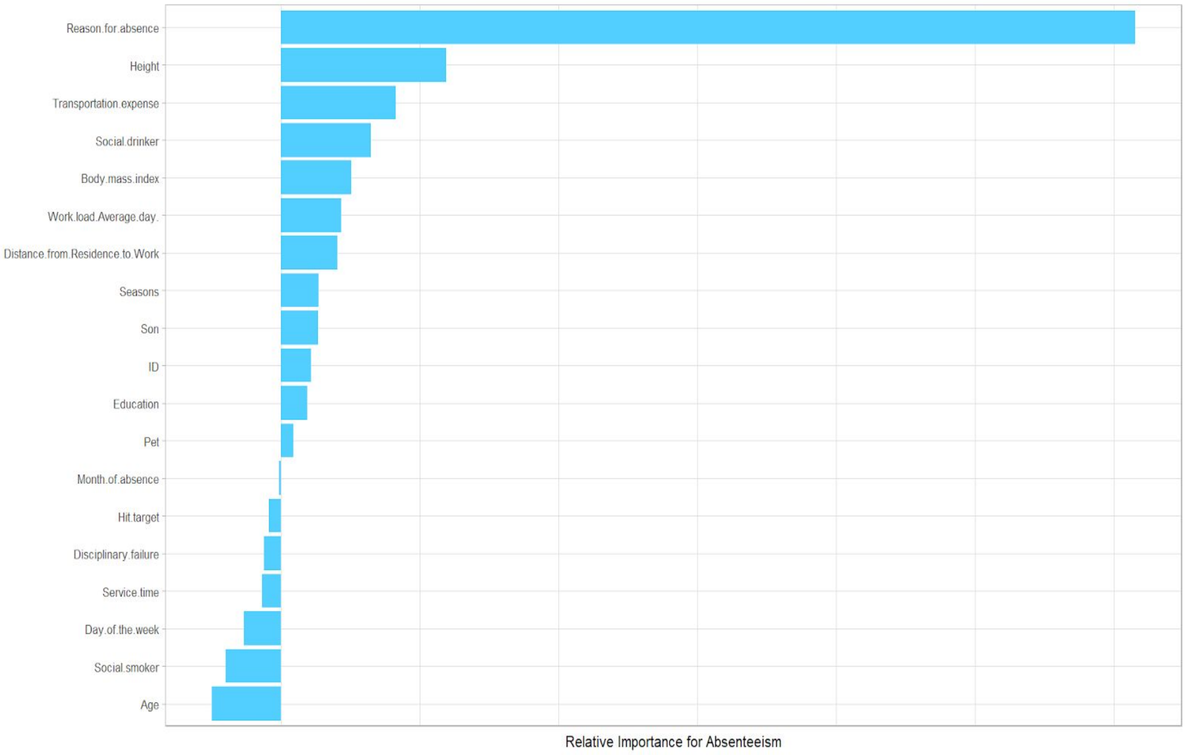
2.2.1 Correlation Plot



We can see Weight and Body mass index have high correlation coefficient(0.9) we can exclude either of the variable for further analysis.

2.2.2 RandomForest Method





As we can see **Reason for absence** plays the most important role in calculating the time of Absenteeism.

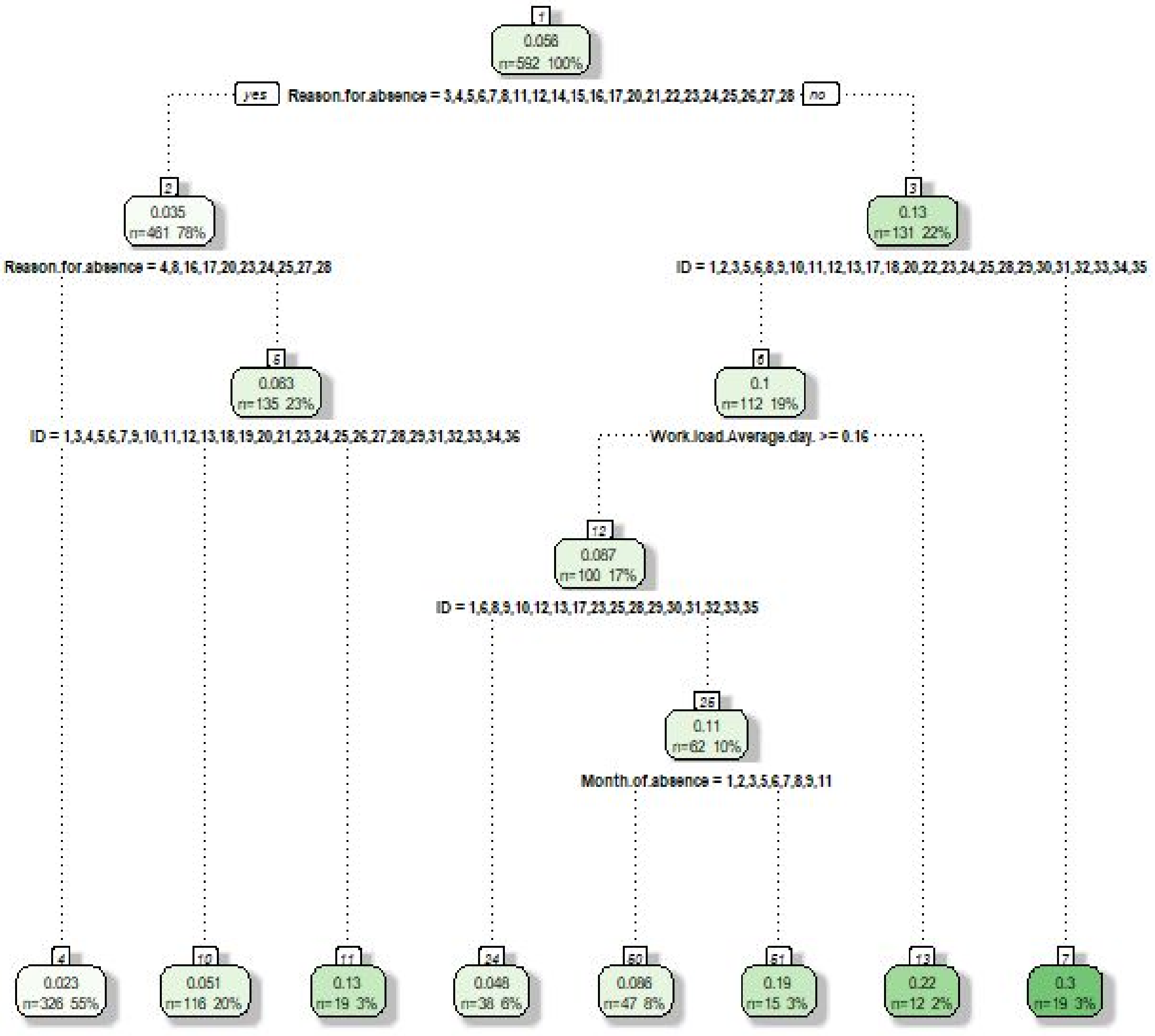
2.3 Model building

2.3.1 Model selection

Our target is to forecast(predict) the absenteeism time with respect to some other features . As the target variable is numeric we can use statistical method like **linear regression** or **Decision Tree Regression** model for prediction. We have used **Decision Tree Regression** for our prediction.

2.3.2 Regression Tree

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. A decision node has two or more branches each representing values for the attribute tested. Leaf node represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. **Decision Tree Regression** is used to handle numerical response variable.


1
0.056
n=592 100%
yes
Reason.for.absence = 3,4,5,6,7,8,11,12,14,15,16,17,20,21,22,23,24,25,26,27,28
no
2
0.035
n=461 78%
Reason.for.absence = 4,8,16,17,20,23,24,25,27,28
3
0.13
n=131 22%
ID = 1,2,3,5,6,8,9,10,11,12,13,17,18,20,22,23,24,25,28,29,30,31,32,33,34,35
5
0.063
n=135 23%
ID = 1,3,4,5,6,7,9,10,11,12,13,18,19,20,21,23,24,25,26,27,28,29,31,32,33,34,36
6
0.1
n=112 19%
Work.load.Average.day. >= 0.16
12
0.087
n=100 17%
ID = 1,6,8,9,10,12,13,17,23,25,28,29,30,31,32,33,35
25
0.11
n=62 10%
Month.of.absence = 1,2,3,5,6,7,8,9,11
4
0.023
n=326 55%
10
0.051
n=116 20%
11
0.13
n=19 3%
24
0.048
n=38 6%
50
0.086
n=47 8%
51
0.19
n=15 3%
13
0.22
n=12 2%
7
0.3
n=19 3%

Rattle 2019-Jan-17 20:44:23 User

2.3.3 Time Series Approach

What is time series modelling?

One such method, which deals with time based data is **Time Series Modeling**. As the name suggests, it involves working on time (years, days, hours, minutes) based data, to derive hidden insights to make informed decision making.

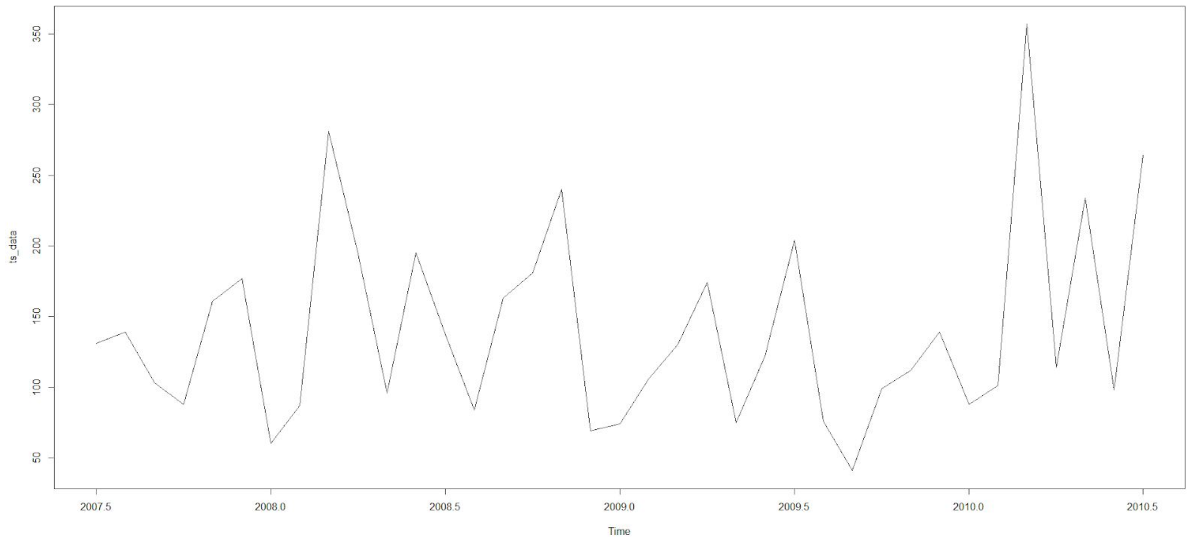
Time series models are very useful models when you have serially correlated data. Most of business houses work on time series data to analyze sales number for the next year, website traffic, competition position and much more. However, it is also one of the areas, which many analysts do not understand.

So, if you aren't sure about complete process of time series modelling, this guide would introduce you to various levels of time series modelling and its related techniques.

In this problem the data is measured periodically every month. So this problem can be classified as time series problem.

As it's 3 years of data from July 2007 to July 2010(37 months) and we don't have a 'Year' variable in the dataset,

including that variable will help in our seasonal analysis.



Checking Stationarity:

```
       Augmented Dickey-Fuller Test

data:  ts_data
Dickey-Fuller = -4.4573, Lag order = 1, p-value = 0.01
alternative hypothesis: stationary
```

Checking Auto-correlation:

```
       Durbin-Watson test

data:  ts_data[-37] ~ ts_data[-1]
DW = 1.9688, p-value = 0.4662
alternative hypothesis: true autocorrelation is greater than 0
```

ARIMA Model:

ARIMA model there are 3 parameters that are used to help model the major aspects of a times series: seasonality, trend, and noise. These parameters are labeled **p, d,** and **q.**

AR: Autoregressive part: Summation of lags, p

I: Integration, degree of differencing: d

MA: Moving Average: Summation of forecasting errors, q

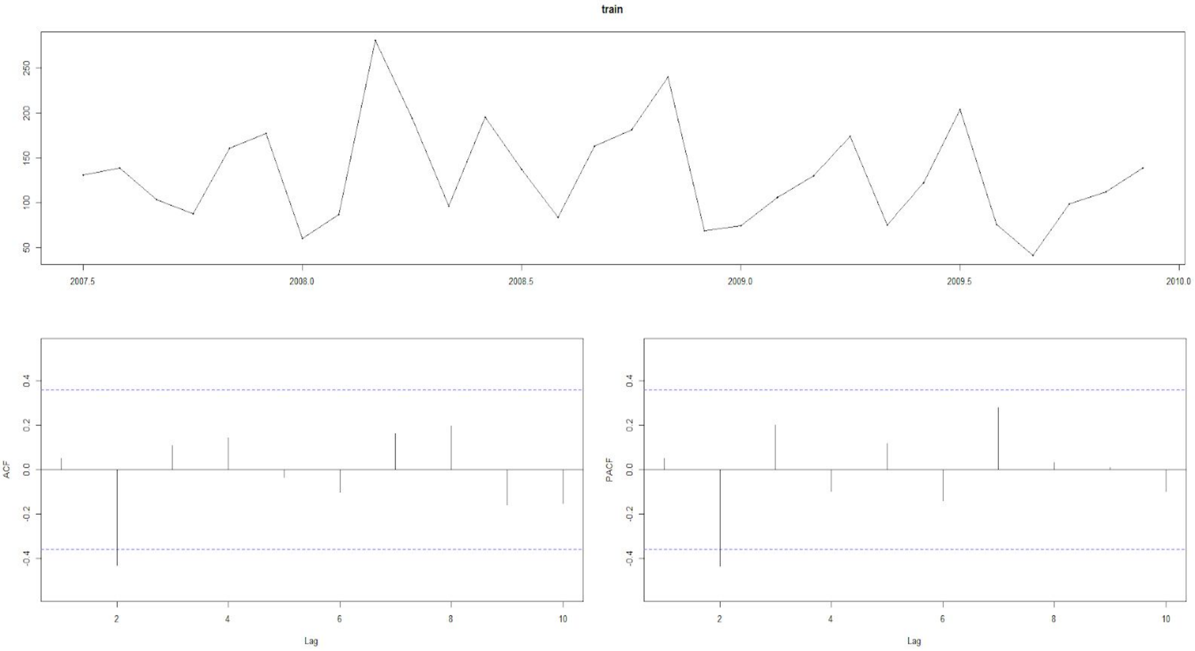
ACF And PACF Plot :

ACF plot tells about q: Moving average part

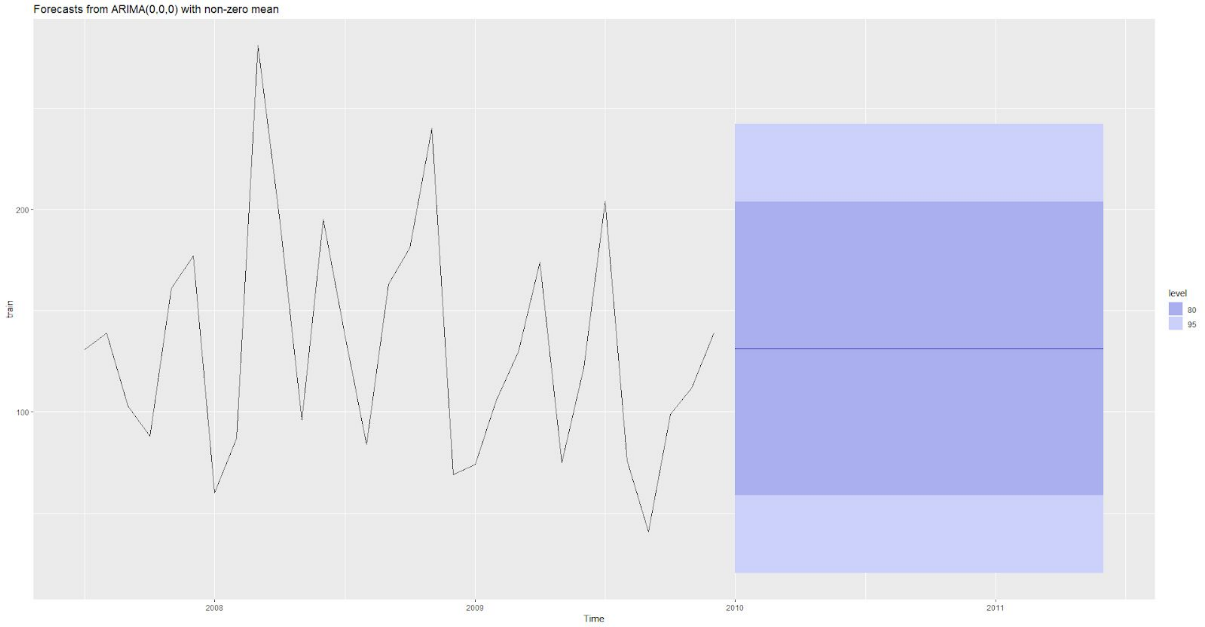
PACF plot tells about p; auto regression part

```
 ARIMA(2,0,2)            with non-zero mean : 333.3774
 ARIMA(0,0,0)            with non-zero mean : 330.696
 ARIMA(1,0,0)            with non-zero mean : 333.0944
 ARIMA(0,0,1)            with non-zero mean : 332.2914
 ARIMA(0,0,0)            with zero mean     : 384.8671
 ARIMA(1,0,0)            with non-zero mean : 333.0944
 ARIMA(0,0,1)            with non-zero mean : 332.2914
 ARIMA(1,0,1)            with non-zero mean : 330.848

 Best model: ARIMA(0,0,0)            with non-zero mean
```



Forecasting:



Conclusion

3.1 Model Evaluation

Now we have few models for predicting target variable, we need to decide which model to choose. The performance of any classification model does not only depend upon its prediction accuracy. Depending upon business understanding we need to consider different kinds of metrics to evaluate our models. The choice of metric completely depends on the type of model and the implementation plan of the model.

There are different types of metrics depend on what kind of problem you are trying to solve:

For Regression Problem :

a. RMSE
b. MSE
c. MAPE

Regression error metrics: We want to know how well the model predicts new data, not how well it fits the data it was trained with. Key component of most regression measures are difference between actual y and predicted y ("error")

- **MSE:** The mean squared error (MSE) or mean squared deviation (MSD) of an estimator measures the average of the squares of the errors or deviations that is, the difference between the estimator and what is estimated.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y_i} - Y_i)^2$$

- **RMSE:** It stands for Root Mean Squared Error/Deviation. Square the errors, find their average, take the square root

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n}}.$$

- **MAE:** The mean absolute error (MAE) is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. The mean absolute error is given by

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i|$$

- **MAPE:** Stands for Mean absolute percentage error. Measures accuracy as a percentage Of error.

$$\mathrm{M} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|,$$

For regression tree(predicting Absenteeism time) :

```
     mae      mape     mase
   0.056     67.51     0.76
```

For ARIMA model :

```
                  mae      mape      mase
Training set    45.25     42.92      0.69
```

```
 Test set         83.61   40.80     1.28
```

Appendix A: Extra Figure

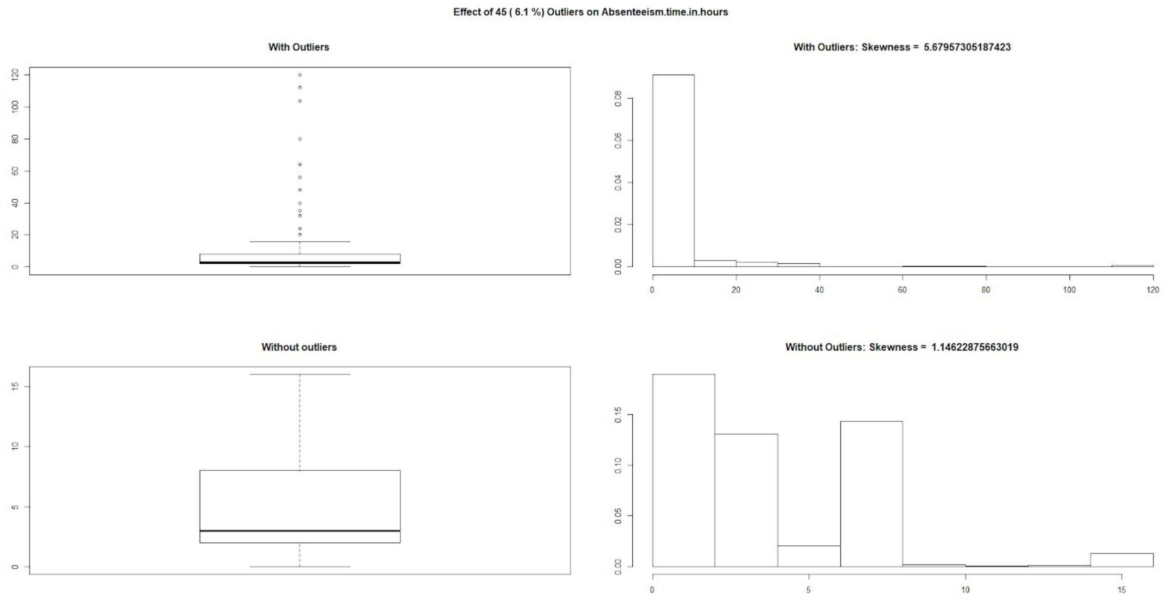


Fig: Outlier effect on Absenteeism in Hours

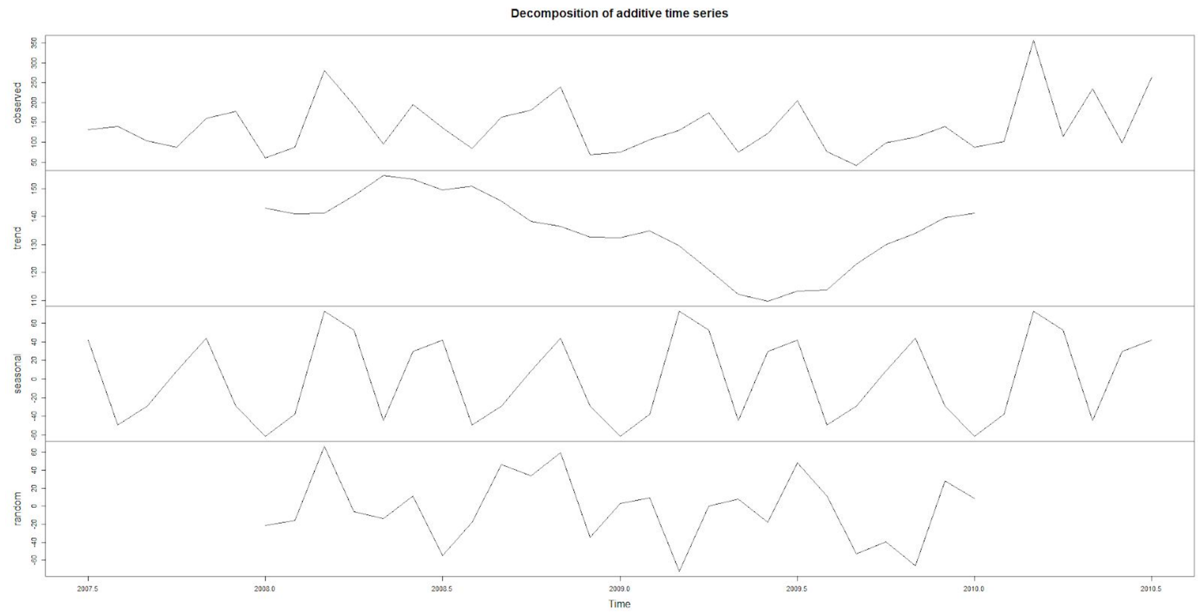


Fig: decomposion of additive time series

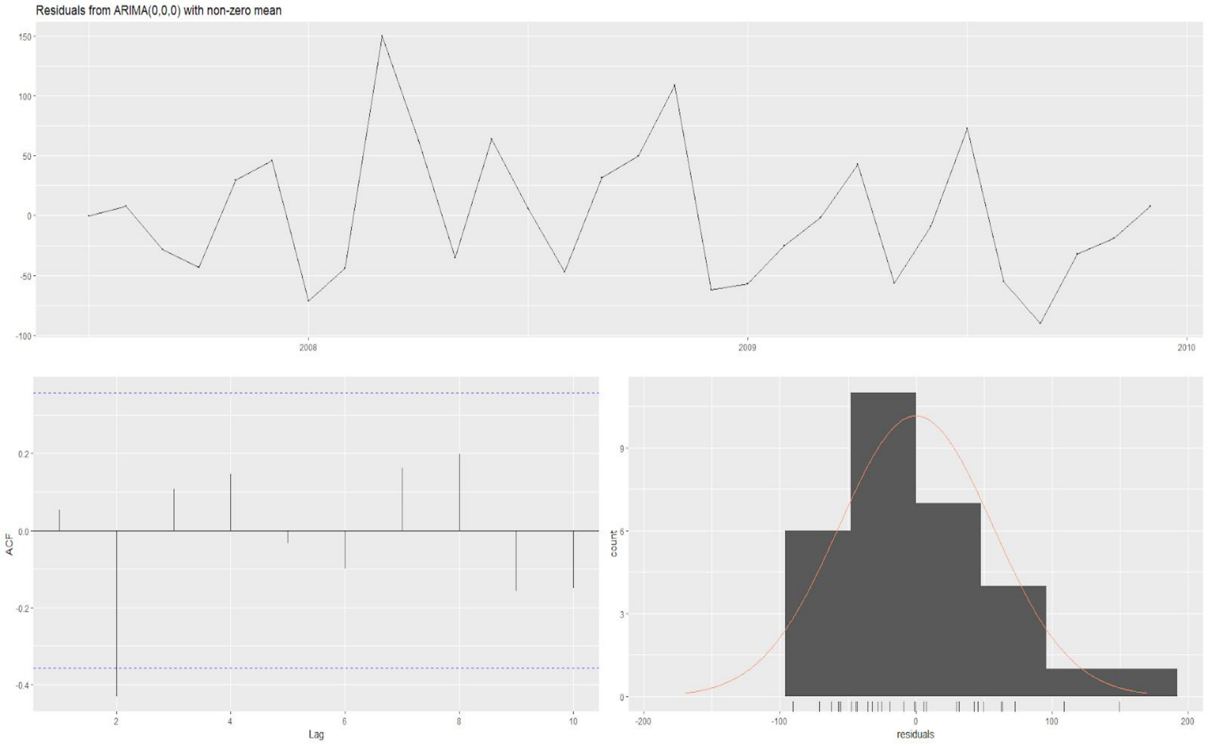


Fig: Residuals for ARIMA model

Appendix B: R code

Complete R code:

```
library(xlsx)
library(naniar)
library(ggplot2)
library(e1071)
library(DMwR)
library(randomForest)
library(dplyr)
library(rpart)
library(MASS)
library(rattle)
library(rpart.plot)
library(RColorBrewer)
library(forecast)
library(tseries)
library(lmtest)
library(plyr)


raw_data <- read.xlsx('Absenteeism_at_work_Project.xls', sheetIndex = 1)

missing_val = data.frame(apply(raw_data,2,function(x){sum(is.na(x))}))
missing_val

data <- raw_data %>% replace_with_na_at(.vars = c("Reason.for.absence","Education"), condition =
~.x == 0)
data$Month.of.absence[data$Month.of.absence<=0] <- 7

data_without_na <- knnImputation(data)
data_without_na <- data.frame(lapply(data_without_na, function(x) as.integer(x)))
```

```
data_without_na$Son <- factor(ifelse(data_without_na$Son > 2, "1", "0"))

data_without_na$Pet <- factor(ifelse(data_without_na$Pet > 0, "with_pet", "without_pet"))


categorical_index = c(1, 2, 3, 4, 5, 12, 13, 14, 15, 16, 17)

categorical_data = data_without_na[,categorical_index]

categorical_cols <- colnames(categorical_data)

data_without_na[,categorical_index] <- lapply(data_without_na[,categorical_index], function(x) as.factor(x))


#######Bar plot

for (i in 1:length(categorical_cols)){

    assign(paste0("bar",i), ggplot(aes_string(y = data_without_na$Absenteeism.time.in.hours, x = data_without_na[,categorical_index[i]]),

                              data = data_without_na)+

         geom_bar(stat = "identity")+

         labs(y = 'Absenteeism in hrs', x = colnames(categorical_data[i]))+

         ggtitle(paste("Bar plot of Absenteeism for ",colnames(categorical_data[i]))))

}

gridExtra::grid.arrange(bar1,bar2,bar3,bar4,bar5,bar6, ncol=2)

gridExtra::grid.arrange(bar7,bar8,bar9,bar10,bar11,ncol=3)

#######


#########Boxplot with outliers

numeric_index <- sapply(data_without_na,is.numeric)

numeric_data <- data_without_na[,numeric_index]

cnames <- colnames(numeric_data)

for (i in 1:length(cnames)){

  assign(paste0("gn",i), ggplot(aes_string(y = (cnames[i])), data = numeric_data)+

         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])+
        ggtitle(paste("Box plot of ",cnames[i])))
}
gridExtra::grid.arrange(gn1,gn2,gn3,gn4,gn5,gn6,gn7,gn8,gn9,gn10,ncol=4)
##########




#### Outlier analysis

outToNa <- function(vl, df) {
  for (i in vl) {
    outlier <- boxplot.stats(df[, i])$out
    df[, i] <- ifelse(df[, i] %in% outlier,
                      NA, df[, i])
  }
  return(df)
}
library(e1071)
outliereff <- function(i, df) {
  total = length(df[, i])
  par(mfrow = c(2, 2), oma = c(0, 0, 3,
                               0))
  boxplot(df[, i], main = "With Outliers")
  hist(df[, i], main = paste("With Outliers: Skewness = ",skewness(df[,i])),
       xlab = NA, ylab = NA, prob = TRUE)
  #skewness(df[,i],)
  df <- outToNa(i, df)
  out <- sum(is.na(df[, i]))
  per <- round((out)/total * 100, 1)
  df <- knnImputation(df)
  boxplot(df[, i], main ="Without outliers")
  hist(df[, i], main = paste("Without Outliers: Skewness = ",skewness(df[,i])),
       xlab = NA, ylab = NA, prob = TRUE)

  title(paste("Effect of", out, "(", per,
             "%)", "Outliers on", colnames(df)[i],
             sep = " "), outer = TRUE)
}

var_list <- list(10,19,21)
for(vl in var_list){
  outliereff(vl,data_without_na)
}
################

correlation_table <- data.frame(cor(numeric_data))



new_data <- subset(data_without_na,select=-c(Weight))

#### normalization
normalized_data <- new_data
numeric_index <- sapply(normalized_data,is.numeric)
numeric_data <- normalized_data[,numeric_index]
cnames <- colnames(numeric_data)
for (i in cnames){
  normalized_data[,i] = (normalized_data[,i] - min(normalized_data[,i]))/
    (max(normalized_data[,i])-min(normalized_data[,i]))
```

```
}


### Feature Engineering

library(randomForest)

feature <- subset(normalized_data,select=-c(Absenteeism.time.in.hours))

rf <- randomForest(feature, normalized_data$Absenteeism.time.in.hours, nTree =100, importance =
T)

imp <- varImpPlot(rf)

featureImportance <- data.frame(Feature=row.names(imp), Importance=imp[,1])


ggplot(featureImportance, aes(x=reorder(Feature, Importance), y=Importance)) +

  geom_bar(stat="identity", fill="#53cfff") +

  coord_flip() +

  theme_light(base_size=16) +

  xlab("") +

  ylab("Relative Importance for Absenteeism") +

  theme(plot.title   = element_text(size=18),

        strip.text.x = element_blank(),

        axis.text.x  = element_blank(),

        axis.ticks.x = element_blank())



########### Decision Tree Regression

ML_data <- normalized_data

train_index <- sample(1:nrow(ML_data), 0.8 * nrow(ML_data))

train <- ML_data[train_index,]

test <- ML_data[-train_index,]

fit <- rpart(Absenteeism.time.in.hours ~ ., data=train, method = "anova")

windows()

fancyRpartPlot(fit)

prediction_DT <- predict(fit, test[,-20])
```

```
########## Error Metric

mape <- function(y,yhat){

  mean(abs((y-yhat)/y)) *100

}

calculateMASE <- function(f,y) { # f = vector with forecasts, y = vector with actuals

  if(length(f)!=length(y)){ stop("Vector length is not equal") }

  n <- length(f)

  return(mean(abs((y - f) / ((1/(n-1)) * sum(abs(y[2:n]-y[1:n-1]))))))

}

mape(prediction_DT,test[,20])

calculateMASE(prediction_DT, test[,20])

mae <- mean(abs(prediction_DT - test[,20]))


############ Time Series Approach



data_without_na[1:113, 'Year'] = 2007

data_without_na[114:358, 'Year'] = 2008

data_without_na[359:570, 'Year'] = 2009

data_without_na[571:740, 'Year'] = 2010

data_without_na = data_without_na[,c(1,2,3,22,4:21)]

total_absent =  ddply(data_without_na, c("Year","Month.of.absence"),

                                   function(x) colSums(x[c("Absenteeism.time.in.hours")]))

ts_data = ts(total_absent$Absenteeism.time.in.hours,

             frequency = 12,start = c(2007, 7))




plot(ts_data)
```

```
train = window(ts_data, start = c(2007,7), end = c(2009,12))
test = window(ts_data, start = c(2010))



plot(decompose(ts_data))



adf.test(ts_data, k = 1)


#Test for autocorrelation
dwtest(ts_data[-37] ~ ts_data[-1])



tsdisplay(train)


#ARIMA model building
arima_fit = auto.arima(train, trace = T)


#Forecasting
arimafore = forecast(arima_fit, h =18)
autoplot(arimafore)


#Accuracy
accuracy(arimafore, test)


#Residuals
checkresiduals(arima_fit)
```

Appendix C - Python code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
import seaborn as sns
from sklearn.cross_validation import train_test_split
import sklearn
import fancyimpute
import statsmodels.api as sm


#For Time series Analyis
from statsmodels.tsa.stattools import adfuller, acf, pacf,arma_order_select_ic
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf
```

```
import statsmodels.api as sm
from statsmodels.regression.linear_model import OLS
from statsmodels.tools import add_constant
from matplotlib import pyplot
from statsmodels.tsa.seasonal import seasonal_decompose

raw_data = pd.read_excel('Absenteeism_at_work_Project.xls')

fig = plt.figure(figsize = (20,20))
ax = fig.gca()
raw_data.hist(ax=ax, facecolor='red',alpha=0.7, rwidth=0.85)
plt.savefig('histogram.png')

cols = ['Reason for absence','Month of absence', 'Education']
raw_data[cols] = raw_data[cols].replace({0:np.nan, 0:7, 0:np.nan})
data = raw_data

data = pd.DataFrame(fancyimpute.KNN(k=3).fit_transform(data), columns=data.columns)

for cols in data.columns:
   data[cols] = data[cols].astype('int64')



categorical_cols = ['ID', 'Reason for absence','Month of absence','Day of the week','Seasons','Disciplinary failure','Education', 'Social drinker','Social smoker','Son', 'Pet']
for cols in categorical_cols:
   data[cols] = data[cols].astype('category')



##### Outlier Analysis
numeric_data = data.drop(categorical_cols,axis=1)
```

```
categorical_data = data.drop(list(numeric_data.columns),axis=1)


“””boxplot”””
for i in range(1,11):
    plt.subplot(4,3,i)
    numeric_data.boxplot(column=numeric_data.columns[i-1], fontsize=15)


for col_names in numeric_data.columns:
    q75,q25 = np.percentile(numeric_data[col_names],[75,25])
    iqr = q75 - q25
    minimum = q25 - (iqr * 1.5)
    maximum = q75 + (iqr * 1.5)
    numeric_data.loc[numeric_data[col_names] < minimum,:col_names] = np.nan
    numeric_data.loc[numeric_data[col_names] > maximum,:col_names] = np.nan


numeric_data = pd.DataFrame(fancyimpute.KNN(k=3).fit_transform(numeric_data), columns=numeric_data.columns)


######correlation matrix
corr = data.corr()
plt.title('Pearson Correlation of features')
correlation_plot = sns.heatmap(corr, linewidths=0.4,vmax=1.0, square=True, cmap="cubehelix", linecolor='k', annot=True)


data = data.drop(['Weight'], axis=1)
#####Normalization
x = data.values
min_max_scaler = sklearn.preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x)
data = pd.DataFrame(x_scaled)
```

```
train, test = train_test_split(data, test_size = 0.2)
fit_DT = sklearn.tree.DecisionTreeRegressor(max_depth=5).fit(train.iloc[:,0:19],train.iloc[:,19])
prediction_DT = fit_DT.predict(test.iloc[:,0:19])

def MAPE(y_true, y_pred):
    mape = np.mean(np.abs((y_true-y_pred)/y_true))
    return mape

MAPE(test.iloc[:,19],prediction_DT)



############# Time series Approach
data.loc[0:112, 'Year'] = 2007
data.loc[113:358, 'Year'] = 2008
data.loc[359:570, 'Year'] = 2009
data.loc[571:740, 'Year'] = 2010

cols_name = data.columns.tolist()
column_to_move = "Year"
new_position = 3
cols_name.insert(new_position, cols_name.pop(cols_name.index(column_to_move)))
data = data[cols_name]

#Grouping the data based on year of month and total absenteeism hour
ts = data.groupby(['Year' , 'Month_of_absence'])['Absenteeism_time_in_hours'].sum()
ts.index=pd.date_range(start = '2007-07-01',end='2010-08-01', freq = 'M')

ts.astype('float')
plt.figure(figsize=(16,8))
plt.title('Total Absenteese hour per month (July 2007 to July 2010)')
plt.xlabel('Year-Month')

plt.ylabel('Absenteese Hour')

plt.plot(ts)



res = sm.tsa.seasonal_decompose(ts.values,freq=12,model="multiplicative")

fig = res.plot()

# Stationarity tests

def test_stationarity(timeseries):


    #Perform Dickey-Fuller test:

    dftest = adfuller(timeseries, autolag='AIC')

    dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of
Observations Used'])

    for key,value in dftest[4].items():

        dfoutput['Critical Value (%s)'%key] = value

    print (dfoutput)


test_stationarity(ts)




train = pd.DataFrame(ts[0:30])

test = pd.DataFrame(ts[30:37])



y_hat_avg = test.copy()

#we are using the same order as used in ARIMA in R with seasonal difference

history = [x for x in train]

fit1 = sm.tsa.statespace.SARIMAX(train.Absenteeism_time_in_hours, order=(0, 0,
0),seasonal_order=(0,1,0,12)).fit()
```

```
y_hat_avg['SARIMA'] = fit1.predict(start="2010-01-31", end= "2011-12-31", dynamic=True)

plt.figure(figsize=(16,8))

plt.plot( train['Absenteeism_time_in_hours'], label='Train')

plt.plot(test['Absenteeism_time_in_hours'], label='Test')

plt.plot(y_hat_avg['SARIMA'], label='SARIMA')

plt.legend(loc='best')

plt.show()
```

References

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 6. Springer.

Wickham, Hadley. 2009. *Ggplot2: Elegant Graphics for Data Analysis*. Springer Science & Business Media

“The Comprehensive R Archive Network”. Retrieved 2018-08-06.
Chatfield, C. (1995). *Problem Solving: A Statistician's Guide* (2nd ed.). Chapman and Hall. ISBN 0412606305

“Forecasting principles and practices” by Rob J Hyndman