Employee Absenteeism Report

-Rahul kumar

Contents

Chapter 1

1.1 Introduction

Project Description

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.

1.2 Problem statement

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?

1.3 Data

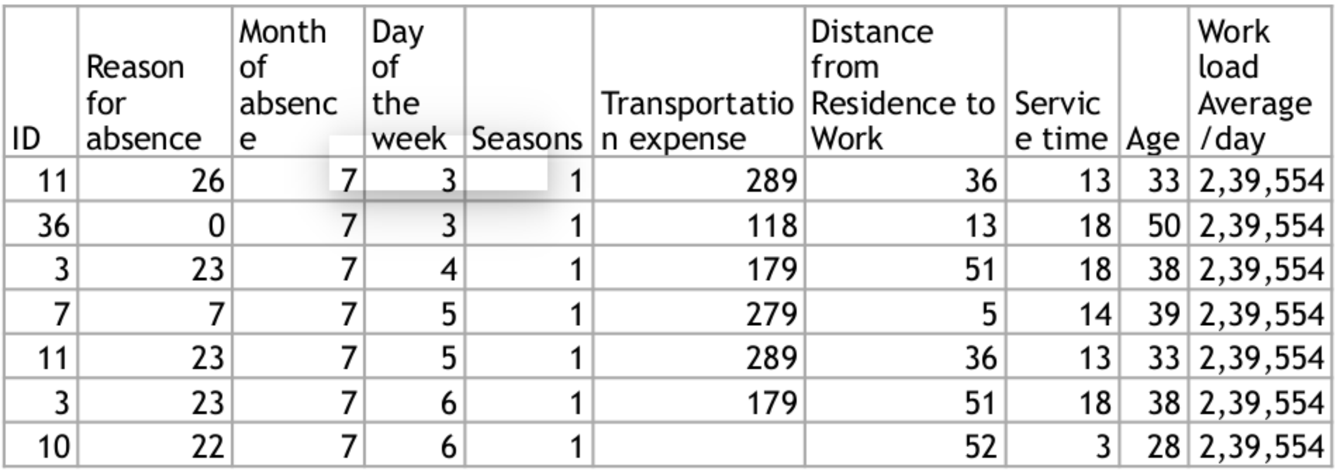
Dataset Details:

Dataset Characteristics: Timeseries Multivariant
Number of Attributes: 21
Missing Values : Yes

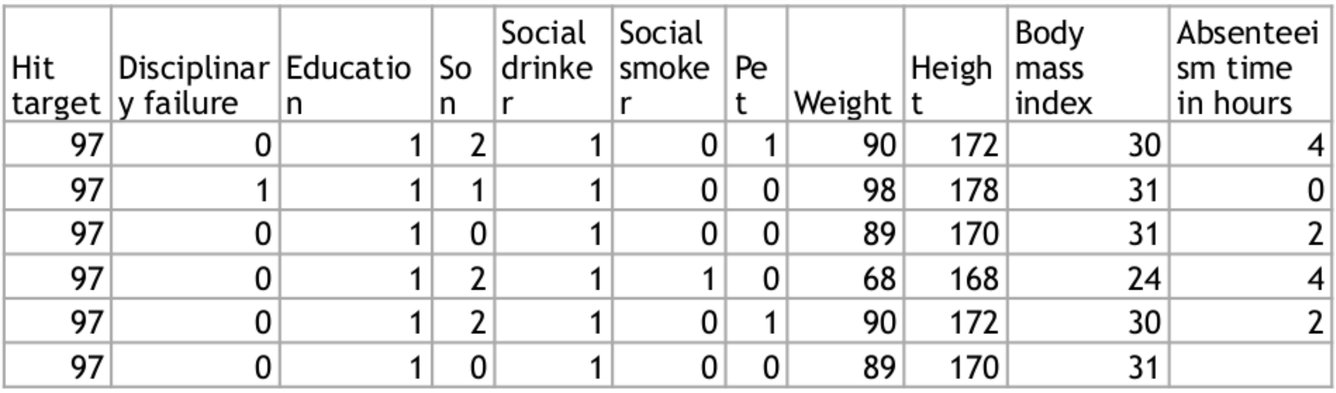
Attribute Information:
1. Individual identification (ID)
2. Reason for absence (ICD) (stratified into 28 categories)
3. Month of absence
4. Day of the week (Monday (2), Tuesday (3), Wednesday (4), Thursday (5), Friday (6))
5. Seasons (summer (1), autumn (2), winter (3), spring (4))
6. Transportation 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)



ID	Reason for absence	Month of absence	Day of the week	Seasons	Transportation expense	Distance from Residence to Work	Service time	Age	Work load Average /day
11	26	7	3	1	289	36	13	33	2,39,554
36	0	7	3	1	118	13	18	50	2,39,554
3	23	7	4	1	179	51	18	38	2,39,554
7	7	7	5	1	279	5	14	39	2,39,554
11	23	7	5	1	289	36	13	33	2,39,554
3	23	7	6	1	179	51	18	38	2,39,554
10	22	7	6	1		52	3	28	2,39,554



Hit target	Disciplinary failure	Education	Son	Social drinker	Social smoker	Pet	Weight	Height	Body mass index	Absenteeism time in hours
97	0	1	2	1	0	1	90	172	30	4
97	1	1	1	1	0	0	98	178	31	0
97	0	1	0	1	0	0	89	170	31	2
97	0	1	2	1	1	0	68	168	24	4
97	0	1	2	1	0	1	90	172	30	2
97	0	1	0	1	0	0	89	170	31	

Chapter 2

2.1 Data Preprocessing

Data preprocessing is a data mining 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 is generally incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data.

Data can be noisy: containing errors or outliers. Inconsistent: containing discrepancies in codes or names.

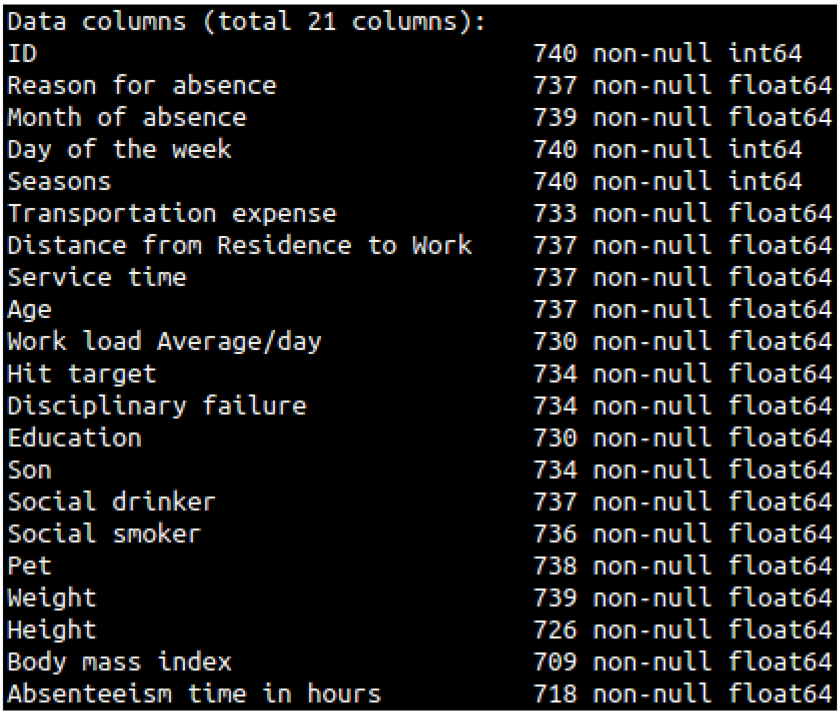
2.2 Data exploration

Data exploration is a methodology like beginning information examination, whereby an information expert uses visual investigation to comprehend what is in a dataset and the attributes of the information, as opposed to through customary information administration frameworks.

These qualities can incorporate size or measure of information, fulfillment of the information, rightness of the information, conceivable connections among information components or documents/tables in the information.

Data exploration needs collections and growing profound comprehension about the information is a standout amongst the most critical aptitude each datum researcher ought to have.

Exploratory information investigation (EDA) is a vital mainstay of information science, a basic advance required to finish each extend paying little heed to the space or the sort of information you are working with. It is exploratory examination that gives us a feeling of what extra work ought to be performed to measure and concentrate bits of knowledge from our information. It likewise illuminates us regarding what the finished result of our explanatory procedure ought to be. However, in the decade that I've been working in examination and information science. Individuals gauge that time spent on these exercises can go as high as 80% of the venture time at times.



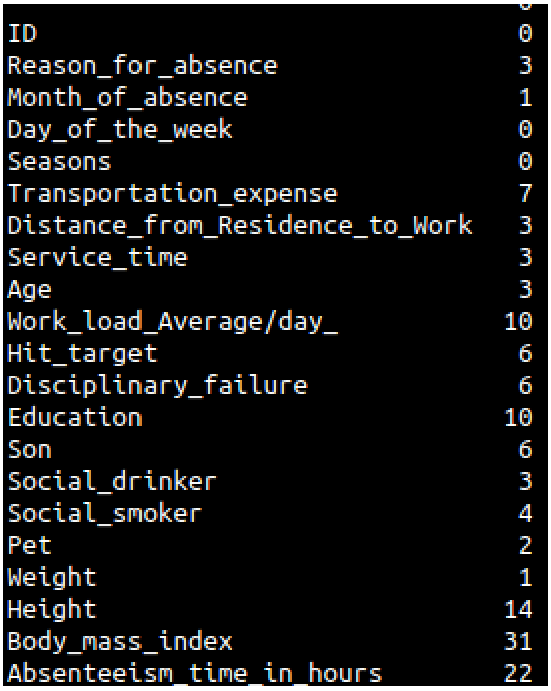
```
Data columns (total 21 columns):
ID                                 740 non-null int64
Reason for absence                 737 non-null float64
Month of absence                   739 non-null float64
Day of the week                    740 non-null int64
Seasons                            740 non-null int64
Transportation expense             733 non-null float64
Distance from Residence to Work    737 non-null float64
Service time                       737 non-null float64
Age                                737 non-null float64
Work load Average/day              730 non-null float64
Hit target                         734 non-null float64
Disciplinary failure               734 non-null float64
Education                          730 non-null float64
Son                                734 non-null float64
Social drinker                     737 non-null float64
Social smoker                      736 non-null float64
Pet                                738 non-null float64
Weight                             739 non-null float64
Height                             726 non-null float64
Body mass index                    709 non-null float64
Absenteeism time in hours          718 non-null float64
```

2.3 Missing values

The missing values in the data can be treated by following ways:

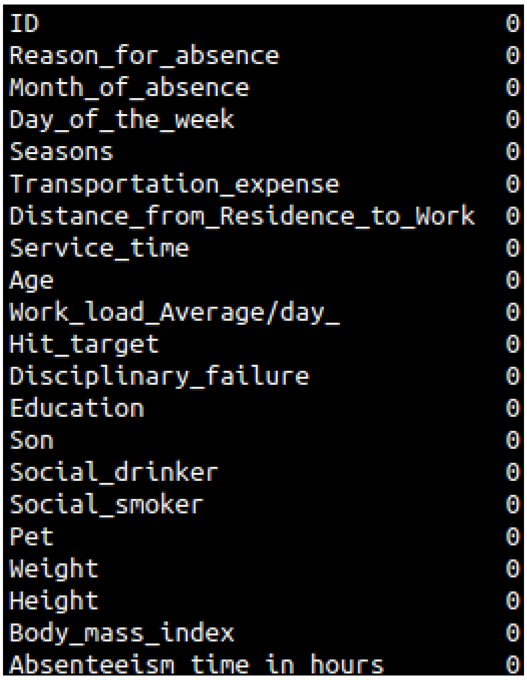
- Imputing missing data by mean, mode or median.
- Imputing missing data predictive models like KNN.

Data before imputations:



```
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
```

When the data goes by imputation these data are substituted.



```
ID                                 0
Reason_for_absence                 0
Month_of_absence                   0
Day_of_the_week                    0
Seasons                            0
Transportation_expense             0
Distance_from_Residence_to_Work    0
Service_time                       0
Age                                0
Work_load_Average/day_             0
Hit_target                         0
Disciplinary_failure               0
Education                          0
Son                                0
Social_drinker                     0
Social_smoker                      0
Pet                                0
Weight                             0
Height                             0
Body_mass_index                    0
Absenteeism_time_in_hours          0
```

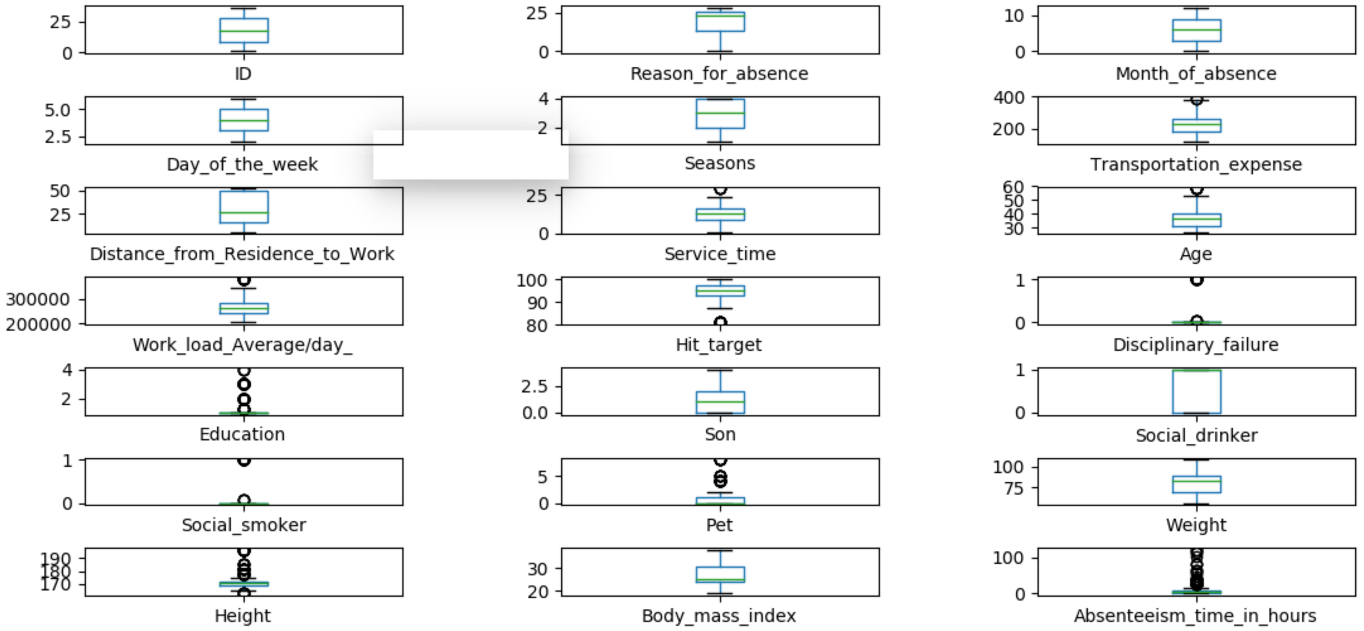
2.4 Outlier Analysis

An Outlier is an uncommon shot of event inside a given informational index. In Data Science, an Outlier is a perception point that is inaccessible from different perceptions. An Outlier might be because of inconstancy in the estimation or it might demonstrate test mistake.

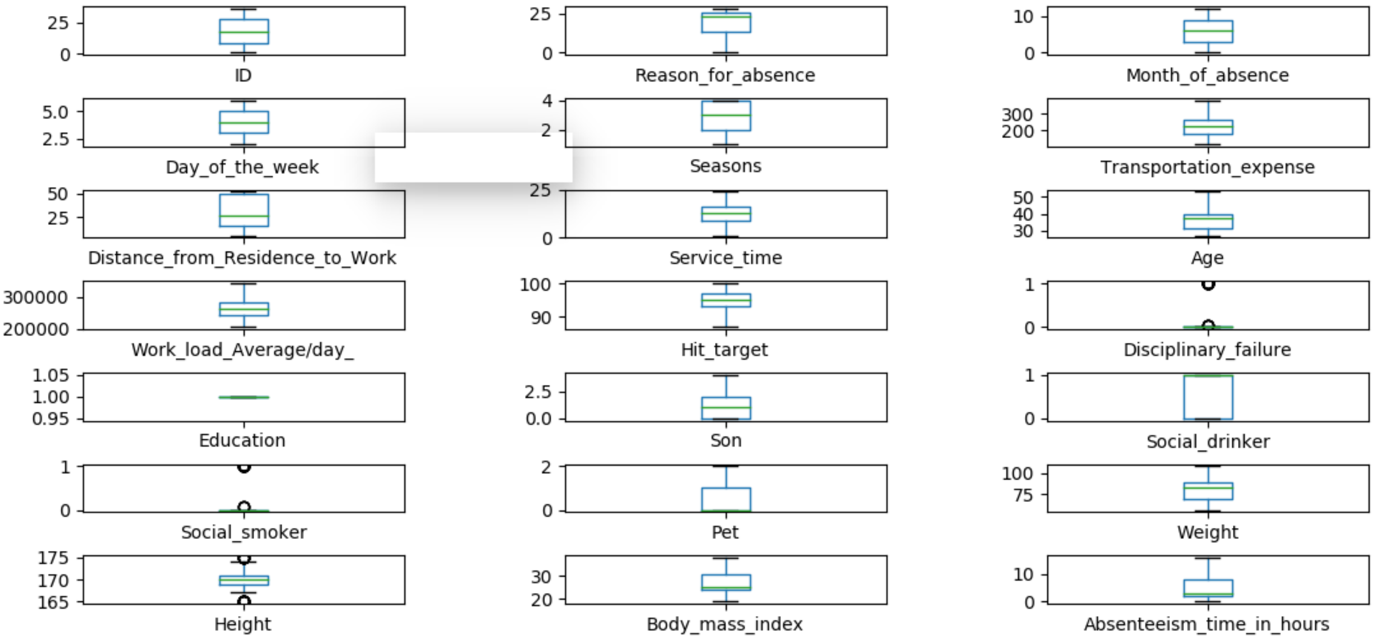
A univariate exception is an information point that comprises of an outrageous incentive on one variable. A portion of the Univariate Outlier Detection Techniques prominently utilized are "The Box Plot Rule", Grubbs Test.

Box plot outline additionally named as Whisker's plot is a graphical technique regularly portrayed by quartiles and bury quartiles that aides in characterizing as far as possible and lower confine past which any information lying will be considered as anomalies.

The plain motivation behind this chart is to recognize anomalies and dispose of it from the information arrangement before mentioning any further objective fact with the goal that the end produced using the investigation gives more precise outcomes not affected by any limits or anomalous qualities.

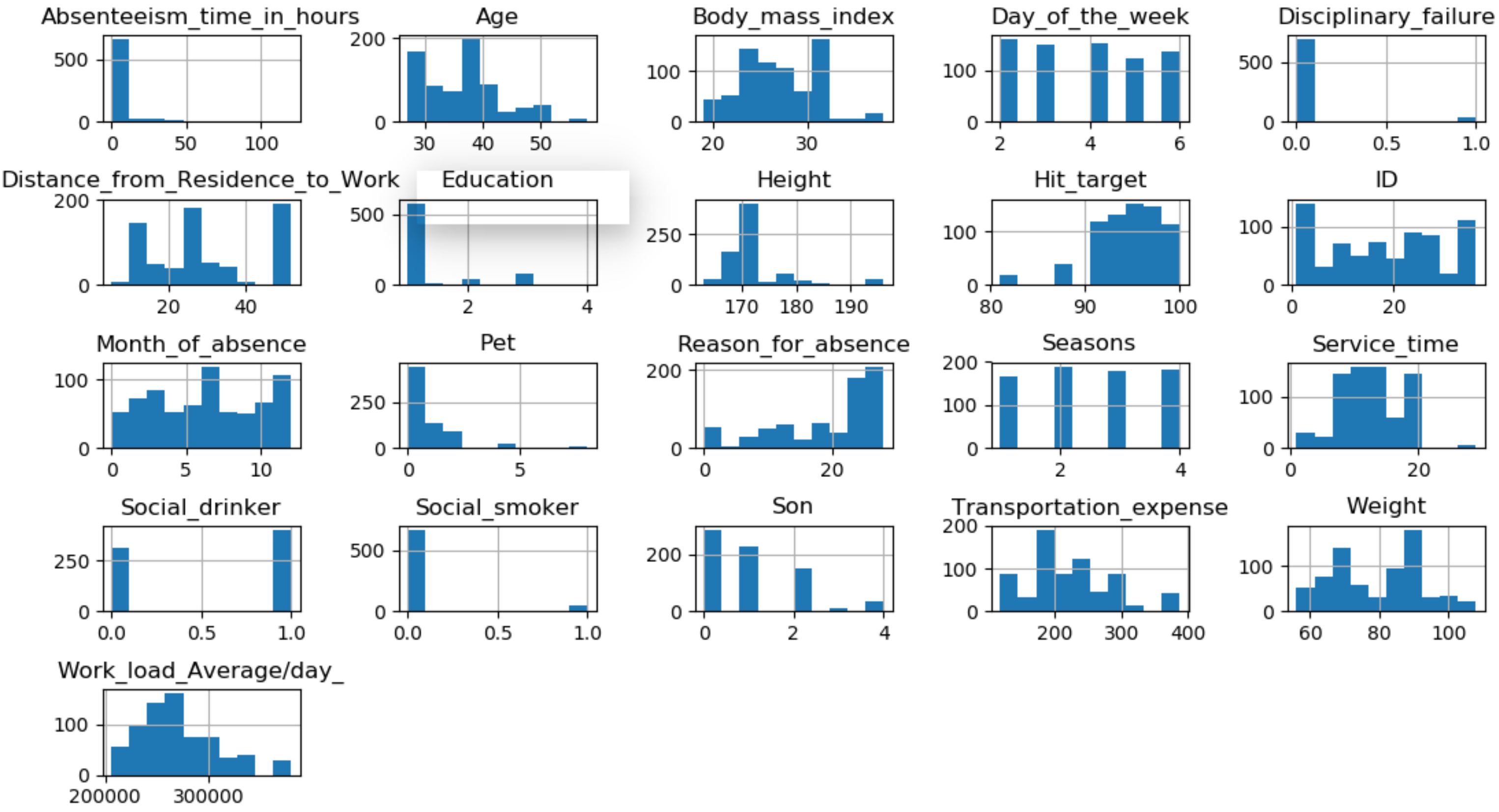


After outlier treatment:



2.5 Histograms Plot

Histograms are an approach to outline a numeric variable. They utilize checks to total comparative qualities together and demonstrate to you the general circulation. Be that as it may, they can be touchy to parameter decisions. We will make you stride by advance through the contemplations with loads of information representations.

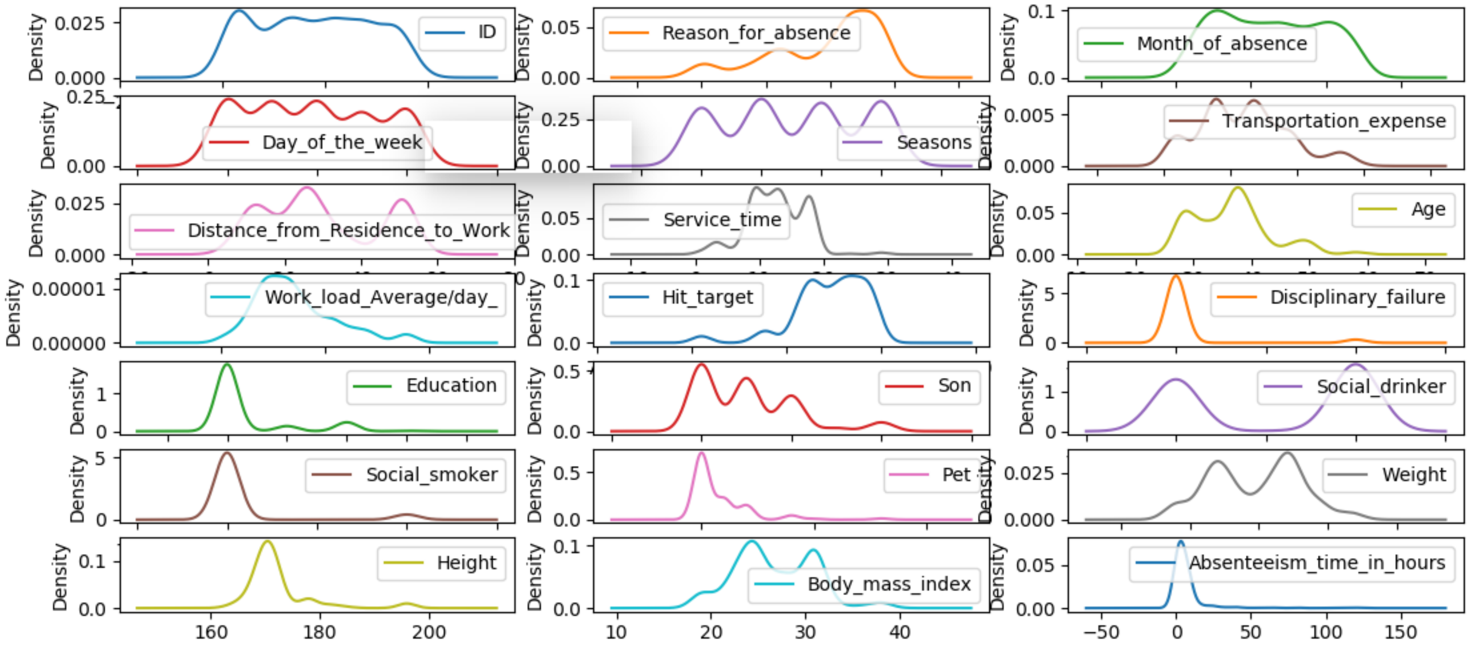


A histogram is a plot that gives you a chance to find, and show, the basic recurrence appropriation (shape) of an arrangement of constant information. This permits the review of the information for its basic circulation (e.g., ordinary dissemination), anomalies, skewness, and so forth.

2.6 Density Plots

A Density Plot envisions the conveyance of information over a ceaseless interim or day and age. This diagram is a variety of a Histogram that utilizations part smoothing to plot esteems, taking into account smoother conveyances by smoothing out the clamor. The pinnacles of a Density Plot help show where esteems are thought over the interim.

Leeway Density Plots have over Histograms is that they're better at deciding the dispersion shape since they're not influenced by the quantity of receptacles utilized (each bar utilized in a run of the mill histogram). A Histogram including just 4 receptacles wouldn't deliver a sufficiently discernable state of dissemination as a 20-container Histogram would. In any case, with Density Plots, this isn't an issue.



None of the above variables are having a similar distributions, each variable follows a random normal distribution.

Chapter 3

3. Modelling

Data Modelling is the way toward reporting a mind boggling programming framework outline as an effortlessly comprehended graph, utilizing content and images to speak to the route data needs to stream. The graph can be utilized as a diagram for the development of new programming or for re-building a heritage application.

Traditionally, information models have been worked amid the examination and configuration periods of an undertaking to guarantee that the necessities for another application are completely understood. A information model can be thought of as a flowchart that represents the connections between information.

Despite the fact that catching all the conceivable connections in an information model can be extremely time-escalated, it's an essential advance that shouldn't be hurried. Well-documented conceptual, legitimate and physical information models allow partners to distinguish mistakes and roll out improvements previously any programming code has been composed.

Information modelers regularly utilize various models to see similar information and guarantee that all procedures, elements, connections and information streams have been recognized.

3.1 Regression

What is Regression

Regression is a factual measure utilized in fund, contributing and different controls that endeavors to decide the quality of the connection between one variable (for the most part meant by Y) and a progression of other evolving factors (known as free factors/independent variables). Relapse helps speculation and money related chiefs to esteem resources and comprehend the connections between factors, for example, ware costs and the supplies of organizations managing in those wares.

3.2 Linear Regression

The two basic types of regression are linear regression and multiple linear regression, although there are non-linear regression methods for more complicated data and analysis. Linear regression uses one independent variable to explain or predict the outcome of the dependent variable Y, while multiple regression uses two or more independent variables to predict the outcome.

Linear Regression: $Y = a + bX + u$

Multiple Regression: $Y = a + b_1X_1{}^{+}\ b_2X_2 + b_3X_3 + ... + b_tX_t + u$

Where:

Y = the variable that you are trying to predict (dependent variable)

X = the variable that you are using to predict Y (independent variable)

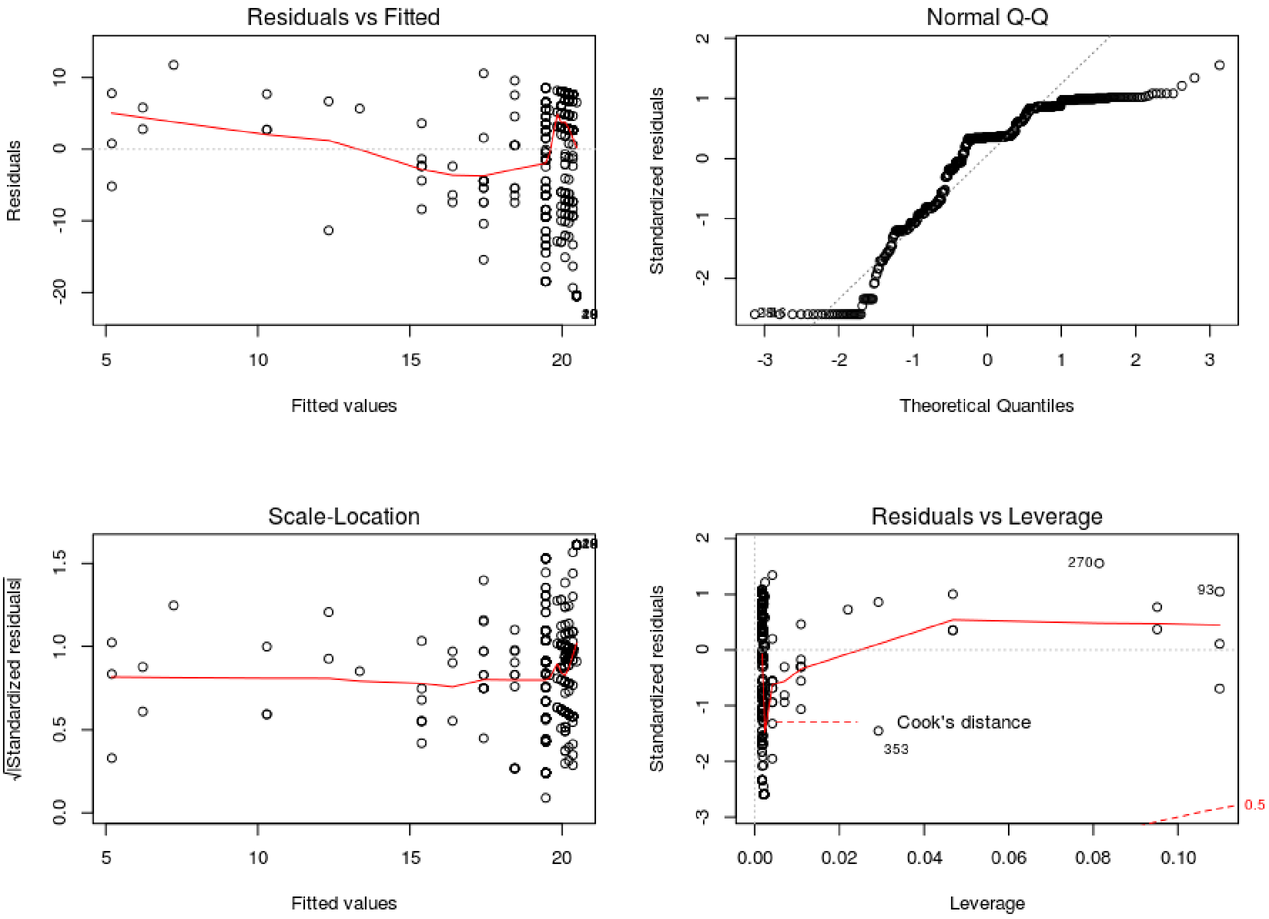
a = the intercept

b = the slope

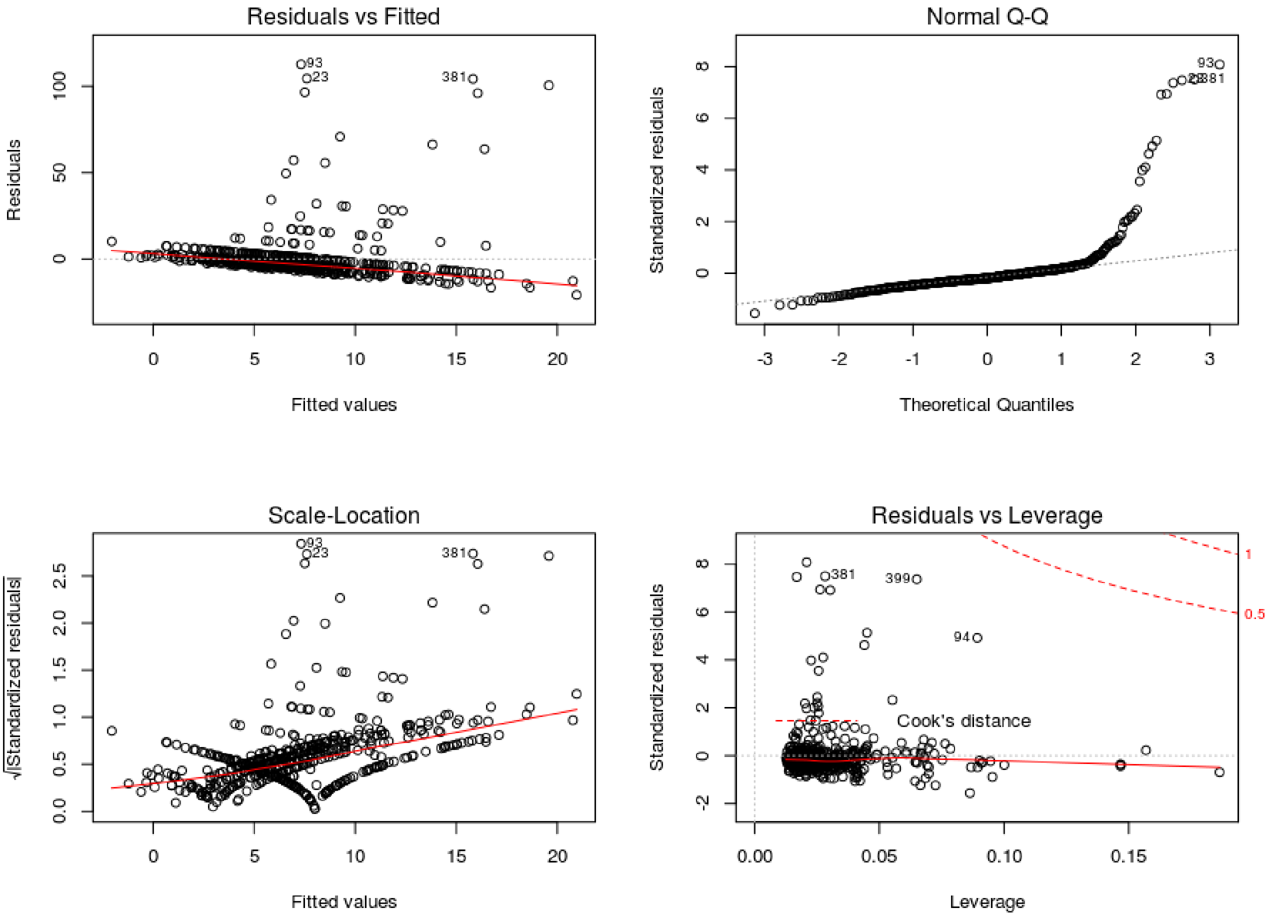
u = the regression residual

Regression takes a group of random variables, thought to be predicting Y, and tries to find a mathematical relationship between them. This relationship is typically in the form of a straight line (linear regression) that best approximates all the individual data points. In multiple regression, the separate variables are differentiated by using numbers with subscript.

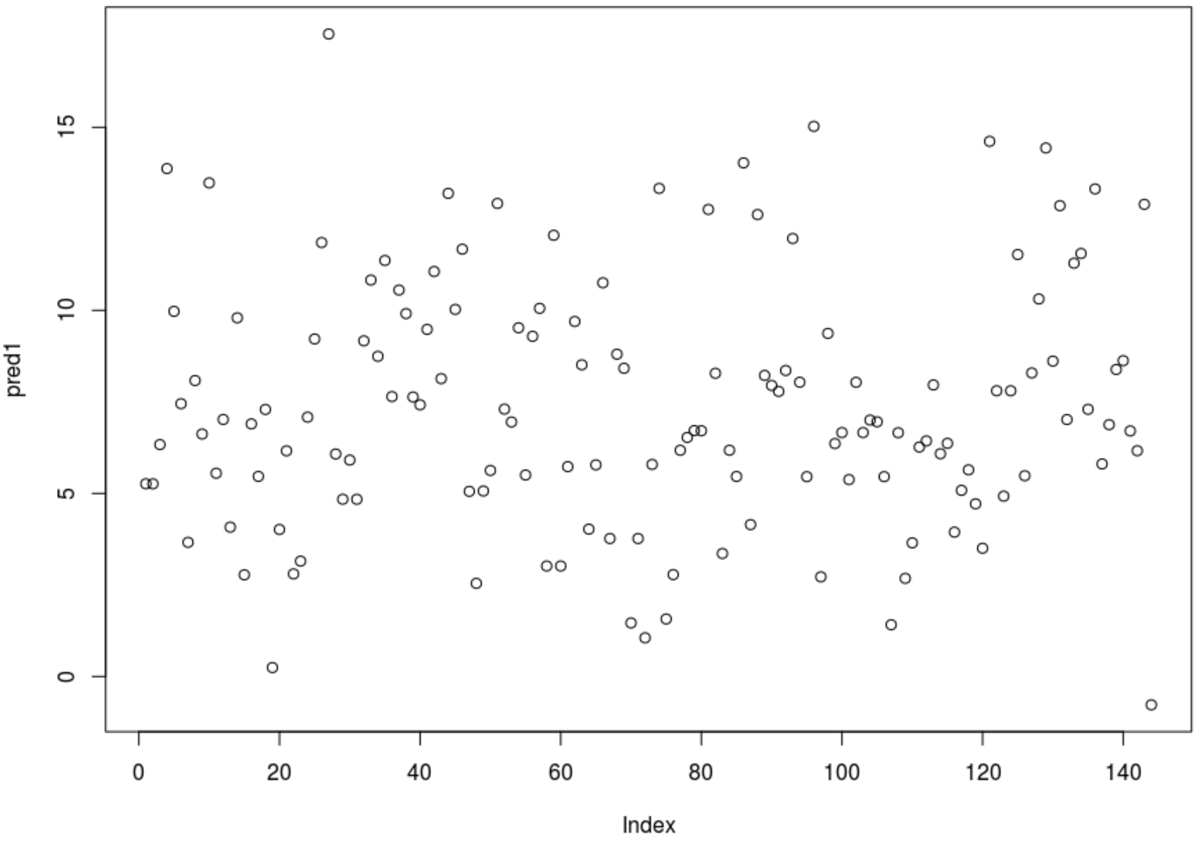
Results with Abseentism hours taken in consideration with reasons of hours:

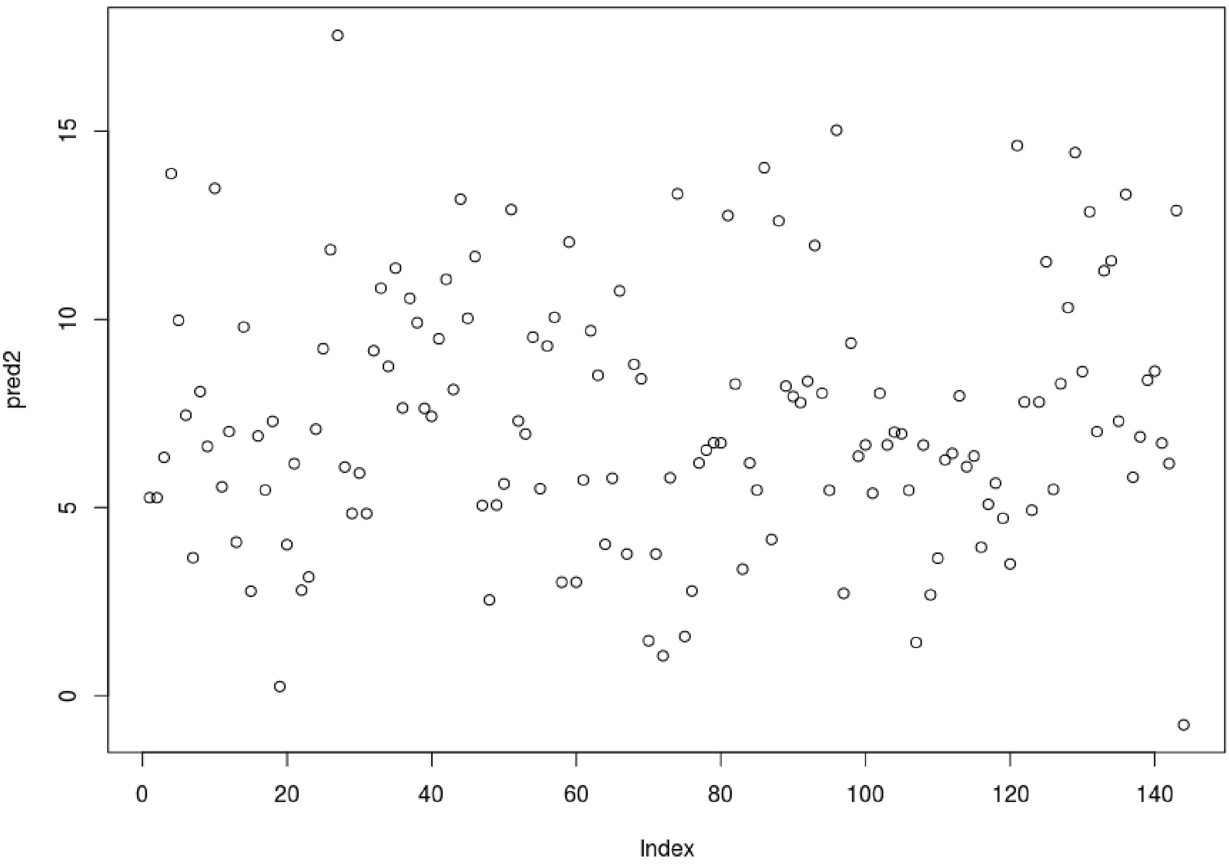


Taking significant variables to fit the model on training data :



Performance on the test data with different models developed under predictions:




pred2
0
5
10
15
Index
0
20
40
60
80
100
120
140

Chapter 4

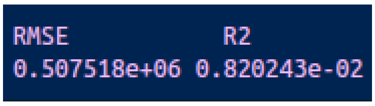
4. Validation

4.1 RMSE, R2

The RMSE is the square base of the change of the residuals. It shows without a doubt the attack of the model to the data– how shut the watched information indicates are the model's anticipated qualities. While R-squared is a relative proportion of fit, RMSE is a flat out proportion of fit. As the square base of a change, RMSE can be translated as the standard deviation of the unexplained difference, and has the helpful property of being in indistinguishable units from the reaction variable. Lower estimations of RMSE demonstrate better fit. RMSE is a decent proportion of how precisely the model predicts the reaction, and it is the most essential measure for fit if the primary reason for the model is expectation.

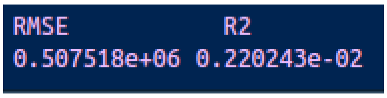
The best proportion of model fit relies upon the scientist's destinations, and more than one are regularly helpful. The insights examined above are relevant to relapse models that utilization OLS estimation. Numerous sorts of relapse models, be that as it may, for example, blended models, summed up straight models, and occasion history models, utilize greatest probability estimation. These measurements are not accessible for such models.

With 1st model:



```
RMSE          R2
0.507518e+06 0.820243e-02
```

With 2nd model:



```
RMSE          R2
0.507518e+06 0.220243e-02
```

Q. *Loss Prediction :*

Loss Prediction can be done as :

Loss = absenteeism time * work load average/day

Appendix

A1 R code

```
#importing library
library("readxl")
library("ggplot2")

#Importing the dfset
df <- read_excel('Absenteeism_at_work_Project.xls')

#getting the names of the column in the dfset
colnames(df)

colnames(df) = gsub(" ", "_", colnames(df))

#getting the dimension of dfset
dim(df)

#getting the summary of the dfset
summary(df)

missing_values_count = data.frame(apply(df,2,function(x)
{sum(is.na(x))}))
names(missing_values_count)[1] =  "Missing_values_count"
print(missing_values_count)

# Droping observation in which "Absenteeism time in hours" has
missing value
```

```r
df = df[!is.na(df$Absenteeism_time_in_hours), ]

#checking null values in target variable
sum(is.na(df$Absenteeism_time_in_hours))

#Checking Dimension again with removed NA's from Target Variable
dim(df)

## Imputing Missing Values using mean to fill the dataframe
fillNAwithMean <- function(x){
  na_index <- which(is.na(x))
  mean_x <- mean(x, na.rm=T)
  x[na_index] <- mean_x
  return(x)
}

(df <- apply(df,2,fillNAwithMean))

#checking if NA exits
sum(is.na(df))

apply(df, 2, function(x) any(is.na(x)))

#Density plots
library("kdensity")
plot(density(df$ID))
plot(density(df$Reason_for_absence))
plot(density(df$Month_of_absence))
plot(density(df$Day_of_the_week))
plot(density(df$Seasons))
plot(density(df$Transportation_expense))
plot(density(df$Distance_from_Residence_to_Work))
plot(density(df$Service_time))
plot(density(df$Age))
plot(density(df$Work_load_Average.day))
plot(density(df$Hit_target))
plot(density(df$Absenteeism_time_in_hours))
plot(density(df$Body_mass_index))
plot(density(df$Height))
plot(density(df$Weight))
plot(density(df$Pet))
plot(density(df$Social_smoker))
plot(density(df$Social_drinker))
plot(density(df$Son))
plot(density(df$Education))
plot(density(df$Disciplinary_failure))

#boxplots
boxplot(ID  ~ Absenteeism_time_in_hours, data = df)
boxplot(Reason_for_absence  ~ Absenteeism_time_in_hours, data = df)
boxplot(Month_of_absence  ~ Absenteeism_time_in_hours, data = df)
boxplot(Day_of_the_week  ~ Absenteeism_time_in_hours, data = df)
boxplot(Seasons  ~ Absenteeism_time_in_hours, data = df)
boxplot(Work_load_Average.day  ~ Absenteeism_time_in_hours, data = df)
boxplot(Age  ~ Absenteeism_time_in_hours, data = df)
boxplot(Hit_target  ~ Absenteeism_time_in_hours, data = df)
boxplot(Disciplinary_failure  ~ Absenteeism_time_in_hours, data = df)
boxplot(Education  ~ Absenteeism_time_in_hours, data = df)
boxplot(Son  ~ Absenteeism_time_in_hours, data = df)
boxplot(Social_smoker  ~ Absenteeism_time_in_hours, data = df)
boxplot(Social_drinker  ~ Absenteeism_time_in_hours, data = df)
boxplot(Pet  ~ Absenteeism_time_in_hours, data = df)
boxplot(Weight  ~ Absenteeism_time_in_hours, data = df)
boxplot(Height  ~ Absenteeism_time_in_hours, data = df)
boxplot(Body_mass_index  ~ Absenteeism_time_in_hours, data =
df)

#histograms for each column
for (col in 2:ncol(df)) {
  hist(df[,col])
  }

#density details
apply(df,  2,  density, kernel="gaussian", bw=15)

#splitting the data into train/test
set.seed(1)
row.number <- sample(1:nrow(df), 0.8*nrow(df))
train = df[row.number,]
#train = data.frame(train)
test = df[-row.number,]
#test = data.frame(test)
dim(train)
dim(test)

#response of the target variable
ggplot(train, aes(Absenteeism_time_in_hours)) +
geom_density(fill="blue")
ggplot(train, aes(log(Absenteeism_time_in_hours))) +
geom_density(fill="green")
ggplot(train, aes(sqrt(Absenteeism_time_in_hours))) +
geom_density(fill="brown")

#linear regression model with first parameter
library(mlbench)
library(caret)
model1 = lm(Reason_for_absence ~ Absenteeism_time_in_hours,
data=train, na.action=na.omit)
summary(model1)
```

```
par(mfrow=c(2,2))
plot(model1)

#removing the redundant features
model2 = update(model1, ~.-Pet-Height-Son-Weight-
Social_smoker-Social_drinker)
summary(model2)
par(mfrow=c(2,2))
plot(model2)


#Lets  make default model and add square term in the model.
model3 =
lm(Absenteeism_time_in_hours~Month_of_absence+Day_of_the
_week+Seasons+Transportation_expense+Distance_from_Residen
ce_to_Work+

Service_time+Work_load_Average.day+Body_mass_index+I(Mo
nth_of_absence^2)+ I(Day_of_the_week^2)+I(Seasons^2)+
I(Transportation_expense^2)+
I(Distance_from_Residence_to_Work^2)+
            I(Service_time^2)+ I(Work_load_Average.day^2)+
I(Body_mass_index^2), data=train)
summary(model3)

##Removing the insignificant variables of model3
model4=update(model3, ~.-Body_mass_index-
I(Body_mass_index^2))
summary(model4)
par(mfrow=c(2,2))
plot(model4)

#predicting the test data performance with different models
developed
pred1 <- predict(model4, newdata = test)
```

```
rmse <- sqrt(sum((exp(pred1) - test$Absenteeism_time_in_hours)^2)/length(test$Absenteeism_time_in_hours))
c(RMSE = rmse, R2=summary(model4)$r.squared)
par(mfrow=c(1,1))
plot(test$Absenteeism_time_in_hours, exp(pred1))

pred2 <- predict(model3, newdata = test)
rmse <- sqrt(sum((exp(pred2) - test$Absenteeism_time_in_hours)^2)/length(test$Absenteeism_time_in_hours))
c(RMSE = rmse, R2=summary(model3)$r.squared)
par(mfrow=c(1,1))
plot(test$Absenteeism_time_in_hours, exp(pred2))
```

A2 Python Code

```
#importing libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

#Importing the dataset
df = pd.read_excel('Absenteeism_at_work_Project.xls')

#describing the dataset to getthe statistics of the dataset
df.describe()

#More information on dataset
df.info()

#getting the dimension of dataset
df.shape

#replacing space in names of columns with underscor
df.columns = df.columns.str.replace(' ', '_')

#check whether there are missing values in the dataframe
count_missing_values = pd.DataFrame(df.isnull().sum())
print(count_missing_values)

# Droping observation in which "Absenteeism time in hours" has missing value
df = df.drop(df[df['Absenteeism_time_in_hours'].isnull()].index, axis=0)
print(df.shape)
```

```
#checking null values in target variable
print(df['Absenteeism_time_in_hours'].isnull().sum())

## Imputing Missing Values using mean to fill
df = df.fillna(df.mean())

#verifing again the presence of missing values in the dataframe
verify_missing_values = pd.DataFrame(df.isnull().sum())
print(verify_missing_values)

#plotting the histograms
df.hist(figsize=(15,12))
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top= 0.9,
wspace=0.5, hspace=0.9)
plt.show()

#checking distribution of the 'variables' with the 'density' plot
df.plot.density(figsize=(15,12), subplots = True , layout=(8,3),
sharex=False)
plt.show()

#checking the outliers with box plot of each variables
df.plot.box(figsize=(15,12), subplots = True , layout=(8,3),
sharex=False, sharey= False)
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top= 0.9,
wspace=0.5, hspace=0.9)
plt.show()

#Removing the outliers
columns_with_outliers =['Service_time', 'Pet', 'Age',
'Work_load_Average/day_', 'Transportation_expense',
       'Hit_target', 'Education' , 'Height',
'Absenteeism_time_in_hours']
```

```
for col in columns_with_outliers:

    quartile_75, quartile_25 = np.percentile(df[col], [75,25])       #
Fetching quartile at 25th and 75th Percentile
    interquartile_range = quartile_75 - quartile_25                #
Finding difference to get interquartile range

    lower_limit = quartile_25 - (interquartile_range*1.5)           #
Defining lower limit
    upper_limit = quartile_75 + (interquartile_range*1.5)           #
Defining upper limit

    df.loc[df[col]< lower_limit,col] = np.nan                       #
Replacing the outliers values with NA.
    df.loc[df[col]> upper_limit,col] = np.nan


df = df.fillna(df.mean())                                    # Imputing
missing values with mean
df.isnull().sum().sum()                                      # Checking if
there is any missing value

#checking the box plot of each variables without outliers
df.plot.box(figsize=(15,12), subplots = True , layout=(8,3),
sharex=False, sharey= False)
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top= 0.9,
wspace=0.5, hspace=0.9)
plt.show()

#Forming the input and target data
X = df[['ID', 'Reason_for_absence', 'Month_of_absence',
'Day_of_the_week','Seasons','Transportation_expense','Distance_f
rom_Residence_to_Work', 'Service_time',
'Age','Work_load_Average/day_', 'Hit_target',
'Disciplinary_failure','Education', 'Son', 'Social_drinker',
'Social_smoker', 'Pet', 'Weight', 'Height', 'Body_mass_index']]
y = df[['Absenteeism_time_in_hours']]

#splitting the datasets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)

##applying model
from sklearn.linear_model import LinearRegression
model = LinearRegression()
fit_model = model.fit(X_train, y_train)
predict_model = fit_model.predict(X_train)
test_predict = fit_model.predict(X_test)

print('\t')

print('Coefficients: \n', model.coef_)

print('\t')

#cross validating through mean_square_error
from sklearn.metrics import mean_squared_error
print("Mean squared error: %.2f"
      % mean_squared_error(y_test, test_predict))
```