Predicting Employee Absenteeism

Satish kumar sahu

19th June 2018

Contents

Introduction

1.1 Problem Statement

Human capital plays an important role in collection, transportation and delivery in a courier company. The aim of the project is to analyze the data and suggest some changes to reduce employee absenteeism so that company can prevent losses. We would like to predict employee absenteeism based on various factors which have been provided in the company's dataset. We would also going to see how much losses every month company would suffer in 2011 if the same trend of absenteeism continues.

1.2 Data

Our task is to build regression models which will predict employee absenteeism depending on multiple Socio-physical factors. Given below is a sample of the data set that we are using to predict employee absenteeism.

Table 1.1: Employee's Sample Data (Columns: 1-9)

1	ID	Reason for absence	Month of absence	Day of the week	Seasons	Transportation expens	Distance from Residence to Wor	Service time	Age
2	11	26	7	3	1	289	36	13	33
3	36	0	7	3	1	118	13	18	50
4	3	23	7	4	1	179	51	18	38
5	7	7	7	5	1	279	5	14	39

Table 1.2: Employee's Sample Data (Columns: 10-22)

Work load Average/day	Hit target	Disciplina	Education	Son	Social drir	Social smc	Pet	Weight	Height	Body mass inde:	Absenteeism time in hours
239,554	97	0	1	2	1	0	1	90	172	30	4
239,554	97	1	1	1	1	0	0	98	178	31	0
239,554	97	0	1	0	1	0	0	89	170	31	2
239,554	97	0	1	2	1	1	0	68	168	24	4

As you can see in the table below we have the following 20 variables, using which we have to correctly predict employee absenteeism. Before making prediction we have to perform correlation analysis for categorical variable, which will help us to remove the unnecessary variables. We will be also using anova for numerical variable.

Table 1.5: Predictor Variables

S.No:	Predictor
1.	Individual identification (ID)
2.	Reason for absence (ICD).
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

Methodology

2.1 Pre Processing

Any predictive modeling requires that we look at the data before we start modeling. However, in data mining terms *looking at data* refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as **Exploratory Data Analysis**. Exploratory Data Analysis includes various processes. We will go through each process involved in the coming subsection.

2.1.1 Missing value analysis

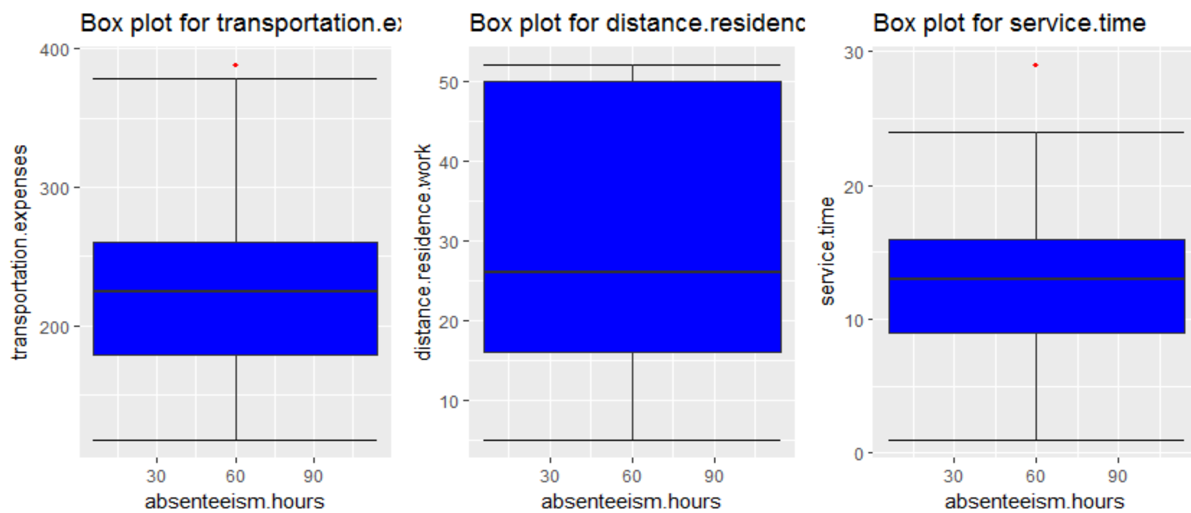
In statistics, **missing data**, or **missing values**, occur when no **data value** is stored for the variable in an observation. **Missing data** are a common occurrence and can have a significant effect on the conclusions that can be drawn from the **data**.
We can clearly see from the graph below that there are values missing from the data. There can be various reason for it. The values which are missing can create a lot of problem in prediction.so we will perform missing value imputation in order to fill the missing value in the dataset. We have used median for missing value imputation because it is easily to implement. We can also perform knn imputation but in a small dataset both knn and median gives the same values for imputation. so, I preferred median for missing value imputation.
From the data provided below we can see that missing value percentage is less than thirty (<30). Therefore, we will have to impute the missing values. For imputation, we have used median for missing value imputation. The code for missing value imputation is given in the appendix

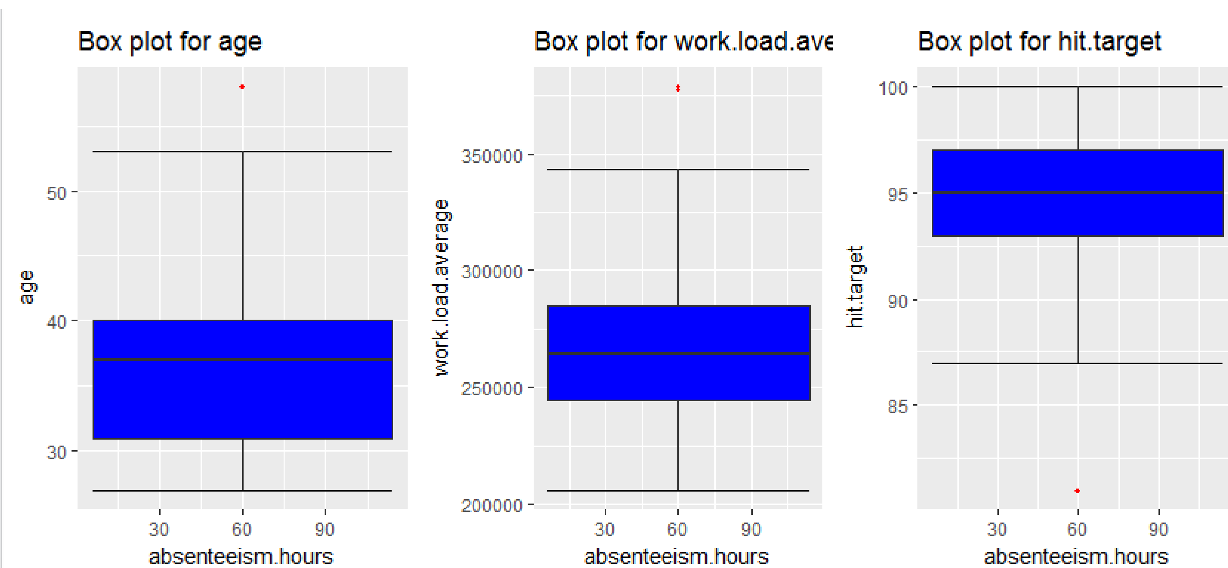
	Variables	Missing_percentage
20	Body mass index	4.1891892
21	Absenteeism time in hours	2.9729730
19	Height	1.8918919
10	Work load Average/day	1.3513514
13	Education	1.3513514
6	Transportation expense	0.9459459
11	Hit target	0.8108108
12	Disciplinary failure	0.8108108
14	Son	0.8108108
16	Social smoker	0.5405405
2	Reason for absence	0.4054054
7	Distance from Residence to Work	0.4054054
8	Service time	0.4054054
9	Age	0.4054054
15	Social drinker	0.4054054
17	Pet	0.2702703
3	Month of absence	0.1351351
18	Weight	0.1351351
1	ID	0.0000000
4	Day of the week	0.0000000
5	Seasons	0.0000000

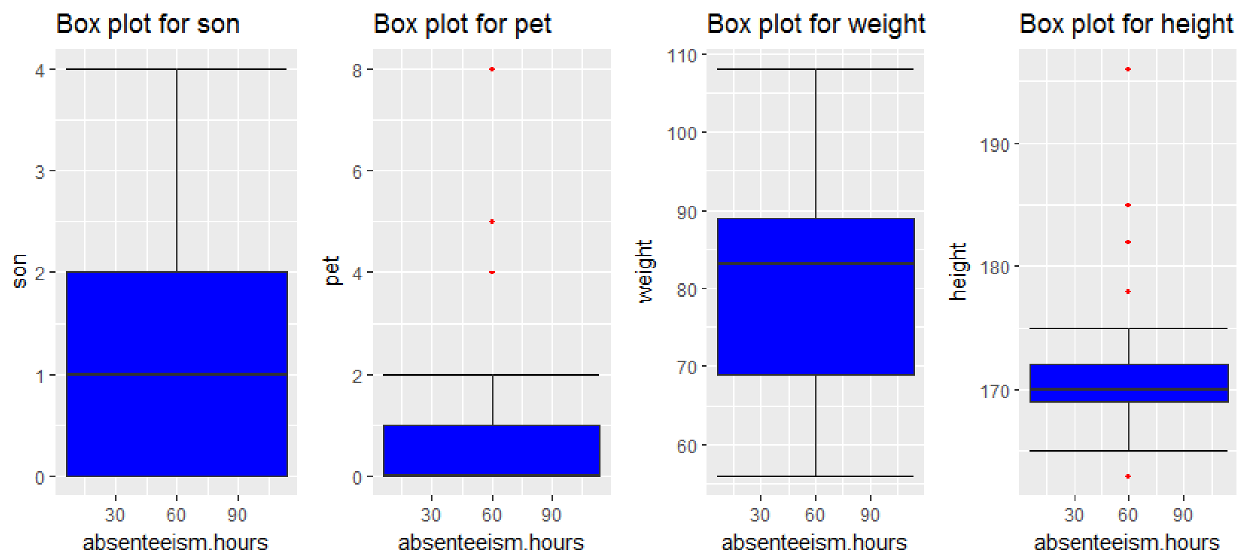
2.1.2 Outlier Analysis

In statistics, an **outlier** is an observation point that is distant from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set. An outlier can cause serious problems in statistical analyses. We can clearly observe from the boxplot that that most of the numerical variables contains red dot, this is clearly the effect of outliers and extreme values.
In this case we use a classic approach of removing outliers, Tukey's *method*. We visualize the

outliers using *boxplots*. In figure , we have plotted the boxplots of the 10 predictor variables with respect to each absenteeism in hours value for columns 6,7,8,9,10,11,14,17,18,19,20. A lot of useful inferences can be made from these plots. First as you can see, we have a lot of outliers and extreme values in each of the data set.







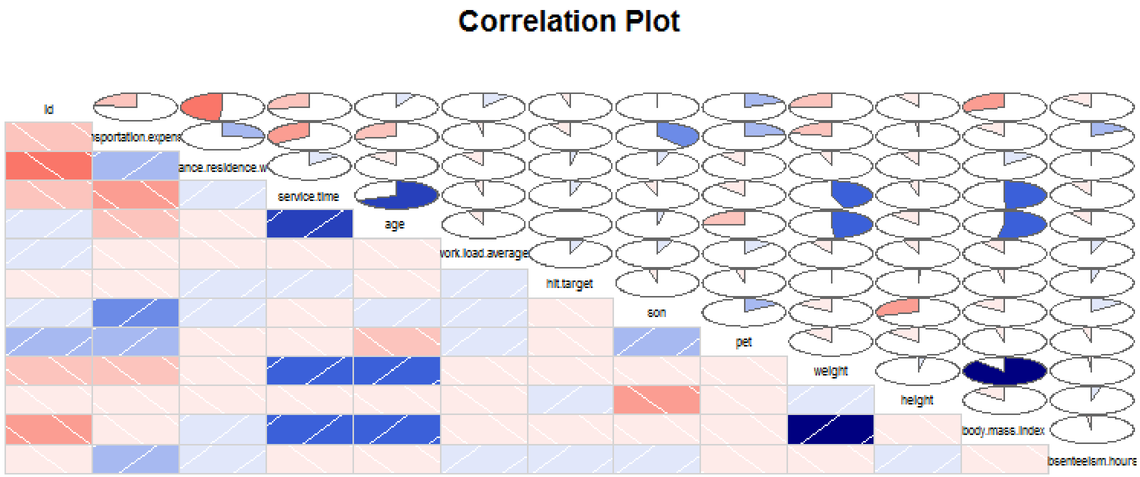
We have replaced outlier with NAs and later used median for imputation. We have replacing by NAs method because there are only 740 observation which is very less. So, we have not removed the outliers instead we have replace with NAs and then imputed the missing values using median.

2.1.3 Feature selection

In machine learning and statistics, **feature selection**, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are used for four reasons:

- Simplification of models to make them easier to interpret by researchers/users.
- shorter training times,
- to avoid the curse of dimensionality,
- enhanced generalization by reducing overfitting (formally, reduction of variance[])

We have used correlation analysis for categorical variables. The correlation plot obtained for categorical variable tells us that there is high correlation between weight and body mass index. So, we have removed body mass index from the data set.

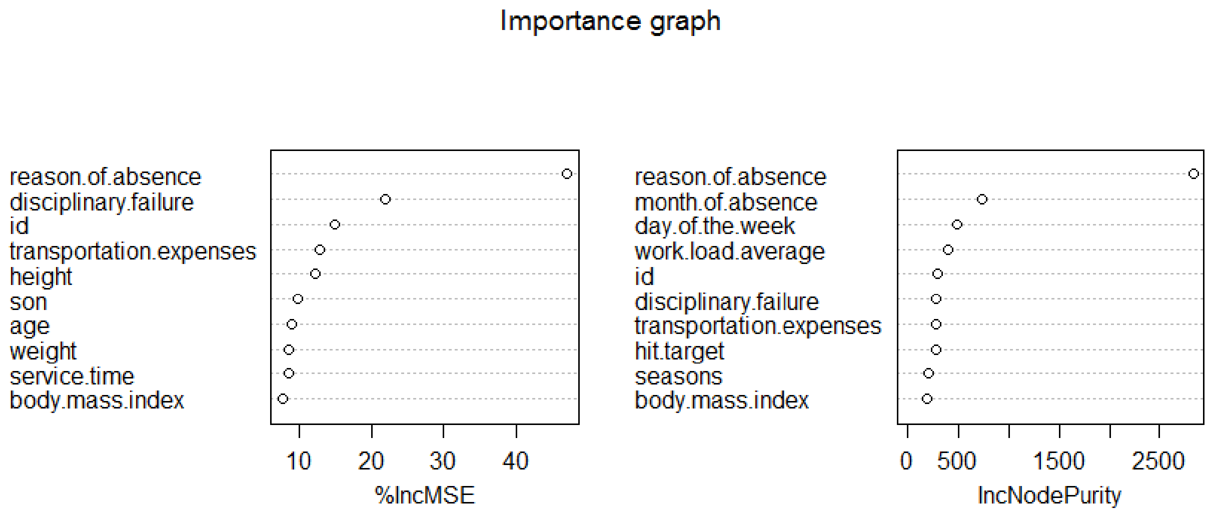


For numerical variable we have used annova to know which have variable are important. Finally, we have removed (id,education,day.of.the.week, pet,hit.target, seasons,social.smoker, social.drinker,weight). After performing correlation analysis, we also performed some hit and trail to increase the r square value and decrease the rsme value.

ANOVA TABLE

```
> summary(result)
                      Df Sum Sq Mean Sq F value Pr(>F)    
reason.of.absence     27   2928  108.43  14.983 <2e-16 ***
month.of.absence      12     89    7.39   1.021  0.428    
day.of.the.week        4     28    7.01   0.969  0.424    
seasons                3     24    8.12   1.121  0.340    
disciplinary.failure   1      5    5.48   0.757  0.384    
education              3      3    1.12   0.155  0.926    
social.smoker          1      8    7.64   1.056  0.305    
social.drinker         1     15   15.33   2.118  0.146    
Residuals            687   4972    7.24                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. There are several methods of doing that. Below we have used *Random Forests* to perform features selection.




```
> varImp(random_model)
                          Overall
id                      14.886564
reason.of.absence       47.040443
month.of.absence         6.619975
day.of.the.week          1.687629
seasons                  3.459802
transportation.expenses 12.802269
distance.residence.work  7.372732
service.time             8.484226
age                      8.823340
work.load.average        5.584042
hit.target               4.497327
disciplinary.failure    21.930673
education                4.133727
son                      9.672532
social.drinker           4.436318
social.smoker            4.414074
pet                      6.888809
weight                   8.525702
height                  12.181620
body.mass.index          7.645196
> 
```

Finally, we have removed (id,education,day.of.the.week, pet,hit.target, seasons,social.smoker, social.drinker,weight). After performing correlation analysis, we also performed some hit and trail to increase the r square value and decrease the rsme value.

2.1.4 Normalization

Normalization is the process of reducing unwanted variation either within or between variables. Normalization is done to bring all the variables into proportion with one another. After performing normalization, the range of data is between 0 to 1. Normalization is sensitive to outlier therefor it is performed after outlier analysis. Moreover, most analysis like regression, require the data to be normally distributed. We can visualize whether the data is normalized by using histogram plot and normalization plot.

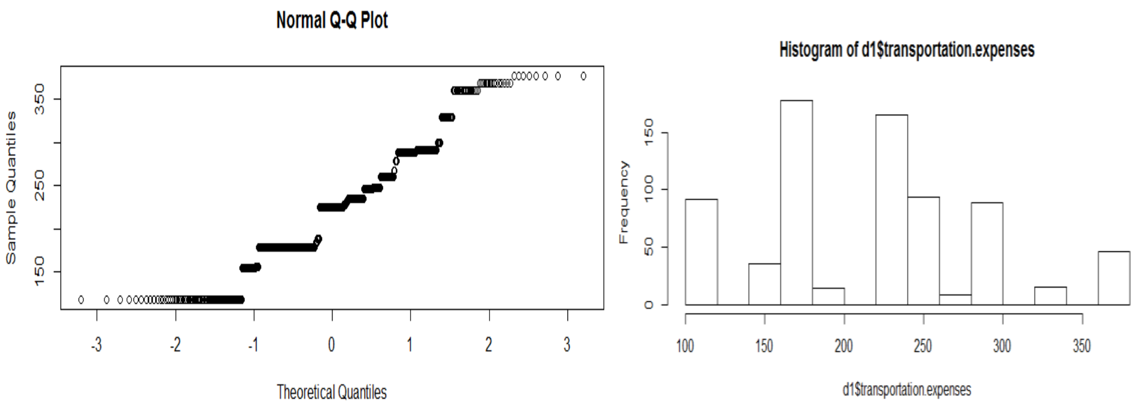


Fig:normalization and histogram plot of transportation expenses

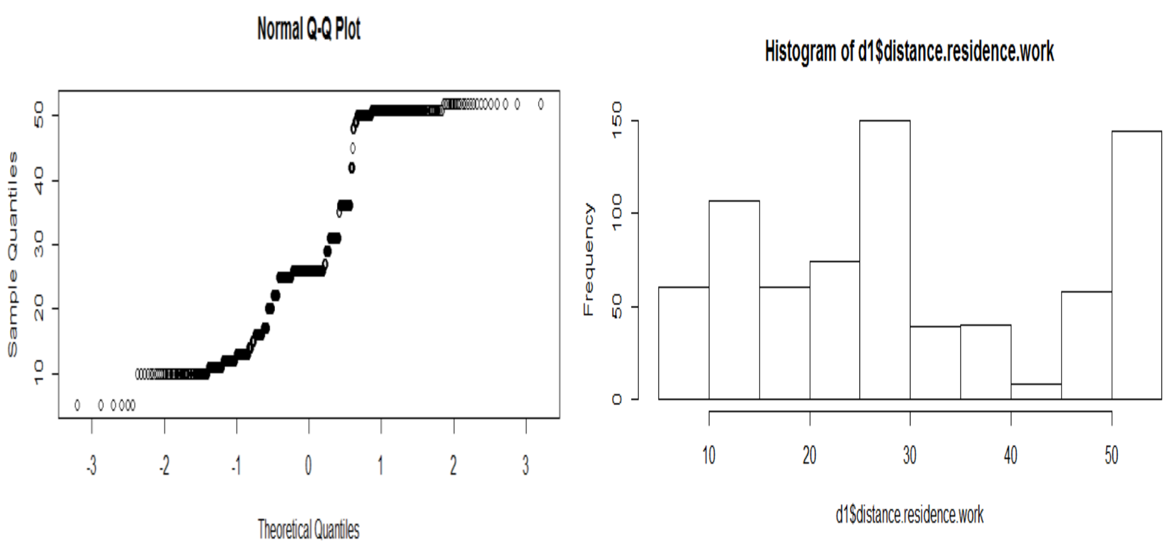


Fig: normalization and histogram plot for distance from residence to work

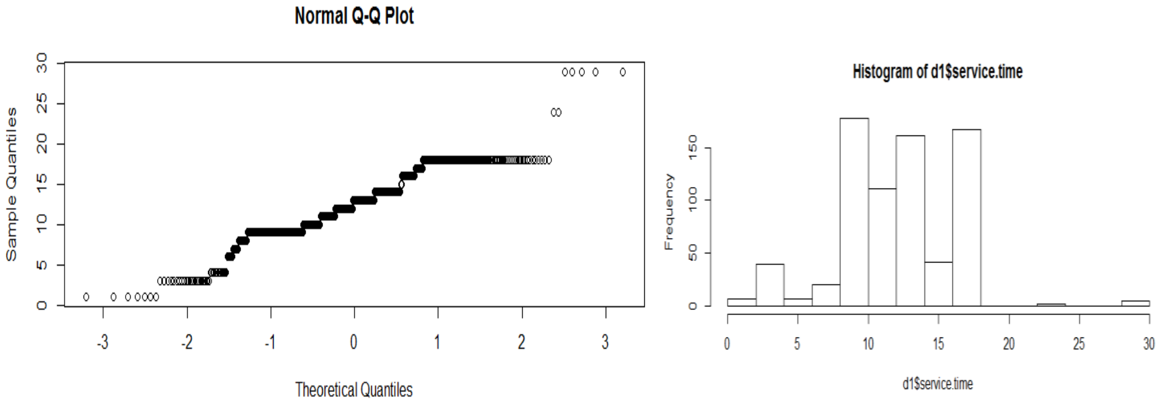


Fig:normalization and histogram plot service time

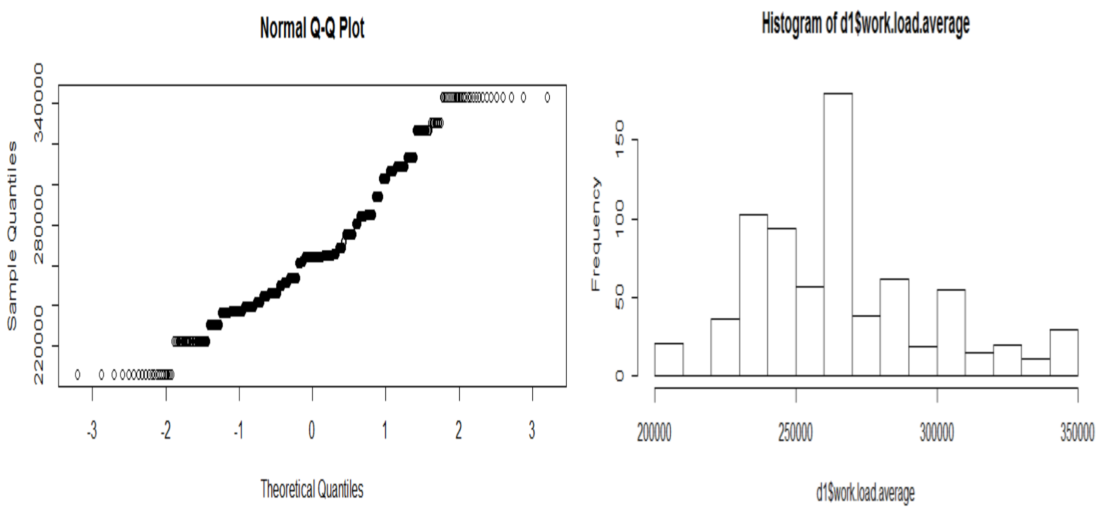
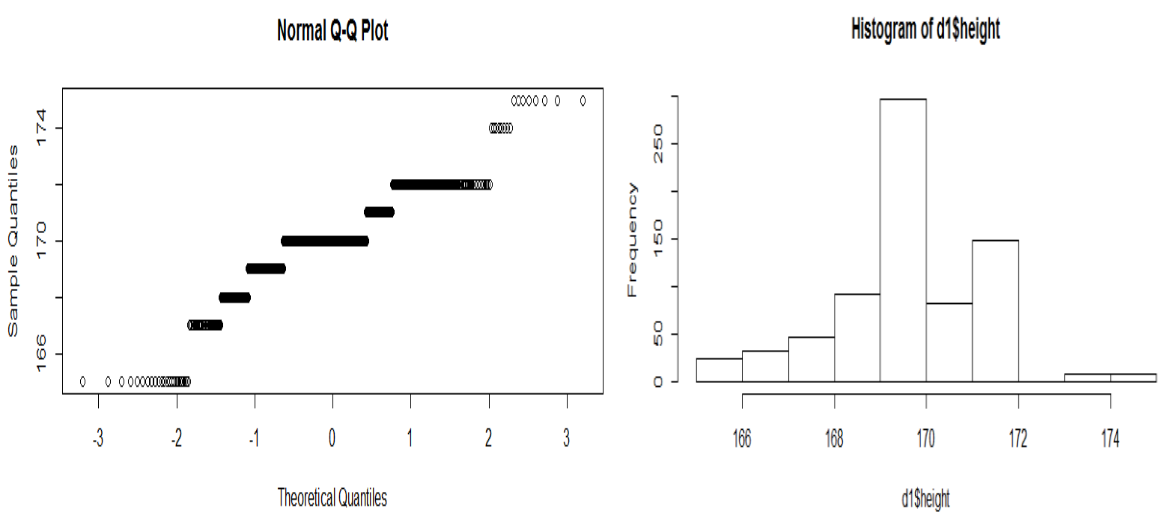


Fig: normalization and histogram plot for work load average



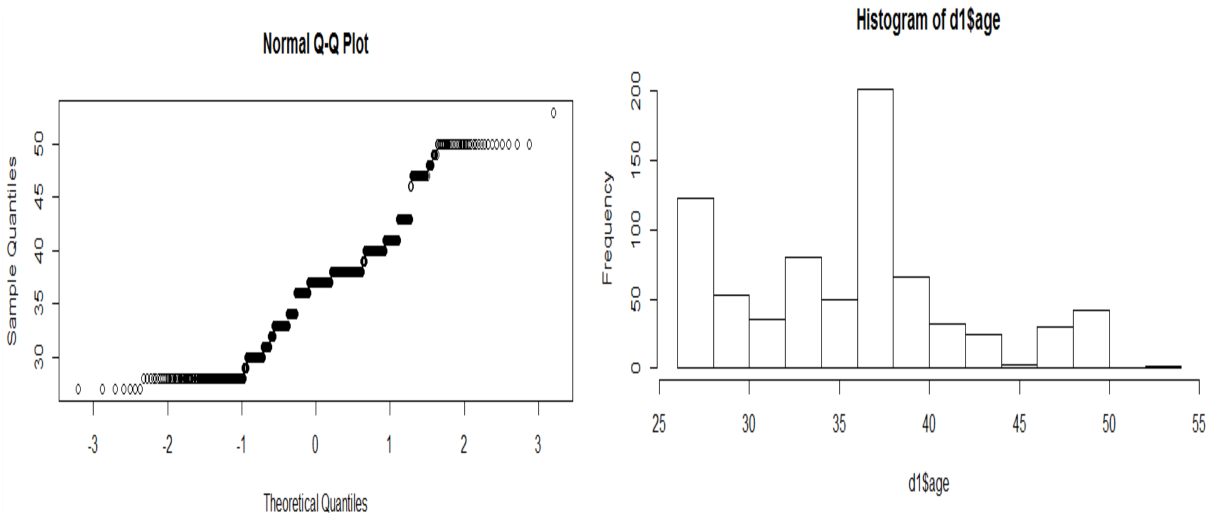


Fig:normalization and histogram plot Age

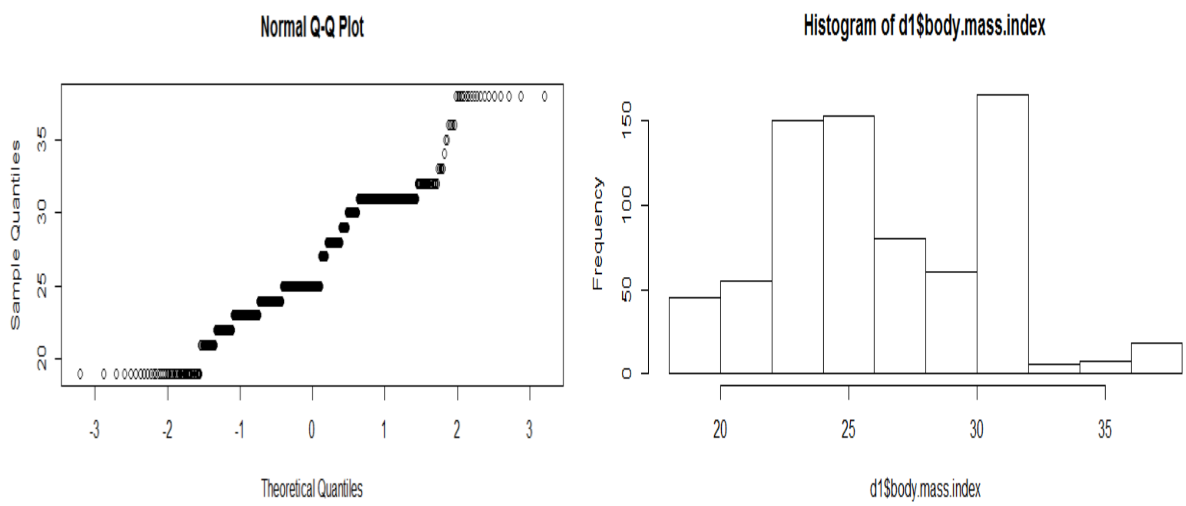


Fig: normalization and histogram plot for body mass index

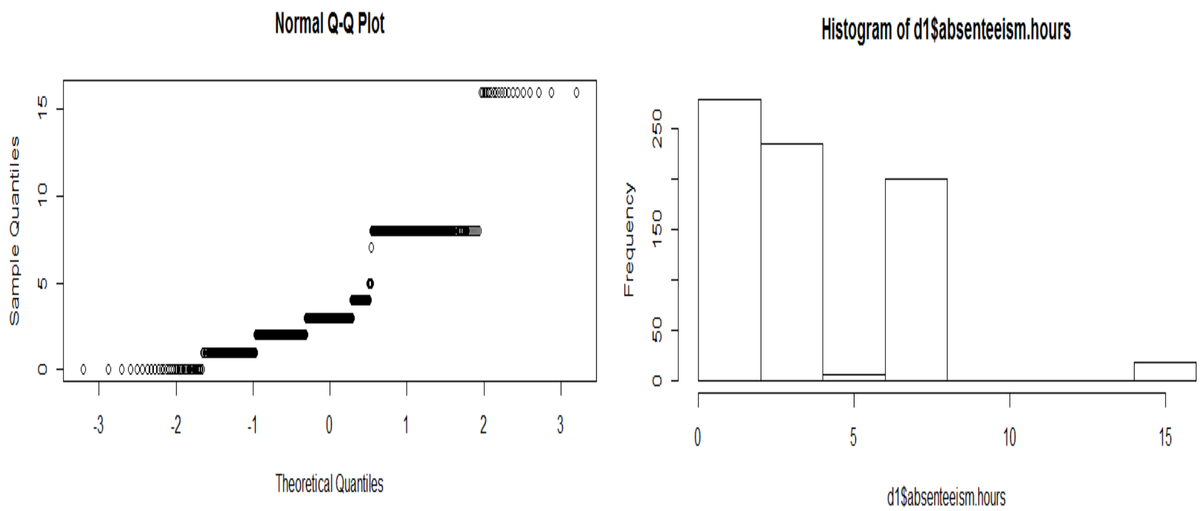


Fig: normalization and histogram plot for absenteeism in hours

From the plot we can find that the data is normalized and there is a huge difference between the ranges of data that variables contains. For example:-the range of age is below 100 but for transportation expenses the range of data is very high. Therefore we have to normalize the data

2.2 Modeling

2.2.1 Model Selection

The dependent variable can fall in either of the four categories:

1. Nominal
2. Ordinal
3. Interval
4. Ratio

If the dependent variable, in our case *absenteeism in hours*, is Interval or Ratio the normal method is to do a **Regression** analysis, or classification after binning. You always start your model building from the simplest to more complex. Therefore we use Multiple Linear Regression.

2.2.2 Multiple linear regression

```
> summary(lm_model)

Call:
lm(formula = absenteeism.hours ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-19.054  -4.861  -1.601   0.995 107.069 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)             27.07776    5.52402   4.902 1.23e-06 ***
reason.of.absence       -0.49037    0.08397  -5.840 8.73e-09 ***
month.of.absence         0.03098    0.16102   0.192 0.847501    
day.of.the.week         -0.78476    0.40627  -1.932 0.053895 .  
transportation.expenses  3.01729    2.62546   1.149 0.250930    
distance.residence.work -8.28146    2.12080  -3.905 0.000105 ***
service.time             5.65143    4.67768   1.208 0.227476    
age                     -1.93125    3.45945  -0.558 0.576888    
work.load.average       -1.08421    2.44543  -0.443 0.657670    
disciplinary.failure   -14.25730    2.88150  -4.948 9.85e-07 ***
social.drinker           5.33883    1.49506   3.571 0.000385 ***
social.smoker            1.53252    2.24638   0.682 0.495375    
weight                  -4.74160    2.91811  -1.625 0.104730    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.17 on 579 degrees of freedom
Multiple R-squared:  0.1131,	Adjusted R-squared:  0.09472 
F-statistic: 6.153 on 12 and 579 DF,  p-value: 3.081e-10
```

As you can see the *Adjusted R-squared* value, we can explain only about 11% of the data using our multiple linear regression model. This is not very impressive, but at least looking at the *F-statistic* and combined p-value we can reject the null hypothesis that target variable does not depend on any of the predictor variables.
Looking at the significance values of some of the predictor we can see that there is some scope of improvement in this model. We can improve this multiple linear regression model using *ANOVA*.

ANOVA TABLE

```
> summary(result)
                         Df Sum Sq Mean Sq F value   Pr(>F)    
reason.of.absence         1   4356    4356  27.251 2.33e-07 ***
month.of.absence          1      4       4   0.026  0.87281    
day.of.the.week           1   1255    1255   7.849  0.00522 ** 
transportation.expenses   1    123     123   0.769  0.38083    
distance.residence.work   1    763     763   4.774  0.02921 *  
service.time              1    741     741   4.636  0.03164 *  
age                       1    384     384   2.400  0.12173    
work.load.average         1      0       0   0.001  0.97148    
disciplinary.failure      1   4597    4597  28.762 1.10e-07 ***
social.drinker            1   1631    1631  10.204  0.00146 ** 
social.smoker             1    202     202   1.262  0.26167    
weight                    1    313     313   1.960  0.16195    
Residuals               727 116198     160                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> summary(linear_model)

Call:
lm(formula = absenteeism.hours ~ ., data = train_lm)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.9979 -1.9421 -0.7792  1.9051 13.1722 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)              1.204e+01  1.730e+00   6.959 9.32e-12 ***
reason.of.absence       -1.289e-01  1.909e-02  -6.753 3.53e-11 ***
month.of.absence         6.005e-03  3.660e-02   0.164  0.86973    
transportation.expenses  1.856e+00  6.201e-01   2.993  0.00288 ** 
distance.residence.work -3.152e-01  4.387e-01  -0.718  0.47275    
service.time            -2.958e-02  4.268e-02  -0.693  0.48864    
age                     -8.848e-01  7.987e-01  -1.108  0.26845    
work.load.average        7.263e-07  4.035e-06   0.180  0.85721    
disciplinary.failure    -6.779e+00  6.629e-01 -10.226  < 2e-16 ***
son                      1.562e+00  5.277e-01   2.960  0.00320 ** 
height                   1.453e+00  7.700e-01   1.887  0.05970 .  
body.mass.index          1.633e+00  7.203e-01   2.268  0.02371 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.966 on 580 degrees of freedom
Multiple R-squared:  0.2047,	Adjusted R-squared:  0.1896 
F-statistic: 13.57 on 11 and 580 DF,  p-value: < 2.2e-16
```

Using ANOVA we saw that many variable which we consider important contribute the least for the reduction of the fitting error of the model. However removing these variables also did not change the predictive power of our regression model.And therefore after many hit and trail and using anova we came up with new subset of variables which are listed above. Therefore, this is the maximum accuracy that we can get from this model.

2.2.3 Regression Trees

Now we will try and use a different regression model to predict our *Quality* target variable. We will use a regression tree to predict the values of our target variable.

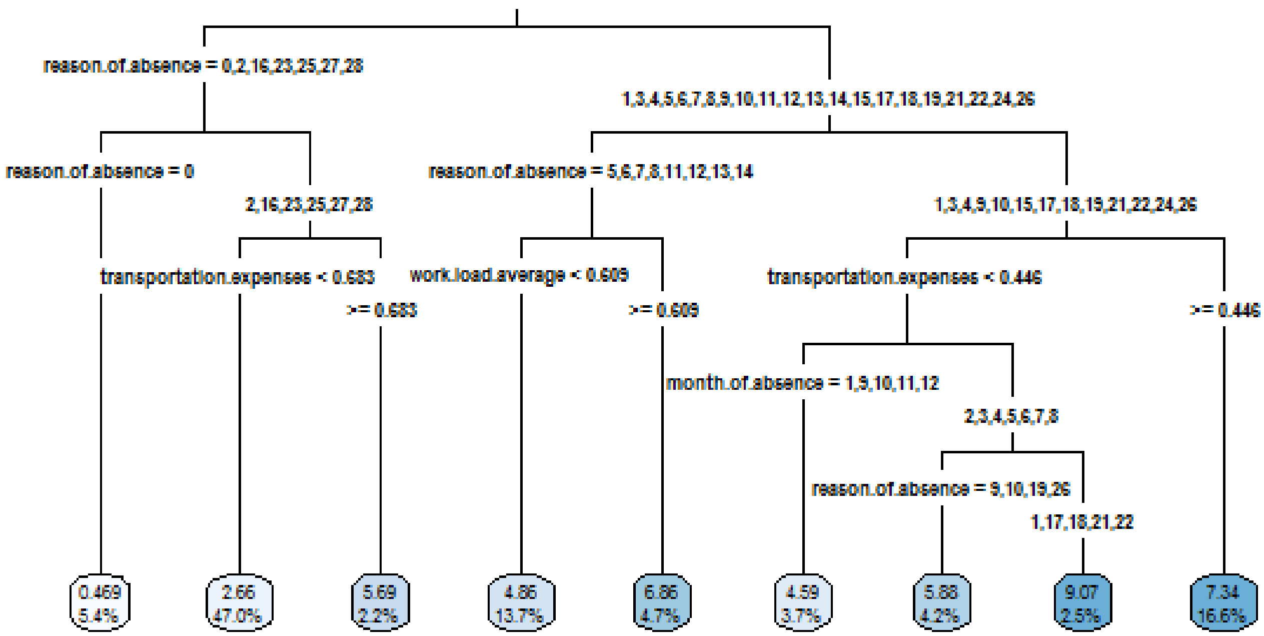


Fig: Regression tree

2.2.4 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set.

2.2.5 Decision tree

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

3.1 Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive Performance
2. Interpretability
3. Computational Efficiency

In our case of employee absenteeism, the latter two, *Interpretability* and *Computation Efficiency*, do not hold much significance. Therefore we will use *Predictive performance* as the criteria to compare and evaluate models. Predictive performance can be measured by comparing Predictions of the models with real values of the target variables, and calculating some average error measure.

3.1.1 Root mean square error (RMSE)

Root Mean Square Error (**RMSE**) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; **RMSE** is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit

RMSE value for decision tree is as follows:

```
RMSE(Error$absenteeism.hours)
```

```
[1] 2.788904
```

RMSE value randomforest is as follows

```
> RMSE(Error$absenteeism.hours)
[1] 2.835388
```

RMSE value for multiple linear regression

```
> RMSE(Error$absenteeism.hours)
[1] 2.708279
```

3.2 Model Selection

We can see that all models perform comparatively on average and therefore we can select either of the two models without any loss of information. But if we want less RMSE value then we should select multiple linear regression because it has lowest RMSE value.

Question and answer

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

Answer: The rules obtained by the decision tree regression we extracted rules which can help us to answer the above question

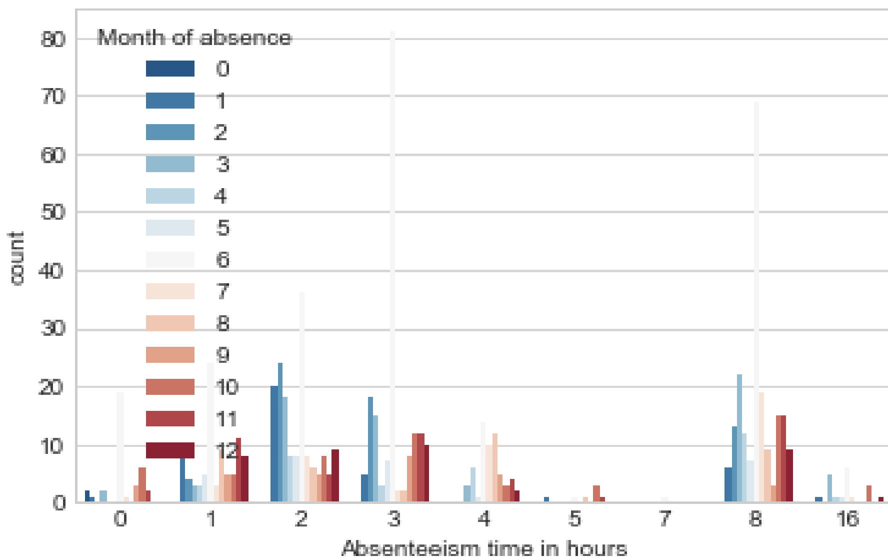
```
> dtree
n= 592

node), split, n, deviance, yval
      * denotes terminal node

 1) root 592 6092.36300 4.2347970
   2) reason.of.absence=0,2,12,16,23,25,27,28 331 1206.50200 2.6132930
     4) reason.of.absence=0 31   48.77419 0.6774194 *
     5) reason.of.absence=2,12,16,23,25,27,28 300 1029.54700 2.8133330
      10) transportation.expenses< 0.6826923 290  722.06900 2.6896550 *
      11) transportation.expenses>=0.6826923 10  174.40000 6.4000000 *
   3) reason.of.absence=1,3,4,5,6,7,8,9,10,11,13,14,15,17,18,19,21,22,24,26 261 2911.870
00 6.2911880
     6) reason.of.absence=4,7,8,9,10,11,13,14,21 127 1494.88200 5.4173230
      12) month.of.absence=4,6,8 32  210.96880 4.0312500 *
      13) month.of.absence=1,2,3,5,7,9,10,11,12 95 1201.72600 5.8842110 *
     7) reason.of.absence=1,3,5,6,15,17,18,19,22,24,26 134 1228.09000 7.1194030
      14) month.of.absence=1,2,4,5,6,9,11,12 74  435.63510 6.3918920 *
      15) month.of.absence=3,7,8,10 60  704.98330 8.0166670
        30) body.mass.index< 0.1842105 9   50.22222 5.4444440 *
        31) body.mass.index>=0.1842105 51  584.70590 8.4705880
          62) body.mass.index>=0.3684211 29  174.55170 7.3448280 *
          63) body.mass.index< 0.3684211 22  324.95450 9.9545450 *
>
```

From the decision tree rules that employee with reason 2,16,23,25,27,28 and normalized transportation expenses constitute large percentage of absenteeism. Company should take care the need of the employee with reason 2,16,23,25,27,28. Because due to this employees com[any could suffer huge loss.

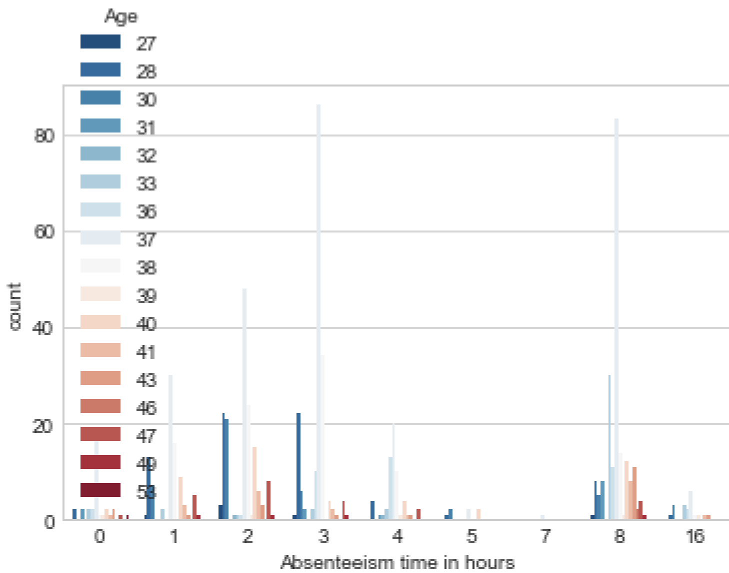
Workloss is directly proportional to service time and absenteeism time. So company should make such policies such that the product of service time and absenteeism hours is minimum. Company should provide health care facilities so that their employee remains fit.

Company should plan there working hours so that work load average per day is less because if the it will be high then employee will remain tensed during the working hours. Holidays should be given to employee during festive seasons.

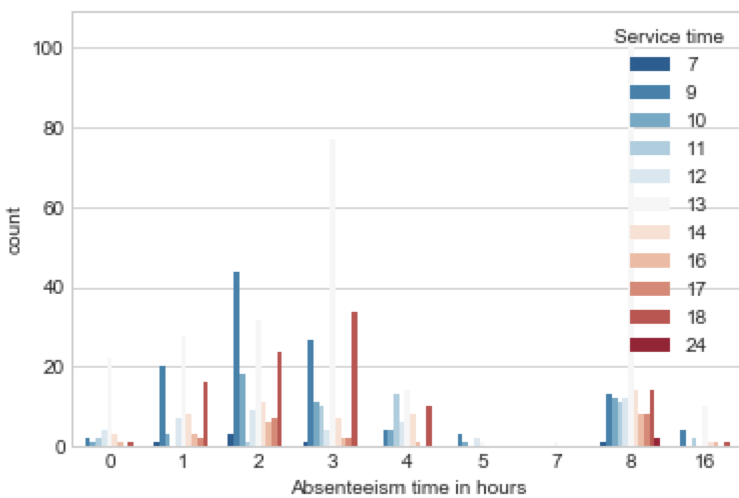
Here are some useful plot which can help us to bring out appropriate changes.



From this graph,we can see that the count of absenteeism is increasing in a particular pattern .in the month 1 and 2,absenteeism is maximum.Company should plan their working hours in suchway that theyhave less working hours in these months.



Age also plays a crucial role in absenteeism we can clearly see that employee of age group 33-38 are very uch involved in absenteeism.This could be due to various reasons.company should hire less number of employee who fall in this age group



Service time is also major factor in employee absenteeism. From the graph, we can clearly see that the main reason of employee absenteeism is long working hours. Employee having long working hours for eg: 7,9,or 11 hrs are deeply involved in absenteeism.

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

Answer:-

If the same trend of absenteeism continues we can project the losses every month that company would in 2011 using our predictive model. Losses are as follows:

	Work Load Loss/Month
No Absent	0
Janaury	3763559
Febraury	4341322
March	7850609
April	3678519
May	2305104
June	21597994
July	4780786
August	2763857
September	1786903
October	5855544
November	4593168
December	3383769

Appendix B - R Code

```
rm(list = ls())

#libraries
x <- c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "C50", "dummies", "e1071",
"Information",
      "MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine', 'inTrees','readxl')


#load packages(x)
lapply(x, require, character.only = TRUE)
rm(x)
setwd("E:/1ST SEM/eng/edwisor_assignments/7.project1")

#data
d1 <- read_excel("Absenteeism_at_work_Project.xls")

str(d1)


#missing value analysis
missing_value = data.frame(apply(d1,2,function(x){sum(is.na(x))}))
missing_value$Columns = row.names(missing_value)
names(missing_value)[2] =  "Variables"
row.names(missing_value)=NULL
names(missing_value)[1] =  "Missing_percentage"
missing_value$Missing_percentage = (missing_value$Missing_percentage/nrow(d1)) * 100
missing_value = missing_value[order(-missing_value$Missing_percentage),]
missing_value = missing_value[,c(2,1)]


#renaming variables
names(d1)[1]="id"
names(d1)[2]="reason.of.absence"
names(d1)[3]="month.of.absence"
names(d1)[4]="day.of.the.week"
```

```
names(d1)[5]="seasons"
names(d1)[6]="transportation.expenses"
names(d1)[7]="distance.residence.work"
names(d1)[8] = "service.time"
names(d1)[9]="age"
names(d1)[10]="work.load.average"
names(d1)[11]="hit.target"
names(d1)[12]="disciplinary.failure"
names(d1)[13]="education"
names(d1)[14]="son"
names(d1)[15]="social.drinker"
names(d1)[16]="social.smoker"
names(d1)[17]="pet"
names(d1)[18]="weight"
names(d1)[19]="height"
names(d1)[20]="body.mass.index"
names(d1)[21]="absenteeism.hours"



# missing values imputation

d1$reason.of.absence[is.na(d1$reason.of.absence)] = median(d1$reason.of.absence, na.rm = T)
d1$month.of.absence[is.na(d1$month.of.absence)] = median(d1$month.of.absence, na.rm = T)
d1$disciplinary.failure[is.na(d1$disciplinary.failure)] = median(d1$disciplinary.failure, na.rm = T)
d1$education[is.na(d1$education)] = median(d1$education, na.rm = T)
d1$social.drinker[is.na(d1$social.drinker)] = median(d1$social.drinker, na.rm = T)
d1$social.smoker[is.na(d1$social.smoker)] = median(d1$social.smoker, na.rm = T)
d1$transportation.expenses[is.na(d1$transportation.expenses)] =
median.default(d1$transportation.expenses, na.rm = T)
d1$distance.residence.work[is.na(d1$distance.residence.work)] = median(d1$distance.residence.work,
na.rm = T)
d1$service.time[is.na(d1$service.time)] = median(d1$service.time, na.rm = T)
d1$age[is.na(d1$age)] = median(d1$age, na.rm = T)
d1$work.load.average[is.na(d1$work.load.average)] = median(d1$work.load.average, na.rm = T)
d1$hit.target[is.na(d1$hit.target)] = median(d1$hit.target, na.rm = T)
d1$son[is.na(d1$son)] = median(d1$son, na.rm = T)
d1$pet[is.na(d1$pet)] = median(d1$pet, na.rm = T)
d1$weight[is.na(d1$weight)] = median(d1$weight, na.rm = T)
d1$height[is.na(d1$height)] = median(d1$height, na.rm = T)
d1$body.mass.index[is.na(d1$body.mass.index)] = median(d1$body.mass.index, na.rm = T)
d1$absenteeism.hours[is.na(d1$absenteeism.hours)] = median(d1$absenteeism.hours, na.rm = T)


#converting variables to their types

d1$reason.of.absence=as.factor(d1$reason.of.absence)
d1$month.of.absence = as.factor(d1$month.of.absence)
d1$day.of.the.week = as.factor(d1$day.of.the.week)
```

```
d1$seasons = as.factor(d1$seasons)
d1$disciplinary.failure = as.factor(d1$disciplinary.failure)



d1$education = as.factor(d1$education)
d1$social.drinker = as.factor(d1$social.drinker)
d1$social.smoker= as.factor(d1$social.smoker)



df= d1
#d1 = df
numeric_index = sapply(d1,is.numeric) #selecting only numeric
numerical = d1[,numeric_index]
Numerical = colnames(numerical)

#creating box plot
for (i in 1:length(Numerical))
{
  assign(paste0("gn",i), ggplot(aes_string(y = (Numerical[i]), x = "absenteeism.hours"), data =
subset(d1))+
        stat_boxplot(geom = "errorbar", width = 0.5) +
        geom_boxplot(outlier.colour="red", fill = "blue" ,outlier.shape=18,
                outlier.size=1, notch=FALSE) +
        theme(legend.position="bottom")+
        labs(y=Numerical[i],x="absenteeism.hours")+
        ggtitle(paste("Box plot for",Numerical[i])))
}

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)


#outlier imputation using NA technique
Out = d1$transportation.expenses[d1$transportation.expenses %in%
boxplot.stats(d1$transportation.expenses)$out]
d1$transportation.expenses[(d1$transportation.expenses %in% Out)] = NA
d1$transportation.expenses[is.na(d1$transportation.expenses)] =
median.default(d1$transportation.expenses, na.rm = T)

Out = d1$distance.residence.work[d1$distance.residence.work %in%
boxplot.stats(d1$distance.residence.work)$out]
d1$distance.residence.work[(d1$distance.residence.work %in% Out)] = NA
d1$distance.residence.work[is.na(d1$distance.residence.work)] = median(d1$distance.residence.work,
na.rm = T)

Out = d1$age[d1$age %in% boxplot.stats(d1$age)$out]
d1$age[(d1$age %in% Out)] = NA
```

```
d1$age[is.na(d1$age)] = median(d1$age, na.rm = T)

Out = d1$ work.load.average [d1$ work.load.average  %in% boxplot.stats(d1$ work.load.average )$out]
d1$ work.load.average [(d1$ work.load.average  %in% Out)] = NA
d1$ work.load.average [is.na(d1$ work.load.average )] = median(d1$ work.load.average , na.rm = T)


Out = d1$hit.target[d1$hit.target %in% boxplot.stats(d1$hit.target)$out]
d1$hit.target[(d1$hit.target %in% Out)] = NA
d1$hit.target[is.na(d1$hit.target)] = median(d1$hit.target, na.rm = T)

Out = d1$pet[d1$pet %in% boxplot.stats(d1$pet)$out]
d1$pet[(d1$pet %in% Out)] = NA
d1$pet[is.na(d1$pet)] = median(d1$pet, na.rm = T)

Out = d1$height[d1$height %in% boxplot.stats(d1$height)$out]
d1$height[(d1$height %in% Out)] = NA
d1$height[is.na(d1$height)] = median(d1$height, na.rm = T)

Out = d1$absenteeism.hours[d1$absenteeism.hours %in% boxplot.stats(d1$absenteeism.hours)$out]
d1$absenteeism.hours[(d1$absenteeism.hours %in% Out)] = NA
d1$absenteeism.hours[is.na(d1$absenteeism.hours)] = median(d1$absenteeism.hours, na.rm = T)

rm(Out)
#rm(df)


#correlation plot
corrgram(d1, order = F,
        upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")



str(d1)

d2 = d1
d1 = d2




#finding the impotant variable using importance graph

library(randomForest)

random_model = randomForest(absenteeism.hours~.,data = d1 , importance= TRUE,ntree = 500)
print(random_model)
attributes(random_model)
varUsed(random_model) # to find which variables used in random forest
varImpPlot(random_model,sort = TRUE,n.var = 10,main=" Importance graph")
varImp(random_model)

# anova for categorical data
```

```
str(d2)
library(ANOVA.TFNs)
library(ANOVAreplication)
result =

aov(formula=absenteeism.hours~reason.of.absence+month.of.absence+day.of.the.week+seasons+discip
linary.failure
           +education+social.smoker+social.drinker, data = d1)
summary(result)


#library(rpart.plot)
#tree <- rpart(absenteeism.hours ~ . , method='class', data = d1)
#printcp(tree)
#plot(tree, uniform=TRUE, main="Main Title")
#text(tree, use.n=TRUE, all=TRUE)
#prp(tree)


d1 = subset(d1, select = -c(id,education,day.of.the.week, pet,hit.target, seasons,social.smoker,
                      social.drinker,weight))
str(d1)



#histogram and normalization plot

qqnorm(d1$transportation.expenses)
hist(d1$transportation.expenses)
qqnorm(d1$distance.residence.work)
hist(d1$distance.residence.work)
qqnorm(d1$service.time)
hist(d1$service.time)
qqnorm(d1$work.load.average)
hist(d1$work.load.average)
qqnorm(d1$height)
hist(d1$height)
qqnorm(d1$age)
hist(d1$age)
qnorm(d1$body.mass.index)
hist(d1$body.mass.index)
qqnorm(d1$absenteeism.hours)
hist(d1$absenteeism.hours)




Numerical_Col =
c("transportation.expenses","age","son","height","body.mass.index","distance.residence.work","work.load.
average",
             "distance.residence.work")

#nomalization
```

```
for(i in Numerical_Col){
  print(i)
  d1[,i] = (d1[,i] - min(d1[,i]))/

(max(d1[,i] - min(d1[,i])))
}

rmExcept(c("d2","d1"))

sample = sample(1:nrow(d1), 0.8 * nrow(d1))
train = d1[sample,]
test = d1[-sample,]

#calculation of rmse

library(caTools)
library(mltools)
RMSE <- function(Error)
{
  sqrt(mean(Error^2))
}

dtree = rpart(absenteeism.hours~.,data = train, method = "anova")
dtree.plt = rpart.plot(dtree,type = 3,digits = 3,fallen.leaves = TRUE)
prediction_dtree = predict(dtree,test[,-12])
actual = test[,12]
predicted_dtree = data.frame(prediction_dtree)
Error = actual - predicted_dtree
RMSE(Error$absenteeism.hours)


# Random forest
random_model = randomForest(absenteeism.hours~.,train,importance = TRUE,ntree = 100)
rand_pred = predict(random_model,test[-12])
actual = test[,12]
predicted_rand = data.frame(rand_pred)
Error = actual - predicted_rand
RMSE(Error$absenteeism.hours)


d1$reason.of.absence=as.numeric(d1$reason.of.absence)
d1$month.of.absence = as.numeric(d1$month.of.absence)
d1$disciplinary.failure= as.numeric(d1$disciplinary.failure)
str(d1)

#droplevels(d1$Reason.for.absence)
sample_lm = sample(1:nrow(d1),0.8*nrow(d1))
train_lm = d1[sample_lm,]
test_lm = d1[-sample_lm,]
linear_model = lm(absenteeism.hours~.,data = train_lm)
summary(linear_model)
vif(linear_model)
```

```
#linear regression
predictions_lm = predict(linear_model,test_lm[,1:11])
Predicted_LM = data.frame(predictions_lm)


Actual = test_lm[,12]
Error = Actual - Predicted_LM
RMSE(Error$absenteeism.hours)



write.csv(d2,"dk.csv",row.names=F)

#part 2

DFF = subset(d2, select = c(month.of.absence,service.time,absenteeism.hours,
                            work.load.average))

DFF["Loss"]=with(DFF,((DFF[,4]*DFF[,3])/DFF[,2]))

for(i in 1:12)
{
  LOSS=DFF[which(DFF["month.of.absence"]==i),]
  print(data.frame(sum(LOSS$Loss)))

}
```

Appendix C – Python code

```
#Load libraries
import os
import pandas as pd
import numpy as np
from fancyimpute import KNN
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency
import seaborn as sns
from random import randrange, uniform

os.chdir("E:/1ST SEM/eng/edwisor_assignments/7.project1")
d1 = pd.read_excel('Absenteeism_at_work_Project.xls')
d1.head()


d2 = d1.copy()

#missing value analysis
```

```
d2['Reason for absence'] = d2['Reason for absence'].fillna(d2['Reason for absence'].median())
d2['Month of absence'] = d2['Month of absence'].fillna(d2['Month of absence'].median())
d2['Transportation expense'] = d2['Transportation expense'].fillna(d2['Transportation expense'].median())
d2['Distance from Residence to Work'] = d2['Distance from Residence to Work'].fillna(d2['Distance from
Residence to Work'].median())
d2['Service time']= d2['Service time'].fillna(d2['Service time'].median())
d2['Age'] = d2['Age'].fillna(d2['Age'].median())
d2['Work load Average/day ']= d2['Work load Average/day '].fillna(d2['Work load Average/day '].median())
d2['Hit target']= d2['Hit target'].fillna(d2['Hit target'].median())
d2['Disciplinary failure']= d2['Disciplinary failure'].fillna(d2['Disciplinary failure'].median())
d2['Education']= d2['Education'].fillna(d2['Education'].median())
d2['Son']= d2['Son'].fillna(d2['Son'].median())
d2['Social drinker']= d2['Social drinker'].fillna(d2['Social drinker'].median())
d2['Social smoker']= d2['Social smoker'].fillna(d2['Social smoker'].median())
d2['Pet']= d2['Pet'].fillna(d2['Pet'].median())
d2['Weight']= d2['Weight'].fillna(d2['Weight'].median())
d2['Height']= d2['Height'].fillna(d2['Height'].median())
d2['Body mass index']= d2['Body mass index'].fillna(d2['Body mass index'].median())
d2['Absenteeism time in hours']= d2['Absenteeism time in hours'].fillna(d2['Absenteeism time in
hours'].median())

d3=d2

#converting dataframe into numeric

d2['Reason for absence'] = d2['Reason for absence'].astype(int)
d2['Month of absence'] = d2['Month of absence'].astype(int)
d2['Transportation expense'] = d2['Transportation expense'].astype(int)
d2['Distance from Residence to Work'] = d2['Distance from Residence to Work'].astype(int)
d2['Service time'] = d2['Service time'].astype(int)
d2['Age'] = d2['Age'].astype(int)
d2['Work load Average/day '] = d2['Work load Average/day '].astype(int)
d2['Hit target'] = d2['Hit target'].astype(int)
d2['Disciplinary failure'] =d2['Disciplinary failure'].astype(int)
d2['Education'] = d2['Education'].astype(int)
d2['Son'] = d2['Son'].astype(int)
d2['Age'] = d2['Age'].astype(int)
d2['Social drinker'] = d2['Social drinker'].astype(int)
d2['Social smoker'] = d2['Social smoker'].astype(int)
d2['Pet'] = d2['Pet'].astype(int)
d2['Weight'] = d2['Weight'].astype(int)
d2['Height'] = d2['Height'].astype(int)
d2['Body mass index'] = d2['Body mass index'].astype(int)
d2['Absenteeism time in hours'] = d2['Absenteeism time in hours'].astype(int)
#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(d2['Transportation expense'], [75 ,25])
#Calculate_IQR
iqr = q75 - q25
#Calculate inner and outer fence
mini = q25 - (iqr*1.5)
maxi = q75 + (iqr*1.5)
```

```
#Replace with NA
d2.loc[d2['Transportation expense'] < mini,:'Transportation expense'] = np.nan
d2.loc[d2['Transportation expense'] > maxi,:'Transportation expense'] = np.nan


#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(d2['Month of absence'], [75 ,25])

#Calculate IQR
iqr = q75 - q25

#Calculate inner and outer fence
mini = q25 - (iqr*1.5)
maxi = q75 + (iqr*1.5)

#Replace with NA
d2.loc[d2['Month of absence'] < mini,:'Month of absence'] = np.nan
d2.loc[d2['Month of absence'] > maxi,:'Month of absence'] = np.nan


# In[ ]:


#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(d2['Distance from Residence to Work'], [75 ,25])

#Calculate IQR
iqr = q75 - q25

#Calculate inner and outer fence
mini = q25 - (iqr*1.5)
maxi = q75 + (iqr*1.5)

#Replace with NA
d2.loc[d2['Distance from Residence to Work'] < mini,:'Distance from Residence to Work'] = np.nan
d2.loc[d2['Distance from Residence to Work'] > maxi,:'Distance from Residence to Work'] = np.nan


# In[ ]:


#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(d2['Service time'], [75 ,25])

#Calculate IQR
iqr = q75 - q25

#Calculate inner and outer fence
mini = q25 - (iqr*1.5)
maxi = q75 + (iqr*1.5)
```

```
#Replace with NA
d2.loc[d2['Service time'] < mini,:'Service time'] = np.nan
d2.loc[d2['Service time'] > maxi,:'Service time'] = np.nan


# In[ ]:


#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(d2['Age'], [75 ,25])

#Calculate IQR
iqr = q75 - q25

#Calculate inner and outer fence
mini = q25 - (iqr*1.5)
maxi = q75 + (iqr*1.5)

#Replace with NA
d2.loc[d2['Age'] < mini,:'Age'] = np.nan
d2.loc[d2['Age'] > maxi,:'Age'] = np.nan


# In[ ]:


#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(d2['Work load Average/day '], [75 ,25])

#Calculate IQR
iqr = q75 - q25

#Calculate inner and outer fence
mini = q25 - (iqr*1.5)
maxi = q75 + (iqr*1.5)

#Replace with NA
d2.loc[d2['Work load Average/day '] < mini,:'Work load Average/day '] = np.nan
d2.loc[d2['Work load Average/day '] > maxi,:'Work load Average/day '] = np.nan


# In[ ]:


#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(d2['Hit target'], [75 ,25])

#Calculate IQR
iqr = q75 - q25
```

```
#Calculate inner and outer fence
mini = q25 - (iqr*1.5)
maxi = q75 + (iqr*1.5)

#Replace with NA
d2.loc[d2['Hit target'] < mini,:'Hit target'] = np.nan
d2.loc[d2['Hit target'] > maxi,:'Hit target'] = np.nan


# In[ ]:


#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(d2['Son'], [75 ,25])

#Calculate IQR
iqr = q75 - q25

#Calculate inner and outer fence
mini = q25 - (iqr*1.5)
maxi = q75 + (iqr*1.5)

#Replace with NA
d2.loc[d2['Son'] < mini,:'Son'] = np.nan
d2.loc[d2['Son'] > maxi,:'Son'] = np.nan


# In[ ]:


#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(d2['Pet'], [75 ,25])

#Calculate IQR
iqr = q75 - q25

#Calculate inner and outer fence
mini = q25 - (iqr*1.5)
maxi = q75 + (iqr*1.5)

#Replace with NA
d2.loc[d2['Pet'] < mini,:'Pet'] = np.nan
d2.loc[d2['Pet'] > maxi,:'Pet'] = np.nan


# In[ ]:


#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(d2['Height'], [75 ,25])
```

```
#Calculate IQR
iqr = q75 - q25

#Calculate inner and outer fence
mini = q25 - (iqr*1.5)
maxi = q75 + (iqr*1.5)

#Replace with NA
d2.loc[d2['Height'] < mini,:'Height'] = np.nan
d2.loc[d2['Height'] > maxi,:'Height'] = np.nan


# In[ ]:


#Detect and replace with NA
#Extract quartiles
qu75, qu25 = np.percentile(d2['Weight'], [75 ,25])

#Calculate IQR
iqr = q75 - q25

#Calculate inner and outer fence
mini = qu25 - (iqr*1.5)
maxi = qu75 + (iqr*1.5)

#Replace with NA
d2.loc[d2['Weight'] < mini,:'Weight'] = np.nan
d2.loc[d2['Weight'] > maxi,:'Weight'] = np.nan


# In[ ]:


#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(d2['Body mass index'], [75 ,25])

#Calculate IQR
iqr = q75 - q25

#Calculate inner and outer fence
mini = q25 - (iqr*1.5)
maxi = q75 + (iqr*1.5)

#Replace with NA
d2.loc[d2['Body mass index'] < mini,:'Body mass index'] = np.nan
d2.loc[d2['Body mass index'] > maxi,:'Body mass index'] = np.nan


#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(d2['Absenteeism time in hours'], [75 ,25])
```

```
#Calculate IQR
iqr = q75 - q25

#Calculate inner and outer fence
mini = q25 - (iqr*1.5)
maxi = q75 + (iqr*1.5)

#Replace with NA
d2.loc[d2['Absenteeism time in hours'] < mini,:'Absenteeism time in hours'] = np.nan
d2.loc[d2['Absenteeism time in hours'] > maxi,:'Absenteeism time in hours'] = np.nan


missing_val
#missing value imputation
d2['Reason for absence'] = d2['Reason for absence'].fillna(d2['Reason for absence'].median())
d2['Month of absence'] = d2['Month of absence'].fillna(d2['Month of absence'].median())
d2['Transportation expense'] = d2['Transportation expense'].fillna(d2['Transportation expense'].median())
d2['Distance from Residence to Work'] = d2['Distance from Residence to Work'].fillna(d2['Distance from
Residence to Work'].median())
d2['Service time']= d2['Service time'].fillna(d2['Service time'].median())
d2['Age'] = d2['Age'].fillna(d2['Age'].median())
d2['Work load Average/day ']= d2['Work load Average/day '].fillna(d2['Work load Average/day '].median())
d2['Hit target']= d2['Hit target'].fillna(d2['Hit target'].median())
d2['Disciplinary failure']= d2['Disciplinary failure'].fillna(d2['Disciplinary failure'].median())
d2['Education']= d2['Education'].fillna(d2['Education'].median())
d2['Son']= d2['Son'].fillna(d2['Son'].median())
d2['Social drinker']= d2['Social drinker'].fillna(d2['Social drinker'].median())
d2['Social smoker']= d2['Social smoker'].fillna(d2['Social smoker'].median())
d2['Pet']= d2['Pet'].fillna(d2['Pet'].median())
d2['Weight']= d2['Weight'].fillna(d2['Weight'].median())
d2['Height']= d2['Height'].fillna(d2['Height'].median())
d2['Body mass index']= d2['Body mass index'].fillna(d2['Body mass index'].median())
d2['Absenteeism time in hours']= d2['Absenteeism time in hours'].fillna(d2['Absenteeism time in
hours'].median())



d2['ID'] = d1['ID']
d2['Day of the week'] = d1['Day of the week']
d2['Seasons'] = d1['Seasons']


d2.info()

d2['ID'] = d2['ID'].astype('category')
d2['Reason for absence'] = d2['Reason for absence'].astype('category')
d2['Month of absence'] = d2['Month of absence'].astype('category')
d2['Day of the week'] = d2['Day of the week'].astype('category')
d2['Seasons'] = d2['Seasons'].astype('category')
d2['Disciplinary failure'] = d2['Disciplinary failure'].astype('category')
d2['Education'] = d2['Education'].astype('category')
```

```
d2['Social drinker'] = d2['Social drinker'].astype('category')
d2['Social smoker'] = d2['Social smoker'].astype('category')
cnames = ['Transportation expense', 'Distance from Residence to Work',
                'Service time', 'Age', 'Work load Average/day ', 'Hit target', 'Son',
                'Pet', 'Weight', 'Height', 'Body mass index']
#correlation plot
df_corr=d2.loc[:,cnames]

%matplotlib inline
#Set the width and hieght of the plot
f, ax = plt.subplots(figsize=(7, 5))

#generate correlation matrix
corr=df_corr.corr()

#Plot using seaborn library
sns.heatmap(corr, mask=np.zeros_like(corr, dtype=np.bool), cmap='rainbow',annot=True,
        square=True, ax=ax)


d2 = d2.drop(['ID','Day of the week','Seasons','Hit target','Education', 'Social drinker','Social smoker', 'Pet',
'Weight'], axis=1)


cnames=["Transportation expense", "Distance from Residence to Work",
                "Service time","Age","Work load Average/day ","Son",
                "Height","Body mass index","Absenteeism time in hours"]


#normalization
for i in cnames:
    print(i)
    d2[i]=(d2[i]-min(d2[i]))/(max(d2[i])-min(d2[i]))

from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeRegressor



#divide data into train and test
train,test=train_test_split(d2,test_size=0.2)


#Decision tree for regresion
fit=DecisionTreeRegressor(max_depth=2).fit(train.iloc[:,0:9],train.iloc[:,9])

#apply model on the test data
prediction_DT=fit.predict(test.iloc[:,0:9])

actual=test['Absenteeism time in hours']
predicted=pd.DataFrame(prediction_DT)
actual=pd.DataFrame(actual)
```

```
predicted["predicted"]=pd.DataFrame(prediction_DT)


#calculate rmse
#def rmse(predict, act):
#    return np.sqrt(((predict- act) ** 2).mean())
def rmse(predictions, targets):

    differences = predictions - targets                       #the DIFFERENCEs.

    differences_squared = differences ** 2                    #the SQUAREs of ^

    mean_of_differences_squared = differences_squared.mean()  #the MEAN of ^

    rmse_val = np.sqrt(mean_of_differences_squared)           #ROOT of ^

    return rmse_val


rmse(predicted["predicted"],actual["Absenteeism time in hours"])


train['Reason for absence']=train['Reason for absence'].astype(float)
train['Month of absence']=train['Month of absence'].astype(float)
train['Disciplinary failure']=train['Disciplinary failure'].astype(float)
train['Height']=train['Height'].astype(float)



#import libraries for LR
import statsmodels.api as sm

#Train the model using the training sets
model=sm.OLS(train.iloc[:,9],train.iloc[:,0:9]).fit()

#print out the statistics
model.summary()from sklearn.model_selection import train_test_split

X = d2.drop('Absenteeism time in hours',axis=1)
y = d2['Absenteeism time in hours']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
from sklearn.linear_model import LinearRegression


lm = LinearRegression()
lm.fit(X_train,y_train)


predictions = lm.predict(X_test)


%matplotlib inline
```

```
sns.distplot(d2['Absenteeism time in hours'])
plt.show()

coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df

plt.scatter(y_test,predictions)


sns.distplot((y_test-predictions),bins=50);

from sklearn import metrics
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))


from sklearn.tree import DecisionTreeRegressor
fit = DecisionTreeRegressor()
fit.fit(X_train,y_train)

prediction_dtree = fit.predict(X_test)


print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, prediction_dtree)))



from sklearn.ensemble import RandomForestRegressor

# Create a random forest Regressor
RFR = RandomForestRegressor(n_estimators=1000, random_state=0, n_jobs=-1)

# Train the classifier
RFR.fit(X_train, y_train)

prediction_RFR = RFR.predict(X_test)

print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, prediction_RFR)))
```

part 2

```
LOSS_DF = d3[['Month of absence','Absenteeism time in hours','Work load Average/day ','Service time']]

LOSS_DF["Loss"]=(LOSS_DF['Work load Average/day ']*LOSS_DF['Absenteeism time in
hours'])/LOSS_DF['Service time']


LOSS_DF["Loss"] = np.round(LOSS_DF["Loss"]).astype('int64')

NO = LOSS_DF[LOSS_DF['Month of absence'] == 0]['Loss'].sum()
Jan = LOSS_DF[LOSS_DF['Month of absence'] == 1]['Loss'].sum()
Feb = LOSS_DF[LOSS_DF['Month of absence'] == 2]['Loss'].sum()
Mar = LOSS_DF[LOSS_DF['Month of absence'] == 3]['Loss'].sum()
April =LOSS_DF[LOSS_DF['Month of absence'] == 4]['Loss'].sum()
```

```
may = LOSS_DF[LOSS_DF['Month of absence'] == 5]['Loss'].sum()
Jun = LOSS_DF[LOSS_DF['Month of absence'] == 6]['Loss'].sum()
```



```
Jul = LOSS_DF[LOSS_DF['Month of absence'] == 7]['Loss'].sum()
Aug = LOSS_DF[LOSS_DF['Month of absence'] == 8]['Loss'].sum()
Sep = LOSS_DF[LOSS_DF['Month of absence'] == 9]['Loss'].sum()
Oct = LOSS_DF[LOSS_DF['Month of absence'] == 10]['Loss'].sum()
Nov = LOSS_DF[LOSS_DF['Month of absence'] == 11]['Loss'].sum()
Dec = LOSS_DF[LOSS_DF['Month of absence'] == 12]['Loss'].sum()

data = {'No Absent': NO, 'Janaury': Jan,'Febraury': Feb,'March': Mar,
    'April': April, 'May': may,'June': Jun,'July': Jul,
    'August': Aug,'September': Sep,'October': Oct,'November': Nov,
    'December': Dec}

WorkLoss = pd.DataFrame.from_dict(data, orient='index')

WorkLoss.rename(index=str, columns={0: "Work Load Loss/Month"})

sns.set_style('whitegrid')
sns.countplot(x='Absenteeism time in hours',hue='month.of.absence',data=d3,palette='RdBu_r')
sns.distplot(d3['Age'],kde=True,color='darkred',bins=30)
sns.set_style('whitegrid')
sns.countplot(x='Absenteeism time in hours',hue='Age',data=Absent,palette='RdBu_r')

sns.set_style('whitegrid')
sns.countplot(x='Absenteeism time in hours',hue='Service time',data=Absent,palette='RdBu_r')
```

References:

https://en.wikipedia.org/wiki/Decision_tree
https://discuss.analyticsvidhya.com/
ieee paper on employee absenteeism