Predicting Employee Absenteeism

Vrushabhkumar S. Jain
3 March 2019

Contents

Chapter 1

Introduction

1.1Problem 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?

What is Employee Absenteeism?

Employee Absenteeism is the absence of an employee from work. Its a major problem faced by almost all employers of today. Employees are absent from work and thus the work suffers. Absenteeism of employees from work leads to back logs, piling of work and thus work delay. There are various laws been enacted for safeguarding the interest of both Employers and Employees but they too have various constraints.

Absenteeism is of two types -

- **Innocent absenteeism -** Is one in which the employee is absent from work due to genuine cause or reason. It may be due to his illness or personal family problem or any other real reason

- **Culpable Absenteeism -** Is one in which a person is absent from work without any genuine reason or cause. He may be pretending to be ill or just wanted a holiday and stay at home.

1.2 Data

We have dataset of employees information with 740 rows and 21 variable including target variable “Absenteeism.time.in.hours”; since our target variable is numeric, given problem is regression problem. Hence, our task is to build regression model which will predict employee absenteeism time based on given attributes. Below are the predictors with which we have to build the model:

1. Individual identification (ID)
2. Reason for absence (ICD).

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

I Certain infectious and parasitic diseases
II Neoplasms
III Diseases of the blood and blood-forming organs and certain disorders involving the

immune mechanism
IV Endocrine, nutritional and metabolic diseases
V Mental and behavioural disorders
VI Diseases of the nervous system
VII Diseases of the eye and adnexa
VIII Diseases of the ear and mastoid process
IX Diseases of the circulatory system
X Diseases of the respiratory system
XI Diseases of the digestive system
XII Diseases of the skin and subcutaneous tissue
XIII Diseases of the musculoskeletal system and connective tissue
XIV Diseases of the genitourinary system
XV Pregnancy, childbirth and the puerperium
XVI Certain conditions originating in the perinatal period
XVII Congenital malformations, deformations and chromosomal abnormalities
XVIII Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
XIX Injury, poisoning and certain other consequences of external causes
XX External causes of morbidity and mortality
XXI Factors influencing health status and contact with health services.
And 7 categories without (CID) 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)

From above predictors, we can categorize them as per their data types.

Categorical variables:

ID, Reason for absence, Month of absence, Day of the week, Seasons, disciplinary failure, Education, Social drinker, Social smoker,

Numerical variables:

Transportation expense, Distance from residence to work, Service time, Age, Work load Average/day, Hit target, Son, Pet, Weight, Height, Body mass index, Absenteeism time in hours

Now let's have a look at sample of the dataset that we are using to predict the employees absenteeism time

Table 1.1: Employee Absenteeism Sample Data (Columns:1:7)

ID	Reason.for. absence	Month.of. absence	Day.of.the. week	Seasons	Transportation. expense	Distance .from. Residence. to. Work
11	26	7	3	1	289	36
36	26	7	3	1	118	13
3	23	7	4	1	179	51
7	7	7	5	1	279	5
11	23	7	5	1	289	36

Table 1.2: Employee Absenteeism Sample Data (Columns:8:14)

Service.time	Age	Work.load.Average. day.	Hit. target	Disciplinary. failure	Education	Son
13	33	239554	97	0	1	2
18	50	239554	97	1	1	1
18	38	239554	97	0	1	0
14	39	239554	97	0	1	2
13	33	239554	97	0	1	2

Table 1.3: Employee Absenteeism Sample Data (Columns:15:21)

Social.drinker	Social.smoker	Pet	Weight	Height	Body.mass. index	Absenteeism.time.in. hours
1	0	1	90	172	30	4
1	0	0	98	172	31	0
1	0	0	89	170	31	2
1	1	0	68	168	24	4
1	0	1	90	172	30	2

Chapter 2

Methodology

2.1 Pre Processing

Data pre-processing 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 behaviours or trends, and is likely to contain many errors. Data pre-processing is a proven method of resolving such issues. Data pre-processing is crucial in any data mining process as they directly impact success rate of the project. This reduces complexity of the data under analysis as data in real world is unclean. Data is said to be unclean if it is missing attribute, attribute values, contain noise or outliers and duplicate or wrong data. Presence of any of these will degrade quality of the results. This is often called as Exploratory Data Analysis.

2.1.1 Data Visualization

Data visualization is the process of extracting and visualizing the data in a very clear and understandable way without any form of reading or writing by displaying the results in the form of pie charts, bar graphs, statistical representation and through graphical forms as well. In Data Visualization, the primary goal is to convey the information efficiently and clearly without any deviations or complexities in the form of statistical graphs, information graphs, and plots. In this process we will try to look at the relation between independent variable and target variable with the train dataset through visualization and will decide the further analysis according to it.

Histogram+kernel density estimation plot of numerical variable:

As we know regression analysis gives best result with normal distribution of continuous variables. We can visualize that in a glance by looking at the probability distributions or probability density functions of the continuous variable.
In **Figure 2.1**, we have plotted the probability density functions of all the employees attribute we have available in the data. The blue lines indicate Kernel Density Estimations (KDE) of the variable. The red lines represent the normal distribution. So as we can see in the figure most variables either very closely or somewhat imitate the normal distribution.

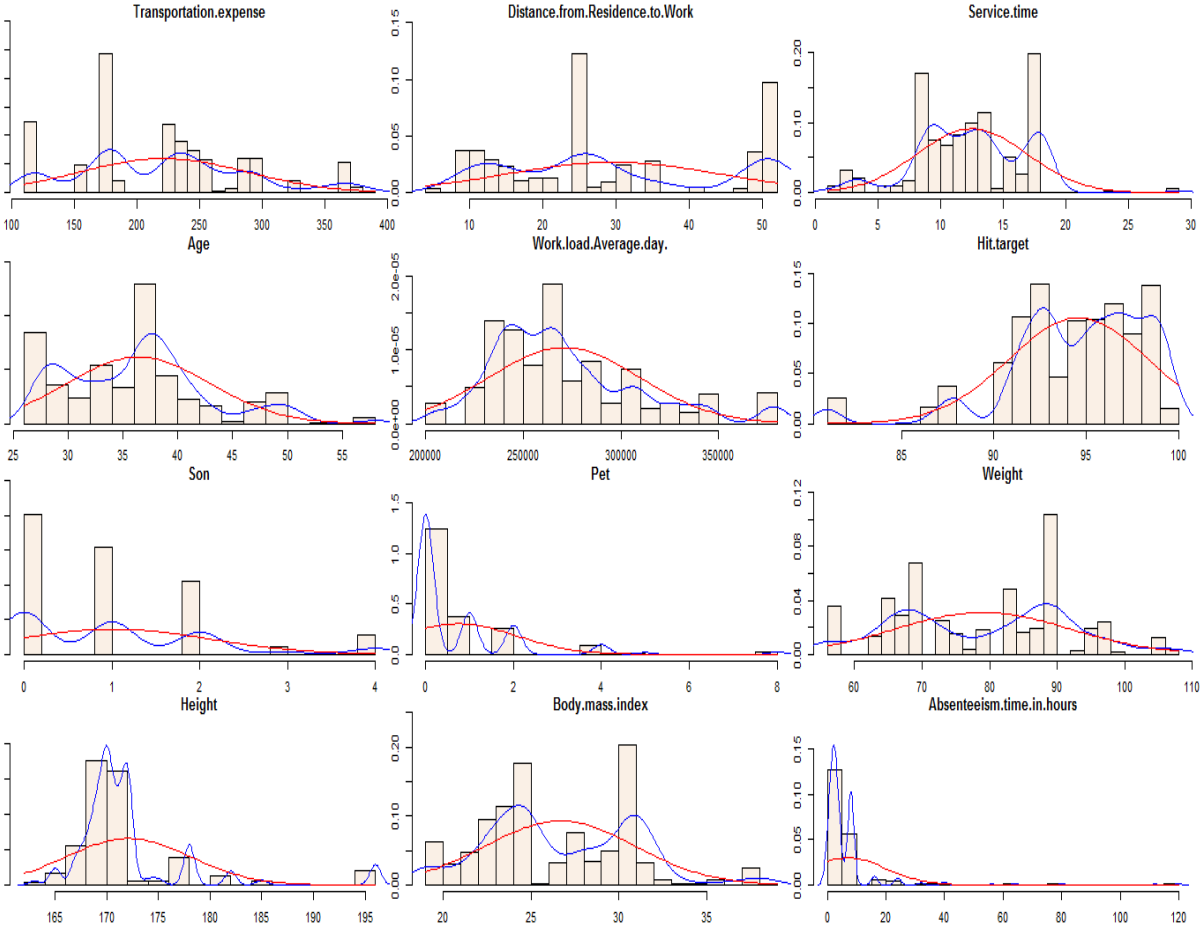
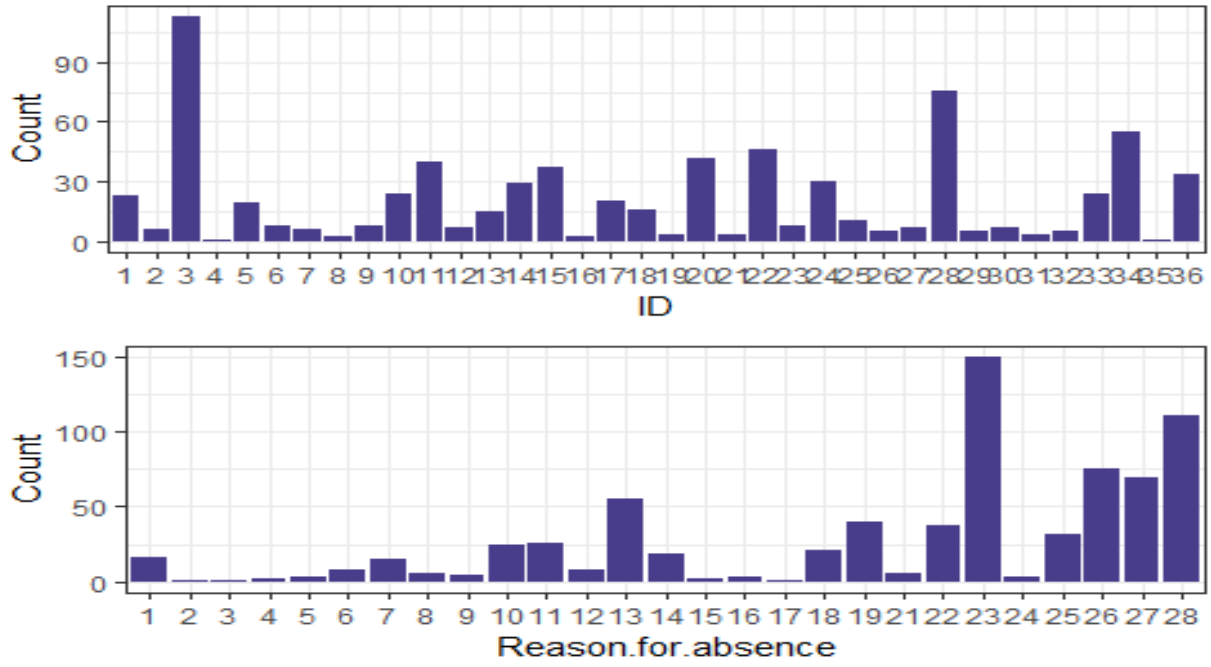


Fig2.1.Histogram+KDE plot of numerical variable

Categorical variable univariate visualization:

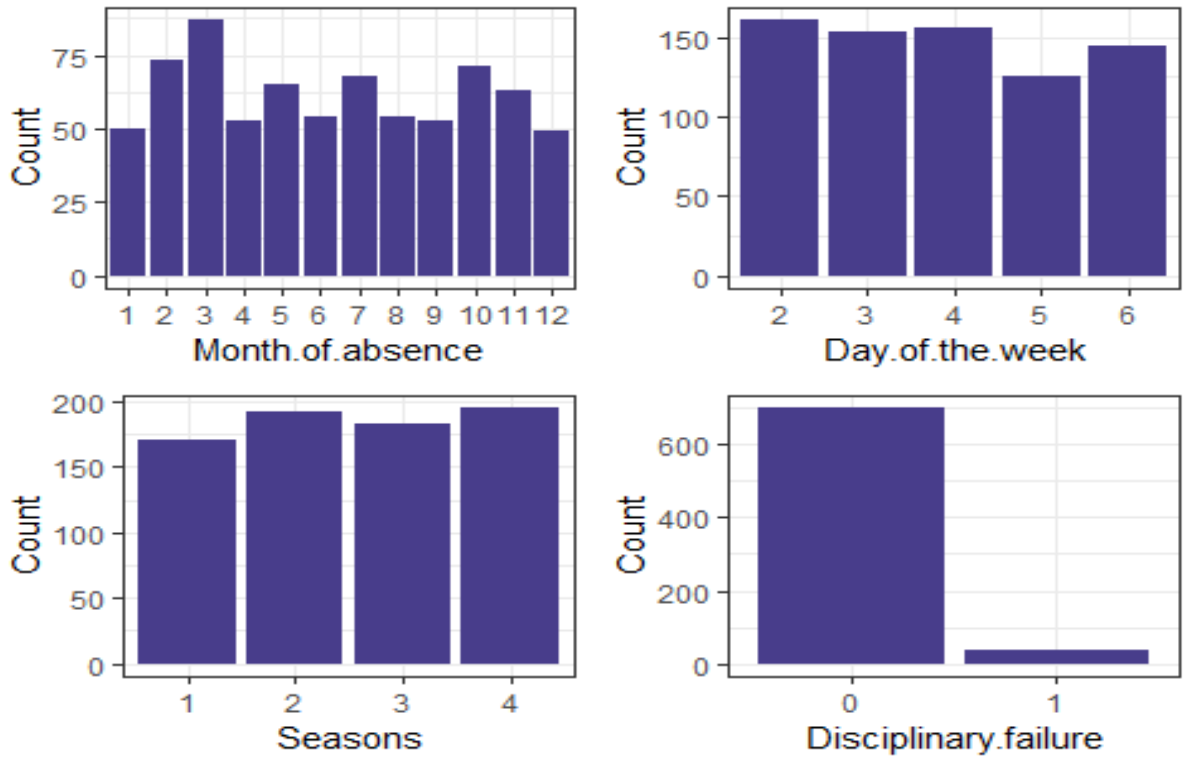
Fig.2.2 For Id and Reason for absence



From the above plots, we can see that,

- ID no. 3 is absent most of the times followed by ID no. 28 and 34
- Also most of the employees are absent for reason 23(**medical consultation)** followed by reason 28(**dental consultation**) , reason 26(**unjustified absence**) and reason 27(**physiotherapy**).

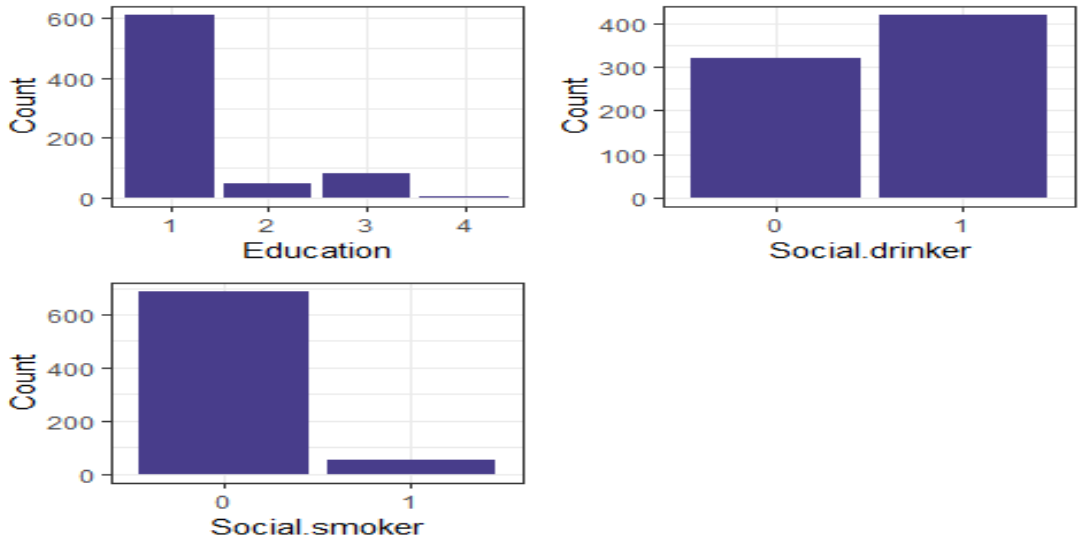
Fig2.3, For month of absence, day of the week, seasons and disciplinary failure



From the above plot we can conclude

- Most of the employees are absent in month 3(March) followed by 2 (**February**),10(**October**) and 11(**November**).
- Most of the employee are absent on 2(**Monday)** followed by 4(**Wednesday**) and 3(**Tuesday**).
- Also Most of the employee are absent in seasons 4(**spring)** followed by season 2(**Autumn**) and season 3 (**Winter**).
- Most of the employees obey disciplines, as most of the employees are disciplined; disciplinary failure is not reason behind their absenteeism.

Fig. 2.4 For Education, Social drinker and Social smoker

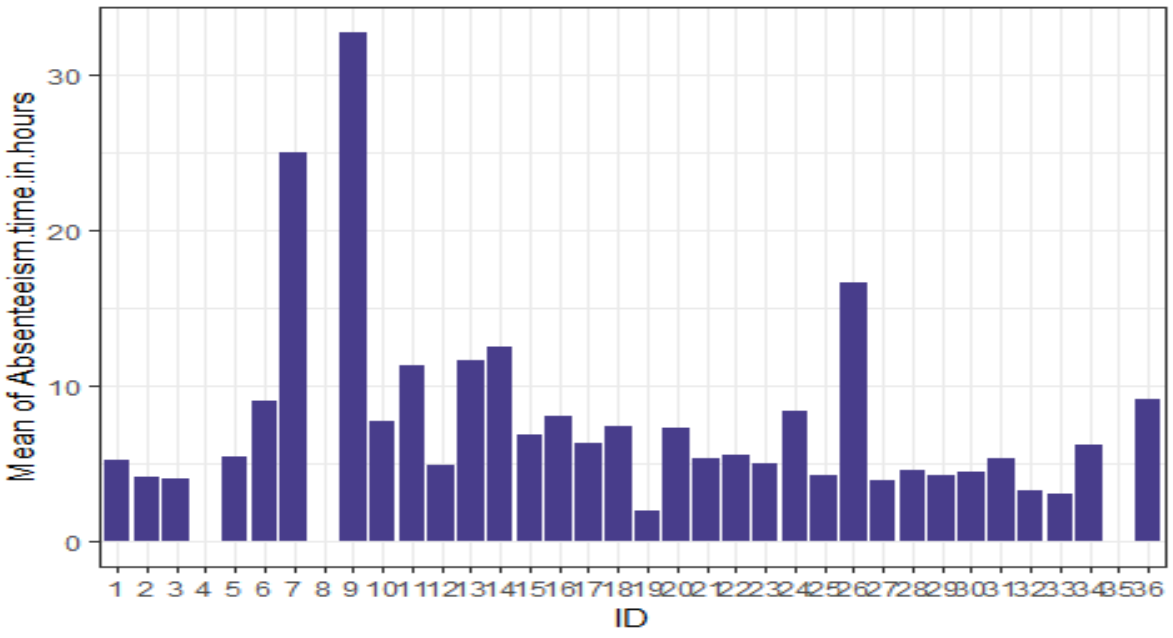


From above bar plots we can conclude:

- About 80-85% of employees are with **high-school**(1) qualification followed by **postgraduate**(3) and **graduate**(2).
- Most of the employees are social drinker.
- Most of the employees are not social smoker.

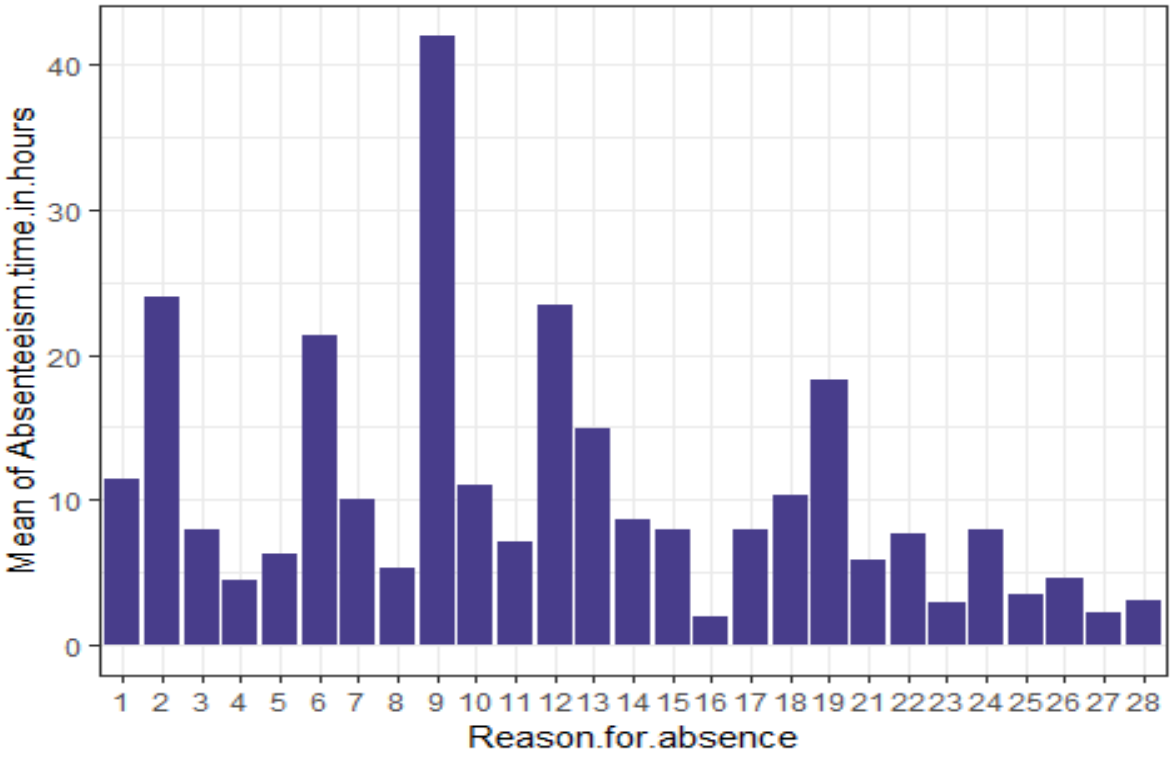
Categorical variable bivariate analysis:

Fig.2.5 Mean absenteeism time in hour per ID



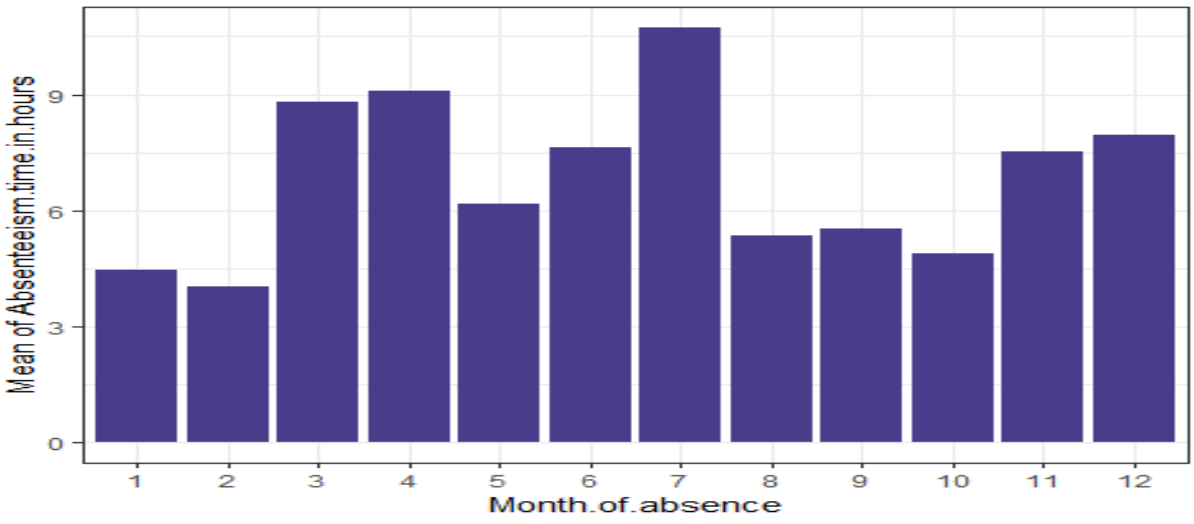
We can see that employee with ID 9 is absent for maximum time followed by ID 7, 26, 14, 13, 11 also we can see that employee with ID 4, 8, 35 are never absent.

Fig.2.6 Mean absenteeism time in hour per Reason for absence



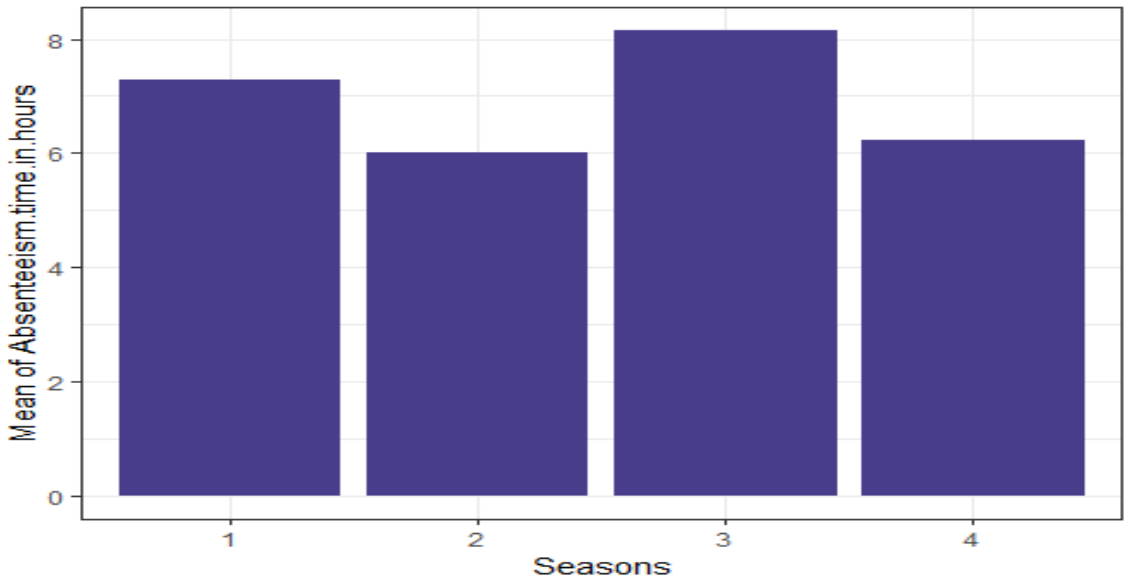
From above plot, it is very clear that employees are absent for maximum time for reason 9(**Diseases of the circulatory system**) followed by reason 12(**Diseases of the skin and subcutaneous tissue**), reason 2(**Neoplasms**), reason 6(**Diseases of the nervous system**) and reason 19(**Injury, poisoning and certain other consequences of external causes**).

Fig.2.7 Mean absenteeism time in hour per Month of absence



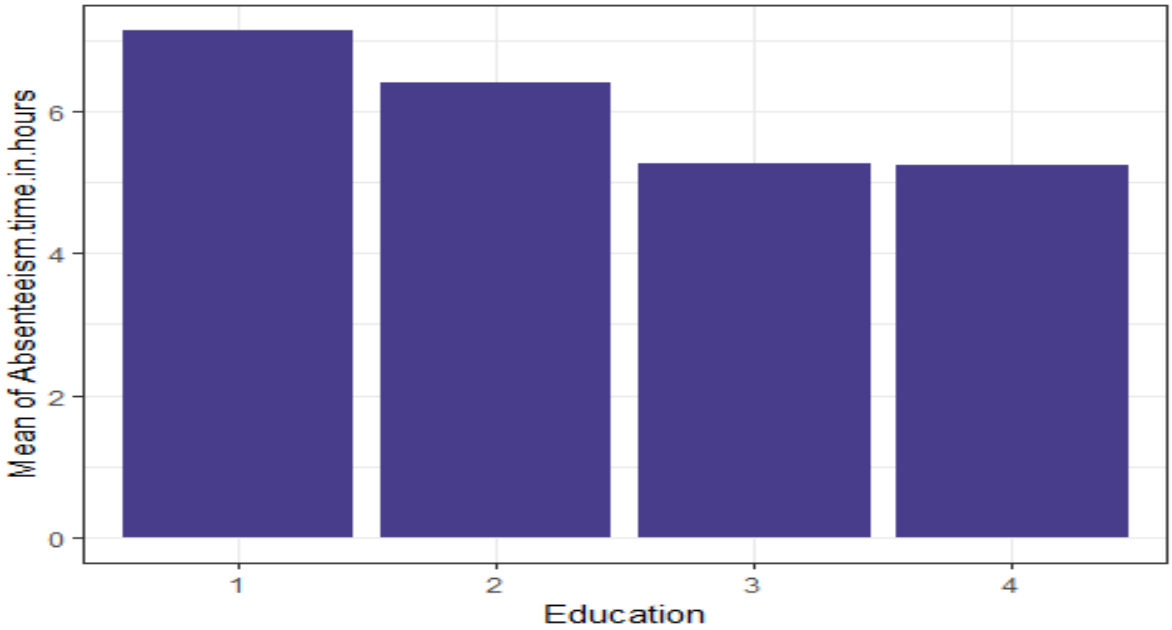
We can see from above plot that, employees are absent for maximum time in month 7(**July**) followed by month 4(**April**), 3(**March**) and 12(**December**).

Fig.2.8 Mean absenteeism time in hour per season



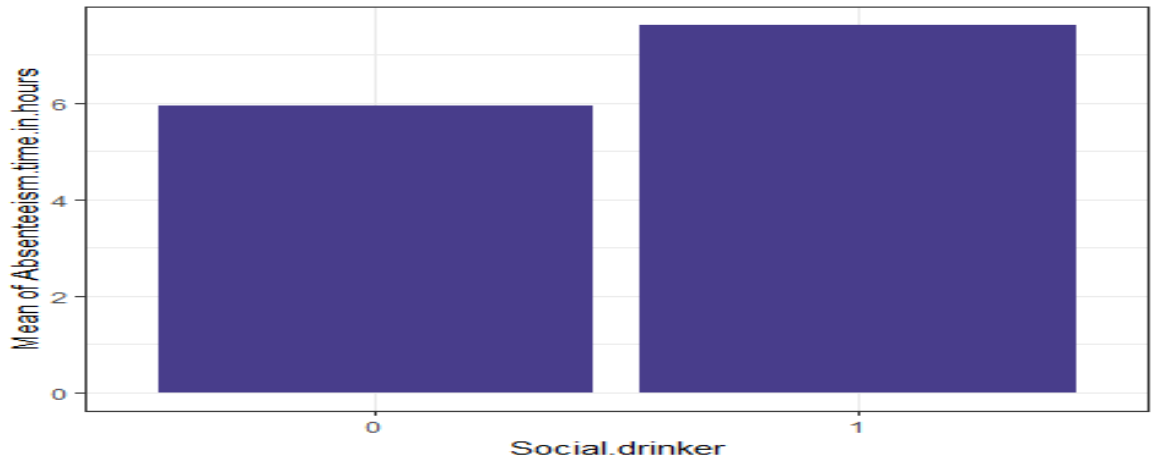
It can be seen from above plot that employees are absent for maximum time in season 3(**winter**) followed by season 1(**summer**) and season 4(**Spring**).

Fig.2.9 Mean absenteeism time in hour as per education



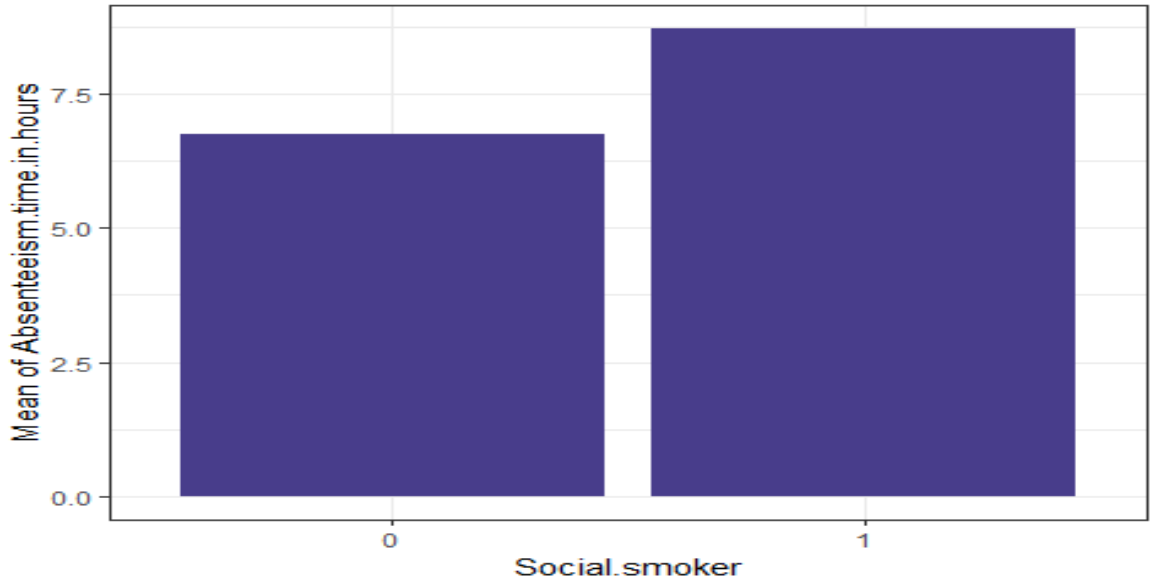
It can be observed from above graph that employees with educational qualification of **high-school** are absent for maximum time followed by **graduate** and **postgraduate** employees.

Fig.2.10 Mean absenteeism time in hour as per his drinking behaviour



Employees who are social drinker are absent for maximum time as compared to those who don't drink.

Fig.2.11 Mean absenteeism time in hour as per his smoking behaviour



Employees who are social smoker are absent for maximum time as compared to those who don't smoke.

2.1.2 Missing Value Analysis

In real world data, there are some instances where a particular element is absent because of various reasons, such as, corrupt data, failure to load the information, or incomplete extraction. Handling the missing values is one of the greatest challenges faced by analysts,

because making the right decision on how to handle it generates robust data models. Let us look at different ways of imputing the missing values.

- **Deleting rows or column**

  In this method we delete the row having missing value also if a particular column has more than 30% of missing value, we delete that column. But this method is done only if we have enough sample as deleting the rows or column would lead to information loss.

- **Replacing with mean/median/mode**

  In this method missing value are imputed based on mean/median/mode method depend upon data type of variable, If the variable numeric we can go with either mean/median and for categorical variable we go with mode method

- **Replacing with KNN imputation**

  In this method missing value are imputed with knn imputation method which is based on Euclidean distance formula or Manhattan distance formula.

- **Predicting missing values**

  In this method based on feature in dataset having no missing value, we can predict missing value with their relation to the variable having missing value. In this method based on relation between two variables we can predict missing value with machine learning algorithm too.

Out of all these method, we are going to use mean/median/mode, KNN imputation and predicting missing value method to impute missing value in our dataset.

Missing Value Imputation:

Below is the horizontal bar plot describing percentage of missing value in each variable of our dataset:

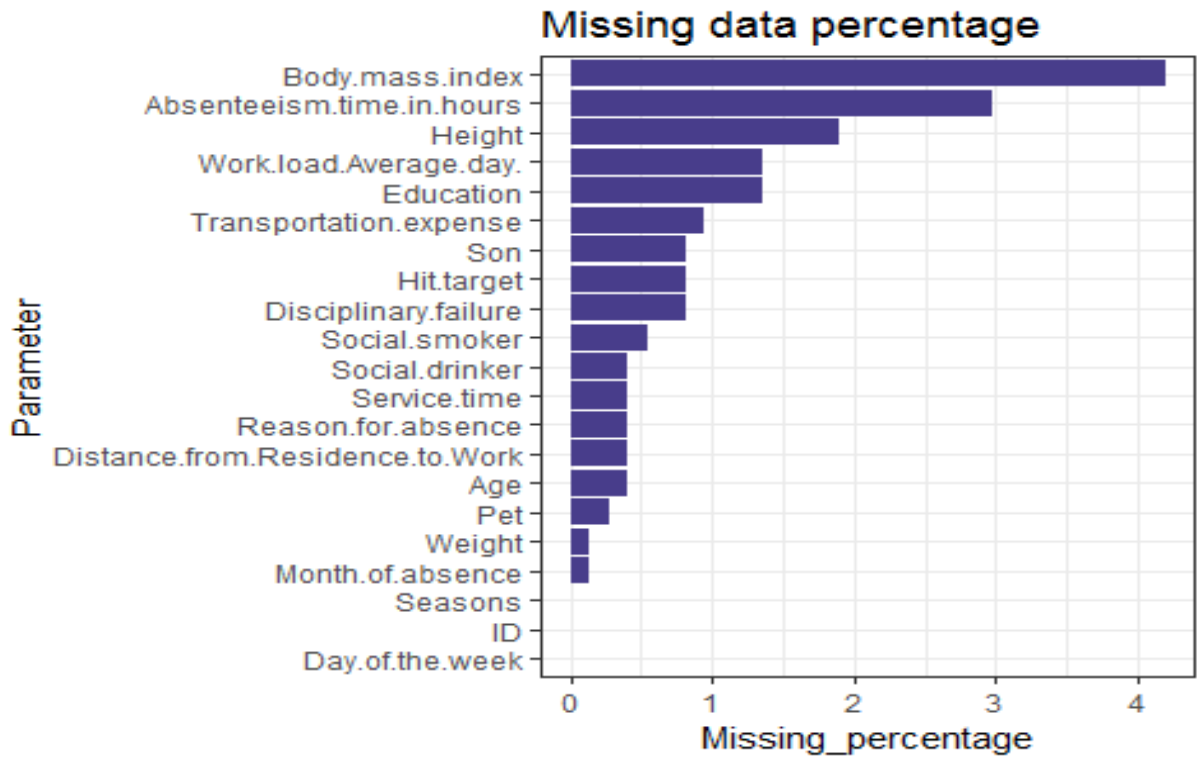


Fig.2.12 percentage of missing value in each column

Now let's see which method is used to impute missing value in particular column

Variable	**Missing value imputation method**
Body.mass.index	As we know, $$\textbf{Body. mass. index} = \frac{\textbf{Weight in kg}}{(\textbf{Height in meter})^2}$$ With above formula, as we have values of weight and height for corresponding null values in Body.mass.index, we have imputed missing value by calculating.
Height, Work.load.Average.day, Son, transportation.expense, Hit.target, Distance.from.Residence.to.work, Service.time, Age, Pet, weight	Missing values are imputed with median method by calculating mean of particular variable corresponding to ID of null value in that variable Suppose we have ID 1 corresponding to null

	value in a particular variable, we have calculated mean value of that variable by considering only those values corresponding to ID 1.
Absenteeism.time.in.hours	Missing values are imputed with median method by calculating median of Absenteeism.time.in.hours corresponding to ID of null value in Absenteeism.time.in.hours
Education, Disciplinary.failure, Social.smoker, Social.drinker, Reason.for.absence Month.of.absence	Missing values are imputed with mode method corresponding to ID of null value in that variable. Missing value is imputed with reference from seasons variable. Also 0 value in Month.of.absence is due to error as Month can't be zero, we have also treated them as null value and imputed with reference from seasons variable.
Reason.for.absence	0 values in Reason.for.absence is due to error as we have no decription of 0 reason, we have also treated them as null value and imputed with reason 26 which is unjustified absence.

In this way, we have imputed all the missing values.

2.1.3 Outlier Analysis

Several definitions of outlier have been presented in the data mining literature. two of them:

i. An outlier is defined as a data point which is very different from the rest of the data based on some measure and

ii. An outlier is a case that does not follow the same model as the rest of the data and appears as though it comes from a different probability distribution.

We can clearly observe from fig. 2.1 the probability distributions that most of the variables are skewed, for e.g. transportation.expense, distance.from.residence.to.work, Height, Pet etc. The skew in these distributions can be most likely explained by the presence of outliers and extreme values in the data. One of the other steps of pre-processing apart from missing value analysis is the presence of outliers. In this case we use a classic approach of removing outliers, Tukey's method in which datapoints falling above and below of 1.5 times inter quartile range are considered as outliers. We can visualize the outliers using boxplots methods.

In figure 2.13, we have plotted the boxplots of the 12 continous variables. A lot of useful inferences can be made from these plots. First as you can see, we have outliers and extreme values in variables like Height, Absenteeism.time.in.hours, Pet etc.

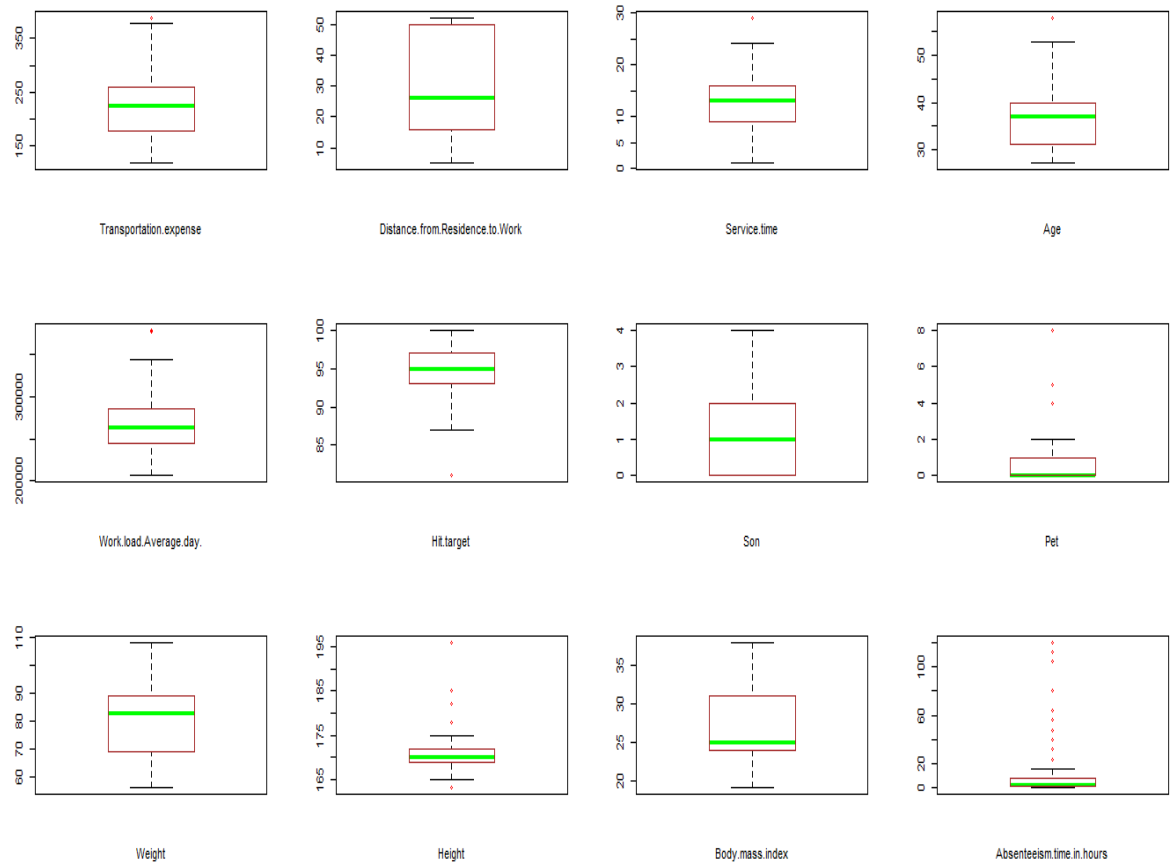


Fig.2.13, Boxplot Visualization

From above boxplots we can see range of variables are very different from each other.Also below we can see values in variables which are considered as outliers.

```
[1] "Transportation.expense"
[1] 388
[1] "Distance.from.Residence.to.Work"
numeric(0)
[1] "Service.time"
[1] 29
[1] "Age"
[1] 58
[1] "Work.load.Average.day."
[1] 378884 377550
[1] "Hit.target"
[1] 81
[1] "Son"
numeric(0)
[1] "Pet"
[1] 4 5 8
[1] "Weight"
numeric(0)
[1] "Height"
[1] 178 196 182 185 163
[1] "Body.mass.index"
numeric(0)
[1] "Absenteeism.time.in.hours"
 [1]  40  32  24  64  56  80 120 112 104  48
```

In this method we have removed outlier and replace them with NA value and then we have imputed them with KNN imputation method.

2.1.4 Feature Selection

Before performing any type of modelling 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. We have performed correlation analysis for features selection. In this method pair of variables having correlation, coefficient greater than 0.8 are considered as correlated features. We have plotted correlation plot using this analysis. From below correlation plot, extreme blue colour indicates highly positively correlated features and extreme red colour indicate highly negatively correlated features; we will eliminate any one variable from a pair of correlated variable.

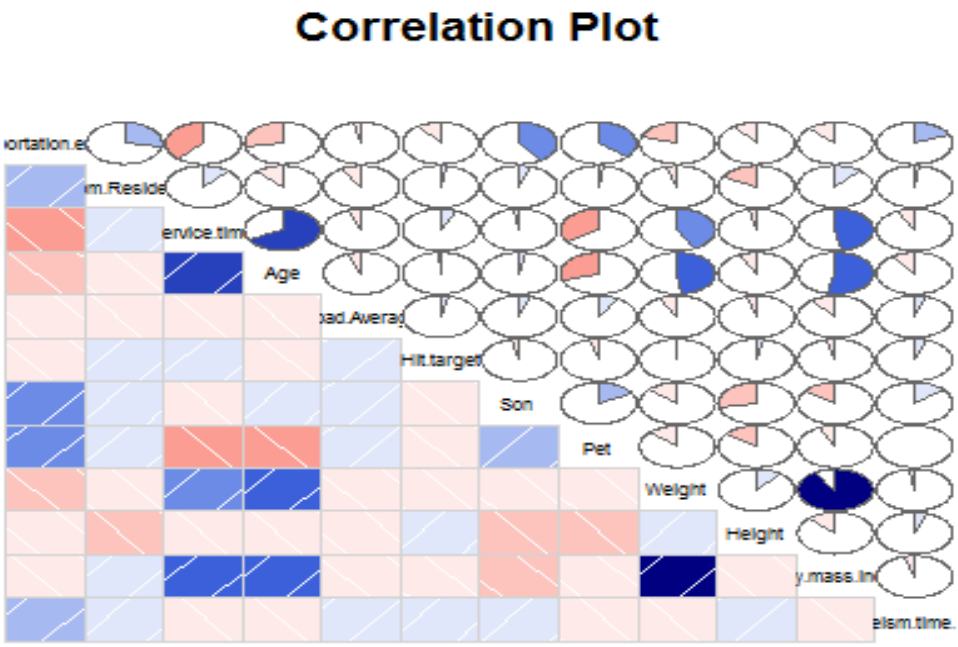


Fig.2.14 Correlation plot

From above correlation plot, features weight and Body.mass.index are highly positively correlated, we will eliminate Body.mass.index feature as Weight is the basic measure.

After feature selection we are left with 740 rows and 20 variables.

2.1.5 Feature Scaling

Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data pre-processing step.
The motivation behind feature scaling is since the range of values of raw data varies widely in some machine learning algorithms, objective functions will not work properly without normalization. For example, the majority of classifiers calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance.
Following are the method to perform feature scaling:

i. **Standardization/ z- score**

It requires data to be normally distributed. It performs scaling based on below formula:

$$z = \frac{x - \mu}{\sigma}$$

Where,
z = difference between raw score and population mean in the units of standard deviation.

$\boldsymbol{x}$ = observation
μ = mean of population
$\boldsymbol{\sigma}$ = standard deviation of population

ii. Normalization

It does not require data to be normally distributed. It scales values between the range of 0-1. It performs scaling based on below formula:

$$\textbf{New Value} = \frac{\textbf{Value} - \textbf{min(Value)}}{\textbf{max(Value)} - \textbf{min(Value)}}$$

To check normality let's have a look at below qqnorm plot

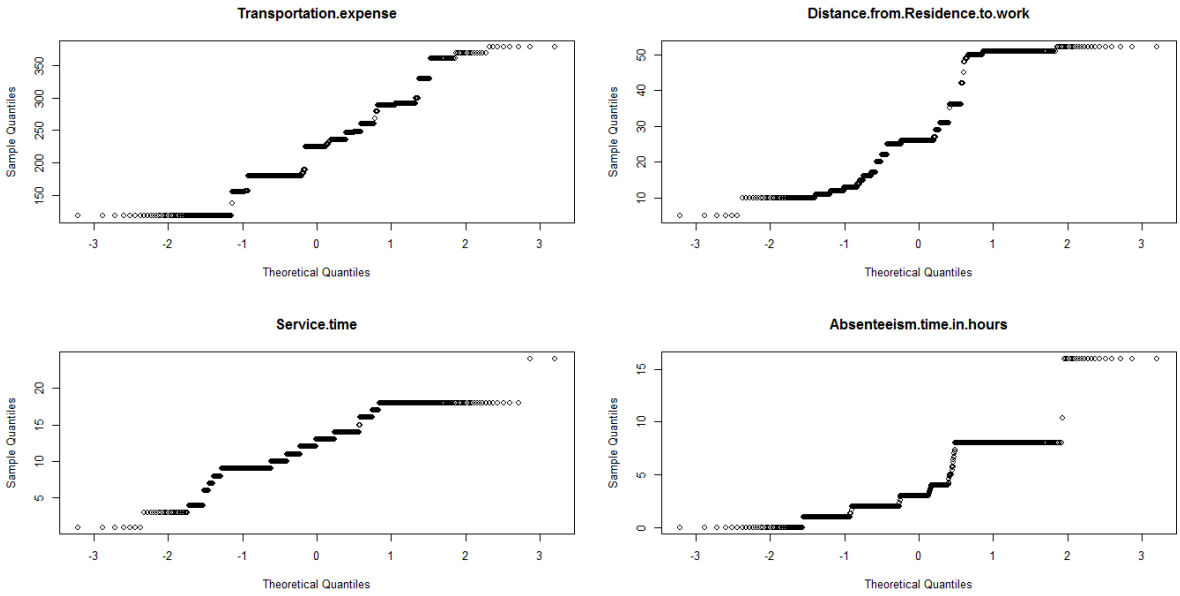


Fig.2.15 qqnorm plot

From above qqnorm plot, we can see variables do not follow normality. Hence we will use normalization approach for all continuous variable except target variable(Absenteeism.time.in.hours).

2.1.6 Principal Component Analysis

As principal component analysis is performed on numerical variables. After feature scaling, we have created dummy variables of categorical variables with one hot encoding method to convert categorical variable in numerical variables. After creating dummy variables, we have 105 variables including target variables. As most of the feature do not provide necessary information to predict target variable and our dataset becomes too complex to handle which leads to bad accuracy, hence dimensionality reduction is performed with principal component analysis. Before performing PCA we have split our dataset into train and test data with 80:20 ration using simple random sampling method.

Principal component analysis is a method of extracting important variables (in form of components) from a large set of variables available in a data set. It extracts low dimensional set of features from a high dimensional data set with a motive to capture as much information

as possible. With fewer variables, visualization also becomes much more meaningful. PCA is more useful when dealing with 3 or higher dimensional data. First we are going to plot cumulative scree plot of component to see how much percentage of variance is explained by how many components.

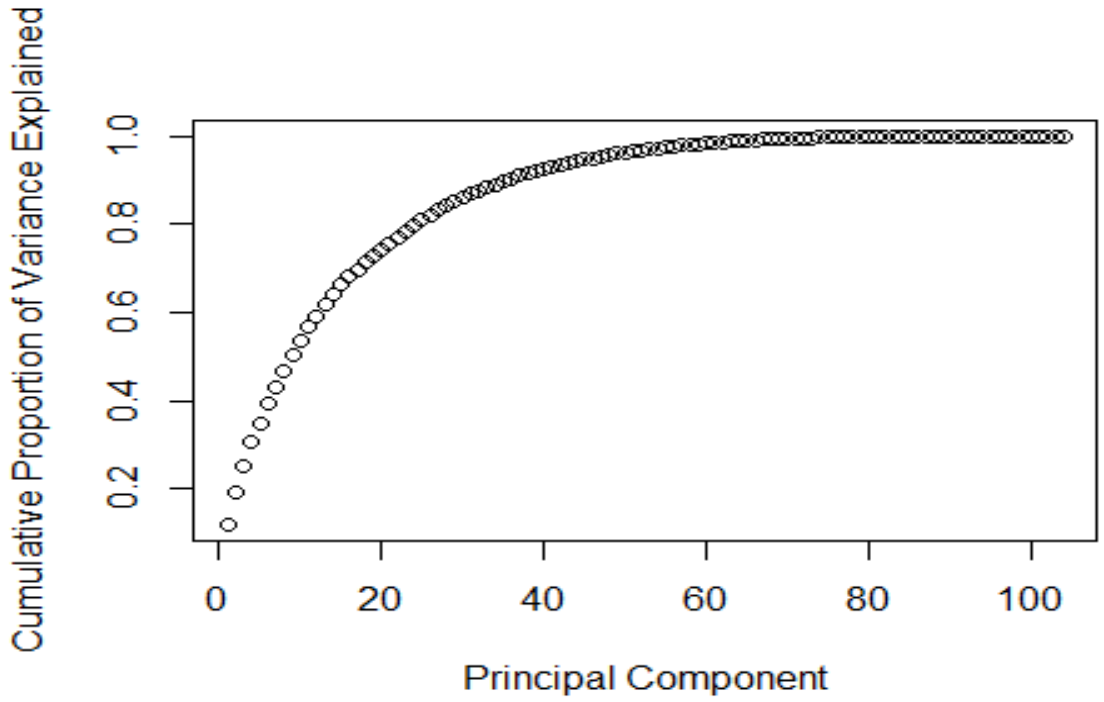


Fig.2.16 Cumulative scree plot

.From above plot we can see almost 95%+ variation is explained by first 45 principal component, we will select only first 45 principal component for our analysis. Now we are left with 46 variables including target variables. Using PCA we have reduced 104 predictors to 45 without compromising on explained variance. This is the power of PCA.

2.2 Modelling

2.2.1 Model Selection

After performing pre-processing technique first step in model building is the selection of model for a particular problem. As our target variable “Absenteeism.time.in.hours” falls in the category of numeric variable, we can use regression approach for the predictive analysis. We always start our model building from the most simplest to more complex. Therefore we are using Decision tree regressor followed by Random Forest regression and Linear Regression.

2.2.2 Decision 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 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.

```
# Building Decision tree
fit = rpart(Absenteeism.time.in.hours ~ ., data = train, method = "anova")
#plot decision tree
plot(fit)
text(fit,pretty = 0)
```

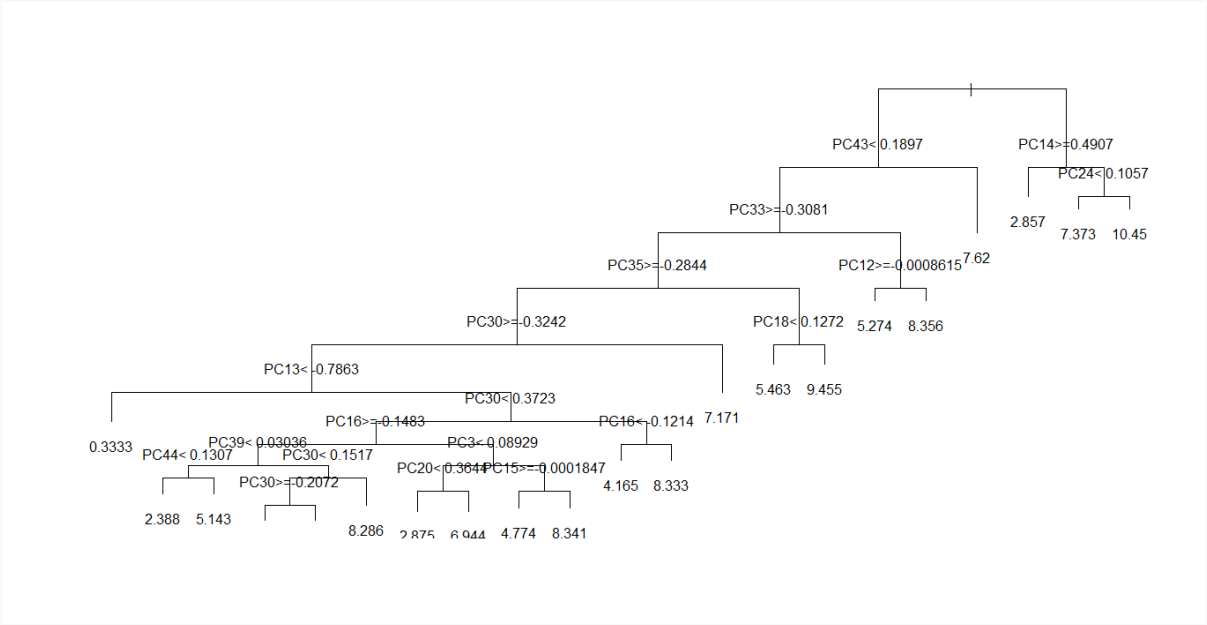


Fig.2.17,Decision tree plot

After building the model on train data, we are going to predict for test case, we are going to e valuate the model with Root Mean Square Error, below is the result of model:

```
#Predict for test cases
predictions_DT = predict(fit, test[,-46])
#Evaluate performance of model with RMSE and R-squared value
print(postResample(pred = predictions_DT, obs = test[,46]))
```

```
##      RMSE  Rsquared       MAE
## 3.4249551 0.1018658 2.6783464
```

2.2.3 Random Forest

The random forest combines hundreds or thousands of decision trees, trains each one on a slightly different set of the observations, splitting nodes in each tree considering a limited

number of the features. The final predictions of the random forest are made by averaging the predictions of each individual tree.

```
#Building random forest
fit_RF = randomForest(Absenteeism.time.in.hours ~., data = train)

fit_RF

##
## Call:
##  randomForest(formula = Absenteeism.time.in.hours ~ ., data = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 15
##
##           Mean of squared residuals: 7.918353
##                     % Var explained: 29.13
```

After building the model on train data, we are going to predict for test case, we are going to e valuate the model with Root Mean Square Error, below is the result of model:

```
#Predict for test cases
predictions_RF = predict(fit_RF,test[,1:45])
# Evaluate performance of model with RMSE and R-squared value(Result)
print(postResample(pred = predictions_RF, obs = test$Absenteeism.time.in.h
ours))

##      RMSE  Rsquared       MAE
## 2.6729269 0.3535375 1.1837144
```

2.2.4 Linear Regression

Linear regression is a basic and commonly used type of predictive analysis. The overall idea of regression is to examine two things:

(1) does a set of predictor variables do a good job in predicting an outcome (dependent) varia ble?

(2) Which variables in particular are significant predictors of the outcome variable, and in wh at way do they–indicated by the magnitude and sign of the beta estimates–impact the outcom e variable?

These regression estimates are used to explain the relationship between one dependent variabl e and one or more independent variables. The simplest form of the regression equation with one dependent and one independent variable is defined by the formula:

$$\mathbf{y} = \mathbf{c} + \mathbf{b} * \mathbf{x}$$

where,

y = estimated dependent variable score,

c = constant,

b = regression coefficient, and

x = score on the independent variable.

```
#Building Linear Regression Model
lm_model = lm(Absenteeism.time.in.hours~.,data = train)

#Plot linear regression model
par(mfrow = c(2,2))
plot(lm_model)
```

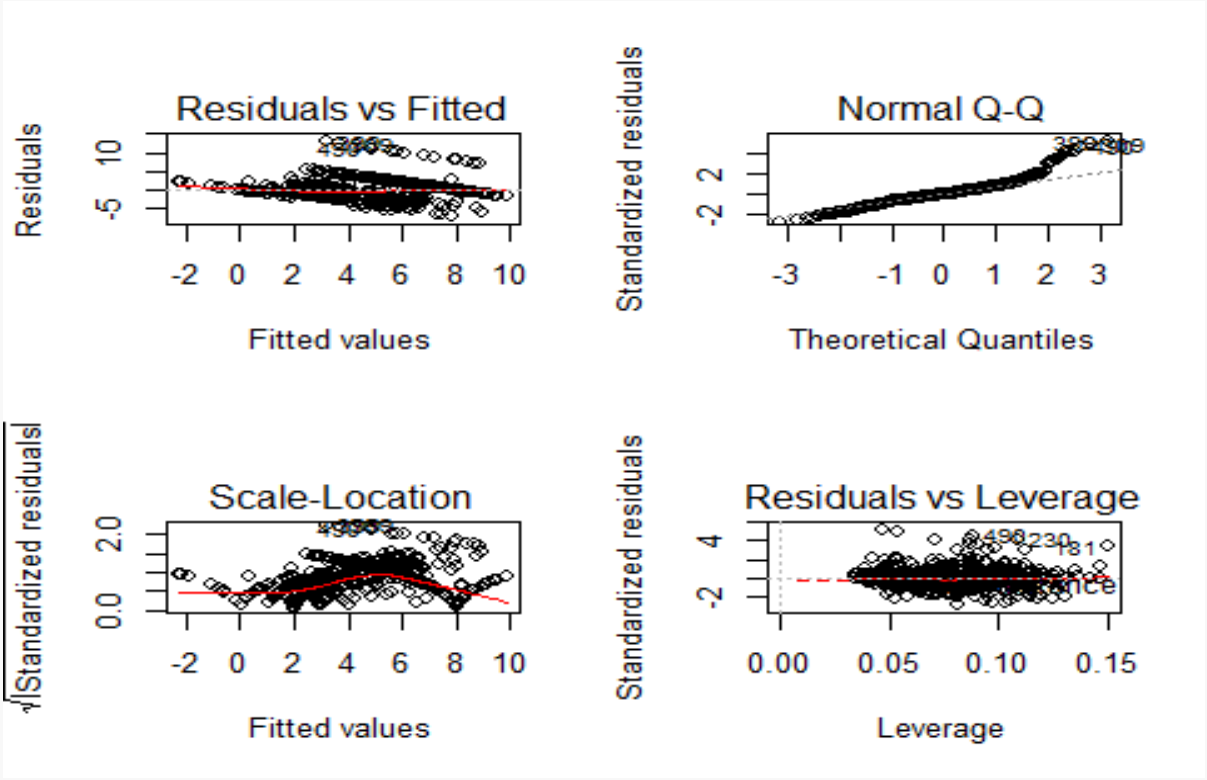


Fig.2.18 Linear Regression model

From Residual vs Fitted plot, we can see there is somewhat non-linear relationship between fi tted and residual value or we can say between dependent and independent variable as red line is not perfectly flat, though we assume linear relationship between dependent and independen t variable to build linear regression model,such cases may not exist in actual, still the model c an be used to predict for test cases.

Also from QQnormal plot, we can see that residual or dependent variable is normal except at extreme point, as we assume normality of residual while building linear regression model, he nce transformation of dependent variable may solve this problem somewhat.

Also linear regression model assume constant variance between fitted and standardized residu al, but we can see from scale-location plot, the curve indicates somewhat non-constant varian ce.

Hence, to fit our model to the assumption we can take log transformation of dependent variab le as transformation of variable solve the problems to some extent, but as our dependent varia ble contain zero's, we will take square root transformation which is quite less effective as co mpared to log transformation.

Let’s build second linear regression model taking square root transformation of dependent var iable

```
#Building Second Linear Regression Model

lm_model2 = lm(sqrt(Absenteeism.time.in.hours)~.,data = train)

#Plot linear regression model
par(mfrow = c(2,2))
plot(lm_model2)
```

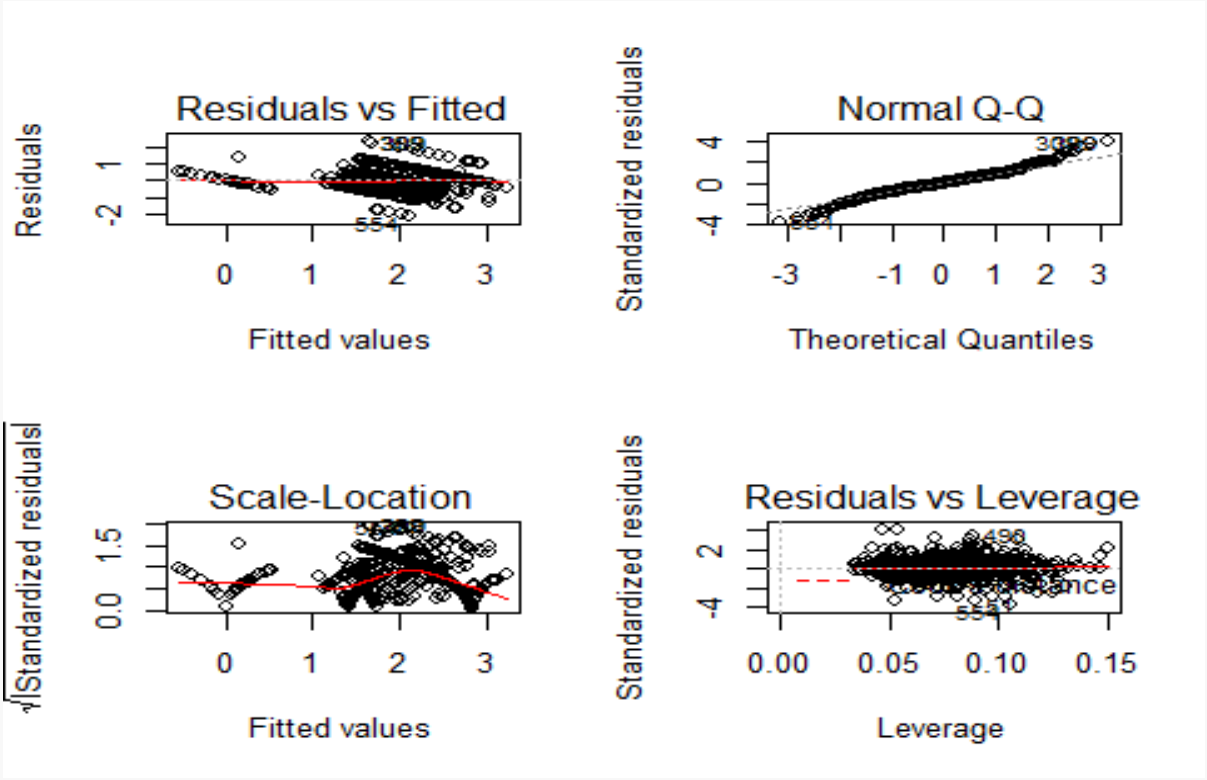


Fig.2.19 Linear regression model_2

As we can see relations are somewhat improved, now we will predict for test cases and see w hat results we get:

```
#Predict for test cases as well as squaring prediction as we have taken sq
uare root of target variable in train data
predictions_lm = (predict(lm_model2,test[,1:45]))^2
#Evaluate performance of model with RMSE and R-squared value
print(postResample(pred = predictions_lm, obs = test[,46]))
```

```
##      RMSE  Rsquared       MAE
## 2.6746195 0.3607024 1.1504976
```

Chapter 3

Conclusion

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

The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) (or sometimes root-mean-squared error) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. The RMSD represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences. These deviations are called residuals when the calculations are performed over the data sample that was used for estimation and are called errors (or prediction errors) when computed out-of-sample.

Though we are building regression models but looking at the characteristics of our dataset, our data is time series problem, and RMSE is generally used for time series problem, hence RMSE seems to be good choice for evaluating performance of our model.

3.2 Model Selection

Below are the RMSE values for models we built:

Table 3.1: Models Result

Sr.no	Model	RMSE	Rsquared value
1	Decision tree	3.42	10.18
2	Random forest	2.67	35.35
3	Linear regression	2.67	36.07

Based on the above values of RMSE and Rsquared value, as Linear regression gives least value of RMSE 2.67 and higher Rsquared value 36.07, we select linear regression model for our problem.

3.3 Solution to problem asked

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

As we have already stated the reasons behind employee's absenteeism in data visualization steps, following are the ways to reduce the number of absenteeism:

- **i.** ID no. 3,28 and 34 are absent for most of the time, company should call these employees and ask for valid reasons and should warn these employees about their absenteeism behaviour.
- **ii.** Most of the employees are absent for either medical consultation or dental consultation, company can arrange the medical camps periodicaly or improve the awareness of employees towards their health through different mediums so that employees absenteeism will get reduced.
- **iii.** Also most of employees are absent with no justification, company can make the rules and policies for leaves and set the limit of absenteeism time to reduce absenteeism.
- **iv.** Employees with ID 9,11,13,14 and 26 are absent for maximum time, company can set the limit of absenteeism. Also company can introduce incentives and performance hike to keep the employees motivated to make them more regular to work.
- **v.** Also we can see employees with high school are absent most, company can hire more qualified employees or should work for increasing morale, engagement and commitment to the organization of employees.
- **vi.** Employees who are social drinker and social smoker are absent for maximum time, company can make the rule for employees to not to drink and smoke during working hours.

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

As we know excessive absences can equate to decreased productivity and can have a major effect on company finances, morale and other factors. With the given dataset, if same trend of

employees occurs in 2011, company will experience work loss which result in money and productivity loss of the company. Below is the formula used to predict monthly work loss:

$$Work\ loss = \frac{Work\ load\ average\ day * Absenteeism\ time\ in\ hours}{Service\ time}$$

Table 3.2:Monthly loss in 2011

Sr. no.	Month of absence	Monthly loss
1	1	4834973
2	2	7987798
3	3	10936303
4	4	6318055
5	5	6871465
6	6	10121617
7	7	11627652
8	8	7156722
9	9	4278465
10	10	7416093
11	11	6773002
12	12	7889856

Monthly loss visualization:

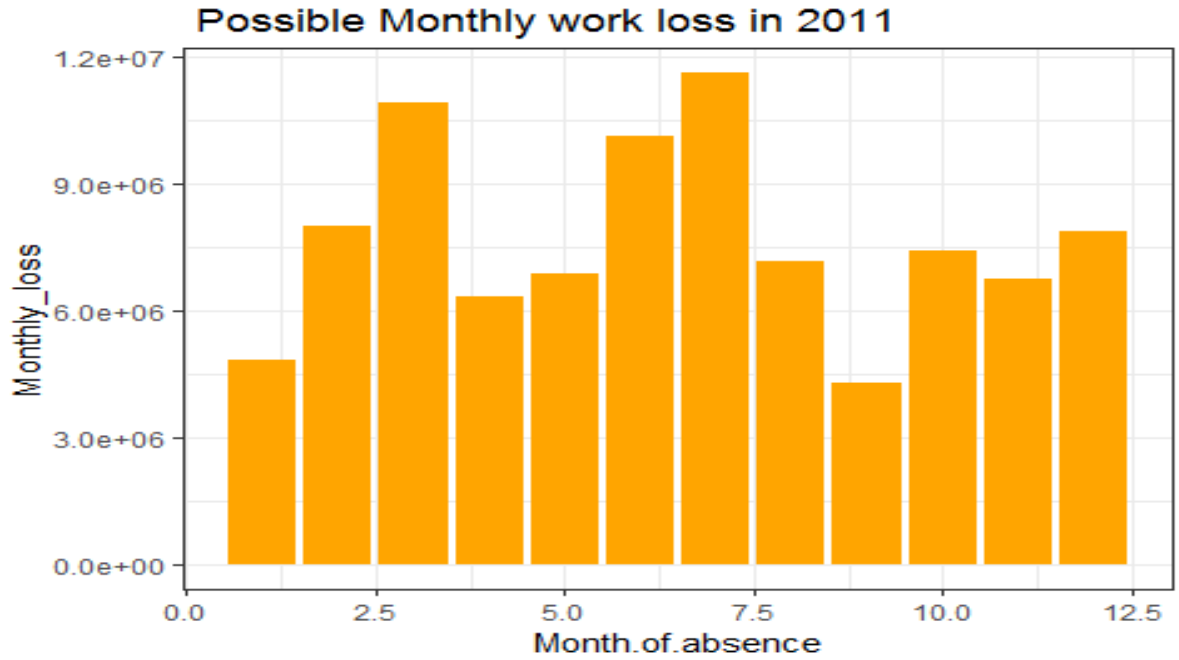


Fig.3.1 Monthly loss in 2011

Appendix A - Extra Figures

Fig.Mean Absenteeism time in hour per Day.of.week

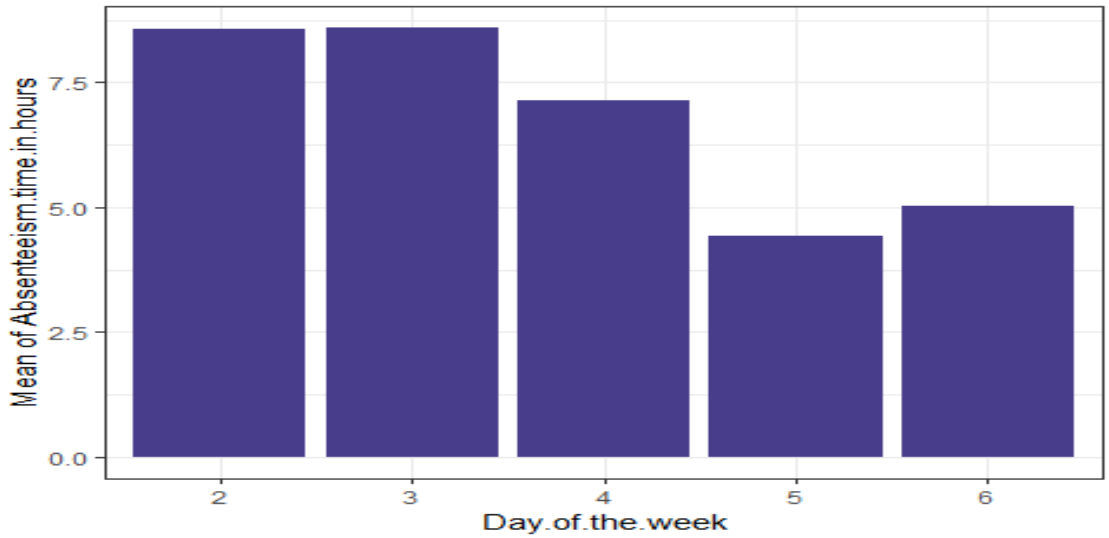
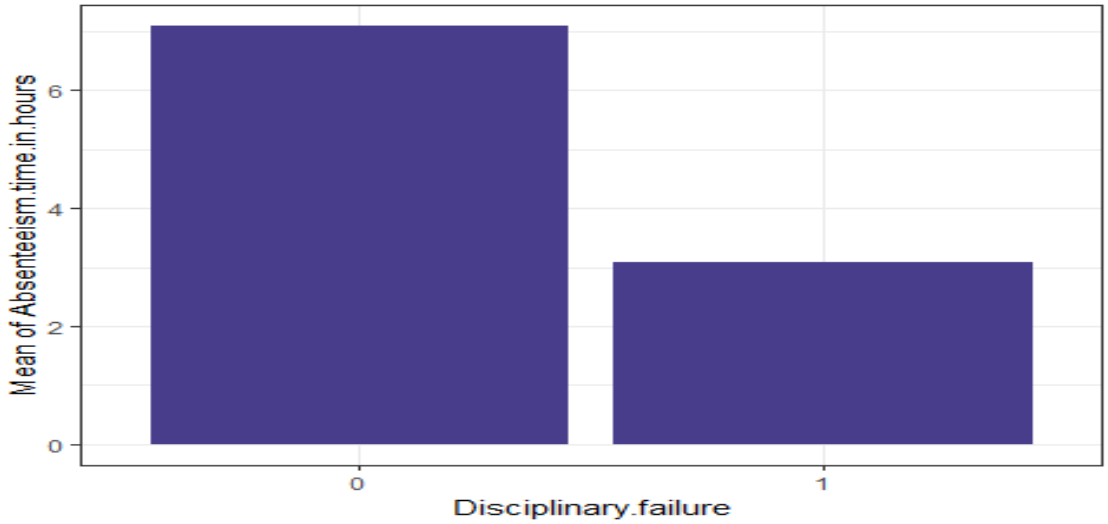


Fig. Mean Absenteeism time in hour per category of disciplinary failure



Appendix B – R Code

Fig2.1.Histogram+KDE plot of numerical variable

```
numeric_index = sapply(absent_train,is.numeric) #selecting only numeric
numeric_data = absent_train[,numeric_index]
cnames = colnames(numeric_data)
multi.hist(numeric_data,nrow = 4,ncol = 3,bcol="linen", dcol=c("blue","red"),dlty=
c("solid","solid"),main=NULL)
```

Fig.2.2 For Id and Reason for absence

```
#Categorical variable univariate analyis
factor_index = sapply(absent_train,is.factor)
factor_data = absent_train[,factor_index]
cat_names = colnames(factor_data)
#Define function to draw barplots
count.show = function(x){
  ggplot(absent_train,aes(x = absent_train[,x])) +
    geom_bar(fill = "darkslateblue") +
    theme_bw() + xlab(x) + ylab("Count")
}
for (i in 1:length(cat_names)){
  assign(paste0("plt",i),count.show(cat_names[i]))
}
grid.arrange(plt1,plt2,nrow =2)
```

Fig2.3, For month of absence, day of the week, seasons and disciplinary failure

```
grid.arrange(plt3,plt4,plt5,plt6,ncol = 2,nrow = 2)
```

Fig. 2.4 For Education, Social drinker and Social smoker

```
grid.arrange(plt7,plt8,plt9,ncol = 2,nrow = 2)
```

Fig.2.5 Mean absenteeism time in hour per ID

```
cat_show = function(x,y){

m = aggregate(absent_train[,x],by = list(category = absent_train[,y]),FUN = mean)
ggplot(m,aes(x = m$category,y = m$x)) + geom_bar(stat = "identity",fill =
"DarkslateBlue") + xlab(y) +
  ylab("Mean of Absenteeism.time.in.hours") +
  theme_bw()
}
cat_show(21,"ID")
```

Fig.2.6 Mean absenteeism time in hour per Reason for absence

```
cat_show(21,"Reason.for.absence")
```

Fig.2.7 Mean absenteeism time in hour per Month of absence

```
cat_show(21,"Month.of.absence")
```

Fig.2.8 Mean absenteeism time in hour per season

```
cat_show(21,"Seasons")
```

Fig.2.9 Mean absenteeism time in hour as per education

```
cat_show(21,"Education")
```

Fig.2.10 Mean absenteeism time in hour as per his drinking behaviour

```
cat_show(21,"Social.drinker")
```

Fig.2.11 Mean absenteeism time in hour as per his smoking behaviour

```
cat_show(21,"Social.smoker")
```

Fig.2.12 percentage of missing value in each column

```
ggplot(data = missing_val[,], aes(x=reorder(Columns, Missing_percentage),y = Missi
ng_percentage))+
  geom_bar(stat = "identity",fill = "DarkslateBlue") +coord_flip()+xlab("Parameter
")+
  ggtitle("Missing data percentage") + theme_bw()
```

Fig.2.13, Boxplot Visualization

```
par(mfrow = c(3,4))
for (i in 1:length(cnames)){
  boxplot(numeric_data[,i],xlab = cnames[i],outcol = "red",boxcol = "brown",medcol
```

```
= "green")
}
```

Fig.2.14 Correlation plot

```
corrgram(absent_train[,numeric_index], order = F,
         upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")
```

Fig.2.15 qqnorm plot

```
par(mfrow = c(2,2))
qqnorm(absent_train$Transportation.expense,main = 'Transportation.expense')
qqnorm(absent_train$Distance.from.Residence.to.Work,main = 'Distance.from.Residenc
e.to.work')
qqnorm(absent_train$Service.time,main = 'Service.time')
qqnorm(absent_train$Absenteeism.time.in.hours,main = 'Absenteeism.time.in.hours')
```

Fig.2.16 Cumulative scree plot

```
plot(cumsum(prop_varex), xlab = "Principal Component",
     ylab = "Cumulative Proportion of Variance Explained",
     type = "b")
```

Fig.2.17,Decision tree plot

```
#Building Decision tree
fit = rpart(Absenteeism.time.in.hours ~ ., data = train, method = "anova")
#plot decision tree
plot(fit)
text(fit,pretty = 0)
```

Fig.2.18 Linear Regression model

```
#Building Linear Regression Model
lm_model = lm(Absenteeism.time.in.hours~.,data = train)
summary(lm_model)

#Plot linear regression model
par(mfrow = c(2,2))
plot(lm_model)
```

Fig.2.19 Linear regression model_2

```
lm_model2 = lm(sqrt(Absenteeism.time.in.hours)~.,data = train)
summary(lm_model2)
```

```
#Plot linear regression model
par(mfrow = c(2,2))
plot(lm_model2)
```

Fig.3.1 Monthly loss in 2011

```
ggplot(Loss_2011,aes(x = Loss_2011$Month.of.absence,y = Loss_2011$Monthly.loss)) +
  geom_bar(stat = "identity",fill = 'orange') + xlab("Month.of.absence") + ylab("M
onthly_loss") +
  theme_bw() + ggtitle(" Possible Monthly work loss in 2011 ")
```

Complete R File:

```
#Remove all object to clear the environment
rm(list=ls(all=T))
#set the working directory
setwd("C:/Users/vrush_000/Desktop/Data science/Project/Main Project/Absenteeism")
#Check the working directory
getwd()

#Load Libraries
x = c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "dummi
es","Information",
      "MASS", "rpart", "gbm", "ROSE", "sampling", "DataCombine", "inTrees","psych"
,"gridExtra","xlsx")

#install.packages(x)
lapply(x, require, character.only = TRUE)

rm(x)

## Read the data
absent_train = read.xlsx("Absenteeism_at_work_Project.xls",header = T,sheetIndex =
1)
##Explore the data
#dimension of data
dim(absent_train)

## [1] 740  21

#structure of data
str(absent_train)

#unique value in each column
data.frame(apply(absent_train, 2, function(x) length(unique(x))))

#conversion of datatype of variable
absent_train[1:5] = lapply(absent_train[1:5],as.factor)
absent_train[12:13] = lapply(absent_train[12:13],as.factor)
absent_train[15:16] = lapply(absent_train[15:16],as.factor)
#Let's have a look at summary of each variable
summary(absent_train)
```

```r
#**********************Missing Values Analysis*****************************#
#checking for missing values
sum(is.na(absent_train))

#Missing value in each column
missing_val = data.frame(apply(absent_train,2,function(x){sum(is.na(x))}))
missing_val$Columns = row.names(missing_val)
names(missing_val)[1] =  "Missing_percentage"
missing_val$Missing_percentage = (missing_val$Missing_percentage/nrow(absent_train
)) * 100
missing_val = missing_val[order(-missing_val$Missing_percentage),]
row.names(missing_val) = NULL
missing_val = missing_val[,c(2,1)]
#store missing value file to system
write.csv(missing_val, "Missing_perc.csv", row.names = F)
ggplot(data = missing_val[,], aes(x=reorder(Columns, Missing_percentage),y = Missi
ng_percentage))+
  geom_bar(stat = "identity",fill = "DarkslateBlue") +coord_flip()+xlab("Parameter
")+
  ggtitle("Missing data percentage") + theme_bw()
```

```r
##Imputing missing values with reference to ID's
#For Reason.for.absence
absent_train$ID[which(is.na(absent_train$Reason.for.absence))]

table(absent_train$ID,absent_train$Reason.for.absence)

#3 = 27 and 6 = 23 and 20 = 28
absent_train$Reason.for.absence[is.na(absent_train$Reason.for.absence) & absent_tr
ain$ID == 3] = 27
absent_train$Reason.for.absence[is.na(absent_train$Reason.for.absence) & absent_tr
ain$ID == 6] = 23
absent_train$Reason.for.absence[is.na(absent_train$Reason.for.absence) & absent_tr
ain$ID == 20] = 28
absent_train$Reason.for.absence[absent_train$Reason.for.absence == 0] = 26

#For Month.of.absence
absent_train$Month.of.absence[is.na(absent_train$Month.of.absence)] = 10
absent_train$Month.of.absence[absent_train$Month.of.absence == 0 & absent_train$Se
asons== 1] = 7
absent_train$Month.of.absence[absent_train$Month.of.absence == 0 & absent_train$Se
asons== 2] = 2
absent_train$Month.of.absence[absent_train$Month.of.absence == 0 & absent_train$Se
asons== 3] = 5

#For Transportation.expense
#imputing with mean vaue of transportation expense for a paticular ID
table(absent_train$ID,absent_train$Transportation.expense)

absent_train$ID[is.na(absent_train$Transportation.expense)]

for (i in c(1,3,10,15,20,22)){
  print(i)
  absent_train$Transportation.expense[is.na(absent_train$Transportation.expense) &
absent_train$ID == i] = mean(absent_train$Transportation.expense[absent_train$ID =
= i],na.rm = T)
}
```

```r
#For Distance.from.Residense.to.work
#Imputing with mean value with refernce to the ID
table(absent_train$ID,absent_train$Distance.from.Residence.to.Work)

absent_train$ID[is.na(absent_train$Distance.from.Residence.to.Work)]

for (i in c(22,28,34)){
  print(i)
  absent_train$Distance.from.Residence.to.Work[is.na(absent_train$Distance.from.Re
sidence.to.Work) & absent_train$ID == i] = mean(absent_train$Distance.from.Residen
ce.to.Work[absent_train$ID == i],na.rm = T)
}

#For Service.time
#Imputing with mean value with refernce to the ID
table(absent_train$ID,absent_train$Service.time)

absent_train$ID[is.na(absent_train$Service.time)]

## [1] 28 34 34
## 36 Levels: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 ... 36

for (i in c(28,34)){
  print(i)
  absent_train$Service.time[is.na(absent_train$Service.time) & absent_train$ID ==
i] = mean(absent_train$Service.time[absent_train$ID == i],na.rm = T)
}

#For Age
#Imputing with mean value with refernce to the ID
table(absent_train$ID,absent_train$Age)

absent_train$ID[is.na(absent_train$Age)]

for (i in c(24,28)){
  print(i)
  absent_train$Age[is.na(absent_train$Age) & absent_train$ID == i] = round(mean(ab
sent_train$Age[absent_train$ID == i],na.rm = T))
}

#For Height
#Imputing with mean value with refernce to the ID
table(absent_train$ID,absent_train$Height)

H_id = absent_train$ID[is.na(absent_train$Height)]
for (i in H_id){
  print(i)
  absent_train$Height[is.na(absent_train$Height) & absent_train$ID == i] = round(m
ean(absent_train$Height[absent_train$ID == i],na.rm = T))
}

#For work.load.average.day
#Imputing with mean value with refernce to the ID
table(absent_train$ID,absent_train$Work.load.Average.day.)

W_id = absent_train$ID[is.na(absent_train$Work.load.Average.day.)]
for (i in W_id){
  print(i)
  absent_train$Work.load.Average.day.[is.na(absent_train$Work.load.Average.day. &
absent_train$ID == i)] = mean(absent_train$Work.load.Average.day.[absent_train$ID
```

```r
== i],na.rm = T)
}

#For Hit.target
#Imputing with mean value with refernce to the ID
table(absent_train$ID,absent_train$Hit.target)

T_id = absent_train$ID[is.na(absent_train$Hit.target)]
for (i in T_id){
  print(i)
  absent_train$Hit.target[is.na(absent_train$Hit.target) & absent_train$ID == i] =
round(mean(absent_train$Hit.target[absent_train$ID == i],na.rm = T))
}

#For Disciplinary.failure
table(absent_train$ID,absent_train$Disciplinary.failure)

absent_train$ID[is.na(absent_train$Disciplinary.failure)]

for (i in c(10,20,22,34)){
  print(i)
  absent_train$Disciplinary.failure[is.na(absent_train$Disciplinary.failure) & abs
ent_train$ID == i] = 0
}

#For Education
table(absent_train$ID,absent_train$Education)

E_id = absent_train$ID[is.na(absent_train$Education)]

for(i in E_id){
  print(i)
  absent_train$Education[is.na(absent_train$Education) & absent_train$ID == i] = 1
}

#For Son
absent_train$ID[is.na(absent_train$Son)]

table(absent_train$ID,absent_train$Son)

#1 = 1,14 = 2,20 = 4,27 = 0,34 = 0
absent_train$Son[is.na(absent_train$Son) & absent_train$ID == 1] = 1
absent_train$Son[is.na(absent_train$Son) & absent_train$ID == 14] = 2
absent_train$Son[is.na(absent_train$Son) & absent_train$ID == 20] = 4
absent_train$Son[is.na(absent_train$Son) & absent_train$ID == 27] = 0
absent_train$Son[is.na(absent_train$Son) & absent_train$ID == 34] = 0
#For Social,drinker
table(absent_train$ID,absent_train$Social.drinker)

absent_train$ID[is.na(absent_train$Social.drinker)]

absent_train$Social.drinker[is.na(absent_train$Social.drinker) & absent_train$ID =
= 10] = 1
absent_train$Social.drinker[is.na(absent_train$Social.drinker) & absent_train$ID =
= 14] = 1
absent_train$Social.drinker[is.na(absent_train$Social.drinker) & absent_train$ID =
= 17] = 0
#For social.smoker
table(absent_train$ID,absent_train$Social.smoker)

absent_train$ID[is.na(absent_train$Social.smoker)]
```

```r
for (i in c(1,11,15,34)){
  print(i)
  absent_train$Social.smoker[is.na(absent_train$Social.smoker) & absent_train$ID ==
= i] = 0
}

#For Pet
table(absent_train$ID,absent_train$Pet)

absent_train$ID[is.na(absent_train$Pet)]

absent_train$Pet[is.na(absent_train$Pet) & absent_train$ID == 1] = 1
absent_train$Pet[is.na(absent_train$Pet) & absent_train$ID == 13] = 0
#For Weight
table(absent_train$ID,absent_train$Weight)

absent_train$ID[is.na(absent_train$Weight)]

absent_train$Weight[is.na(absent_train$Weight) & absent_train$ID == 27] = 58
#For BMI
##We will use the formula of BMI (BMI = weight in kg/(Height in m^2)) to impute mi
ssing value in BMI column
absent_train$bmi = NA
absent_train$bmi = (absent_train$Weight)/((absent_train$Height/100)^2)
absent_train$Body.mass.index = ifelse(is.na(absent_train$Body.mass.index),absent_t
rain$bmi,absent_train$Body.mass.index)
absent_train$bmi = NULL
#For Absenteeism.time.in.hours
#we will compare mean, mediam and knn and will select most suited method
#absent_train[3,21]
#absent_train[3,21] = NA
#Actual value = 2
#Mean = 3
#Median = 2
#knn = 4
#Mean method
table(absent_train$Reason.for.absence,absent_train$Absenteeism.time.in.hours)

T_reason = absent_train$Reason.for.absence[is.na(absent_train$Absenteeism.time.in.
hours)]
#for (i in T_reason){
#  print(i)
#  absent_train$Absenteeism.time.in.hours[is.na(absent_train$Absenteeism.time.in.h
ours) & absent_train$Reason.for.absence == i] = round(mean(absent_train$Absenteeis
m.time.in.hours[absent_train$Reason.for.absence == i],na.rm = T))
#}
#Median Method
for (i in T_reason){
  print(i)
  absent_train$Absenteeism.time.in.hours[is.na(absent_train$Absenteeism.time.in.ho
urs) & absent_train$Reason.for.absence == i] = round(median(absent_train$Absenteei
sm.time.in.hours[absent_train$Reason.for.absence == i],na.rm = T))
}

anyNA(absent_train)

# kNN Imputation
#absent_train = knnImputation(absent_train, k = 3)
#we have selected mediam method as it gives closer value to actual value
#**********************Data Visualisation****************************************
```

```
**#
#Histogram of numeric variables
#checking normality and skewness
numeric_index = sapply(absent_train,is.numeric) #selecting only numeric
numeric_data = absent_train[,numeric_index]
cnames = colnames(numeric_data)
multi.hist(numeric_data,nrow = 4,ncol = 3,bcol="linen", dcol=c("blue","red"),dlty=
c("solid","solid"),main=NULL)

#Box plot distribution of numeric variables
par(mfrow = c(3,4))
for (i in 1:length(cnames)){
  boxplot(numeric_data[,i],xlab = cnames[i],outcol = "red",boxcol = "brown",medcol
= "green")
}

par(mfrow = c(1,1))
#Categorical variable univariate analyis
factor_index = sapply(absent_train,is.factor)
factor_data = absent_train[,factor_index]
cat_names = colnames(factor_data)
#Define function to draw barplots
count.show = function(x){
  ggplot(absent_train,aes(x = absent_train[,x])) +
    geom_bar(fill = "darkslateblue") +
    theme_bw() + xlab(x) + ylab("Count")
}
for (i in 1:length(cat_names)){
  assign(paste0("plt",i),count.show(cat_names[i]))
}
grid.arrange(plt1,plt2,nrow =2)

grid.arrange(plt3,plt4,plt5,plt6,ncol = 2,nrow = 2)

grid.arrange(plt7,plt8,plt9,ncol = 2,nrow = 2)

#Compairing categories with mean of target variable(Bivariate analysis)
#Define function to draw bar plots
cat_show = function(x,y){
m = aggregate(absent_train[,x],by = list(category = absent_train[,y]),FUN = mean)
ggplot(m,aes(x = m$category,y = m$x)) + geom_bar(stat = "identity",fill = "Darksla
teBlue") + xlab(y) +
  ylab("Mean of Absenteeism.time.in.hours") +
  theme_bw()
}
cat_show(21,"ID")

cat_show(21,"Reason.for.absence")

cat_show(21,"Month.of.absence")

cat_show(21,"Day.of.the.week")

cat_show(21,"Seasons")

cat_show(21,"Disciplinary.failure")

cat_show(21,"Education")

cat_show(21,"Social.drinker")
```

```r
cat_show(21,"Social.smoker")

#**********************Outlier Analysis*************************************
**#
#BoxPlots - Distribution and Outlier Check

numeric_index = sapply(absent_train,is.numeric) #selecting only numeric
numeric_data = absent_train[,numeric_index]
cnames = colnames(numeric_data)

for (i in 1:length(cnames))
{
  assign(paste0("gn",i), ggplot(aes_string(y = (cnames[i])), data = subset(absent_
train))+
           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 time for",cnames[i])))
}


# ## Plotting plots together
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,gn12,ncol=3)

#Outlier values in each column
for (i in cnames){
  print(i)
  print(unique(boxplot.stats(absent_train[,i])$out))
}

#Replace outlier with NA's
for(i in cnames){
  val = absent_train[,i][absent_train[,i] %in% boxplot.stats(absent_train[,i])$out
]
  print(length(val))
  absent_train[,i][absent_train[,i] %in% val] = NA
}

#Imputing missing values resulting from outlier
#absent_train$Transportation.expense[2] = 118
#Actual value = 118
#Mean = 220.79
#median = 225
#knn = 137
#Mean Method
#absent_train$Transportation.expense[is.na(absent_train$Transportation.expense)] =
mean(absent_train$Transportation.expense,na.rm = T)
#Median method
#absent_train$Transportation.expense[is.na(absent_train$Transportation.expense)] =
median(absent_train$Transportation.expense,na.rm = T)
#KNN Imputation
absent_train = knnImputation(absent_train, k = 3)
```

```r
#we select knn as it best fits our data
#store clean file into the system
write.csv(absent_train, "clean_file_R.csv", row.names = F)
#*****************************Feature Selection*****************************
***#

#Correlation Plot

corrgram(absent_train[,numeric_index], order = F,
         upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")



#As weight and BMI are correlated, we will eliminate BMI as weight is the basic me
sure
## Dimension Reduction
absent_train = subset(absent_train, select = -Body.mass.index)

#*********************************Feature scaling*****************************
***#
#Normality check
par(mfrow = c(2,2))
qqnorm(absent_train$Transportation.expense,main = 'Transportation.expense')
qqnorm(absent_train$Distance.from.Residence.to.Work,main = 'Distance.from.Residenc
e.to.work')
qqnorm(absent_train$Service.time,main = 'Service.time')
qqnorm(absent_train$Absenteeism.time.in.hours,main = 'Absenteeism.time.in.hours')

par(mfrow = c(1,1))
# As our variables are not normally distributed, we will use normalization approac
h
#store cotinous variables name
numeric_index = sapply(absent_train,is.numeric)
numeric_data = absent_train[,numeric_index]
cnames = colnames(numeric_data)
#drop target variable
cnames = cnames[-11]
#Performing feature scaling with normalization
#Normalization formula
for(i in cnames){
  print(i)
  absent_train[,i] = (absent_train[,i] - min(absent_train[,i]))/
    (max(absent_train[,i] - min(absent_train[,i])))
}

#If the distribution is normal , we can use below formula for feature scaling
# #Standardisation
#for(i in cnames){
#print(i)
#absent_train[,i] = (absent_train[,i] - mean(absent_train[,i]))/
# sd(absent_train[,i])
#}

#Remove the unnecessary object
rmExcept("absent_train")

#Creating dummies for categorical variable
library(dummies)
#create list of categorical variables
Factor_index = sapply(absent_train,is.factor)
```

```r
Factor_data = absent_train[,Factor_index]
cnames = colnames(Factor_data)
#apply dummies package for one hot encoding store the result in data object
data = dummy.data.frame(absent_train, names = cnames)
#***********************************Sampling*********************************
******#
#Divide the data into train and test in ratio of 80:20 using simple random samplin
g
train_index = sample(1:nrow(data), 0.8 * nrow(data))
train = data[train_index,]
test = data[-train_index,]
#As PCA is unsupervised technique, we will remove target variable from both train
#and test and will save them in different object and later will readd them after
#performing PCA
#getting target variable from both train and test dataset
T_train = subset(train,select = Absenteeism.time.in.hours)
T_test = subset(test,select = Absenteeism.time.in.hours)
#drop target variable from both train and test
train = subset(train,select = -Absenteeism.time.in.hours)
test = subset(test,select = -Absenteeism.time.in.hours)
#***************************Principal component analysis************************
********#
prin_comp = prcomp(train)
names(prin_comp)

#outputs the mean of variables
prin_comp$center

#Dimension of data
dim(prin_comp$x)

#compute standard deviation of each principal component
std_dev = prin_comp$sdev
#compute variance
pr_var = std_dev^2
#check variance of first 5 components
pr_var[1:5]

#proportion of variance explained
prop_varex = (pr_var/sum(pr_var))
prop_varex[1:10]

#cumulative scree plot
plot(cumsum(prop_varex), xlab = "Principal Component",
     ylab = "Cumulative Proportion of Variance Explained",
     type = "b")

#Add a training set with principal components
train = data.frame(prin_comp$x)
#we are interested in first 45 PCAs as 95 % of variation is explained by first 45
PC's
train = train[,1:45]
#Add target variable back to train data
train = cbind(train,T_train)
#transform test into PCA
test  =  predict(prin_comp, newdata = test)
test = as.data.frame(test)
#select the first 45 components
test = test[,1:45]
#Add target variable back to test data
```

```r
test = cbind(test,T_test)
#*****************************Model Building*************************************
***#
set.seed(1234)
#*********************Decision tree (rpart) for regression************************
*******#
# building Decision tree
fit = rpart(Absenteeism.time.in.hours ~ ., data = train, method = "anova")
#plot decision tree
plot(fit)
text(fit,pretty = 0)

#Predict for test cases
predictions_DT = predict(fit, test[,-46])
#Evaluate performance of model with RMSE and R-squared value
print(postResample(pred = predictions_DT, obs = test[,46]))

#RMSE = 3.42
#Rsquared = 10.18
#*****************************Random Forest**************************************
********#
#Building random forest
fit_RF = randomForest(Absenteeism.time.in.hours ~., data = train)
fit_RF

#Predict for test cases
predictions_RF = predict(fit_RF,test[,1:45])
# Evaluate performance of model with RMSE and R-squared value(Result)
print(postResample(pred = predictions_RF, obs = test$Absenteeism.time.in.hours))

#RMSE = 2.67
#Rsquared = 35.35
#*************************Linear regression Model*********************************
*******#
#Building Linear Regression Model
lm_model = lm(Absenteeism.time.in.hours~.,data = train)
summary(lm_model)

#Plot linear regression model
par(mfrow = c(2,2))
plot(lm_model)
```

```r
##Building second linear model with square root transformation of dependent variab
le to improve it's normality
lm_model2 = lm(sqrt(Absenteeism.time.in.hours)~.,data = train)
summary(lm_model2)

#Plot linear regression model
par(mfrow = c(2,2))
plot(lm_model2)

#Predict for test cases as well as squaring prediction as we have taken square roo
t of target variable in train data
predictions_lm = (predict(lm_model2,test[,1:45]))^2
#Evaluate performance of model with RMSE and R-squared value
print(postResample(pred = predictions_lm, obs = test[,46]))

#RMSE = 2.67
#RSquared = 36.07
```

```r
#*****************************Monthly work loss*********************************
******#
#Get the clean file
loss = read.csv("clean_file_R.csv", header = T)
#check for any missing value
anyNA(loss)

#selct the column to calculate monthly work loss
Loss_df = loss[,c("Month.of.absence","Service.time","Work.load.Average.day.","Abse
nteeism.time.in.hours")]
#Formula to calculate monthly work loss
#work loss = (work.load.average.day * Absenteeism.time.in.hours)/(Service.time)
#create a new column with monthly_loss
Loss_df$Monthly.loss = (Loss_df$Work.load.Average.day. * Loss_df$Absenteeism.time.
in.hours)/(Loss_df$Service.time)
#Total work loss per month
Loss_2011 = aggregate(Loss_df$Monthly.loss, by = list(Month.of.absence = Loss_df$M
onth.of.absence),FUN = sum)
#Rename variable name
names(Loss_2011)[2]  ="Monthly.loss"
#Visualise monthly work loss
ggplot(Loss_2011,aes(x = Loss_2011$Month.of.absence,y = Loss_2011$Monthly.loss)) +
  geom_bar(stat = "identity",fill = 'orange') + xlab("Month.of.absence") + ylab("M
onthly_loss") +
  theme_bw() + ggtitle(" Possible Monthly work loss in 2011 ")
```

References

Jared P. Lander,2014. R for Everyone: Advanced Analytics and Graphics

Winston chang,2012.R Graphics Cookbook

www.analyticsvidhya.com, principal component analysis