Employee Absenteeism

Rishabh Tiwari

27th *July 2019*

Contents

Chapter 1

Introduction

1.1 Problem Statement

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

1. What changes company should bring to reduce the number of absenteeism?
2. How much losses every month can we project in 2011 if same trend of absenteeism continues?

1.2 Sample Data & Variables

There are 21 variables in our data in which 20 are independent variables and 1 (Absenteeism time in hours) is dependent variable.

Variable Information:
1. Individual identification (ID)

2. Reason for absence (ICD).
- Absences attested by the International Code of Diseases (ICD) stratified into 21 categories (I to XXI) as follows:

I Certain infectious and parasitic diseases
II Neoplasms
III Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism
IV Endocrine, nutritional and metabolic diseases
V Mental and behavioural disorders
VI Diseases of the nervous system
VII Diseases of the eye and adnexa
VIII Diseases of the ear and mastoid process
IX Diseases of the circulatory system
X Diseases of the respiratory system
XI Diseases of the digestive system
XII Diseases of the skin and subcutaneous tissue

XIII Diseases of the musculoskeletal system and connective tissue
XIV Diseases of the genitourinary system
XV Pregnancy, childbirth and the puerperium
XVI Certain conditions originating in the perinatal period
XVII Congenital malformations, deformations and chromosomal abnormalities
XVIII Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
XIX Injury, poisoning and certain other consequences of external causes
XX External causes of morbidity and mortality
XXI Factors influencing health status and contact with health services.
And 7 categories without (ICD) 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 (KMs)

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)

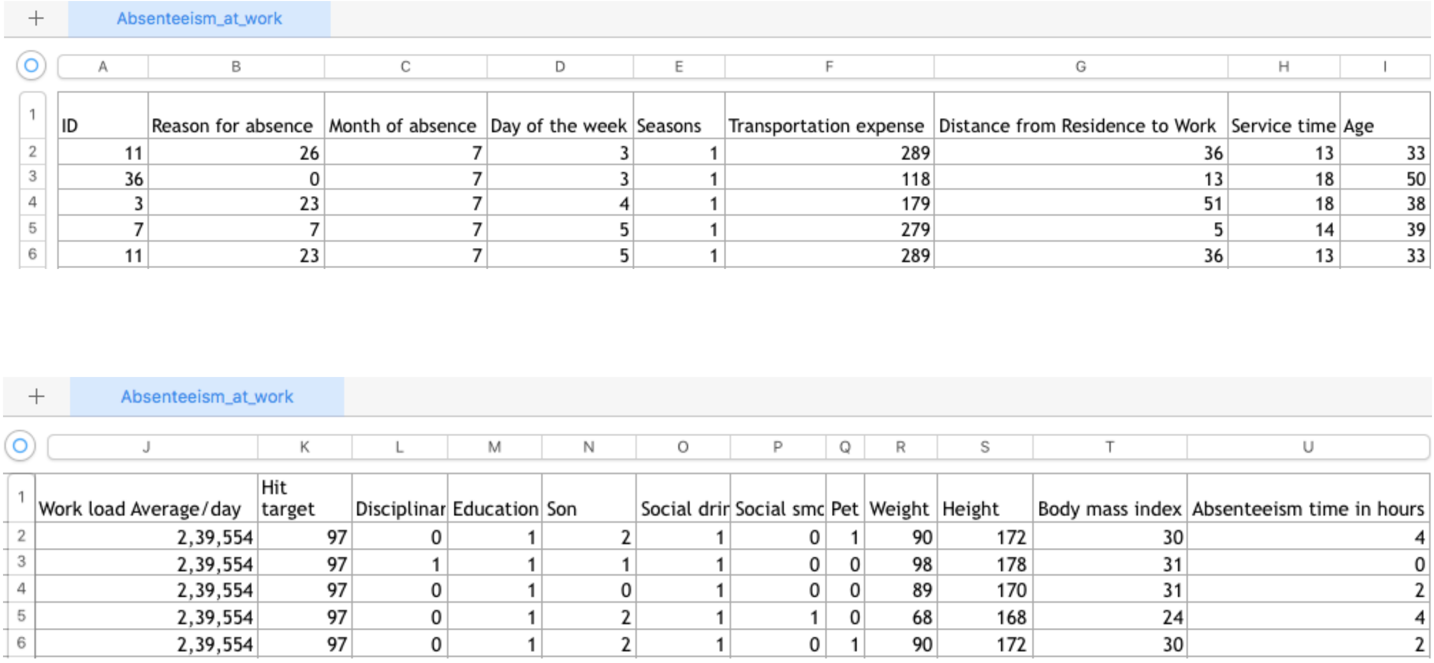
17. Pet (number of pet)

18. Weight

19. Height

20. Body mass index

21. Absenteeism time in hours (target)



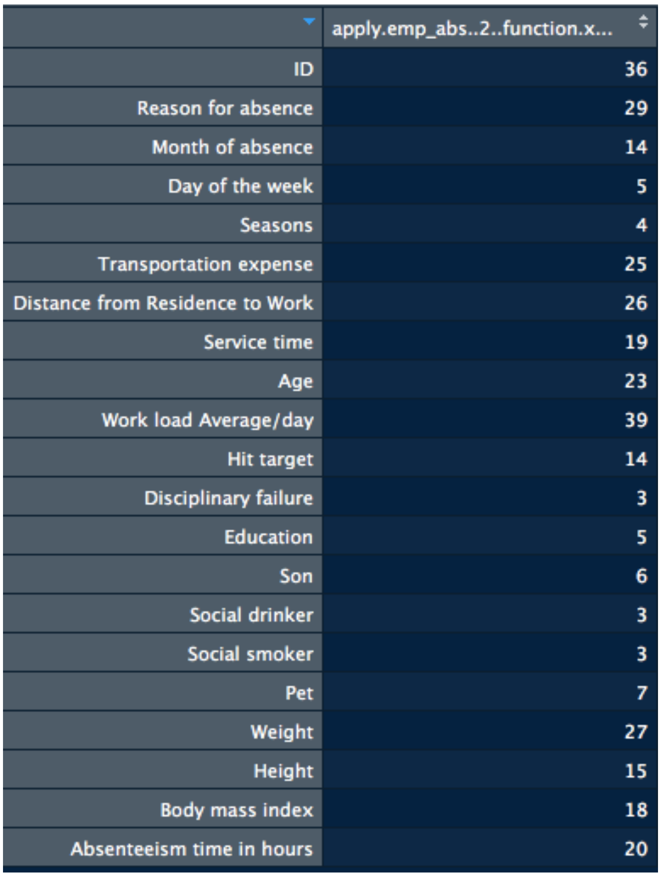
+ Absenteeism_at_work

	A	B	C	D	E	F	G	H	I
1	ID	Reason for absence	Month of absence	Day of the week	Seasons	Transportation expense	Distance from Residence to Work	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
6	11	23	7	5	1	289	36	13	33

+ Absenteeism_at_work

	J	K	L	M	N	O	P	Q	R	S	T	U
1	Work load Average/day	Hit target	Disciplinar	Education	Son	Social drir	Social smc	Pet	Weight	Height	Body mass index	Absenteeism time in hours
2	2,39,554	97	0	1	2	1	0	1	90	172	30	4
3	2,39,554	97	1	1	1	1	0	0	98	178	31	0
4	2,39,554	97	0	1	0	1	0	0	89	170	31	2
5	2,39,554	97	0	1	2	1	1	0	68	168	24	4
6	2,39,554	97	0	1	2	1	0	1	90	172	30	2

Fig. I [Sample Data]



	apply.emp_abs..2..function.x...
ID	36
Reason for absence	29
Month of absence	14
Day of the week	5
Seasons	4
Transportation expense	25
Distance from Residence to Work	26
Service time	19
Age	23
Work load Average/day	39
Hit target	14
Disciplinary failure	3
Education	5
Son	6
Social drinker	3
Social smoker	3
Pet	7
Weight	27
Height	15
Body mass index	18
Absenteeism time in hours	20

Fig. II [Unique Values Counts]

Chapter 2

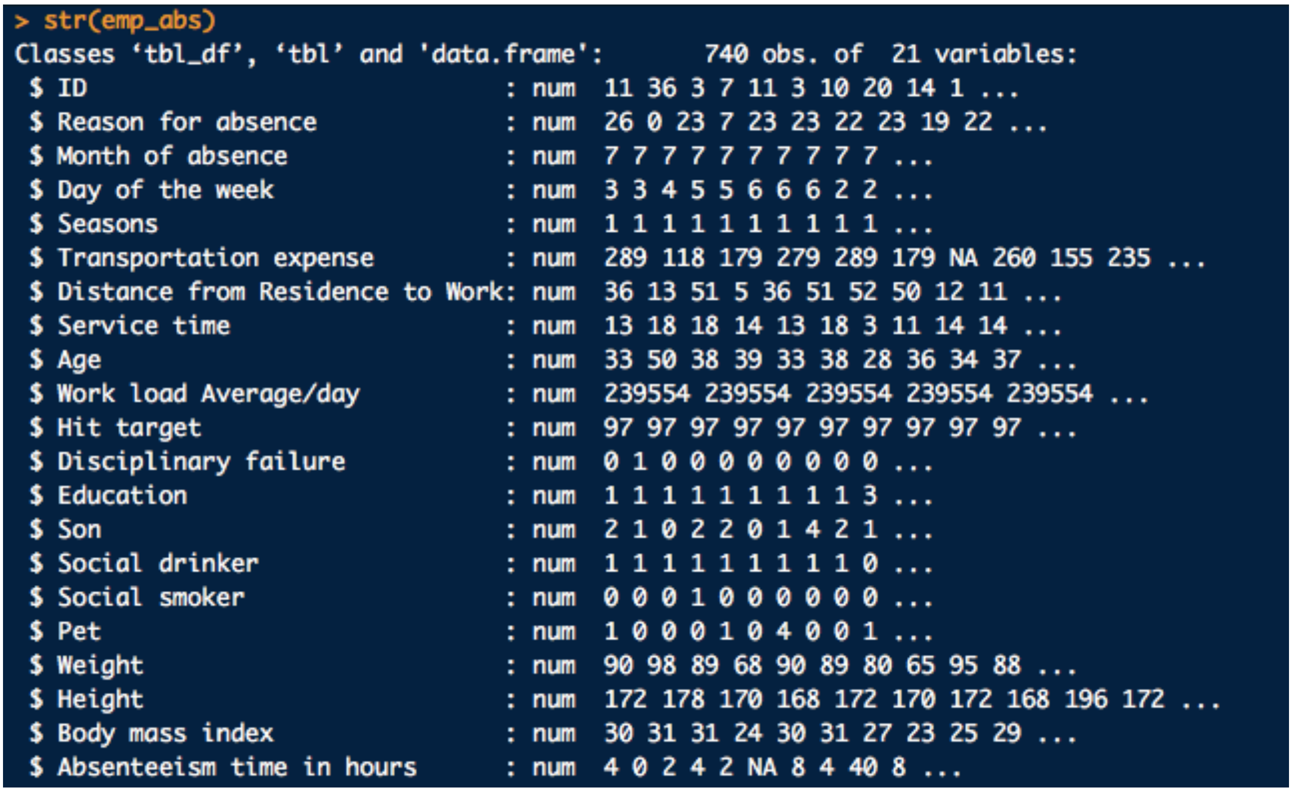
Methodology

2.1 Exploratory Data Analysis

Any predictive modelling requires a scrutiny of the data before we start modelling. However, in data mining terms, looking at data refers to exploring the data, cleaning the data as well as visualising the data through graphs and plots. This is often called as **Exploratory Data Analysis.**

Under **Exploratory Data Analysis,** we check for the structure of the dataset, whether the data is in proper shape, or entries are as per the defined datatype, i.e. numeric, character, factor, or list.

> str(dfmain)



```
> str(emp_abs)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':	740 obs. of  21 variables:
 $ ID                             : num  11 36 3 7 11 3 10 20 14 1 ...
 $ Reason for absence             : num  26 0 23 7 23 23 22 23 19 22 ...
 $ Month of absence               : num  7 7 7 7 7 7 7 7 7 7 ...
 $ Day of the week                : num  3 3 4 5 5 6 6 6 2 2 ...
 $ Seasons                        : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Transportation expense         : num  289 118 179 279 289 179 NA 260 155 235 ...
 $ Distance from Residence to Work: num  36 13 51 5 36 51 52 50 12 11 ...
 $ Service time                   : num  13 18 18 14 13 18 3 11 14 14 ...
 $ Age                            : num  33 50 38 39 33 38 28 36 34 37 ...
 $ Work load Average/day          : num  239554 239554 239554 239554 239554 ...
 $ Hit target                     : num  97 97 97 97 97 97 97 97 97 97 ...
 $ Disciplinary failure           : num  0 1 0 0 0 0 0 0 0 0 ...
 $ Education                      : num  1 1 1 1 1 1 1 1 1 3 ...
 $ Son                            : num  2 1 0 2 2 0 1 4 2 1 ...
 $ Social drinker                 : num  1 1 1 1 1 1 1 1 1 0 ...
 $ Social smoker                  : num  0 0 0 1 0 0 0 0 0 0 ...
 $ Pet                            : num  1 0 0 0 1 0 4 0 0 1 ...
 $ Weight                         : num  90 98 89 68 90 89 80 65 95 88 ...
 $ Height                         : num  172 178 170 168 172 170 172 168 196 172 ...
 $ Body mass index                : num  30 31 31 24 30 31 27 23 25 29 ...
 $ Absenteeism time in hours      : num  4 0 2 4 2 NA 8 4 40 8 ...
```

Fig. III

As per the mentioned problem statement and detailed dataset, we required to convert certain variables datatypes and add some variables as they are, inspite of being categorical variable, stored as numeric or converted. We will implement this further under Feature Engineering.

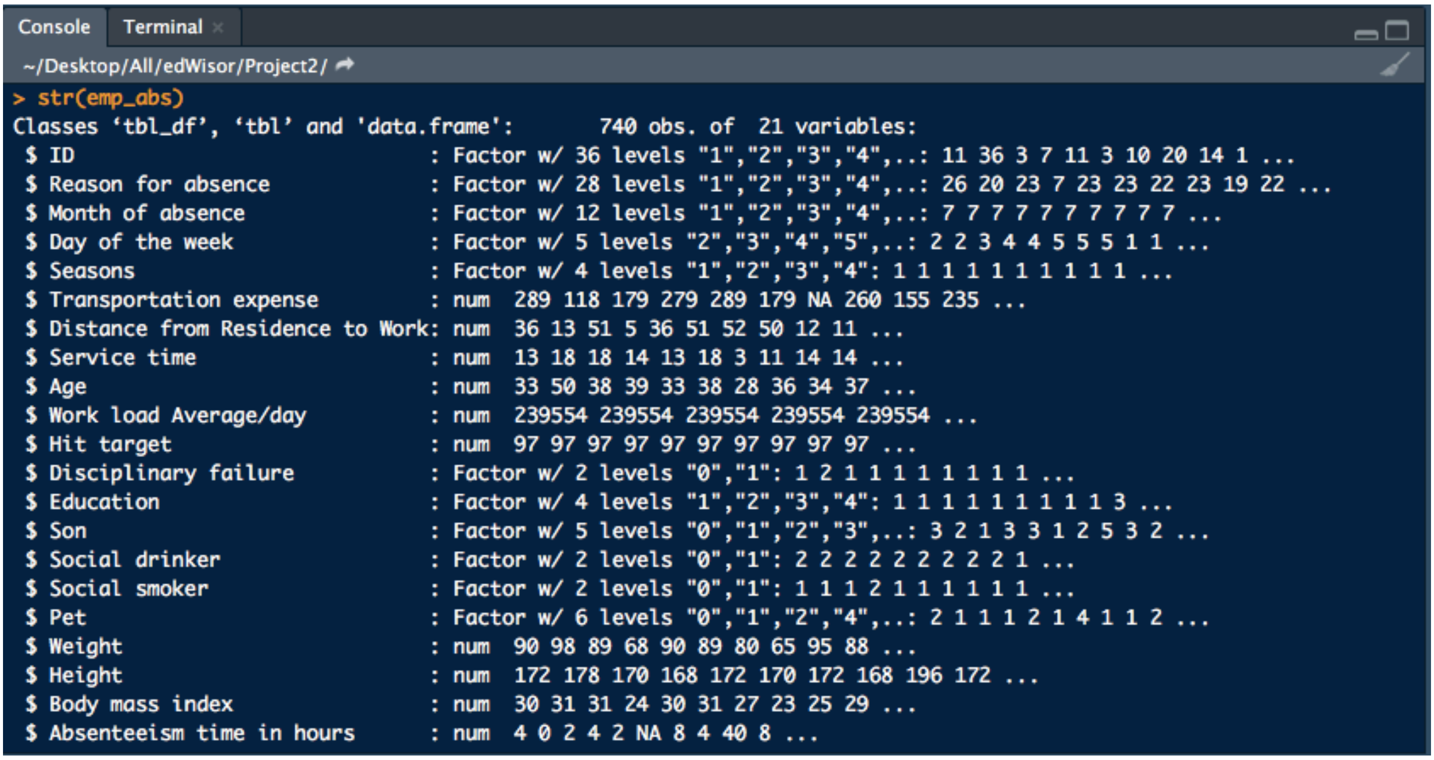
2.1.1 Feature Engineering:

Feature engineering refers to a process of selecting and transforming variables when creating a predictive model using machine learning or statistical modelling. The process involves a combination of data analysis, applying rules of thumb, and judgement. It is sometimes referred to as pre-processing, although that term can have a more general meaning.

We check for any incorrect entries in the data. After scrutiny we get to know that there isn't any entries with category '20' in 'Reason of Absence' variable. There are 46 entries in 'Reason of absence' with such entries and we replace them with corresponding category '20' in the employee dataset. Similarly, there were few wrong entries in 'Month of absence' with entries '0', which were replaced by 'NA'.

Further, as per the given description of the dataset variables, variables **'ID', 'Reason for absence', 'Month of absence', 'Day of the week', 'Seasons', 'Disciplinary failure', 'Education', 'Son', 'Social drinker', 'Social smoker' & 'Pet'** are converted to factor datatypes, as they are the categorical variables present in the given dataset.

> str (emp_abs)



```
> str(emp_abs)
Classes 'tbl_df', 'tbl' and 'data.frame':	740 obs. of  21 variables:
 $ ID                             : Factor w/ 36 levels "1","2","3","4",..: 11 36 3 7 11 3 10 20 14 1 ...
 $ Reason for absence             : Factor w/ 28 levels "1","2","3","4",..: 26 20 23 7 23 23 22 23 19 22 ...
 $ Month of absence               : Factor w/ 12 levels "1","2","3","4",..: 7 7 7 7 7 7 7 7 7 7 ...
 $ Day of the week                : Factor w/ 5 levels "2","3","4","5",..: 2 2 3 4 4 5 5 5 1 1 ...
 $ Seasons                        : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
 $ Transportation expense         : num  289 118 179 279 289 179 NA 260 155 235 ...
 $ Distance from Residence to Work: num  36 13 51 5 36 51 52 50 12 11 ...
 $ Service time                   : num  13 18 18 14 13 18 3 11 14 14 ...
 $ Age                            : num  33 50 38 39 33 38 28 36 34 37 ...
 $ Work load Average/day          : num  239554 239554 239554 239554 239554 ...
 $ Hit target                     : num  97 97 97 97 97 97 97 97 97 97 ...
 $ Disciplinary failure           : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
 $ Education                      : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 3 ...
 $ Son                            : Factor w/ 5 levels "0","1","2","3",..: 3 2 1 3 3 1 2 5 3 2 ...
 $ Social drinker                 : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 1 ...
 $ Social smoker                  : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
 $ Pet                            : Factor w/ 6 levels "0","1","2","4",..: 2 1 1 1 2 1 4 1 1 2 ...
 $ Weight                         : num  90 98 89 68 90 89 80 65 95 88 ...
 $ Height                         : num  172 178 170 168 172 170 172 168 196 172 ...
 $ Body mass index                : num  30 31 31 24 30 31 27 23 25 29 ...
 $ Absenteeism time in hours      : num  4 0 2 4 2 NA 8 4 40 8 ...
```

Fig. IV

2.2 Data Pre-Processing

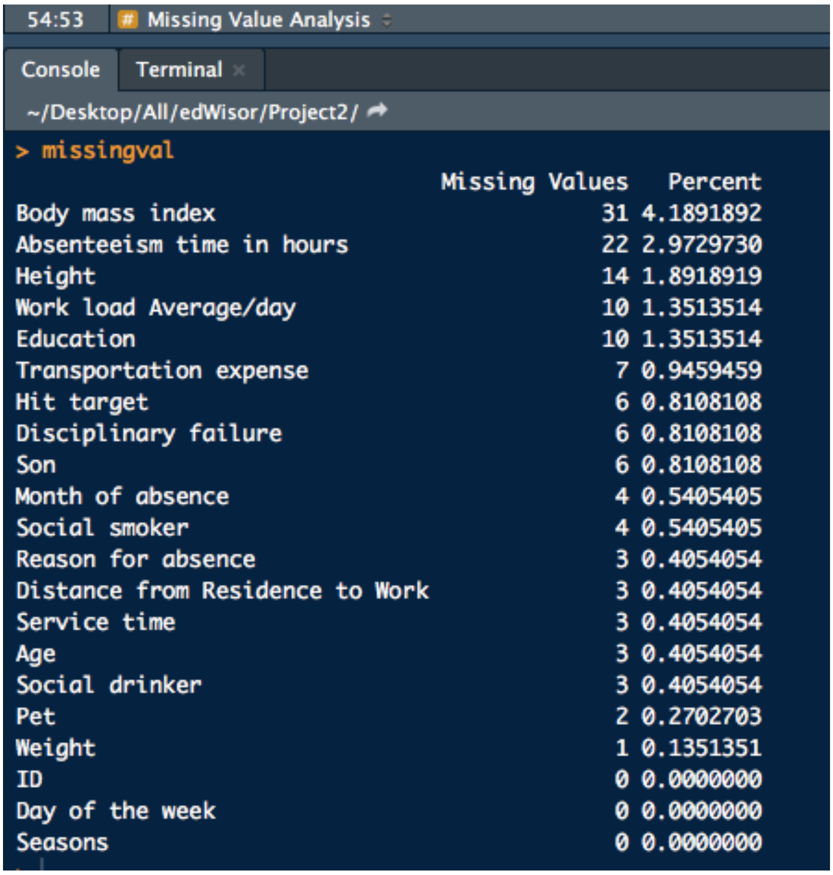
Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviours or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing. Data Pre-Processing is necessary and also plays a major role before model development.

Under data pre-processing, we check for Missing-Values first, followed by Outlier Analysis and Feature Selection.

2.2.1 Missing Value Analysis:

The concept of missing values is important to understand in order to successfully manage data. If the missing values are not handled properly then we may end up drawing an inaccurate inference about the data. Due to improper handling, the result obtained will differ from ones where the missing values are present. In case the percentage of the missing values present in the variable exceeds 30%, we simply drop that variable from our dataset.

We checked for the missing values percentage in the given dataset. As a result, none of the columns have a high percentage of missing values. The maximum missing percentage is 4.18% for Body Mass Index variable.



```
> missingval
                                 Missing Values   Percent
Body mass index                              31 4.1891892
Absenteeism time in hours                    22 2.9729730
Height                                       14 1.8918919
Work load Average/day                        10 1.3513514
Education                                    10 1.3513514
Transportation expense                        7 0.9459459
Hit target                                    6 0.8108108
Disciplinary failure                          6 0.8108108
Son                                           6 0.8108108
Month of absence                              4 0.5405405
Social smoker                                 4 0.5405405
Reason for absence                            3 0.4054054
Distance from Residence to Work               3 0.4054054
Service time                                  3 0.4054054
Age                                           3 0.4054054
Social drinker                                3 0.4054054
Pet                                           2 0.2702703
Weight                                        1 0.1351351
ID                                            0 0.0000000
Day of the week                               0 0.0000000
Seasons                                       0 0.0000000
```

Fig. V

The distribution of the missing values percentage is given below:

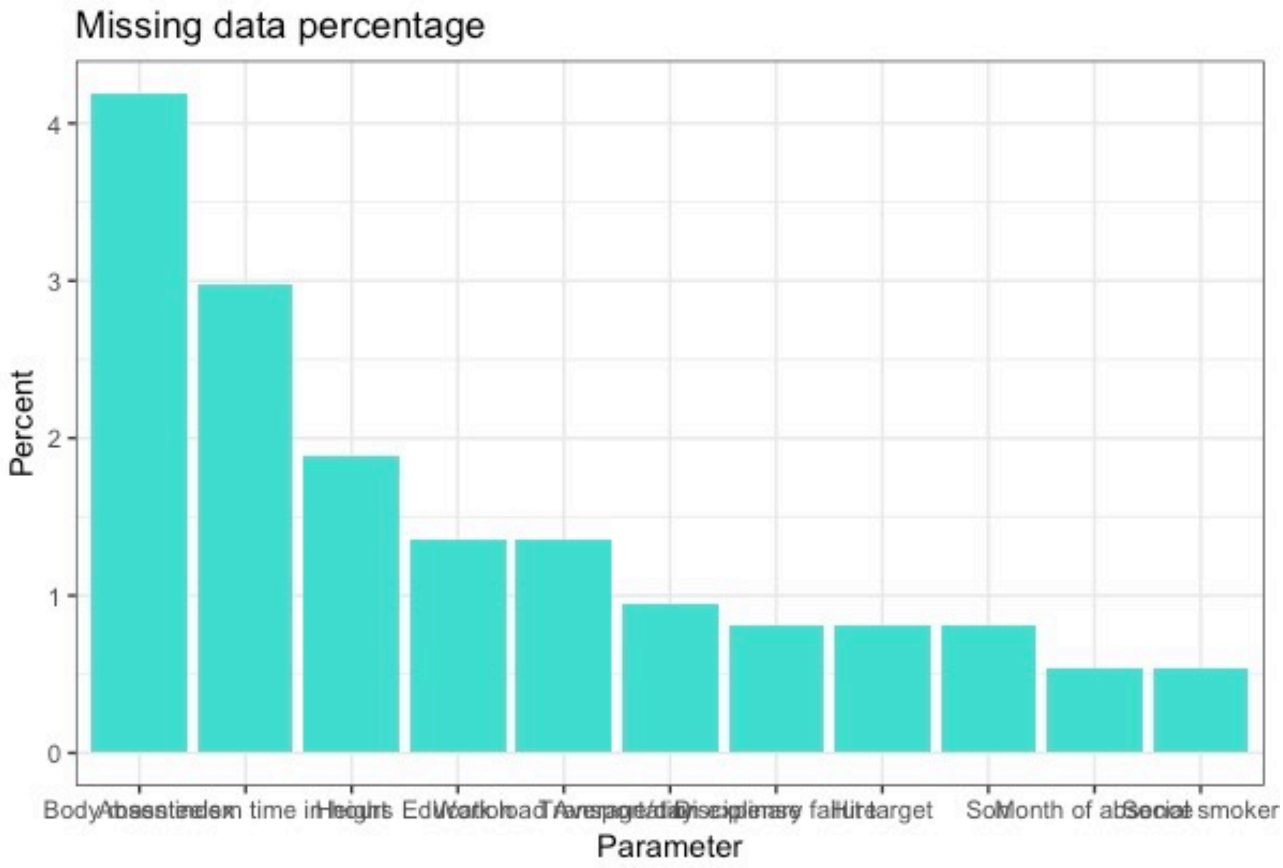


Fig. VI

To replace the missing values, we compute different methods like, Mean, Median and KNN Imputation Method. We select the best possible method which gives the closest result for the missing values. The missing values have been computed using KNN computation method.

2.2.2 Outlier Analysis:

An O**utlier** is a data point that differs significantly from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error. Outliers can occur by chance in any distribution, but they often indicate either measurement error or that the population has a heavy-tailed distribution.

After implementing Boxplots for each non-categorical variables, we found out Outliers were present in all the variables except ‘Distance from residence to work’, ‘Weight’ and ‘Body mass index’.

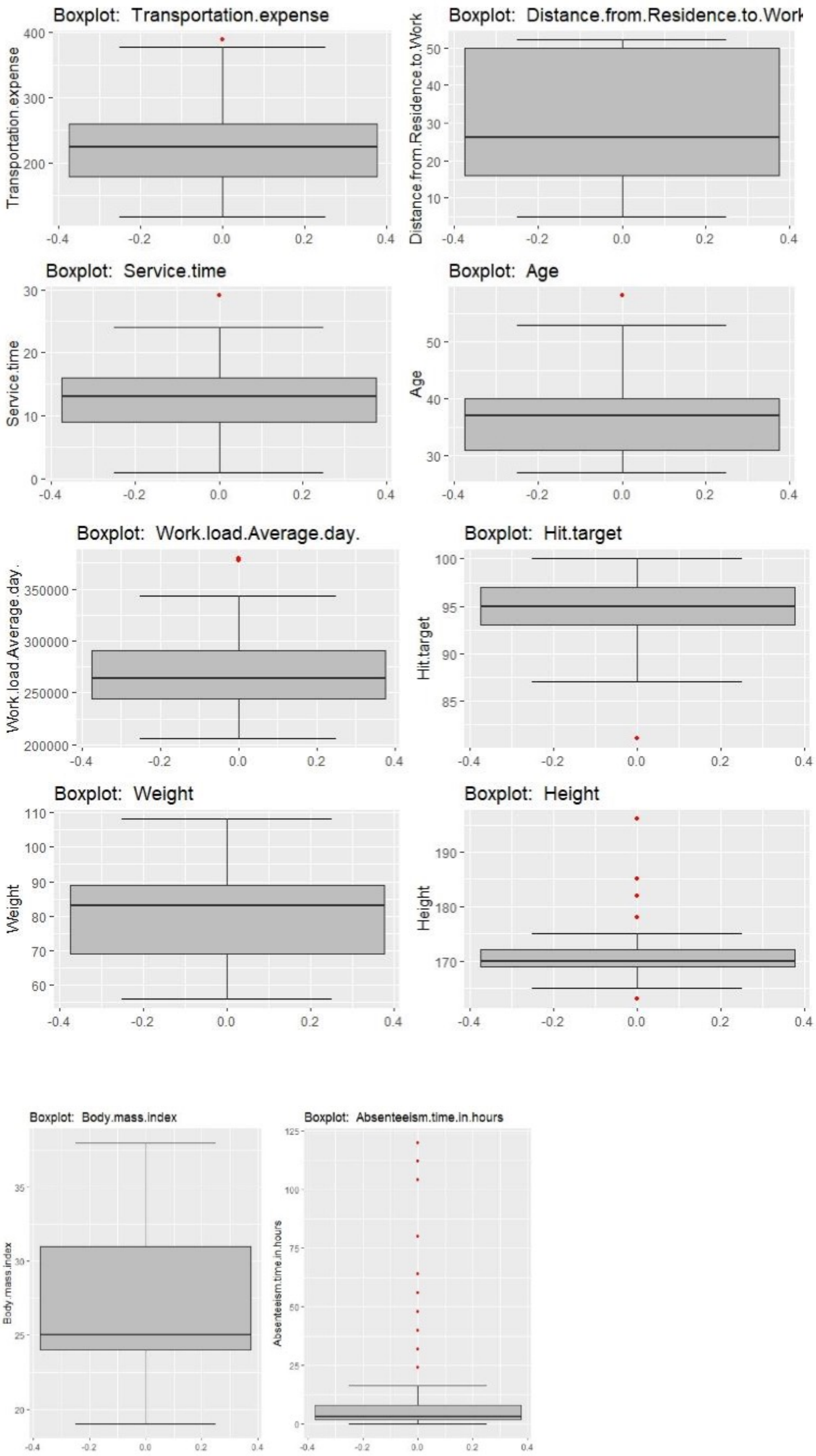
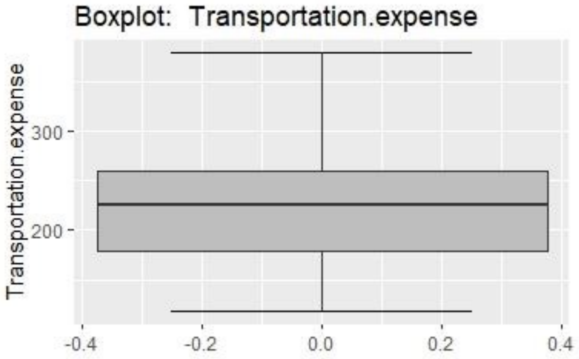
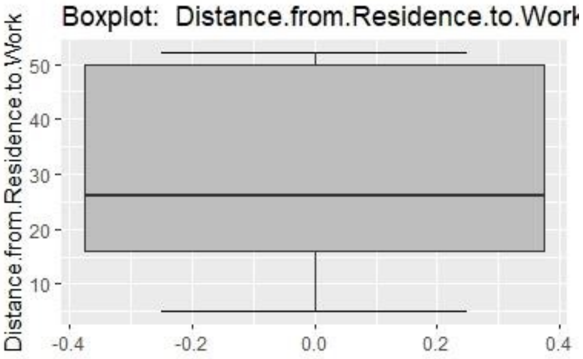
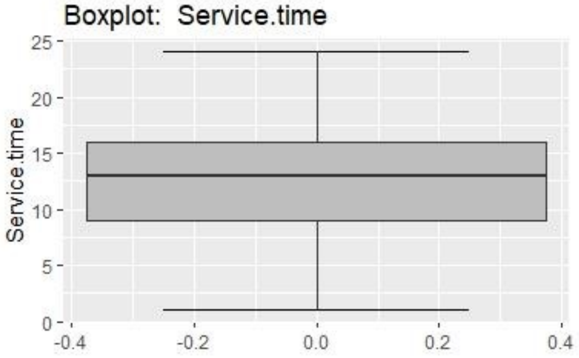


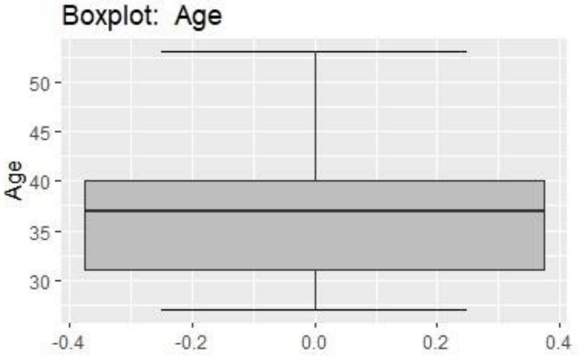
Fig. VII (with Outliers)

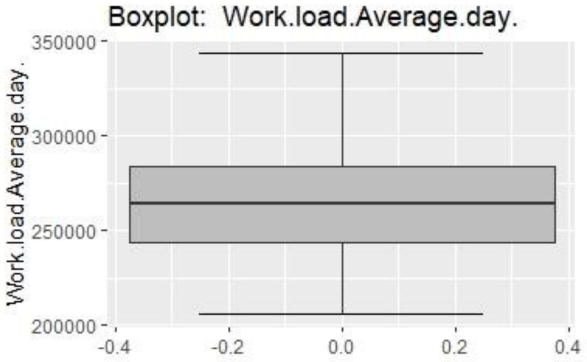
Missing values as obtained and observed above are first converted to NA values, so that we can impute such values under missing value analysis. Hence, after converting, we replace them by using KNN imputation method, as it came out to be the most accurate.
Below is the distribution of the variables after removing the outliers:

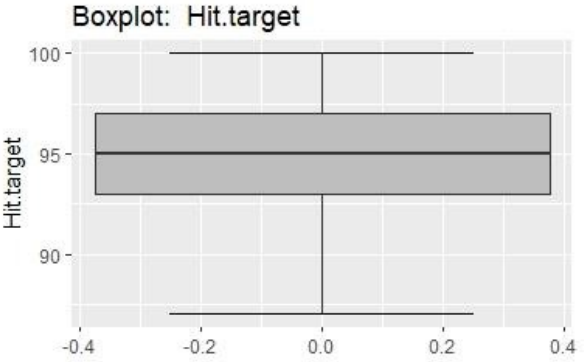


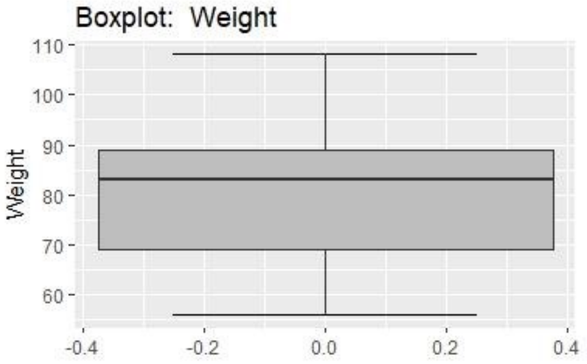


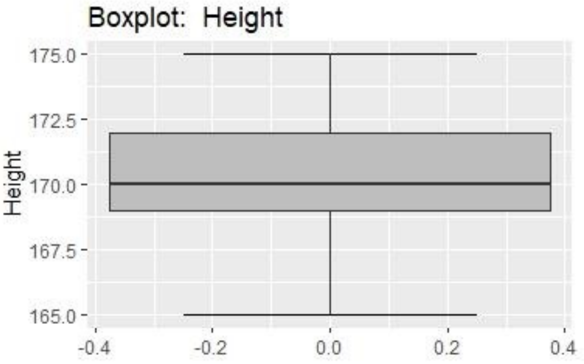












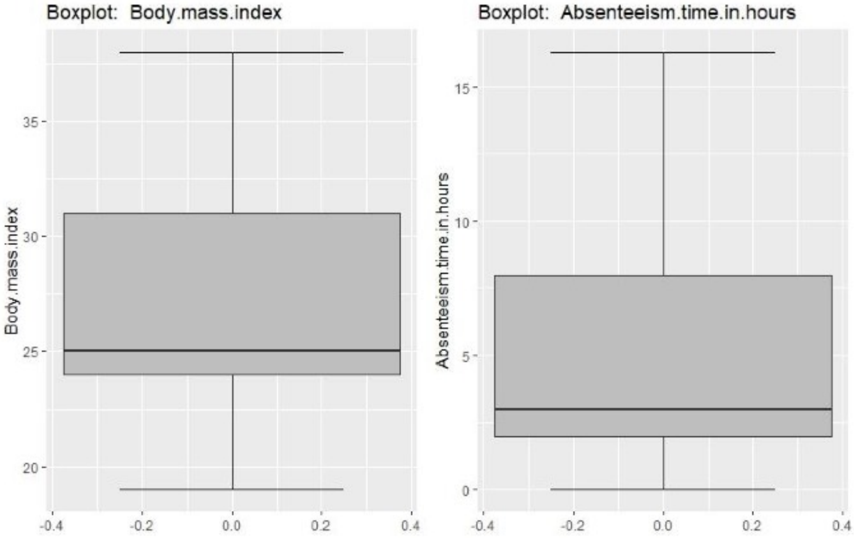
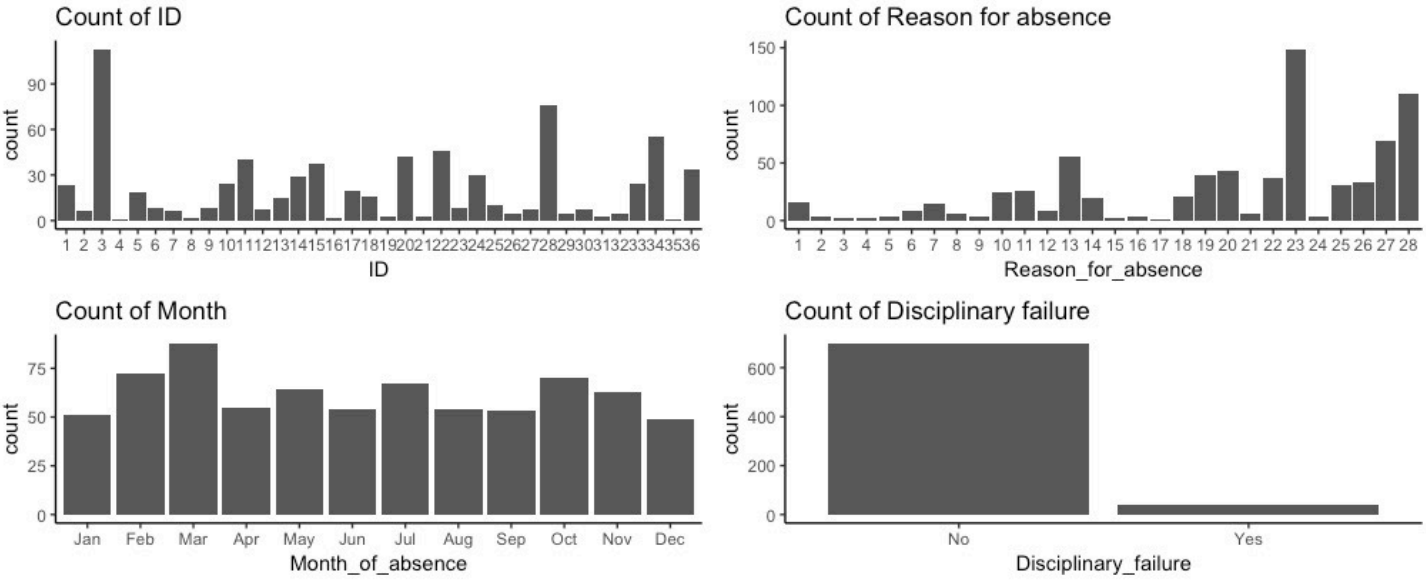


Fig. VIII
(after removing Outliers)

2.2.3 Variable Distribution:

Before further exploratory data analysis, we prefer observing the pattern and frequency distribution of all the variables of our dataset. Hence, we plotted the bar-plots for all the categorical variables, whereas for continuous variables we have plotted the histograms.
By observing the distributions charts, it can be easily seen that the variables are normally distributed.

For Categorical Variables, the bar-plot distributions are as follows:



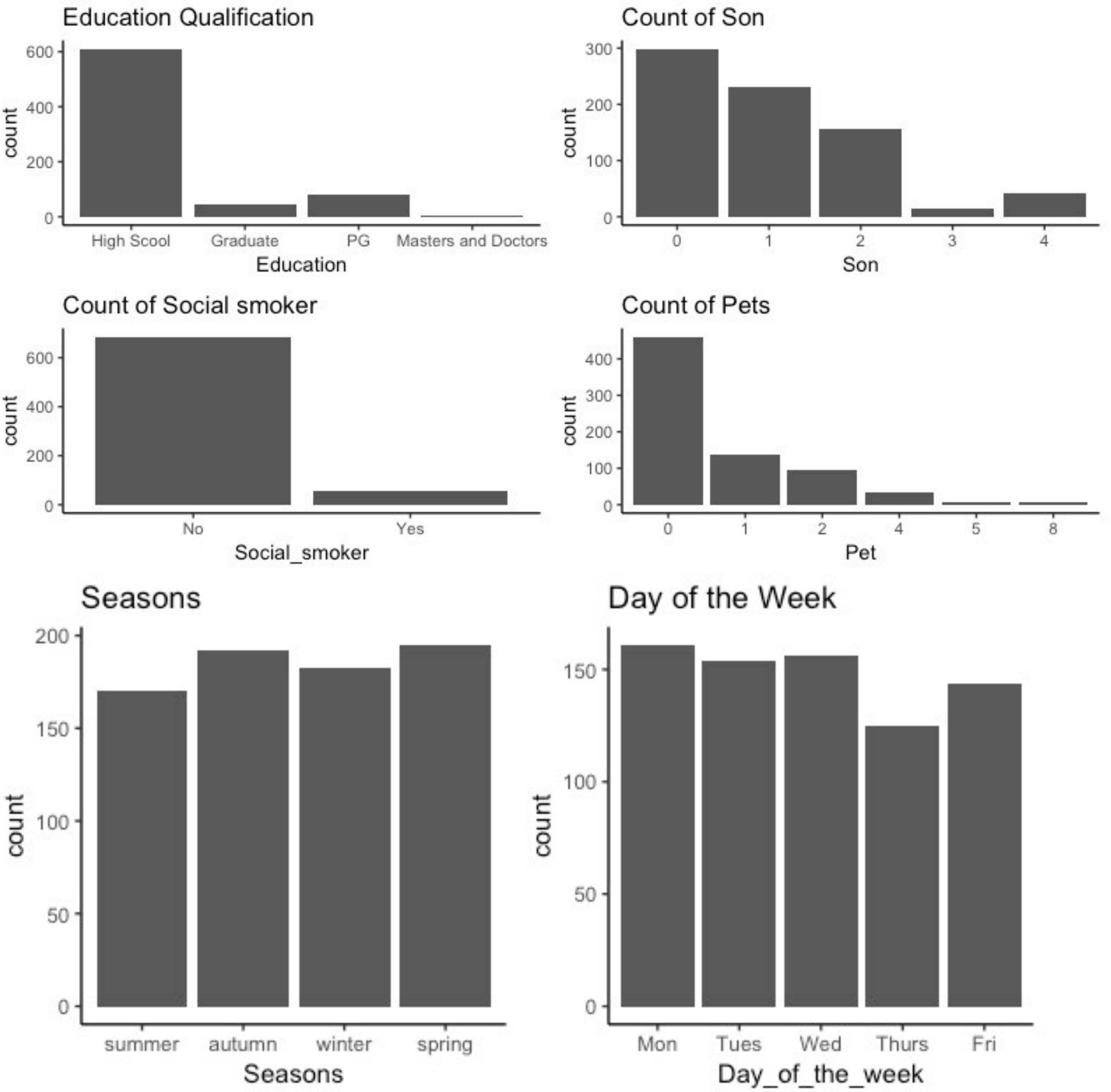
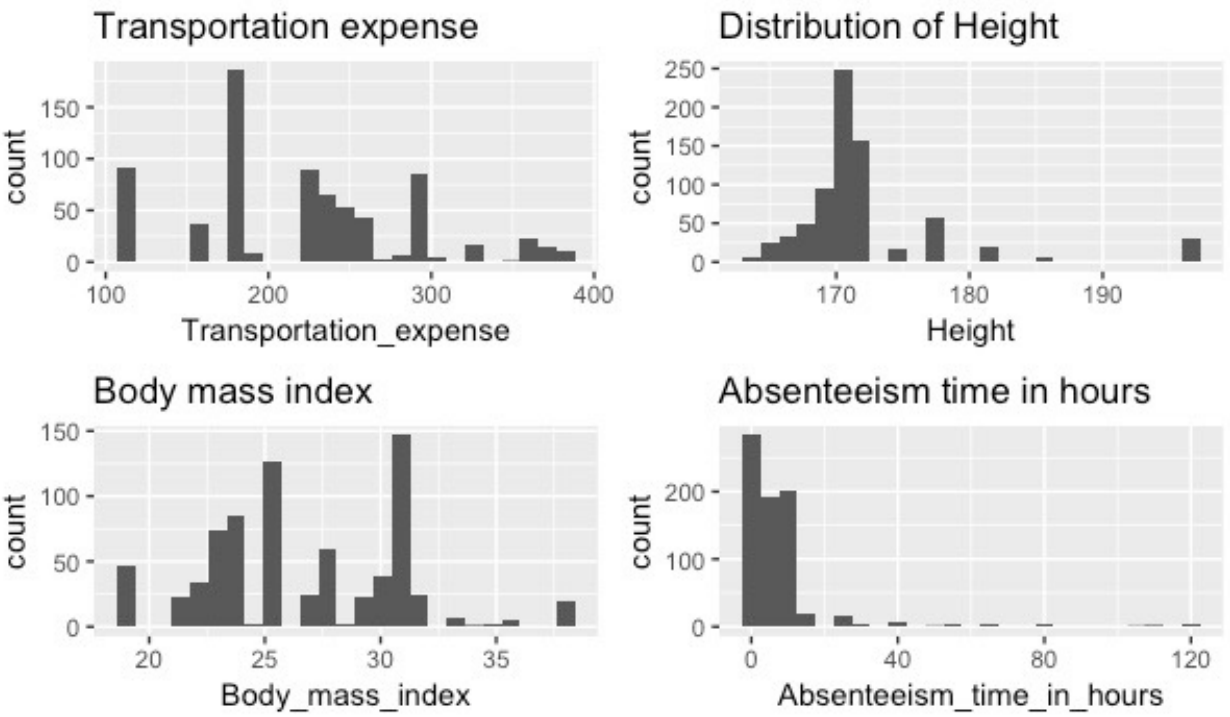
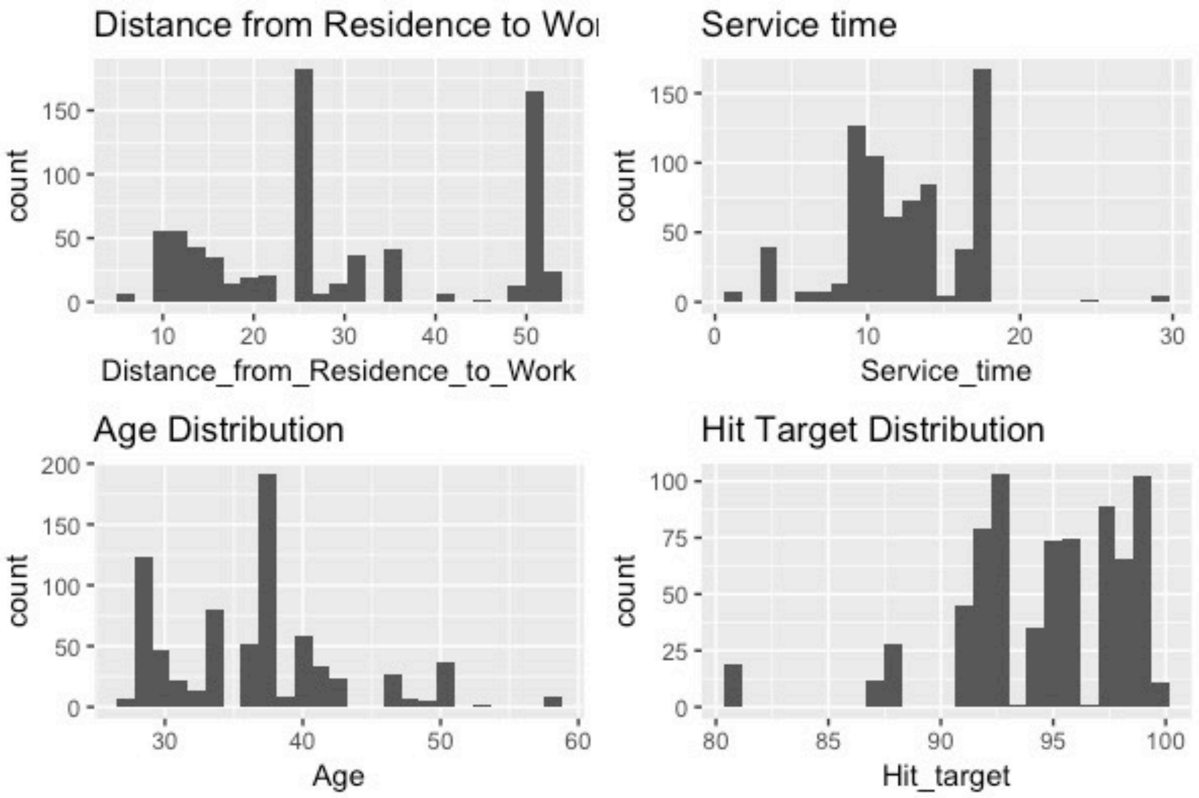


Fig. IX
(Categorical Variables Distribution)

For Continuous Variables, the histogram plot distributions are as follows:





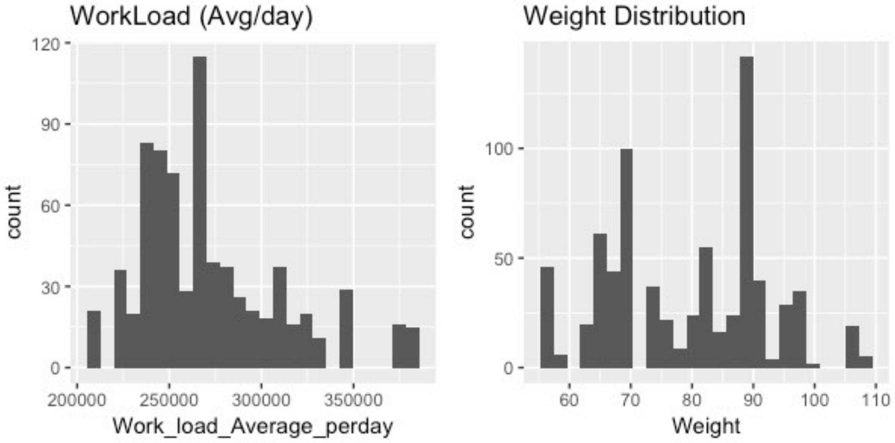


Fig. X
(Continuous Variables Distribution)

2.2.4 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.

For Numerical variables, we check for the dependencies between the variables. To do so, we go for the correlation analysis and plotting the correlational plot for all the numeric variables.

The correlation plot for all the nominal numeric variables is given below:

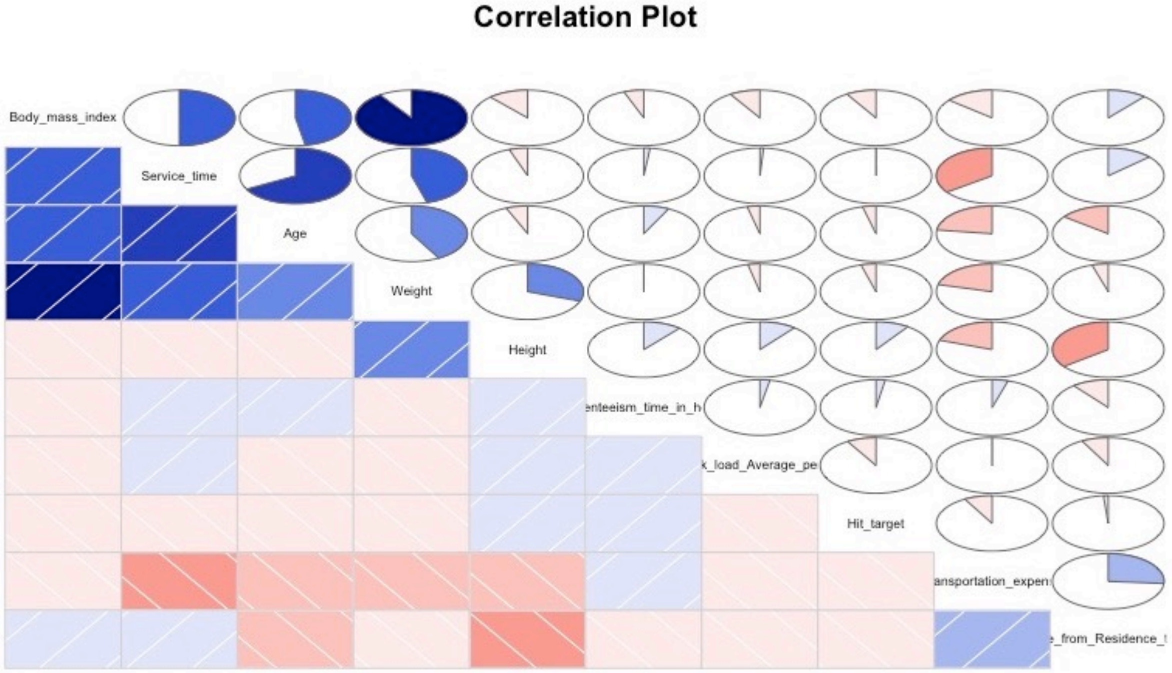


Fig. XI
(Correlation Plot)

From correlation analysis we found out that **Weight** and **Body Mass Index** has high correlation, so we have excluded the **Body Mass Index** variable from out dataset.

2.2.5 Feature Scaling :

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing and is also known as data-normalisation or data-standardisation.
Most 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 feature. Therefore, the range of all features should be normalised so that each feature contributes proportionately to the final distance. Since our data is not uniformly distributed, we will use **Normalization as Feature Scaling Method.**

2.2.6 Principal Component Analysis :

Principal component analysis (**PCA**) is a procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called **principal components**. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.

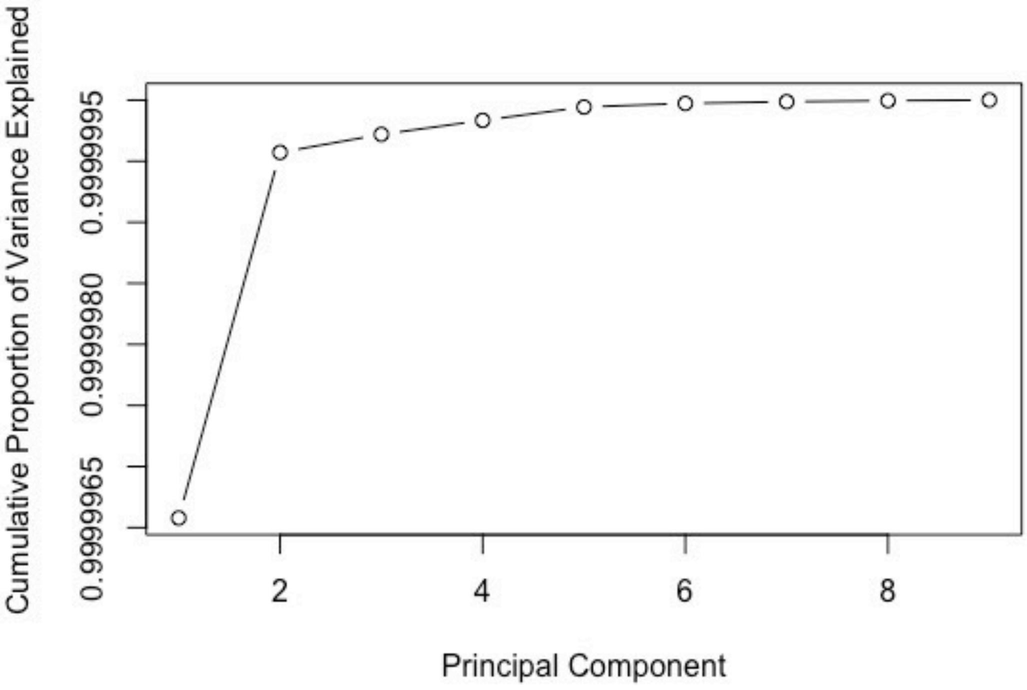


Fig. XII

After applying PCA algorithm and observing the Cumulative Scree Plot, it can be observed that almost 95% of the data can be explained by 45 variables out of 116. Hence, we choose only 45 variables as input to the models.

Chapter 3

Model Development

3.1 Selection

Data modelling is a process that uses data mining and probability to forecast outcomes. Each model is made up of a number of predictors, which are variables that are likely to influence future results. Once data has been collected for relevant predictors, a statistical model is formulated. The model may employ a simple linear equation, or it may be a complex neural network, mapped out by sophisticated software.

As per the structure of our variables distribution and our target variable in the dataset which is a continuous variable (Absenteeism_time_in_hours), we choose and implemented Linear Regression, Decision Tree and Random Forest algorithms on our dataset. The error metric chosen to select the best suitable model for our predictive analysis is Root Mean Square Error (RMSE).

3.2 Decision Tree

Decision Tree algorithm belongs to the family of supervised learning algorithms. Decision trees are used for both classification and regression problems.
A decision tree is a tree where each node represents a feature(attribute), each link(branch) represents a decision(rule) and each leaf represents an outcome (categorical or continues value). The general motive of using Decision Tree is to create a training model which can use to predict class or value of target variables by learning decision rules inferred from prior data (training data).

The RMSE values and R^2 values for the given project in R and Python are:

DECISION TREE	**RMSE**	**R^2**
R	0.442	0.978
PYTHON	0.0353	0.9998

3.3 Random Forest

Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because it's simplicity and the fact that it can be used for both classification and regression tasks. We implemented the same in Python to check and compare the model development as per our desired prediction model for bike rent.

RANDOM FOREST	RMSE	R^2
R	0.480	0.978
PYTHON	0.0445	0.9998

3.4 Linear Regression

It is rare that a dependent variable is explained by only one variable. In this case, an we use multiple regression, which attempts to explain a dependent variable using more than one independent variable. Multiple regressions can be linear and nonlinear. Multiple regressions are based on the assumption that there is a linear relationship between both the dependent and independent variables. It also assumes no major correlation between the independent variables.

We implemented the Multiple Linear regression model on our train data to train the model as per our dataset. After training, we implemented the same on our test data to test our model.

LINEAR REGRESSION	RMSE	R^2
R	0.003	0.9999
PYTHON	0.0013	0.9999

Chapter 4

Conclusion

4.1 Model Evaluation

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and overfitted models.

Now that we have a models implemented for predicting our target variable, we need to decide which will be best suitable for our problem statement. There are several criteria that exist for evaluating and comparing the models. We can compare the models using any of the following criteria:

- Predictive Performance
- Interpretability
- Computational Efficiency

In our case of Employee Absenteeism, we prefer using Predictive Performance, as the latter two doesn't hold much significance.
Under Predictive Performance analysis, we subjected Root Mean Square Error (RMSE) and R-Squared Value of different models. RMSE is a measure of how spread out the residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Lower values of RMSE and higher value of R-Squared Value indicates best fit model.

4.2 Best Model

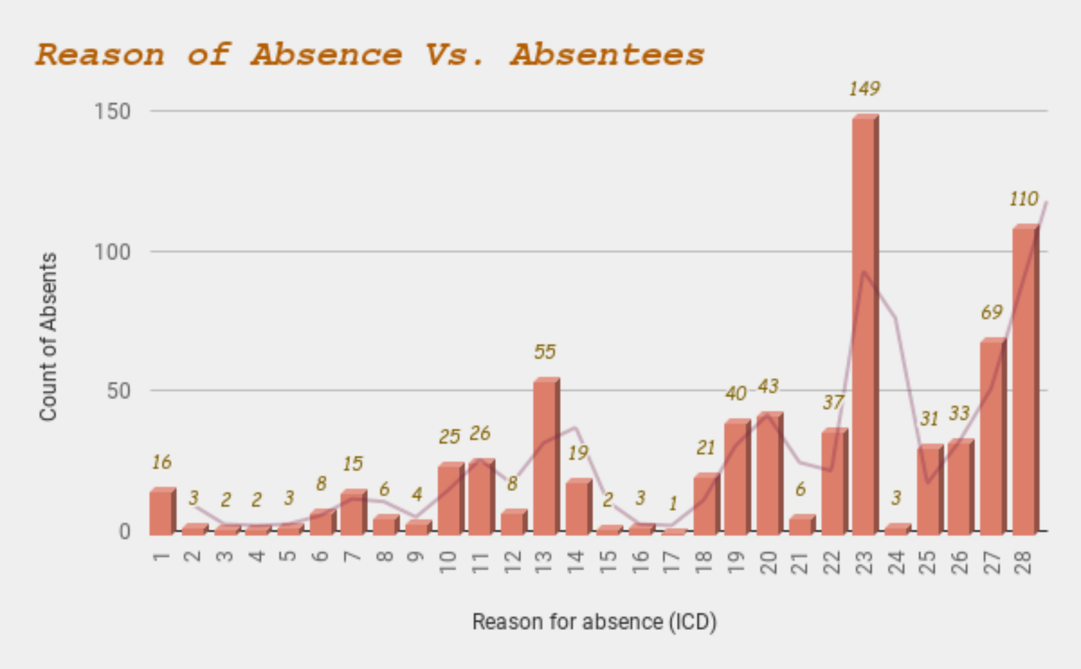
From the observation of all RMSE Value and R-Squared Value we have concluded that **Linear Regression Model** has minimum value of RMSE and its R-Squared Value is also maximum.

4.3 Solutions of Problem Statement

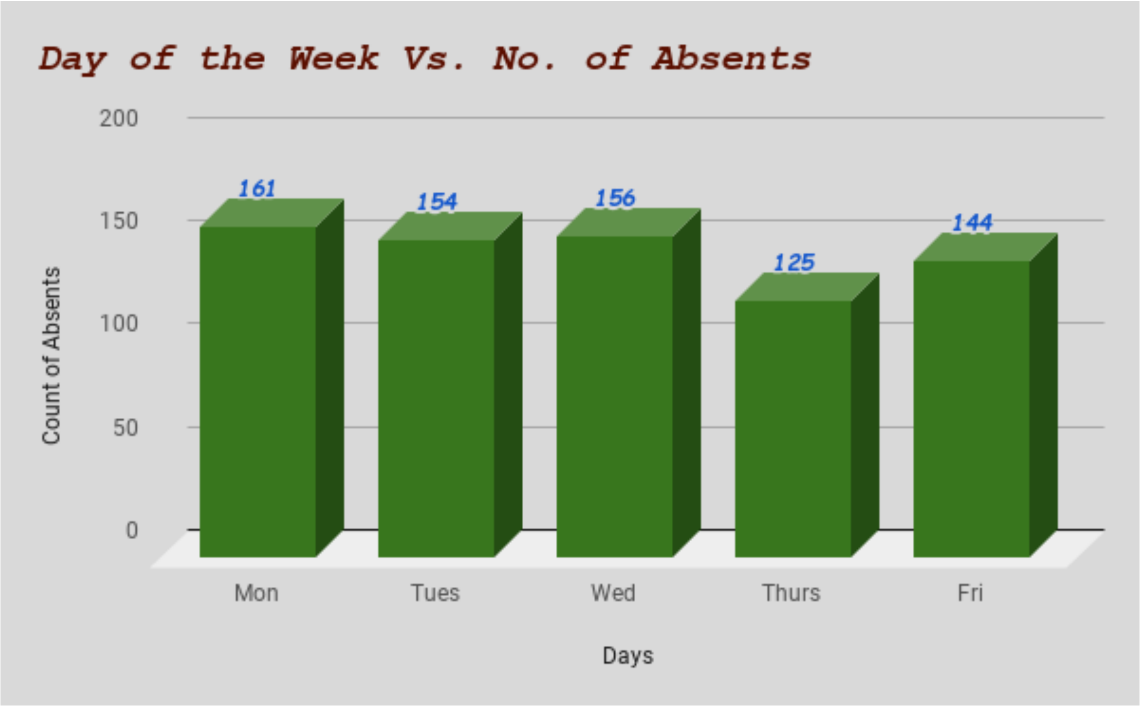
4.3.1 What changes company should bring to reduce the number of absenteeism?

Solution:

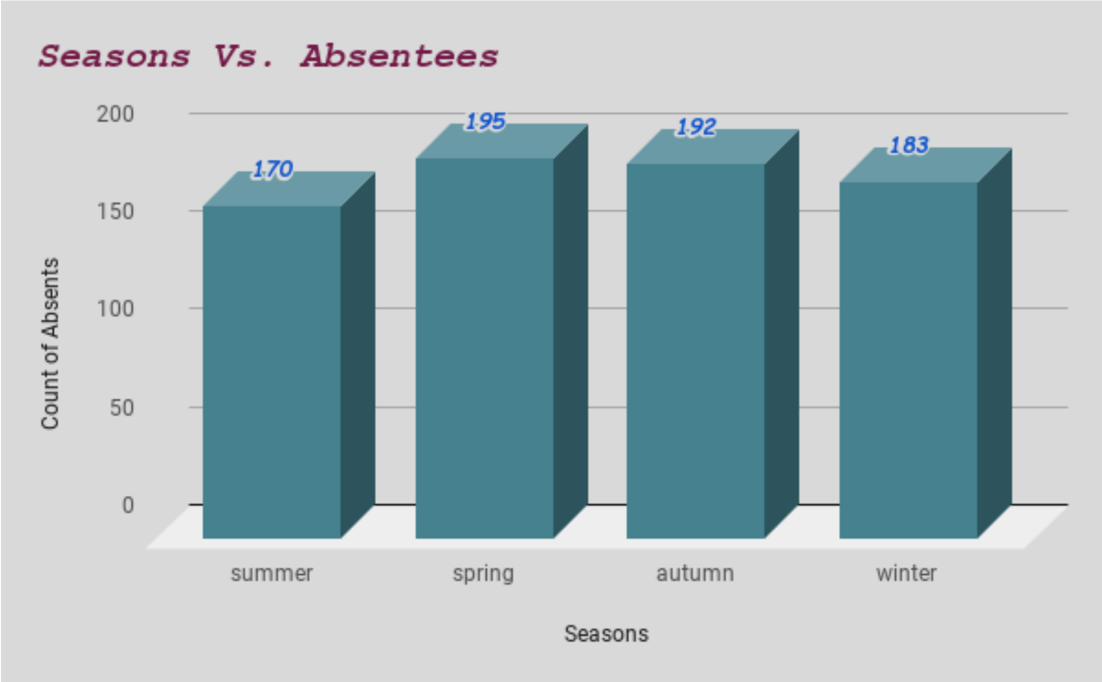
[A] The reasons under the ICD 13, 20, 23 and 28 are the most common and used categories of diseases taken by employees to be absent.These reasons include Medical consultation, Dental appointments, morbidity, mortality and diseases of musculoskeletal system and connective tissue. Respective company can help in informing pre cautious and safety measures to employees on how to keep themselves healthier.



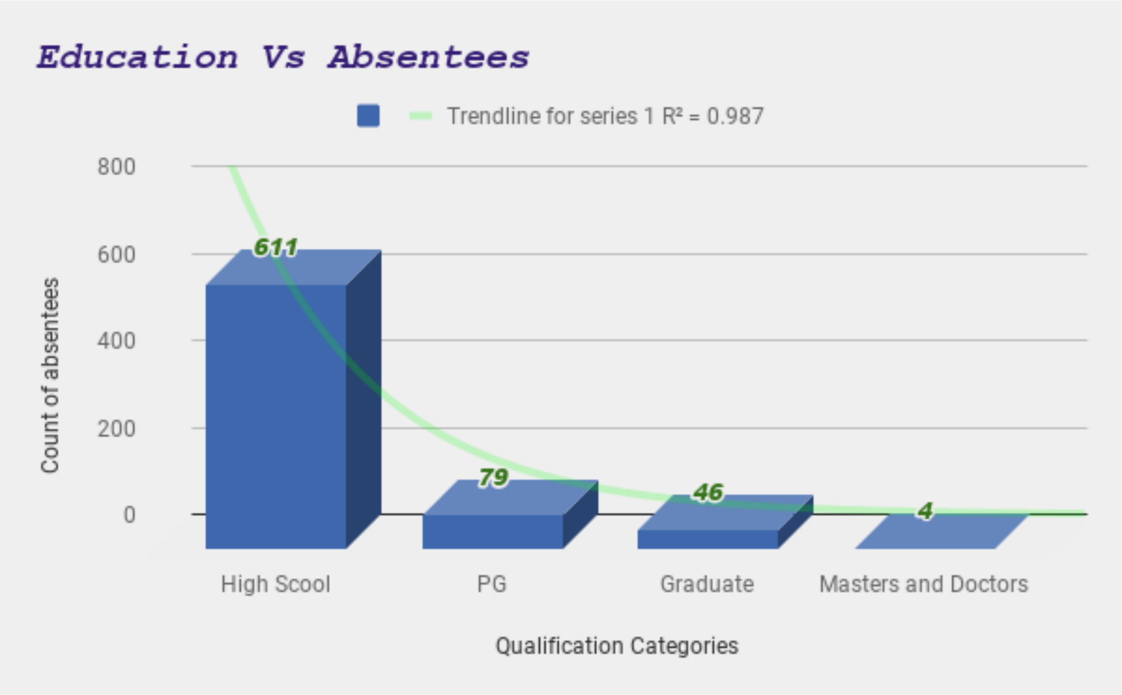
[B] Employees are mostly absent on Mondays followed by Tuesdays. XYZ can inform employees to not take as many absent hours on such days.



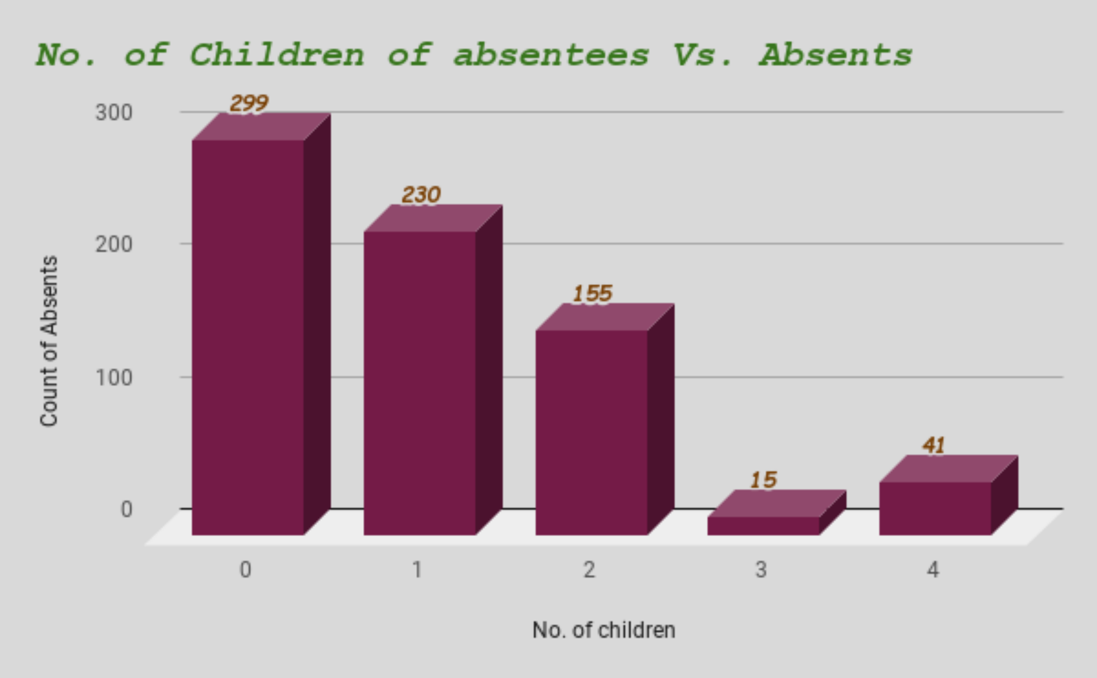
[C] Maximum employees are absent during Spring Season.



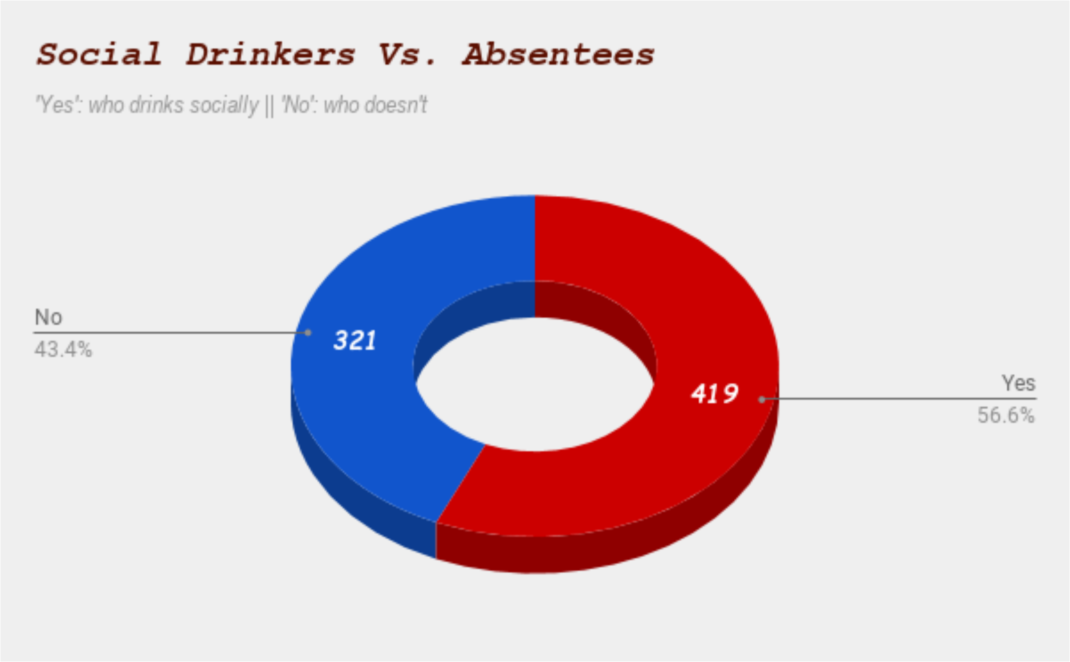
[D] Employees having education only till high school tend to take more absent hours than others. Hence, the company can either hire employees who are graduated from college or guide the employees to be atleast high school education qualified, to reduce the number of abseentism.



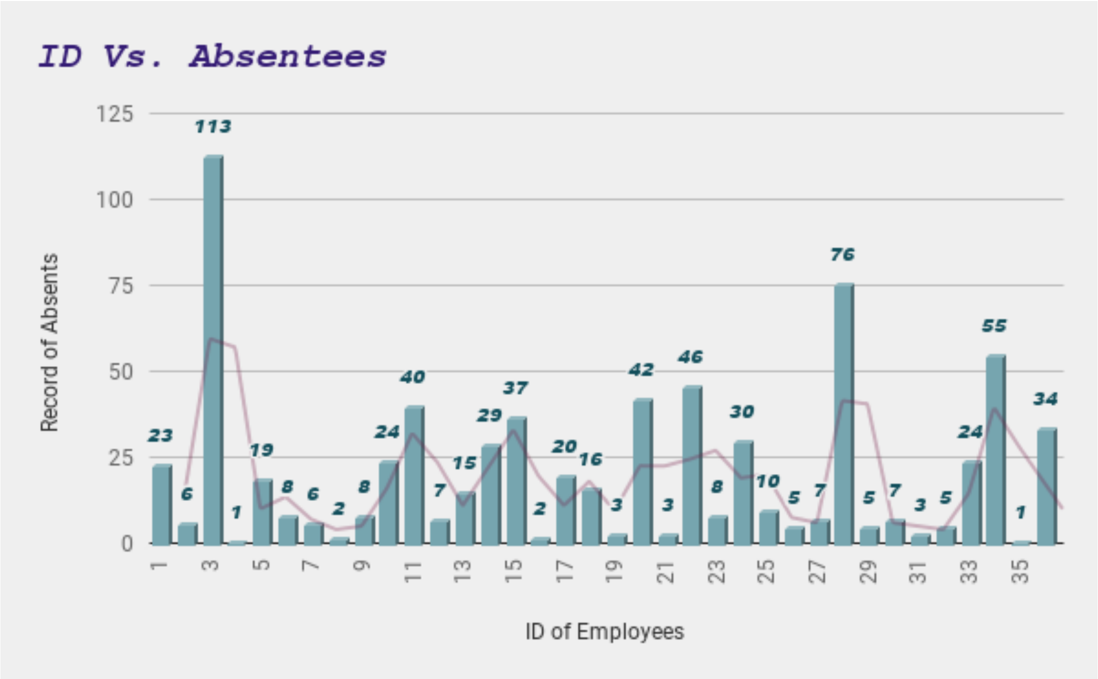
[E] Employees having two children or no child at all are usually more in the list of absenteeism.



[F] People having tendency to drink more socially are more absent than those who don’t drink. Hence, the company XYZ should keep a scrutiny of those people and inform or guide such to reduce alcohol intake during working days.



[G] Employees with ID 3, 28 and 34 are few of the employees who are majorly and exceptionally absent with respect to others. The



company should warn such employees to reduce being absent a lot, else severe actions may be taken against them.

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

Solution:

Considering the losses to be the absenteeism time in hours, if the same trend of absenteeism continues, then the total total losses per month is as shown in the graph below.
Employees are absent the most in the month of March, with total Absenteeism hours equal to 458.2 hours. Employees are absent the least in the month of January, with total Absenteeism hours equal to 173.6.

Below table shows the monthly losses of absenteeism hours:

Month	**Absent Hours**
January	173.6
February	275.4
March	458.2
April	244.7
May	266.7
June	251
July	375.8
August	254.3
September	190.2
October	295.2
November	266
December	200.3

Chapter 5

RCode

```
rm(list = ls())

#setting working directory
setwd("/Users/rishi/Desktop/All/edWisor/Project2")
getwd()
##################################Package
Installation#####################################
#installing required packages
install.packages("readr", "readxl",
"MLmetrics","plyr","dplyr","ggplot2","rpart","DMwR","randomForest","usdm","corrgram","DataC
ombine","xlsx")
l <- c("readr", "readxl",
"MLmetrics","plyr","dplyr","ggplot2","rpart","DMwR","randomForest","usdm","corrgram","DataC
ombine","xlsx")
libraries(l)

#Importing the Dataset
emp_abs <- read_excel("Absenteeism_at_work_Project.xls")
emp_abs_dummy <- read_excel("Absenteeism_at_work_Project.xls")
View(emp_abs)
str(emp_abs)

#Checking for unique values count in each variable
unique_counts <- data.frame(apply(emp_abs, 2, function(x){length(unique(x))}))
View(unique_counts)

#Feature Engineering (converting categorical variables)
colnames(emp_abs)
emp_abs$ID <- as.factor(emp_abs$ID)

emp_abs$`Reason for absence`[emp_abs$`Reason for absence` == 0] <- 20
emp_abs$`Reason for absence` <- as.factor(emp_abs$`Reason for absence`)
levels(emp_abs$`Reason for absence`)

emp_abs$`Month of absence`[emp_abs$`Month of absence` == 0] <- NA
emp_abs$`Month of absence` <- as.factor(emp_abs$`Month of absence`)

emp_abs$`Day of the week` <- as.factor(emp_abs$`Day of the week`)
emp_abs$Seasons <- as.factor(emp_abs$Seasons)
emp_abs$`Disciplinary failure` <- as.factor(emp_abs$`Disciplinary failure`)
emp_abs$Education <- as.factor(emp_abs$Education)
emp_abs$Son <- as.factor(emp_abs$Son)
emp_abs$`Social drinker` <- as.factor(emp_abs$`Social drinker`)
emp_abs$`Social smoker` <- as.factor(emp_abs$`Social smoker`)
emp_abs$Pet <- as.factor(emp_abs$Pet)

str(emp_abs)
#making a copy of dataset
df_emp_abs <- emp_abs
```

```
############################## Missing Value Analysis
#####################################
#Getting number of missing values in each variables
missingval <- data.frame(apply(df_emp_abs, 2, function(x){sum(is.na(x))}))
missingval
missingval$Variables <- row.names(missingval)
names(missingval)[1] <- "Missing Values"
missingval$Percent <- (missingval$`Missing Values`/nrow(df_emp_abs)) *100 #getting missing
value percentage
missingval <- missingval[order(-missingval$Percent),]  #sorting as per percentage
missingval

#plotting missing values graph
ggplot(data = missingval[1:11,], aes(x=reorder(Variables, -Percent),y = Percent))+
  geom_bar(stat = "identity",fill = "turquoise")+xlab("Parameter")+
  ggtitle("Missing data percentage") + theme_bw()


#Best method to compute Missing Values
#creating Missing Value
View(df_emp_abs$`Body mass index`)
df_emp_abs$`Body mass index`[10] <- NA #Actual Value = 29

df_emp_abs$`Body mass index`[10] = mean(df_emp_abs$`Body mass index`, na.rm = T)
df_emp_abs$`Body mass index`[10] #By Mean Method = 26.680

df_emp_abs$`Body mass index`[10] = median(df_emp_abs$`Body mass index`, na.rm = T)
df_emp_abs$`Body mass index`[10] #By Median Method = 25

df_emp_abs <- knnImputation(df_emp_abs, k = 5) #By KNNImputation = 29


##################################### OUTLIER ANALYSIS
######################################
#Getting only numeric variables
numeric_var <- sapply(df_emp_abs, is.numeric)
numeric_data <- df_emp_abs[,numeric_var]

#Getting only categorical variables
category_data <- df_emp_abs[,!numeric_var]

#Check for outliers using boxplots
for(i in 1:ncol(numeric_data)) {
  assign(paste0("box",i), ggplot(data = df_emp_abs, aes_string(y = numeric_data[,i])) +
          stat_boxplot(geom = "errorbar", width = 0.5) +
          geom_boxplot(outlier.colour = "red", fill = "grey", outlier.size = 1) +
          labs(y = colnames(numeric_data[i])) +
          ggtitle(paste("Boxplot: ",colnames(numeric_data[i]))))
}

gridExtra::grid.arrange(box1, box2, box3, box4, ncol(4))
gridExtra::grid.arrange(box5,box6,box7,box8, ncol(4))
gridExtra::grid.arrange(box9,box10,ncol(2))

#Replacing Outliers
for(i in numeric_columns){
  val = df_emp_abs[,i][df_emp_abs[,i] %in% boxplot.stats(df[,i])$out]
  print(paste(i,length(val)))
  df_emp_abs[,i][df_emp_abs[,i] %in% val] = NA
}
```

```
sapply(df_emp_abs,function(x){sum(is.na(x))})

#Missing values i.e. NA after replacing outliers
out_missingval <- data.frame(sapply(df_emp_abs,function(x){sum(is.na(x))}))
out_missingval$Columns <- row.names(out_missingval)
row.names(out_missingval) <- NULL
names(out_missingval)[1] <- "miss_percentage"
out_missingval$miss_percentage <- ((out_missingval$miss_percentage/nrow(emp_absent)) *100)
out_missingval <- out_missingval[order(-out_missingval$miss_percentage),]
out_missingval

#KNN imputation on all NA values
df_emp_abs <- knnImputation(df_emp_abs, k = 5)
sum(is.na(df))

############################ Graphical Representation
#################################
#Distribution of categorical variables
#changing variables names by removing spaces
names(df_emp_abs)[2] <- "Reason_for_absence"
names(df_emp_abs)[3] <- "Month_of_absence"
names(df_emp_abs)[4] <- "Day_of_the_week"
names(df_emp_abs)[6] <- "Transportation_expense"
names(df_emp_abs)[7] <- "Distance_from_Residence_to_Work"
names(df_emp_abs)[8] <- "Service_time"
names(df_emp_abs)[10] <- "Work_load_Average_perday"
names(df_emp_abs)[11] <- "Hit_target"
names(df_emp_abs)[12] <- "Disciplinary_failure"
names(df_emp_abs)[15] <- "Social_drinker"
names(df_emp_abs)[16] <- "Social_smoker"
names(df_emp_abs)[20] <- "Body_mass_index"
names(df_emp_abs)[21] <- "Absenteeism_time_in_hours"

#setting the levels of the variables for plotting
levels(df_emp_abs$Month_of_absence) <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
"Aug", "Sep", "Oct", "Nov", "Dec")
levels(df_emp_abs$Disciplinary_failure) <- c("No", "Yes")
levels(df_emp_abs$Education) <- c("High Scool", "Graduate", "PG", "Masters and Doctors")
levels(df_emp_abs$Social_drinker) <- c("No", "Yes")
levels(df_emp_abs$Social_smoker) <- c("No", "Yes")
levels(df_emp_abs$Day_of_the_week) <- c("Mon", "Tues", "Wed", "Thurs", "Fri")
levels(df_emp_abs$Seasons) <- c("summer", "autumn", "winter", "spring")

#Creating graphs
bar1 <- ggplot(data = df_emp_abs, aes(x = ID)) + geom_bar() + ggtitle("Count of ID") +
theme_classic()
bar2 <- ggplot(data = df_emp_abs, aes(x = Reason_for_absence)) + geom_bar() +
  ggtitle("Count of Reason for absence") + theme_classic()
bar3 <- ggplot(data = df_emp_abs, aes(x = Month_of_absence)) + geom_bar() + ggtitle("Count of
Month") + theme_classic()
bar4 <- ggplot(data = df_emp_abs, aes(x = Disciplinary_failure)) + geom_bar() +
  ggtitle("Count of Disciplinary failure") + theme_classic()
bar5 <- ggplot(data = df_emp_abs, aes(x = Education)) + geom_bar() + ggtitle("Education
Qualification") + theme_classic()
bar6 <- ggplot(data = df_emp_abs, aes(x = Son)) + geom_bar() + ggtitle("Count of Son") +
theme_classic()
bar7 <- ggplot(data = df_emp_abs, aes(x = Social_smoker)) + geom_bar() +
  ggtitle("Count of Social smoker") + theme_classic()
```

```
bar8 <- ggplot(data = df_emp_abs, aes(x = Pet)) + geom_bar() + ggtitle("Count of Pets") +
theme_classic()
bar9 <- ggplot(data = df_emp_abs, aes(x = Seasons)) + geom_bar() + ggtitle("Seasons") +
theme_classic()
bar10 <- ggplot(data = df_emp_abs, aes(x = Day_of_the_week)) + geom_bar() + ggtitle("Day of the
Week") +theme_classic()

#Arranging and plotting graphs
gridExtra::grid.arrange(bar1,bar2,bar3,bar4,ncol=2)
gridExtra::grid.arrange(bar5,bar6,bar7,bar8,ncol=2)
gridExtra::grid.arrange(bar9,bar10,ncol=2)

#Check the distribution of numerical data using histogram
hist1 <- ggplot(data = numeric_data, aes(x = Transportation_expense)) +
  ggtitle("Transportation expense") + geom_histogram(bins = 25)
hist2 <- ggplot(data = numeric_data, aes(x =Height)) +
  ggtitle("Distribution of Height") + geom_histogram(bins = 25)
hist3 <- ggplot(data = numeric_data, aes(x = Body_mass_index)) +
  ggtitle("Body mass index") + geom_histogram(bins = 25)
hist4 <- ggplot(data = numeric_data, aes(x =Absenteeism_time_in_hours)) +
  ggtitle("Absenteeism time in hours") + geom_histogram(bins = 25)
hist5 <- ggplot(data = numeric_data, aes(x =Distance_from_Residence_to_Work)) +
  ggtitle("Distance from Residence to Work") + geom_histogram(bins = 25)
hist6 <- ggplot(data = numeric_data, aes(x = Service_time)) +
  ggtitle("Service time") + geom_histogram(bins = 25)
hist7 <- ggplot(data = numeric_data, aes(x = Age)) +
  ggtitle("Age Distribution") + geom_histogram(bins = 25)
hist8 <- ggplot(data = numeric_data, aes(x = Work_load_Average_perday)) +
  ggtitle("WorkLoad (Avg/day)") + geom_histogram(bins = 25)
hist9 <- ggplot(data = numeric_data, aes(x = Hit_target)) +
  ggtitle("Hit Target Distribution") + geom_histogram(bins = 25)
hist10 <- ggplot(data = numeric_data, aes(x = Weight)) +
  ggtitle("Weight Distribution") + geom_histogram(bins = 25)


gridExtra::grid.arrange(hist1,hist2,hist3,hist4,ncol=2)
gridExtra::grid.arrange(hist5,hist6,hist7,hist9,ncol=2)
gridExtra::grid.arrange(hist8,hist10,ncol=2)

######################################### Feature Selection
############################################
#Check for multicollinearity using VIF
vifcor(numeric_data)

#Plotting Correlation Plot
corrgram(numeric_data, order = T, upper.panel=panel.pie,
        text.panel=panel.txt, main = "Correlation Plot")

#Removing 'Body_mass_index' variable
df_emp_abs <- subset.data.frame(df_emp_abs, select = -c(Body_mass_index))

#copy of pre-processed data
processed_data <- df_emp_abs
write_csv(processed_data, "processed_data.csv")

######################################### Feature Scaling
############################################
hist(processed_data$Absenteeism_time_in_hours)

#Removing dependent variable
```

```
numeric_var <- sapply(processed_data, is.numeric)
numeric_data <- processed_data[,numeric_var]
numeric_cols <- names(numeric_data)
numeric_cols <- numeric_cols[-9] #i.e. 'Absenteeism time in hours' variable

#Normalizing continuous variables
for(i in numeric_cols){
  print(i)
  processed_data[,i] = (processed_data[,i] - min(processed_data[,i]))/
    (max(processed_data[,i]) - min(processed_data[,i]))
}

#Getting names of caegorical variables
category_cols <- names(category_data)


##################################### PCA for Dimensionality reduction
########################################
#splitting the data into Training and Test datasets
set.seed(123)
train_index = sample(1:nrow(processed_data), 0.8*nrow(processed_data))
train = processed_data[train_index,]
test = processed_data[-train_index,]

#Principal component analysis
prin_comp = prcomp(numeric_data)
pr_stdev = prin_comp$sdev #std deviation calculation
pr_var = pr_stdev^2 #variance calculation

#Variance Proportion
prop_var = pr_var/sum(pr_var)

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

#Adding training set with principal components
train.data = data.frame(Absenteeism_time_in_hours = train$Absenteeism_time_in_hours,
prin_comp$x)

#selecting 45 components
train.data =train.data[,1:45]

#Transforming test into PCA
test.data = predict(prin_comp, newdata = test)
test.data = as.data.frame(test.data)
test.data=test.data[,1:45] #for 45 components only


##################################### DECISION TREE
########################################

#RMSE: 0.442
#MAE: 0.301
#R squared: 0.978

#Implementing Decision Tree Model
dt_model = rpart(Absenteeism_time_in_hours ~., data = train.data, method = "anova")
```

```
#Predicting on test cases
dt_predictions = predict(dt_model,test.data)

#DF for actual and predicted values
df_pred = data.frame("actual"=test[,115], "dt_pred"=dt_predictions)
head(df_pred)

#Calcuating Error
print(postResample(pred = dt_predictions, obs = test$Absenteeism_time_in_hours))

#Plot a graph for actual vs predicted values
plot(test$Absenteeism_time_in_hours,type="l",lty=2,col="green")
lines(dt_predictions,col="blue")


##################################### RANDOM FOREST
########################################

#RMSE: 0.480
#MAE: 0.264
#R squared: 0.978

#Implementing Random Forest Model
rf_model = randomForest(Absenteeism_time_in_hours~., data = train.data, ntrees = 500)

#Predicting on test cases
rf_predict = predict(rf_model,test.data)

#DF for actual and predicted values
df_pred = cbind(df_pred,rf_predict)
head(df_pred)

#Calcuating Error
print(postResample(pred = rf_predict, obs = test$Absenteeism_time_in_hours))

#Plot a graph for actual vs predicted values
plot(test$Absenteeism_time_in_hours,type="l",lty=2,col="green")
lines(rf_predict,col="blue")


##################################### LINEAR REGRESSION
########################################

#RMSE: 0.003
#R squared: 0.999
#MAE: 0.002


#Implementing Linear Regression Model
lr_model = lm(Absenteeism_time_in_hours ~ ., data = train.data)
summary(lr_model)

#Predicting on test cases
lr_predictions = predict(lr_model,test.data)

#DF for actual and predicted values
df_pred = cbind(df_pred,lr_predictions)
head(df_pred)

#Calcuating Error
```

```
print(postResample(pred = lr_predictions, obs =test$Absenteeism_time_in_hours))

#Plot a graph for actual vs predicted values
plot(test$Absenteeism_time_in_hours,type="l",lty=2,col="green")
lines(lr_predictions,col="blue")
```