Employee Absenteeism

RITU KUMARI
12/28/2018

Contents

Introduction

1.1 Problem Description

Employee Absenteeism is the absence of an employee from work. It's a major problem faced by almost 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.

Absenteeism 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 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?

1.3 Data

The data is a Time-Series data but instead we will approach it as Regression Problem. Our task is to build a regression model which will predict the absenteeism in hours per employee based on the employee attributes and information in their work place and general information available to the company about them.

	ID	Reason for absence	Month of absence	Day of the week	Seasons	T
0	11	26.0	7.0	3	1	
1	36	0.0	7.0	3	1	
2	3	23.0	7.0	4	1	
3	7	7.0	7.0	5	1	
4	11	23.0	7.0	5	1	

Transportation expense	Distance from Residence to Work	Service time	Age	Work load Average/day
289.0	36.0	13.0	33.0	239,554
118.0	13.0	18.0	50.0	239,554
179.0	51.0	18.0	38.0	239,554
279.0	5.0	14.0	39.0	239,554
289.0	36.0	13.0	33.0	239,554

Disciplinary failure	Education	Son	Social drinker	Social smoker	Pet	Weight	Height
0.0	1.0	2.0	1.0	0.0	1.0	90.0	172.0
1.0	1.0	1.0	1.0	0.0	0.0	98.0	178.0
0.0	1.0	0.0	1.0	0.0	0.0	89.0	170.0
0.0	1.0	2.0	1.0	1.0	0.0	68.0	168.0
0.0	1.0	2.0	1.0	0.0	1.0	90.0	172.0

Height	Body mass index	Absenteeism time in hours
172.0	30.0	4.0
178.0	31.0	0.0
170.0	31.0	2.0
168.0	24.0	4.0
172.0	30.0	2.0

There are 21 variables in our data in which 20 are independent variables and 1 (Absenteeism time in hours) is dependent variable. As we can see in the table above, using those 20 variables we have to correctly predict the 'Absenteeism in hours' for the employees.

Predictor Variables

S.No	Predictor	S.No	Predictor
1	ID	11	Social drinker
2	Reason for absence	12	Social smoker
3	Distance from residence to work	13	Pet
4	Service time	14	Day of the week
5	Age	15	Weight
6	Work load average/day	16	Height
7	Hit target	17	Body mass index
8	Disciplinary failure	18	Transportation expense
9	Education	19	Month of absence
10	Son	20	Seasons

1.4 Performance Metric

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

Also, since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. So, RMSE becomes more useful when large errors are particularly undesirable. So, Roost Mean Square value seems like a perfect choice for our problem at hand.

Methodology

2.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the first step in our data analysis process. We do this by taking a broad look at patterns, trends, outliers, unexpected results and so on in our existing data, using visual and quantitative methods to get a sense of the story this tells. To start with this process, we will first have a look at univariate analysis like plotting Box plot and whiskers for individual features, Histogram plots, Bar plots and Kernel Density Estimation for the same for the same. Then we will proceed to Multivariate analysis like Bar and Histogram and Bar plot using group-by function and Pivot table for the features with respect to the target variable.

2.1.1 Data Visualisation

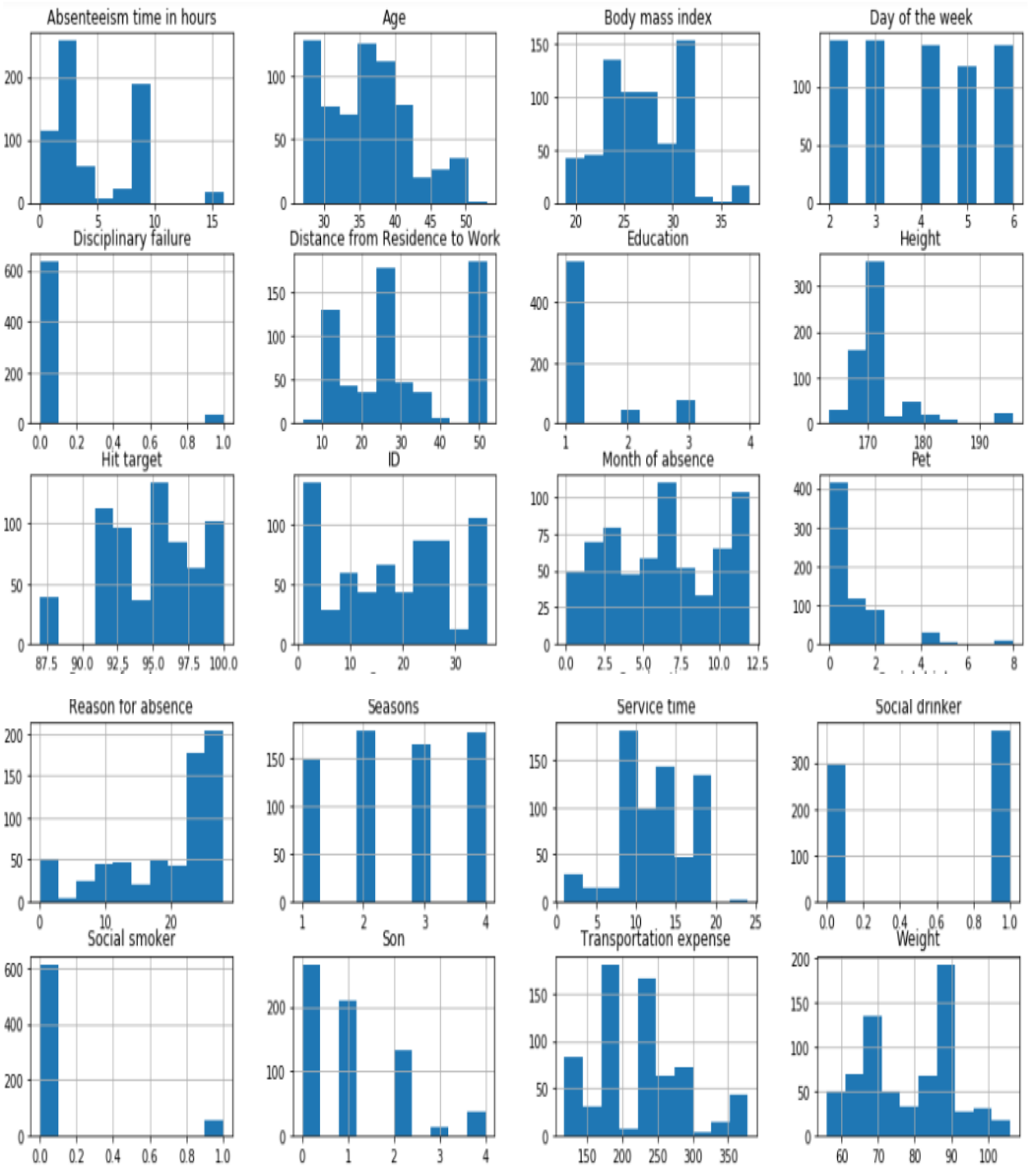
Data visualisation helps us to get better insights of the data. By visualising data, we can identify areas that need attention or improvement and also clarify which factors influence customer behaviour and how the resources are used by the customers.

2.1.1.1 Univariate Analysis

Univariate analysis is the simplest form of data analysis where the data being analyzed contains only one variable. Since it's a single variable it doesn't deal with causes or relationships. The main purpose of univariate analysis is to describe the data and find patterns that exist within it.

So, let's have a look at histogram plot, to identify the characteristic of the features and the data.

Histogram plot for distribution of features in the data



Histograms are constructed by binning the data and counting the number of observations in each bin. The objective of plotting Histogram plot is usually to visualize the shape of the distribution. The

number of bins needs to be large enough to reveal interesting features and small enough not to be too noisy.

From the above histogram plot, we can clearly observe that none of the features in our data are actually skewed. Although feature like ‘work load average/day’ seems like it is right skewed a little. Also if observed properly, It is worth noting the following points:

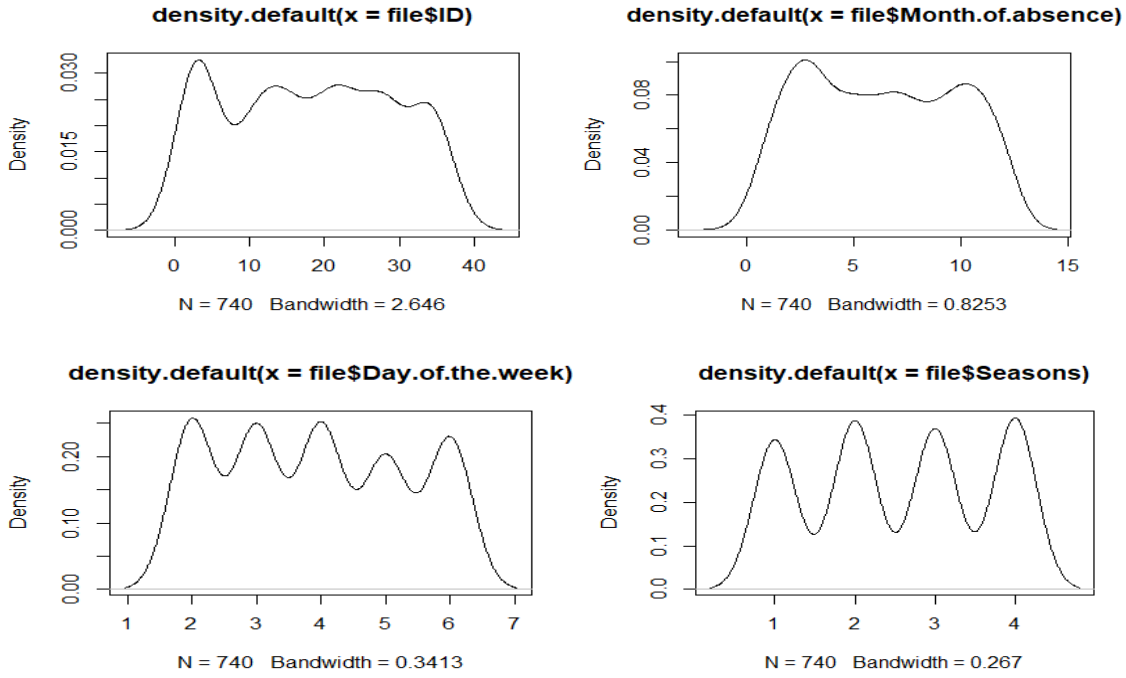
1. Majority of the employees working in the company have age below 40 years.

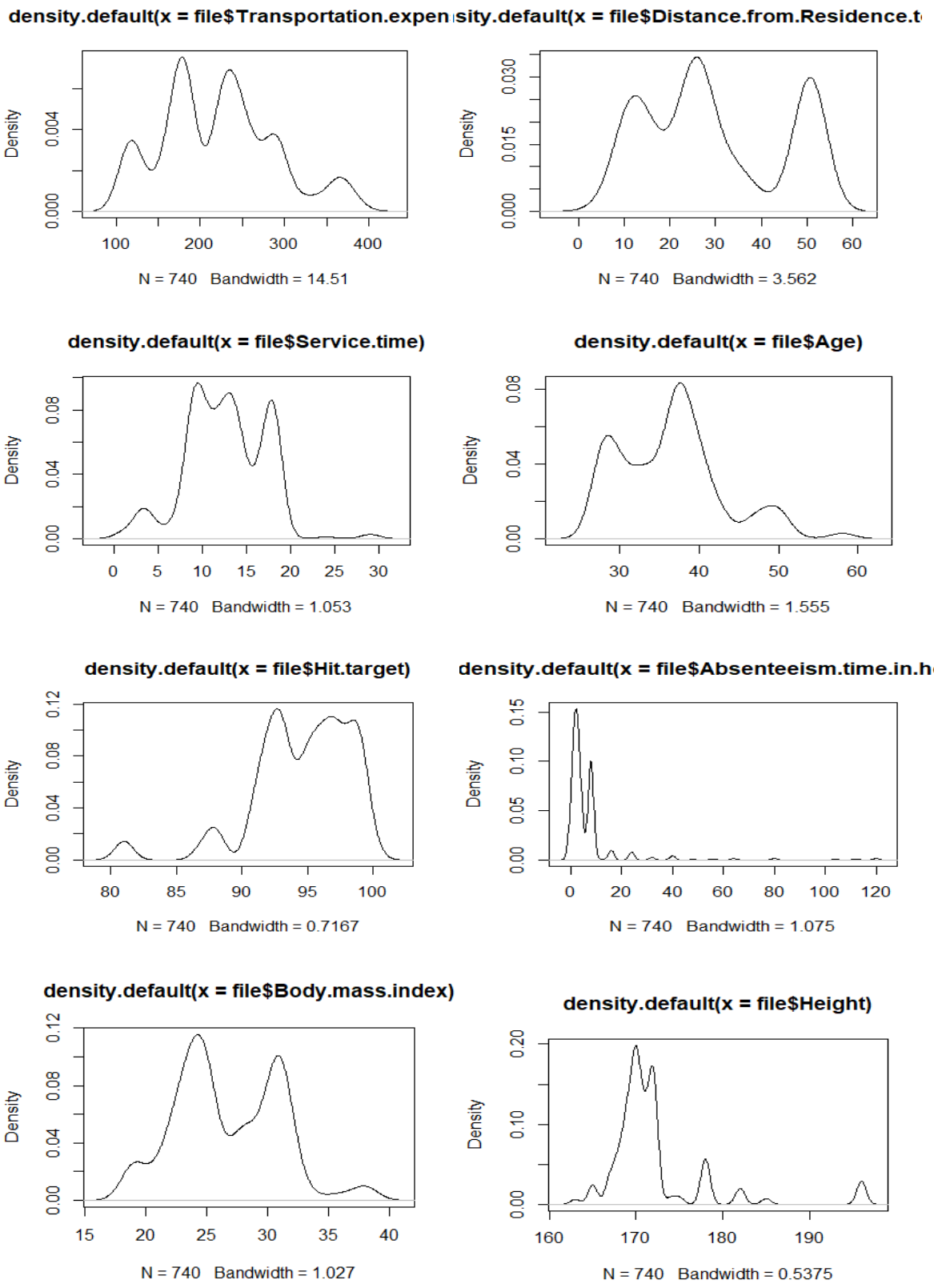
2. A very large portion of the population has only passed ‘High School’.

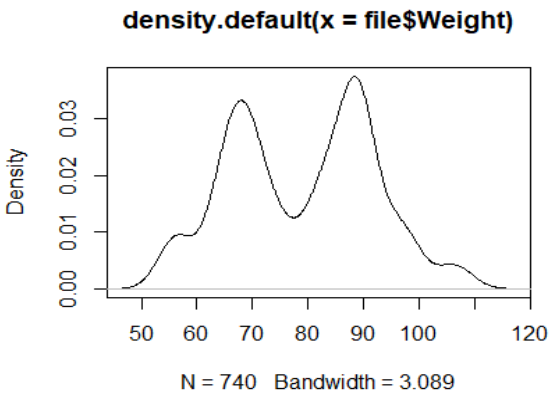
3. More than half of the employees in the company are ‘social drinker’.

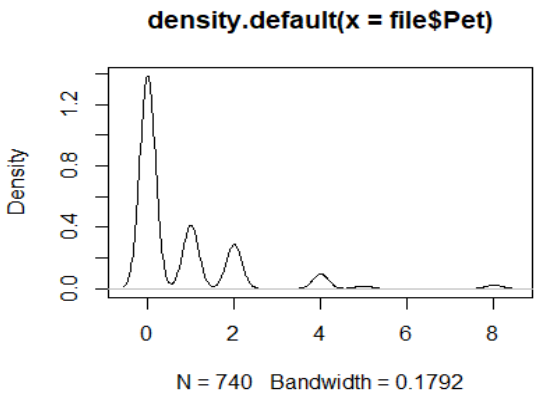
4. Only a very few portion of the employees in the company are ‘social smoker’.

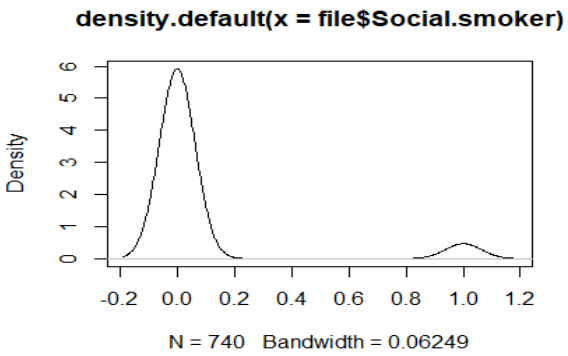
KDE plot for distribution of features in the data

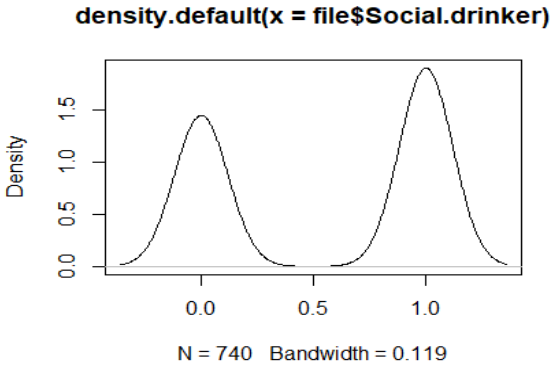


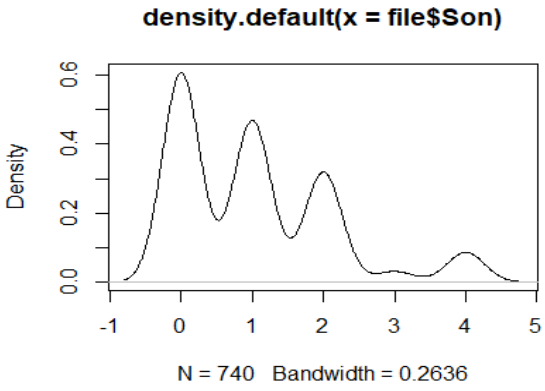

density.default(x = file$Transportation.expen
Density
0.000
0.004
100
200
300
400
N = 740 Bandwidth = 14.51
sity.default(x = file$Distance.from.Residence.t
Density
0.000
0.015
0.030
0
10
20
30
40
50
60
N = 740 Bandwidth = 3.562
density.default(x = file$Service.time)
Density
0.00
0.04
0.08
0
5
10
15
20
25
30
N = 740 Bandwidth = 1.053
density.default(x = file$Age)
Density
0.00
0.04
0.08
30
40
50
60
N = 740 Bandwidth = 1.555
density.default(x = file$Hit.target)
Density
0.00
0.04
0.08
0.12
80
85
90
95
100
N = 740 Bandwidth = 0.7167
density.default(x = file$Absenteeism.time.in.ho
Density
0.00
0.05
0.10
0.15
0
20
40
60
80
100
120
N = 740 Bandwidth = 1.075
density.default(x = file$Body.mass.index)
Density
0.00
0.04
0.08
0.12
15
20
25
30
35
40
N = 740 Bandwidth = 1.027
density.default(x = file$Height)
Density
0.00
0.10
0.20
160
170
180
190
N = 740 Bandwidth = 0.5375

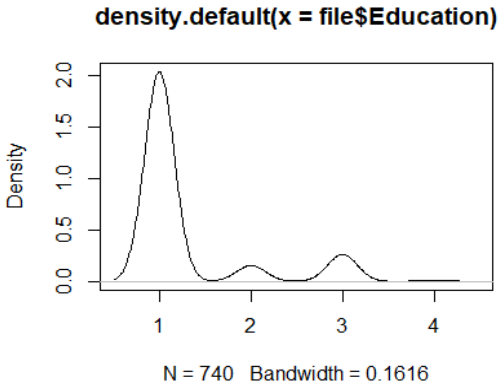

density.default(x = file$Weight)
Density
0.00 0.01 0.02 0.03
50 60 70 80 90 100 120
N = 740 Bandwidth = 3.089

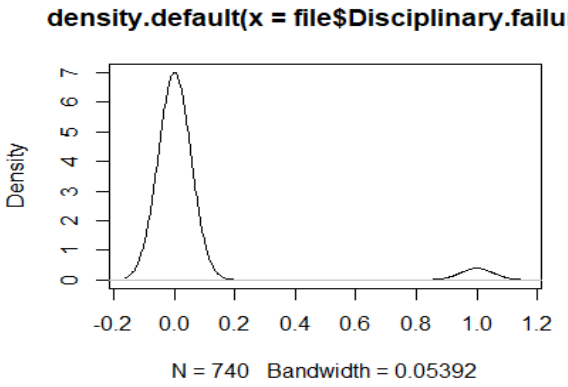

density.default(x = file$Pet)
Density
0.0 0.4 0.8 1.2
0 2 4 6 8
N = 740 Bandwidth = 0.1792


density.default(x = file$Social.smoker)
Density
0 1 2 3 4 5 6
-0.2 0.0 0.2 0.4 0.6 0.8 1.0 1.2
N = 740 Bandwidth = 0.06249


density.default(x = file$Social.drinker)
Density
0.0 0.5 1.0 1.5
0.0 0.5 1.0
N = 740 Bandwidth = 0.119


density.default(x = file$Son)
Density
0.0 0.2 0.4 0.6
-1 0 1 2 3 4 5
N = 740 Bandwidth = 0.2636

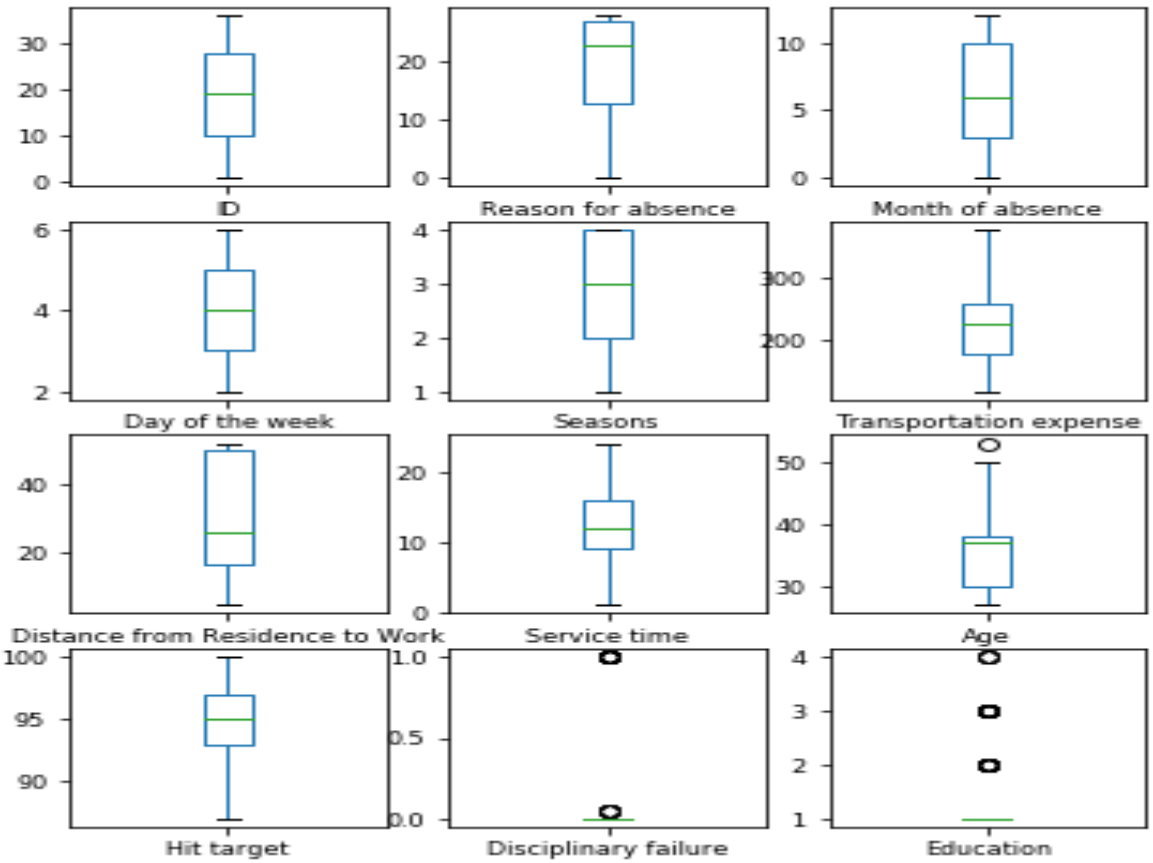

density.default(x = file$Education)
Density
0.0 0.5 1.0 1.5 2.0
1 2 3 4
N = 740 Bandwidth = 0.1616

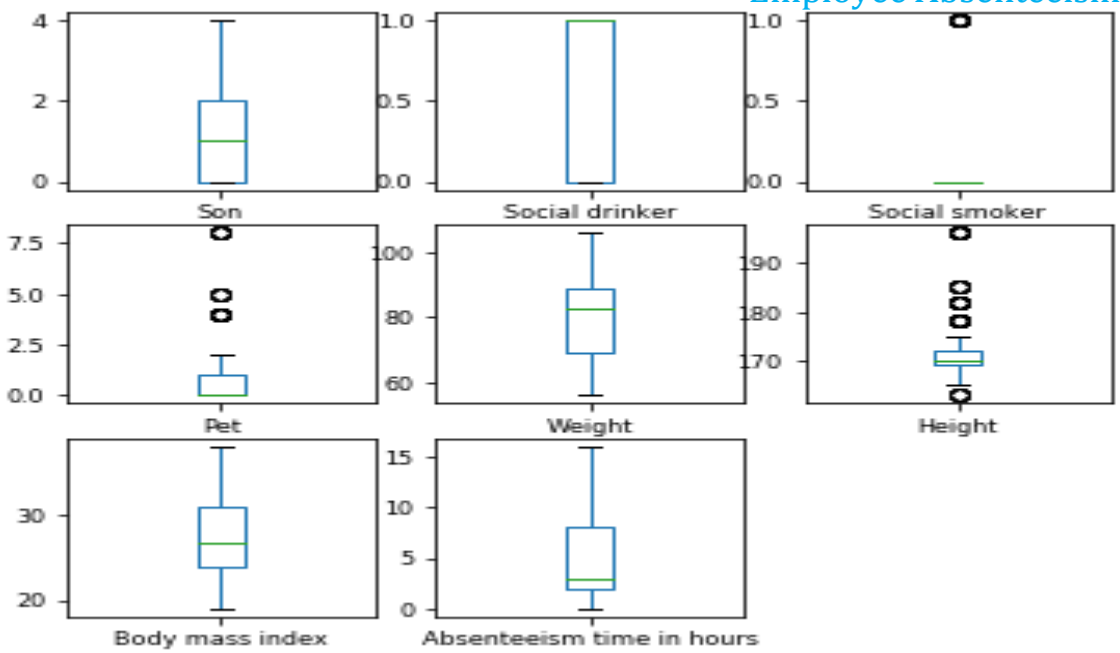

density.default(x = file$Disciplinary.failure)
Density
0 1 2 3 4 5 6 7
-0.2 0.0 0.2 0.4 0.6 0.8 1.0 1.2
N = 740 Bandwidth = 0.05392

A Density Plot visualizes the distribution of data over a continuous interval or time period. Density plots can be thought of as plots of smoothed histograms. An advantage Density Plots have over Histograms is that they're better at determining the distribution shape because they're not affected by the number of bins used.

So, looking at the above density plot, we can observe that none of the features follow Gaussian distribution. Few of the features like ‘Disciplinary failure’, ‘Social smoker’, ‘Work load average/day’ seems to follow Gaussian distribution at first sight but they either have long tail at the left or right or they are either jiggered at the end.

Box and Whiskers plot of features in the data





From the above Box and whisker plots, we can observe that not all the features contain outliers. Continuous features like ‘Weight’, ‘Distance from residence to work’ does not contain any outliers at all. Few features like ‘work load average/day’, ‘Hit target’ and ‘Height’ have a very few outliers.

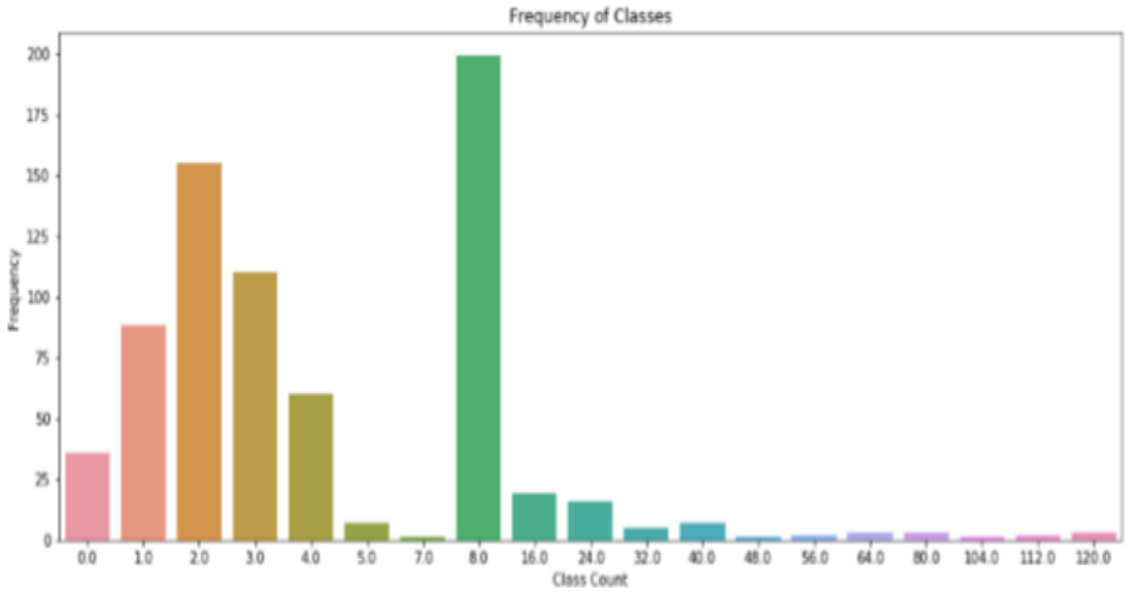
It is also evident from the above plot that none of the features are symmetric to the median and it can easily be interpreted that none of the features follow symmetric distribution. Also, it can also be observed that Median of the feature ‘Body mass index’ is very close to 25th percentile value which means median of this feature is almost equal to 25th percentile.

2.1.1.2 Bivariate Analysis

Bivariate analysis refers to the analysis of bivariate data. It is one of the simplest forms of statistical analysis, used to find out if there is a relationship between two sets of values. It usually involves one predictor variable and one target variable.

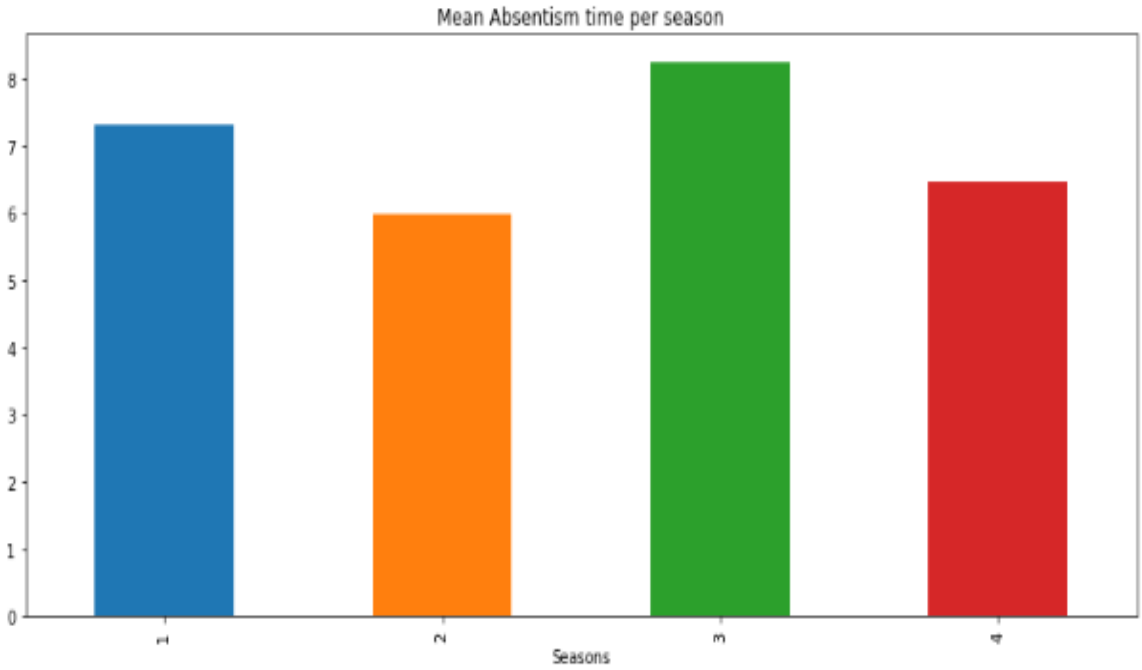
So, let's have a look at the Histogram and Bar Plots to understand the Employee behavior better.

Bar plot for Distribution of Target class



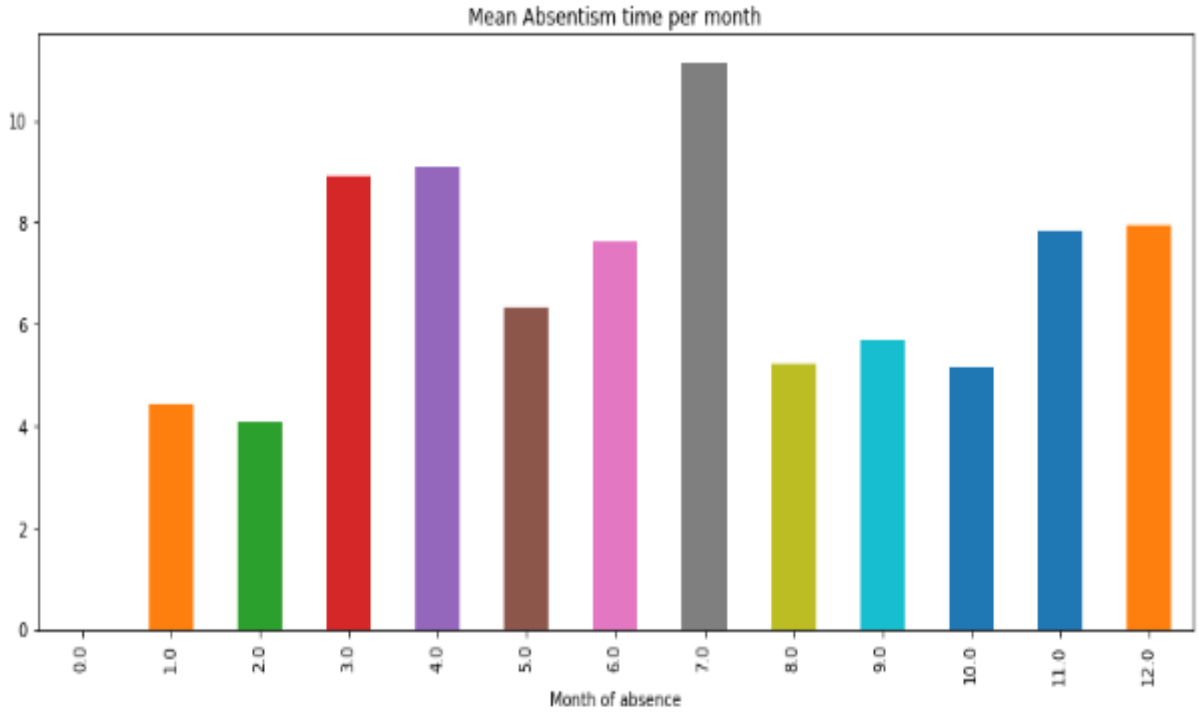
So, we can clearly observe from the above histogram plot that, most of the data points in our data are labeled as '8', followed by '2', '3', '1', '4', '0' etc. Also, it is worth noticing that any label after '8' is a multiple of 8 and have the least occurrence within the data.

Bar Plot for Mean absenteeism time per season



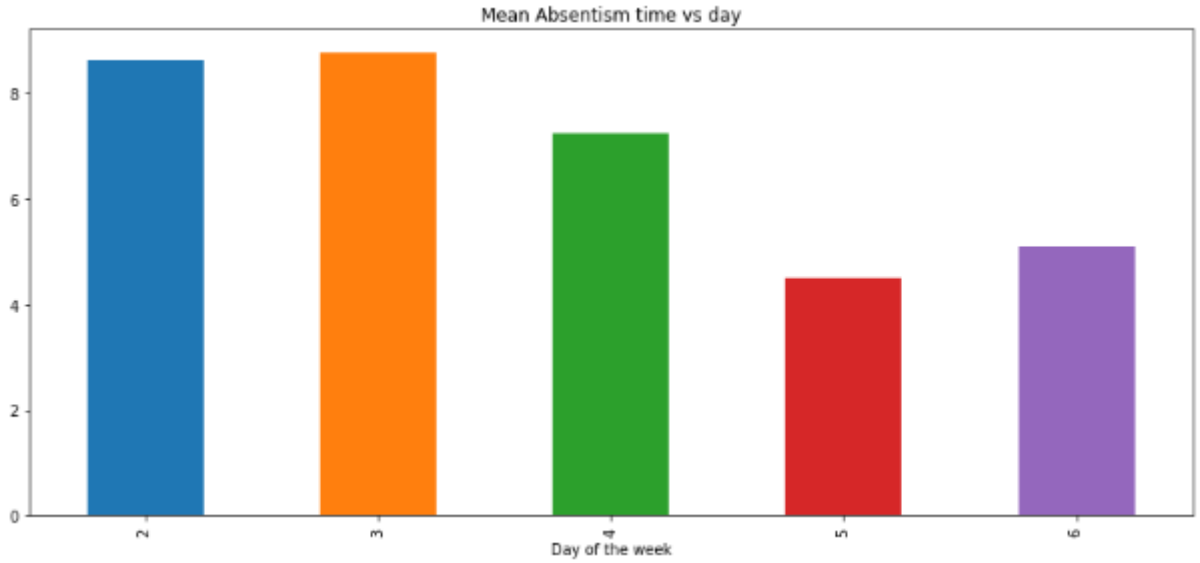
From the above Bar plot, it can be observed that the 'Absenteeism rate' is maximum in Season 3 : Winter followed by Season 1 : Summer , Season 4 : Spring, Season 2 : Autumn.

Bar Plot for Mean absenteeism time per month



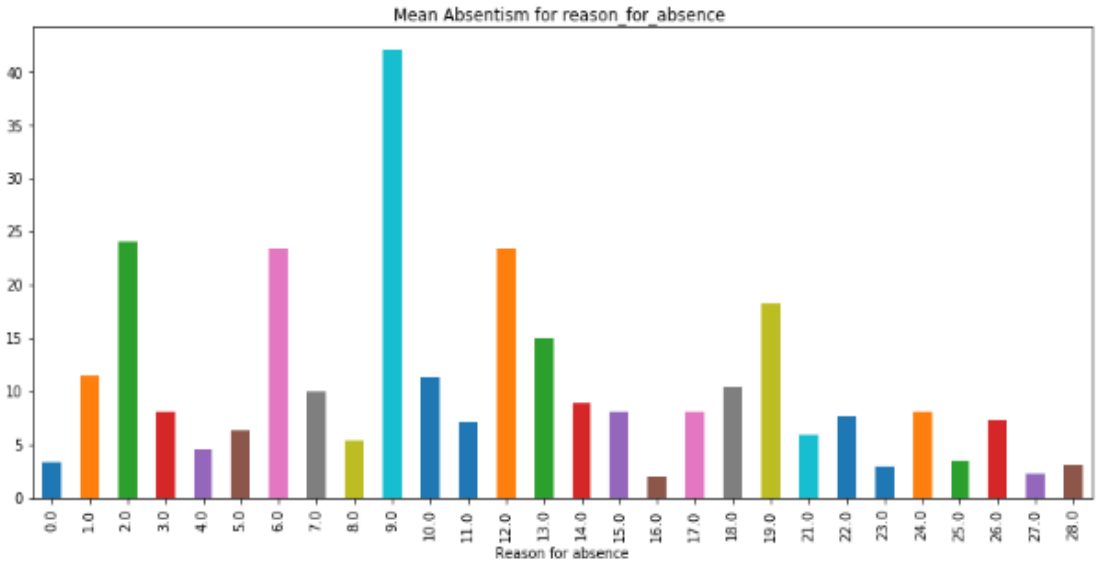
From the above Bar plot , we can clearly observe that the 'Absenteeism rate' is maximum in Month 7 : July followed by Month 4 : April , Month 3 : March, Month 12 : December, Month 11 : November , Month 6 : June , Month 5 : May etc.

Bar Plot for Mean absenteeism time per day



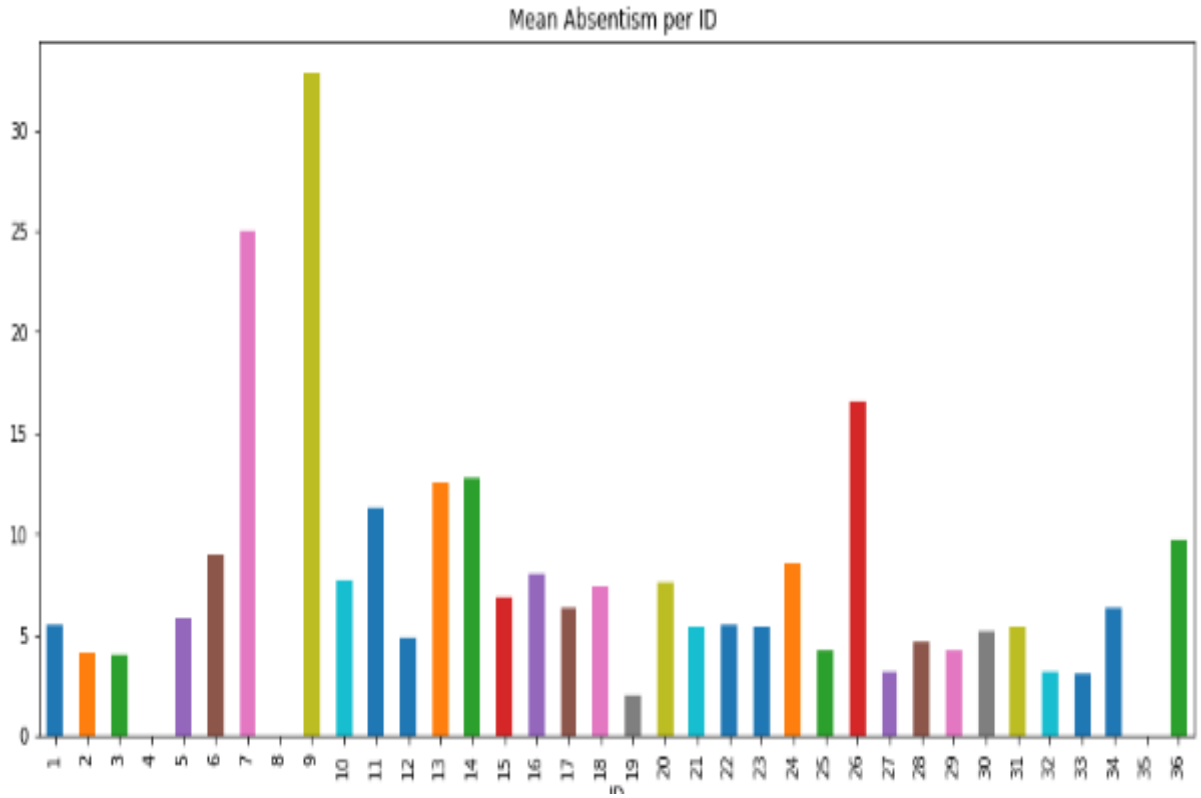
From the above Bar plot, it can clearly be observed that the 'Absenteeism rate' is maximum on the third day of the week i.e Day 3 : Tuesday followed by Day 2 : Monday, Day 4 : Wednesday. Also, the 'absenteeism rate' is lowest on Day 6 : Saturday followed by Day 5 : Friday.

Bar Plot for Mean absenteeism for reason_of_absence



From the above plot, we can observe that ‘9 : Diseases of the circulatory system’ is the most frequent reason for the absence of the employees. The second most frequent reason given by the employees for their absence is ‘2 : Neoplasms’ followed by ‘6 : Diseases of the nervous system’, ‘12: Diseases of the skin and subcutaneous tissue’, ’19 : Injury, poisoning and certain other consequences of external causes’ etc.

Bar Plot for Mean absenteeism time per ID

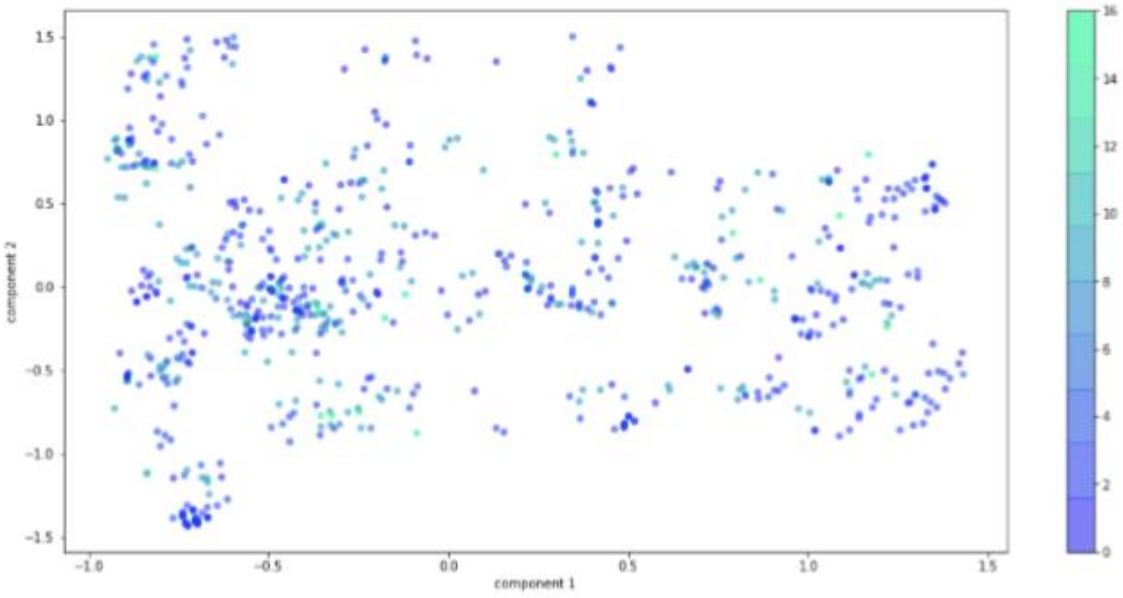


From the above plot, It can be observed that the absence rate is maximum for employee with ID : 9, followed by employees with ID : 7,26,14,13, 36, 11 and 6. Also, it can be observed that Employee with employee ID : 4,8 and 35 never absents and are very much regular to work.

2.1.1.3 Multivariate Analysis

Multivariate analysis is the analysis of more than one variable in a dataset. Multivariate analysis becomes important when we have large dimensional data to visualize and it becomes very difficult to visualize every predictor variable individually. It also helps us to identify the dominant patterns and clusters in the data.

So, we will first look at one of the most widely used algorithm to visualize high dimensional data based on Eigen Value and Eigen vectors which is called as Principal Component Analysis. In PCA visualization, we have plotted the scatter plot along two dimensions.



In the above plot, 100% of the variance in the data is explained by the principal component 1 and 0% of the variance in the data is explained by the principal component 2. Also, it is hard to interpret anything from the above plot except the fact that the data is spread unevenly across the space. The data does not seem to follow any patter or any kind of linear or polynomial relationship.

2.1.2 Data Preparation and Cleaning

2.1.2.1 Missing Value Analysis

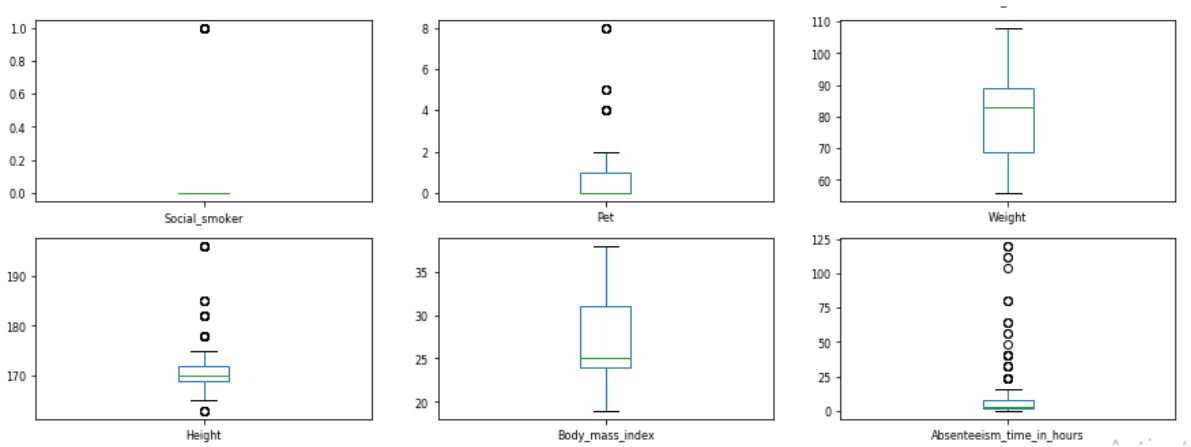
One of the most common problems I have faced in Data Cleaning/Exploratory Data Analysis is handling the missing values. Firstly, there is no good way to deal with missing data. But still missing value analysis helps address several concerns caused by incomplete data. If cases with missing values are systematically different from cases without missing values, the results can be misleading. Also, missing data may reduce the precision of calculated statistics because there is less information than originally planned. Another concern is that the assumptions behind many statistical procedures are based on complete cases, and missing values can complicate the theory required.

So, in our data, there are plenty of missing values available in different variables. So, after computing the percentage of missing data that is available to us in the dataset, it accounts to around 12% of the data. It is also important to note that, the missing value has been calculated after removing the missing values within the target variable. Also, as we have very less data available to us, we impute the missing values in other columns using KNN imputation, because that fits the best after trying various other imputation techniques like Mean, Median and Random value imputation.

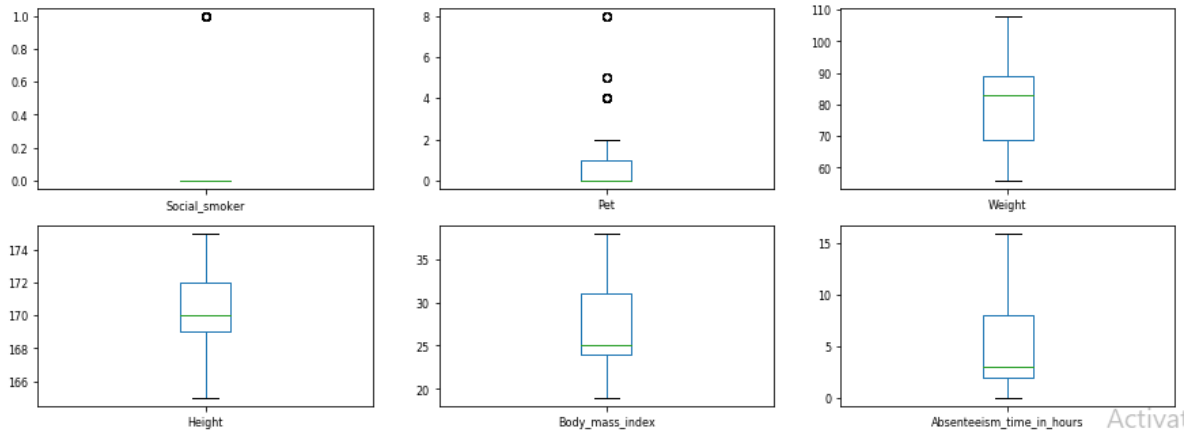
2.1.2.2 Outliers Analysis

In statistics, an outlier is an observation point that is distant from other observations. In layman terms; we can say that an outlier is something which is separated/different from the crowd. Also, Outlier analysis is very important because they affect the mean and median which in turn affects the error (absolute and mean) in any data set. When we plot the error we might get big deviations if outliers are in the data set.

Visualization With outliers



Visualization Without outliers



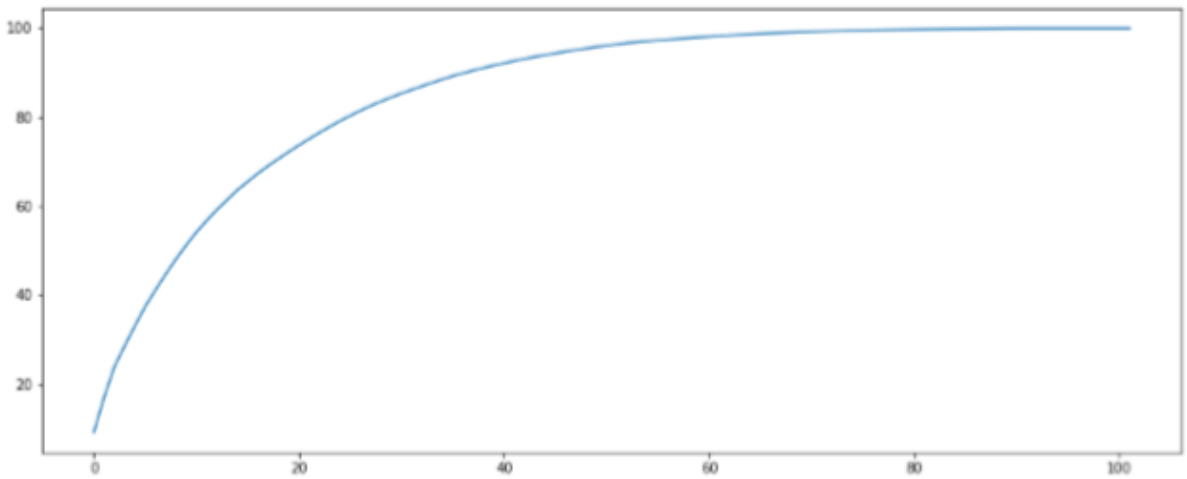
In Box plots analysis of individual features, we can clearly observe from these boxplots that, not every feature contains outliers and many of them even have very few outliers. Also, given the constraint that, we have only 640 data-points and after removing the outliers, the data gets decreased by almost 25%. So, dropping the outliers is probably not the best idea.

Instead we will try to visualize and find out the outliers using box plots and will fill them with NA that means we have created 'missing values' in place of outliers within the data. Now, we can treat these outliers like missing values and impute them using standard imputation techniques. In our case, we use KNN imputation to impute these missing values.

2.1.2.3 Feature Selection

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. There are several methods of doing that. Here, we do this in two steps : Firstly, we find and remove the correlated features and then we use a more advanced technique for dimensionality reduction called PCA .While doing this, we first plot a cumulative distribution function plot to observe how much percentage of variance is explained by how many variables (Principle Components). The CDF plot for the same is plotted below:

CDF Plot for 'Variance Explained' Vs 'Principle components'



It is very clear from the above CDF plot for 'Variance Explained' Vs 'Principle components', that almost 95%+ variance is explained by just 45 variables (Principle components). We, can imagine how powerful PCA is, It just shrank down our feature space to just 45 from a total of 107 features. So, we will keep only 45 principle components in the data and will perform modeling on it.

Modeling

We always start our model building from the simplest to more complex. Therefore we start with KNN Regression.

2.2.1 KNN Regression

KNN regression is one of the simplest algorithms in the whole of Machine learning. It gives a weighted average of the regression function in a local space (k nearest points to a given point). So, we first try to implement and fit KNN regression to our Data and got following results after tuning the hyper-parameter k:

```
Train Data
n_neighbours : 30  ----KNN rmse: 2.2706908685134333
Test Data
n_neighbours : 30  ----KNN rmse: 2.756796390255527
```

So, as we can see from the above results that, our model gives a RMSE (Root Mean Square Value) of 2.27 for the train data and a RMSE value of 2.75 for the test data. Looking at the train and test error, we can say that our model seems like overfitting a little. But still we can consider it as a pretty good score given the shortage of data to us.

2.2.2 Ordinary Least Squares

Now we will try to implement Multiple Linear Regression algorithm using Ordinary Least Squares, the simplest of all. Ordinary least squares (OLS) minimize the squared distances between the observed and the predicted dependent variable. So, we get the following results after implementing the model:

```
Train Data
Ordinary Least Squares rmse: 2.551463189617839
Test Data
Ordinary Least Squares rmse: 2.7433491590595773
```

So, as we can see from the above results that, our model gives a RMSE (Root Mean Square Value) of 2.55 for the train data and a RMSE value of 2.74 for the test data. Looking at the train and test

error, We can say that our model seems like overfitting a little but it still it overfits less than KNN. Also, we can observe that Ordinary Least's Squares perform a little but better then the KNN Regression model

2.2.3 Decision Tree Regression

Decision tree builds regression, 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. So, after we implement Decision Tree Regression on our data, we get the following results:

```
Train Data
Decision Tree rmse: 3.31035213395237
Test Data
Decision Tree rmse: 3.272149876154806
```

So, we can see from the above results that, our model gives a RMSE (Root Mean Square Value) of 3.31 for the train data and a RMSE value of 3.27 for the test data. Although, it seems like the model doesn't overfit on the train data, Decision Tree Regression does not give impressive results and our other linear algorithms gives much lower error.

2.2.7 Gradient Boosting Decision Tree Regression

GBDT for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalises them by allowing optimisation of an arbitrary differentiable loss function. So, after we implement Gradient Boosting Decision Trees on our data, we get the following results:

```
Train Data
GBDT rmse: 3.31035213395237
Test Data
GBDT rmse: 3.272149876154806
```

So, we can see from the above results that, our model gives a RMSE (Root Mean Square Value) of 3.31 for the train data and a RMSE value of 3.27 for the test data. Although, it seems like the model doesn't overfit on the train data, Gradient Boosting Decision Tree Regression does not give impressive results and our other linear algorithms gives much lower error. Also, It can be noticed that, the result of Decision Trees and GBDT are exactly same.

2.2.8 Random Forest Regression

Random Forest Regression or Regression Trees are known to be very unstable, in other words, a small change in our data may drastically change your model. The Random Forest uses this instability as an advantage through bagging resulting on a very stable model. So, after we implement Random Forest Regression or Regression Trees on our data, we get the following results:

```
Train Data
Random Forest rmse: 1.1209862159378214
Test Data
Random Forest rmse: 2.7611642280442057
```

So, we can see from the above results that, our model gives a RMSE (Root Mean Square Value) of 1.12 for the train data and a RMSE value of 2.76 for the test data. Looking at the train and test error, We can say that our model seems like overfitting a lot. But still we can consider it gives a pretty good score on the test data. Also, it out performs many algorithms that we have seen earlier.

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 Value

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

Also, Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. So, RMSE becomes more useful when large errors are particularly undesirable. So, Roost Mean Square value seems like a perfect choice for our problem at hand.

3.2 Model Selection

We saw that both models Random Forest along with SVR and Ridge Regression perform comparatively on RMSE (Root Mean Square Error) , Although Random Forest gives the best results on the test data but it is unstable and it overfits on the train data. So, It will be a wise decision to use either SVR or Ridge Regression for deployment. Model comparison table is given below.

Model	rmse
KNN Regression	2.75
Ordinary Least Square	2.74
Decision Trees	3.19
GBDT	3.19
Random Forest	2.58

3.3 Answer to the asked Questions

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

Looking at the exploratory data analysis of the features, we observe and make following conclusion:

- The rate of Absenteeism is maximum in Season 3: Winter followed by Season 1: Summer , Season 4 : Spring, Season 2 : Autumn.
- Also, We can say that the 'Absenteeism rate' is maximum in Month 7 : July

followed by Month 4 : April , Month 3 : March, Month 12 : December, Month 11 : November , Month 6 : June , Month 5 : May etc.

- Looking at the Bar plot of 'Absenteeism rate' Vs 'Day of the week', it c
- an clearly be observed that the 'Absenteeism rate' is maximum on the third day of the week i.e Day 3 : Tuesday followed by Day 2 : Monday, Day 4 : Wednesday. Also, the 'absenteeism rate' is lowest on Day 6 : Saturday followed by Day 5 : Friday.
- From the Bar plot of 'Absenteeism rate' Vs 'Reason of absence' we can observe that '9 : Diseases of the circulatory system' is the most frequent reason for the absence of the employees. The second most frequent reason given by the employees for their absence is '2 : Neoplasm's' followed by '6 : Diseases of the nervous system', '12: Diseases of the skin and subcutaneous tissue', '19 : Injury, poisoning and certain other consequences of external causes' etc.
- Looking at the Bar plot of 'Absenteeism rate' Vs 'ID' It can be observed that the absence rate is maximum for employee with ID: 9, followed by employees with ID: 7,26,14,13, 36, 11 and 6. Also, it can be observed that Employee with employee ID: 4, 8 and 35 never absents and are very much regular to work.

So, Now that we have understood the behavior of Employee attributes against their Mean Absenteeism rate in the company, we can introduce following changes to reduce the number of Absenteeism:

- Firstly, we can start by Increasing the employee morale, engagement, and commitment to the organization.

- We can set a certain threshold for minimum number of absence for employees during the workdays, and employees not meeting the criteria can be questioned.

- As Absenteeism rate is maximum in Season of 'winter' and in month of July, April and March respectively, we can issue special notices regarding the Absenteeism scenario around the company.
- A Health care facility can be introduced in the company, so that the employees can have regular Medical check-ups to keep them fit and Working. Also, it would help with the company's reputation to have taken the responsibility of their Employees.
- Also, we can introduce person to person phone calls if off sick and return to work interviews. This way, Employee would feel responsible for their action towards the company goal and achievements.
- We can also we can come up with other ideas like: An incentive or conversion scheme for unused sick days.
- Also, strict action could be taken towards the employee with high absence rate in the workplace without any valid reason for the absence and Employees with no absence or a minimum absence can be rewarded with perks.
- Lastly, we can apply performance policies to act at the root of the problem. In some cases, absence rate might be reduced by clear specification of employees' responsibilities and targets.

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

Employee absence, whether caused by sickness or pressures outside of the workplace, can cost employers a large amount of money if not properly managed. Lowering absence levels across a business not only leads to a reduction in money being lost by the business, but also a happier, productive and content workforce.

To calculate Loss per month, We introduce the following formula:

Loss = (Work load average/day *Absenteeism time in hours)/ Service Time

So, below chart represents loss per month and more likely, the same trend could follow in 2011:

Month	Loss Per Month
No Absent	0
January	4856265
February	8003207
March	10174728
April	6350698
May	6242207
June	10254968
July	11650434
August	6400651
September	4327941
October	7227177
November	6337747
December	7692128

So, looking at the above table, we can say that the company incurred maximum loss in the month of July, followed by June and March etc. So, the same trend of loss can be expected in the year 2011 given the attributes of the employees.

References

Andrew Ng's Machine Learning course on Coursera

Ieee research paper of employee absentism

Github

Analytics vidhya Regression Problem

MIT 18.05, Introduction to Probability and Statistics, taught by Jeremy Orloff and Jonathan Bloom. Provides intuition for probabilistic reasoning & statistical inference, which is invaluable for understanding how machines think, plan, and make decisions.

Appendix A – R code

```
rm(list = ls())


#Getting Current working directory.

getwd()


#Setting working directory

setwd("C:/Users/anupr/Desktop/employee_sentism")


getwd()


x = c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "C50", "dummies",
"e1071", "Information",

    "MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine', 'inTrees','readxl')


lapply(x, require, character.only = TRUE)


## Read the data

file= read.csv("data.csv", header = T, na.strings = c(" ", "", "NA"))




#Getting the dimensions of data

dim(file)



# Fetting Structure Of data
```

```
str(file)


#Retrieving Column names of train and test data.

colnames(file)



#Removing Spaces between the column names

names(file) =  gsub(" ", "_", names(file))


############ Distribution pf Target Variable ###############


# For train data

library(ggplot2)

pl = ggplot(file ,aes(x = "Absenteeism time in hours")) + ggtitle("Absenteeism time in hours")

print(pl + geom_bar(fill = 'blue'))



############################################## Missing Value Analysis
##############################################


missing_val = data.frame(apply(file,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(file)) * 100

missing_val = missing_val[order(-missing_val$Missing_percentage),]

row.names(missing_val) = NULL

missing_val = missing_val[,c(2,1)]
```

```
write.csv(missing_val, "Miising_perc.csv", row.names = F)


# kNN Imputation

file = knnImputation(file, k = 3)

sum(is.na(file))



### Imputing Missing Values

require(DMwR)

cnames = colnames(file)

newdata = data.frame()

newdata = data.frame(file)

file = knnImputation(newdata, k = 3)

sum(is.na(file))

anyNA(file)


############################################## Data Visusalisation
#######################################################


#### Box-Plot for data

boxplot(file[,c("ID",
"Reason.for.absence","Month.of.absence","Day.of.the.week","Seasons","Transportation.expense",
"Distance.from.Residence.to.Work")])

boxplot(file[,c("Service.time", "Age","Work.load.Average.day","Hit.target","Disciplinary.failure"
,"Education" ,"Son")])

boxplot(file[,c("Social.drinker","Social.smoker","Pet", "Weight", "Height",
"Body.mass.index","Absenteeism.time.in.hours")])


#### KDE plot
```

```
library("kdensity")

plot(density(file$'ID'))

plot(density(file$'Reason.for.absence'))

plot(density(file$'Month.of.absence'))

plot(density(file$'Day.of.the.week'))

plot(density(file$'Seasons'))

plot(density(file$'Transportation.expense'))

plot(density(file$'Distance.from.Residence.to.Work'))

plot(density(file$'Service.time'))

plot(density(file$'Age'))

plot(density(file$'Work.load.Average.day'))

plot(density(file$'Hit.target'))

plot(density(file$'Absenteeism.time.in.hours'))

plot(density(file$'Body.mass.index'))

plot(density(file$'Height'))

plot(density(file$'Weight'))

plot(density(file$'Pet'))

plot(density(file$'Social.smoker'))

plot(density(file$'Social.drinker'))

plot(density(file$'Son'))

plot(density(file$'Education'))

plot(density(file$'Disciplinary.failure'))


## Normality Check

qqnorm(file$ID)

qqnorm(file$Reason.for.absence)

qqnorm(file$Month.of.absence)
```

```
qqnorm(file$Day.of.the.week)

qqnorm(file$Seasons)

qqnorm(file$Transportation.expense)

qqnorm(file$Distance.from.Residence.to.Work)

qqnorm(file$Service.time)

qqnorm(file$Age)

qqnorm(file$Work.load.Average.day)

qqnorm(file$Hit.target)

qqnorm(file$Absenteeism.time.in.hours)

qqnorm(file$Body.mass.index)

qqnorm(file$Height)

qqnorm(file$Weight)

qqnorm(file$Pet)

qqnorm(file$Social.drinker)

qqnorm(file$Social.smoker)

qqnorm(file$Son)

qqnorm(file$Education)

qqnorm(file$Disciplinary.failure)



#### PCA Visualisation

library(ggfortify)

autoplot(prcomp(file), data = file, colour = 'Absenteeism.time.in.hours')



###################### Outlier Analysis ######################
```

```
cnames =c('Service.time', 'Age', 'Work.load.Average.day', 'Transportation.expense','Hit.target',
'Height', 'Absenteeism.time.in.hours', 'Weight')

df = file


#Replace all outliers with NA and impute

for(i in cnames)

{

 val = df[,i][df[,i] %in% boxplot.stats(df[,i])$out]

 #print(length(val))

 df[,i][df[,i] %in% val] = NA

}


# Imputing missing values

df = knnImputation(df,k=3)

file = df

anyNA(file)



################################### CREATING DUMMIES FOR CATEGORICAL VARIABLES
#####################################

file = fastDummies::dummy_cols(data , select_columns = "Seasons" , remove_first_dummy = TRUE)

file = fastDummies::dummy_cols(data , select_columns = "Month.of.absence" ,
remove_first_dummy = TRUE)

file = fastDummies::dummy_cols(data , select_columns = "Day.of.the.week" , remove_first_dummy
= TRUE)

file = fastDummies::dummy_cols(data , select_columns = "Reason.for.absence" ,
remove_first_dummy = TRUE)

file = fastDummies::dummy_cols(data , select_columns = "ID" , remove_first_dummy = TRUE)
```

```
file = fastDummies::dummy_cols(data , select_columns = "Education" , remove_first_dummy = TRUE)

file = fastDummies::dummy_cols(data , select_columns = "Pet" , remove_first_dummy = TRUE)

file = fastDummies::dummy_cols(data , select_columns = "Son" , remove_first_dummy = TRUE)


knitr::kable(file)


# Deleting the columns for which dummies are created

file = subset(file, select = -c(Seasons,Month.of.absence, Day.of.the.week,Reason.for.absence,ID, Education, Pet, Son ))


############################################## Scaling the data
########################################


cnames1 =c('Service.time', 'Age', 'Work.load.Average.day', 'Transportation.expense','Hit.target', 'Height', 'Weight')



for(i in cnames1){

  print(i)

  file[,i] = (file[,i] - min(file[,i]))/

    (max(file[,i] - min(file[,i])))

}


# Normalization

for(i in cnames1)

{

  print(i)
```

```
  file[,i] = (file[,i] - min(file[,i]))/(max(file[,i])-min(file[,i]))
}

############################################ Feature Selection
#######################################

##### Using Correlation plot

library(corrgram)

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

# Weight seems to be correlated, so Deleting weight from the data

file = subset(file , select = -c(Weight))

##### Splitting the data into train and test

n = nrow(file)

trainIndex = sample(1:n, size = round(0.8*n), replace=FALSE)

train = file[trainIndex ,]

test = file[-trainIndex ,]

X_train = subset(train,select = -c(Absenteeism.time.in.hours))

y_train = subset(train,select = c(Absenteeism.time.in.hours))
```

```
X_test = subset(test,select = -c(Absenteeism.time.in.hours))

y_test = subset(test,select = c(Absenteeism.time.in.hours))


##### Using PCA


#principal component analysis

prin_comp = prcomp(X_train)

prcomp(file[,-1])


#compute standard deviation of each principal component

std_dev = prin_comp$sdev


#compute variance

pr_var = std_dev^2


#proportion of variance explained

prop_varex = pr_var/sum(pr_var)


#cdf plot for principle components

plot(cumsum(prop_varex), xlab = "Principal Component",

    ylab = "Cumulative Proportion of Variance Explained",

    type = "b")


#add a training set with principal components

X_train.data = data.frame( prin_comp$x)


# From the above plot selecting 45 components since it explains almost 95+ % data variance
```

```
X_train.data =X_train.data[,1:45]


#transform test into PCA

X_test.data = predict(prin_comp, newdata = X_test)

X_test.data = as.data.frame(X_test.data)


#select the first 45 components

X_test.data=X_test.data[,1:45]


########################################## Machine learning
model##########################################


X_train.data$Absenteeism_time_in_hours = paste(y_train$Absenteeism_time_in_hours)

X_test.data$Absenteeism_time_in_hours = paste(y_test$Absenteeism_time_in_hours)



library(mlbench)

#### KNN

#Develop Model on training data

fit_LR = knnreg(Absenteeism_time_in_hours ~ ., data = X_train.data)

#Lets predict for testing data

pred_LR_test = predict(fit_LR,X_test.data)

# Results

print(postResample(pred = pred_LR_test, obs =y_test$Absenteeism_time_in_hours))
```

```
###### Multiple Linear Regression

#Develop Model on training data

set.seed(100)

#Develop Model on training data

fit_LR = lm(Absenteeism.time.in.hours ~ ., data = X_train.data)

#Lets predict for testing data

pred_LR_test = predict(fit_LR,X_test.data)

# Results

print(postResample(pred = pred_LR_test, obs =y_test$Absenteeism_time_in_hours))




###### Decision Tree


#Develop Model on training data


fit_DT = rpart(Absenteeism.time.in.hours ~., data = X_train.data, method = 'anova')

pred_DT_test = predict(fit_DT,X_test.data)

# for getting Results

print(postResample(pred = pred_DT_test, obs = y_test$Absenteeism.time.in.hours))




###### GBDT


#Develop Model on training data
```

```
fit_GBDT = gbm(Absenteeism_time_in_hours~., data = X_train.data, n.trees = 500,
interaction.depth = 2)

#Lets predict for testing data

pred_GBDT_test = predict(fit_GBDT,X_test.data, n.trees = 500)

# For testing data

print(postResample(pred = pred_GBDT_test, obs = y_test$Absenteeism.time.in.hours))
```

```
###### Random Forest

#Develop Model on training data

fit_DT = randomForest(Absenteeism_time_in_hours ~., data = X_train.data)

pred_DT_test = predict(fit_DT,X_test.data)

# for getting Results

print(postResample(pred = pred_DT_test, obs = y_test$Absenteeism.time.in.hours))
```