Employee Absenteeism

Sayan Nag

04 December 2018

Contents

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 loss every month can we project in 2011 if same trend of absenteeism continues?

We would be analyzing the historic data using tools like R and Python and predicting the causes which have been responsible for causing the absenteeism – and how the company may be able to remediate it.

1.2 Data

Our task is to design regression models which will predict the employee absenteeism based on other factors (variables as per company dataset). We have modified the dataset variables as per company guidelines. Given below is a sample of the data set that we are using to predict the employee absenteeism:

Table 1.1: Employee Absenteeism Sample Data (Columns: 1-2)

ID	Reason.for.absence
1	patient follow-up
1	medical consultation
1	unjustified absence
1	Diseases of the eye and adnexa
1	Diseases of the musculoskeletal system and connective tissue
1	medical consultation

Table 1.2: Employee Absenteeism Sample Data (Columns: 3-7)

Month.of.absence	Day.of.the.week	Seasons	Transportation.expense	Distance.from.Residence.to.Work
7	Monday	summer	235	11
8	Thursday	summer	235	11
12	Wednesday	spring	235	11
4	Friday	winter	235	11
6	Friday	winter	235	11
8	Tuesday	summer	235	11

Table 1.3: Employee Absenteeism Sample Data (Columns: 8-19)

Service.time	Age	Work.load.Average.day.	Hit.target	Disciplinary.failure	Education
14	37	239554	97	no	postgraduate
14	37	205917	92	no	postgraduate
14	37	261306	97	no	postgraduate
14	37	326452	96	no	postgraduate
14	37	264249	94	no	postgraduate
14	37	265615	94	no	postgraduate

Son	Social.drinker	Social.smoker	Pet	Weight	Height
1	no	no	1	88	172
1	no	no	1	88	172
1	no	no	1	88	172
1	no	no	1	88	172
1	no	no	1	88	172
1	no	no	1	88	172

Table 1.4: Employee Absenteeism Sample Data (Columns: 20-21)

Body.mass.index	Absenteeism.time.in.hours
29	8
29	4
29	8
29	3
29	16
29	1

There are 20 independent variables as per the given dataset as follows:

1. **ID**
2. **Reason.for.absence**
3. **Month.of.absence**
4. **Day.of.the.week**
5. **Seasons**
6. **Transportation.expense**
7. **Distance.from.Residence.to.Work**
8. **Service.time**
9. **Age**
10. **Work.load.Average.day**
11. **Hit.target**
12. **Disciplinary.failure**
13. **Education**
14. **Son**
15. **Social.drinker**
16. **Social.smoker**
17. **Pet**
18. **Weight**
19. **Height**
20. **Body.mass.index**

The dependent variable is **Absenteeism.time.in.hours**, which is **continuous** and to be predicted using the data of the independent variables.

Out of the 20 independent variables, following are the continuous and categorical variables:

Table 1.5: Continuous and Categorical independent variables

Continuous variables	**Categorical variables**
Transportation.expense	ID
Distance.from.Residence.to.Work	Reason.for.absence
Service.time	Month.of.absence
Age	Day.of.the.week
Work.load.Average.day	Seasons
Hit.target	Disciplinary.failure
Weight	Education
Height	Social.drinker
Body.mass.index	Social.smoker
Son	
Pet	

2. Methodology

2.1 Data Pre-Processing

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

We will be executing various pre-processing techniques such as **missing value imputation**, **outlier analysis**, **feature selection** and **feature sampling** in order to clean the data and make it suitable for becoming the input to our model. More often than not, data pre-processing can be challenging since this is the most important and time-consuming part of data analysis and modeling – and any small mistake can even get critical, resulting in an underperforming model.

2.1.1 Missing Value Analysis and Imputation

Mostly in real-time data, there are some observations which will be blank or missing. These missing values tend to make the data more difficult to analyze since the users find it difficult to apply statistical techniques like mean, mode, median, correlation, etc. on variables having missing data. Moreover, it is also not advisable to feed an input dataset with missing values to a model, since that may cause high anomalies in the analysis, which may cause wrong predictions altogether.
In our dataset, we have the following variables with missing values:

1. **Reason.for.Absence** : 3 missing values
2. **Month.of.absence** : 1 missing value
3. **Transportation.expense** : 7 missing values
4. **Distance.from.Residence.to.Work** : 3 missing values
5. **Service.time** : 3 missing values
6. **Age** : 3 missing values
7. **Work.load.Average.day** : 10 missing values
8. **Hit.target** : 6 missing values
9. **Disciplinary.failure** : 6 missing values
10. **Education** : 10 missing values
11. **Son** : 6 missing values
12. **Social.drinker** : 3 missing values
13. **Social.smoker** : 4 missing values
14. **Pet** : 2 missing values
15. **Weight** : 1 missing value
16. **Height** : 14 missing values
17. **Body.mass.index** : 31 missing values
18. **Absenteeism.time.in.hours** : 22 missing values

There are a few techniques by which missing values may be imputed in the dataset:

- **Dropping the observations with missing values** - This technique may be followed, however the percentage of missing values should be lesser than 30% of the total number of observations. Also, dropping observations may lead to loss of important information which may affect the behavior of the model.

- **Imputation with mean, median and mode –** This technique is quite useful and simple, wherein missing values are imputed with the respective mean or median of the variables (for numeric and continuous variables) and with mode of the variables (for categorical variables).
- **KNN imputation** – This technique uses a search algorithm with respect to distance in the data (determined by the value of k), wherein the missing field values get imputed with the most occurring element within that distance defined by the value of "k". This technique is quite slow as the volume of data grows, since the mechanism searches the entire dataset for imputing the missing values. Moreover, in order to avoid issues of duplicity, the values of "k" are always selected as odd positive numbers.

In our case, we have first replaced the missing values with zeroes (0s) and then imputed the continuous variables with their respective medians and the categorical variables with mode. Below are the sample code snippets for the same:

Code snippet:

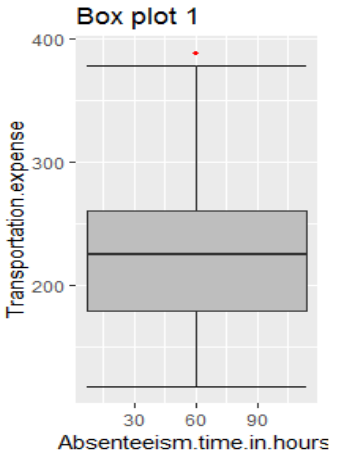
```
# Replacing 0s with mode in categorical variables


getmode = function(x) {
  uniqv = unique(x)
  uniqv[which.max(tabulate(match(x, uniqv)))]
}


data$Reason.for.absence =
replace(data$Reason.for.absence,data$Reason.for.absence=='0',getmode(data$Reason.for.absence))
data$Month.of.absence =
replace(data$Month.of.absence,data$Month.of.absence=='0',getmode(data$Month.of.absence))
data$Day.of.the.week =
replace(data$Day.of.the.week,data$Day.of.the.week=='0',getmode(data$Day.of.the.week))
data$Seasons = replace(data$Seasons,data$Seasons=='0',getmode(data$Seasons))


# Replacing 0s with median in continuous variables


data$Work.load.Average.day. =
replace(data$Work.load.Average.day.,data$Work.load.Average.day.==0,median(data$Work.load.Average
.day.))
data$Hit.target = replace(data$Hit.target,data$Hit.target==0,median(data$Hit.target))
data$Absenteeism.time.in.hours =
replace(data$Absenteeism.time.in.hours,data$Absenteeism.time.in.hours==0,median(data$Absenteeism.t
ime.in.hours))
```

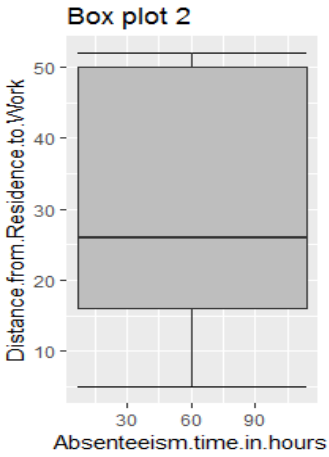
```
# Sorting data with ID
data = data[order(data$ID),]
# View(data)

#Converting ID and Month of Absence as factor

data$ID = as.factor(data$ID)
unique(data$ID)
length(unique(data$ID))
for (i in range(1,length(unique(data$ID)))) {
   data$Transportation.expense =
replace(data$Transportation.expense,data$Transportation.expense==0,aggregate(data$Transportation.ex
pense, by=list(ID=data$ID), FUN=median)[2][i,])
   data$Distance.from.Residence.to.Work =
replace(data$Distance.from.Residence.to.Work,data$Distance.from.Residence.to.Work==0,aggregate(dat
a$Distance.from.Residence.to.Work, by=list(ID=data$ID), FUN=median)[2][i,])
   data$Service.time = replace(data$Service.time,data$Service.time==0,aggregate(data$Service.time,
by=list(ID=data$ID), FUN=median)[2][i,])
   data$Age = replace(data$Age,data$Age==0,aggregate(data$Age, by=list(ID=data$ID),
FUN=median)[2][i,])
```

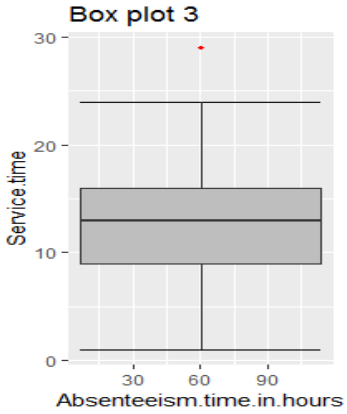
2.1.2 Outlier analysis

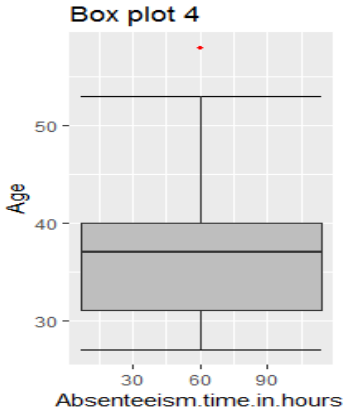
This is a technique for ensuring that the **continuous** variables in the data do not contain values which vary largely from the other sets of values. Whenever we are attempting to design a model, the input data should be devoid of outliers so that the output predicted by the model does not get negatively affected by the outliers.

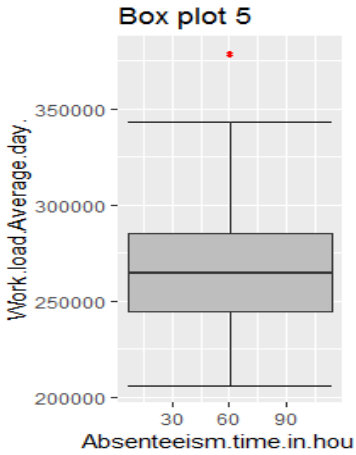
In our case, we have performed a graphical analysis using **box-plots** to check the outliers for all the continuous variables, shown as below:

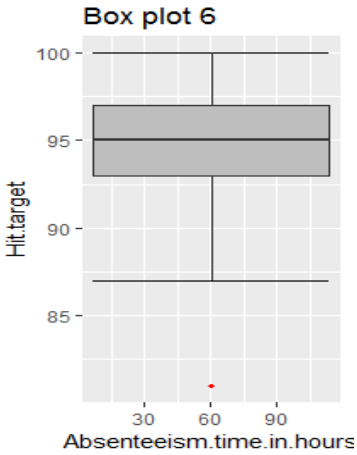

Box plot 1
Transportation.expense
400
300
200
30
60
90
Absenteeism.time.in.hours

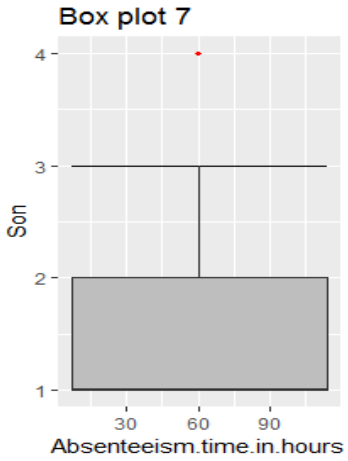

Box plot 2
Distance.from.Residence.to.Work
50
40
30
20
10
30
60
90
Absenteeism.time.in.hours

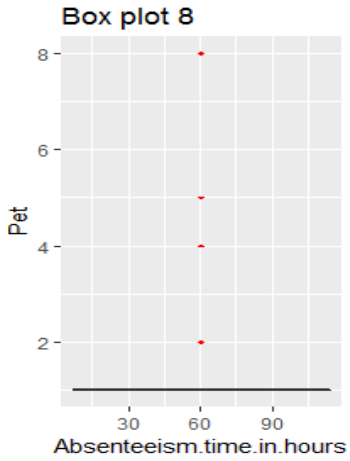

Box plot 3
Service.time
30
20
10
0
30
60
90
Absenteeism.time.in.hours

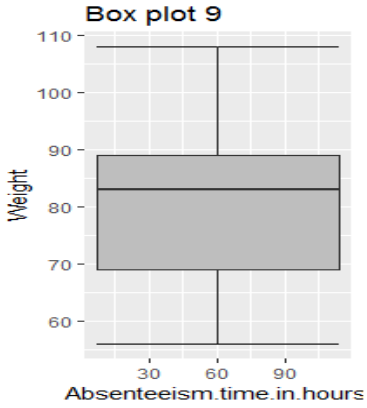

Box plot 4
Age
50
40
30
30
60
90
Absenteeism.time.in.hours

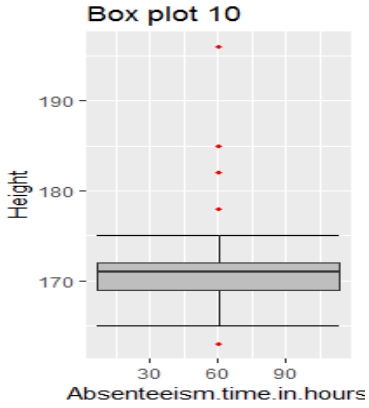

Box plot 5
Work.load.Average.day.
350000
300000
250000
200000
30
60
90
Absenteeism.time.in.hou

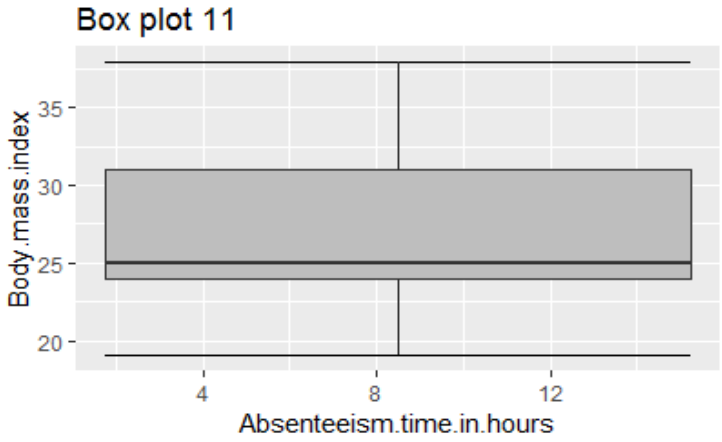

Box plot 6
Hit.target
100
95
90
85
30
60
90
Absenteeism.time.in.hours











After performing graphical analysis of outliers, we have replaced the outliers with zeroes (0s) in the dataset and then replaced those zeroes with median (for continuous variables) and mode (for categorical variables), exactly in the same way that we have done for missing value imputation.

Following is the code snippet for replacing the outliers with zeroes in the dataset:

Code-snippet:

```
df = data
data = df

#Replace all outliers with 0 and impute

for(i in cnames){
  #print(i)
  val = data[,i][data[,i] %in% boxplot.stats(data[,i])$out]
  print(length(val))
  data[,i][data[,i] %in% val] = 0
}
```

2.1.3 Feature Selection

Post performing missing value and outlier analyses, we need to perform feature selection. Often, the dataset provided by the client/company contains many unnecessary or unimportant variables which do not impact the target/dependent variable. If we keep these variables and pass the input to our model, it causes unnecessary overheads with respect to memory consumption and also model performance. That is why it is absolutely necessary to perform feature selection and determine which independent variables actually impact the target variable and accordingly, we may design the input for the model.

In this project, the following feature selection techniques have been utilized:

a) **Correlation analysis**: This is a test performed essentially only on **continuous numeric** variables and is used to visualize the correlation co-efficients of each independent numeric variable among themselves and also with respect to the dependent or target variable. This test helps one to determine which variables to keep in the dataset and which ones to drop. It is suggested that numeric variables with zero correlation co-efficients with respect to target variable need to be dropped as there is no dependency between those variables and the target variable.

   Following is the code snippet:

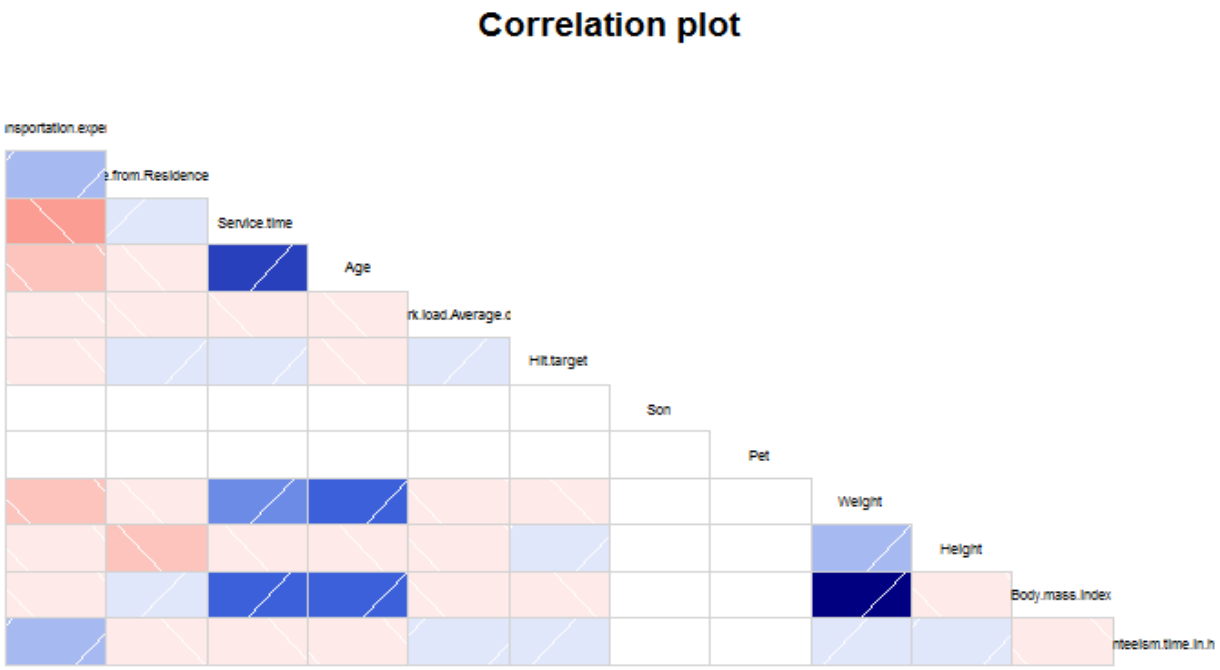
```
value = round(cor(as.matrix(data[,numerics])),2)

value

library(corrgram)
corrgram(data[,numerics], order=FALSE, lower.panel=panel.shade,
         upper.panel=NULL, text.panel=panel.txt,
         main="Correlation plot")
```

Following is the summary of the correlation test on the given dataset:

```
> value
                                 Month.of.absence Transportation.expense Di
Month.of.absence                             1.00                    0.15
Transportation.expense                       0.15                    1.00
Distance.from.Residence.to.Work             -0.01                    0.27
Service.time                                -0.08                   -0.39
Age                                          0.01                   -0.27
Work.load.Average.day.                      -0.17                   -0.03
Hit.target                                  -0.48                   -0.08
Son                                           NA                      NA
Pet                                           NA                      NA
Weight                                       0.02                   -0.21
Height                                      -0.03                   -0.08
Body.mass.index                              0.05                   -0.13
Absenteeism.time.in.hours                    0.01                    0.19
                                 Work.load.Average.day. Hit.target Son Pet
Month.of.absence                                -0.17       -0.48  NA  NA
Transportation.expense                          -0.03       -0.08  NA  NA
Distance.from.Residence.to.Work                 -0.09        0.03  NA  NA
Service.time                                    -0.05        0.07  NA  NA
Age                                             -0.06        0.00  NA  NA
Work.load.Average.day.                           1.00        0.05  NA  NA
Hit.target                                       0.05        1.00  NA  NA
Son                                                NA          NA   1  NA
Pet                                                NA          NA  NA   1
Weight                                          -0.08        0.00  NA  NA
Height                                          -0.05        0.02  NA  NA
Body.mass.index                                 -0.10       -0.03  NA  NA
Absenteeism.time.in.hours                        0.06        0.02  NA  NA
>
```

```
> value
                                 Distance.from.Residence.to.Work Service.time   Age
Month.of.absence                                           -0.01        -0.08  0.01
Transportation.expense                                      0.27        -0.39 -0.27
Distance.from.Residence.to.Work                             1.00         0.11 -0.11
Service.time                                                0.11         1.00  0.67
Age                                                        -0.11         0.67  1.00
Work.load.Average.day.                                     -0.09        -0.05 -0.06
Hit.target                                                  0.03         0.07  0.00
Son                                                           NA           NA    NA
Pet                                                           NA           NA    NA
Weight                                                     -0.04         0.42  0.48
Height                                                     -0.20        -0.10 -0.05
Body.mass.index                                             0.12         0.46  0.52
Absenteeism.time.in.hours                                   0.00        -0.08 -0.07
                                 Weight Height Body.mass.index Absenteeism.time.in.hours
Month.of.absence                   0.02  -0.03           0.05                      0.01
Transportation.expense            -0.21  -0.08          -0.13                      0.19
Distance.from.Residence.to.Work   -0.04  -0.20           0.12                      0.00
Service.time                       0.42  -0.10           0.46                     -0.08
Age                                0.48  -0.05           0.52                     -0.07
Work.load.Average.day.            -0.08  -0.05          -0.10                      0.06
Hit.target                         0.00   0.02          -0.03                      0.02
Son                                  NA     NA             NA                        NA
Pet                                  NA     NA             NA                        NA
Weight                             1.00   0.15           0.88                      0.00
Height                             0.15   1.00          -0.13                      0.04
Body.mass.index                    0.88  -0.13           1.00                      0.00
Absenteeism.time.in.hours          0.00   0.04           0.00                      1.00
>
```

Following is the visualization plot of the correlation test:



In the above correlation plot, deep red signifies high negative correlation and deep blue signifies high positive correlation.

From the given dataset, after performing correlation test, we find that the dependent variable (**Absenteeism.time.in.hours**) is having zero correlation with respect to the numeric variables **Distance.from.Residence.to.Work**, **Weight** and **Body.mass.index**. Moreover, the correlation is NA for the variables **Son** and **Pet**.

Thus it may be assumed that these variables do not have any impact or effect on the target variable **Absenteeism.time.in.hours** and hence, have been dropped.

b) **ANOVA (Analysis of Variance) test**: This test is useful to determine the dependence of categorical independent variables and continuous dependent variable. It uses two hypothesis principles, viz., **null** hypothesis and **alternate** hypothesis.

The null hypothesis assumes that the selected variables are independent of each other whereas the alternate hypothesis assumes that the selected variables under consideration are not independent of each other. It is guided based on a parameter known as the probability factor or more popularly, as the p-value.

If the p-value is lesser than 0.05, then the null hypothesis is rejected with the assumption that the variable under consideration is having dependency with the target variable, i.e., the variable carries information to describe the target variable.

In the given dataset, ANOVA test has been performed to select the categorical variables and following is the result:

```
> summary(anova_result)
                     Df Sum Sq Mean Sq F value Pr(>F)
Reason.for.absence   26   2394   92.06  12.877 <2e-16 ***
Month.of.absence     11     82    7.49   1.048  0.402
Day.of.the.week       4     27    6.67   0.933  0.444
Seasons               3     26    8.76   1.225  0.300
Disciplinary.failure  1      0    0.11   0.016  0.901
Education             3      5    1.60   0.223  0.880
Social.drinker        1     11   10.86   1.518  0.218
Social.smoker         1      4    4.37   0.611  0.435
Residuals           689   4926    7.15
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

Following is the code snippet of the ANOVA test:

```
# Anova test for categorical independent and continuous dependent variable

data$Month.of.absence = as.factor(data$Month.of.absence)
data$Reason.for.absence = as.factor(data$Reason.for.absence)
data$Day.of.the.week = as.factor(data$Day.of.the.week)
data$Seasons = as.factor(data$Seasons)
data$Disciplinary.failure = as.factor(data$Disciplinary.failure)
data$Education = as.factor(data$Education)
data$Social.drinker = as.factor(data$Social.drinker)
data$Social.smoker = as.factor(data$Social.smoker)

facts = sapply(data, is.factor)
facts_cols = data[,facts]
cnames_facts = colnames(facts_cols)

anova_result <- aov(Absenteeism.time.in.hours~Reason.for.absence*Month.of.absence*Day.of.the.week
                    *Seasons*Disciplinary.failure*Education*Social.drinker*Social.smoker , data = data)
summary(anova_result)
```

Since the p-value for **Reason.for.absence** is 2e-16, which is way lesser than 0.05, we reject the null hypothesis for this variable and confirm that this variable contains information to define the target variable(**Absenteeism.time.in.hours**). For the other variables, the p-values are greater than 0.05 and hence, the alternate hypothesis becomes true and thus, the variables are dropped.

Thus, post feature selection, the following 12 independent variables are dropped from the original dataset:

1) Month.of.absence
2) Day.of.the.week
3) Seasons
4) Disciplinary.failure
5) Education
6) Social.drinker
7) Social.smoker
8) Distance.from.Residence.to.Work
9) Son
10) Pet
11) Weight
12) Body.mass.index

Thus, the dataset now contains (21-12) = 9 variables of which, 8 are independent variables and the 9th variable is the target/dependent variable.

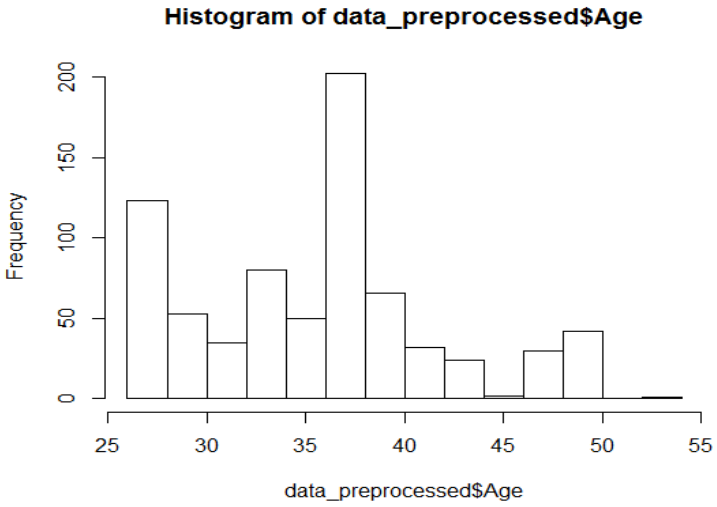
2.1.4 Feature Sampling

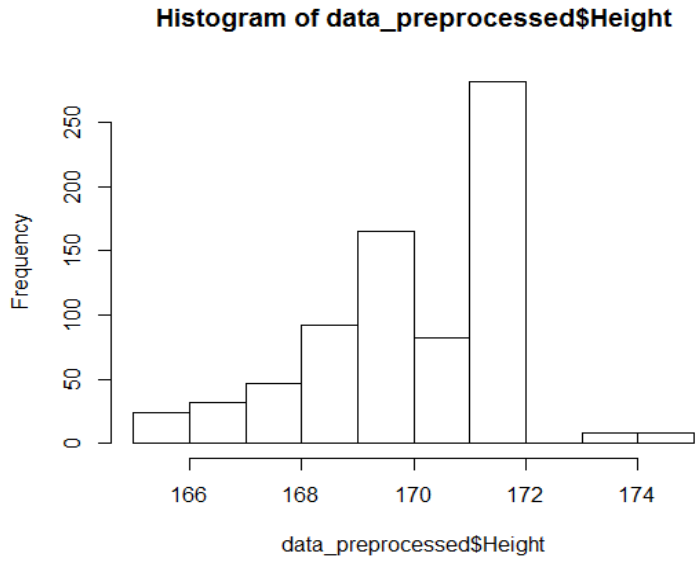
It is a technique which modifies the range of the variables in the dataset as per some pre-defined limits. Many a times one or more variables may vary between a huge range. If those variables are not sampled, it may cause biasing effects in the model wherein, the model performance will be more inclined towards the high-ranged variable and therefore, the effect of the other variable/s may be reduced or shielded.

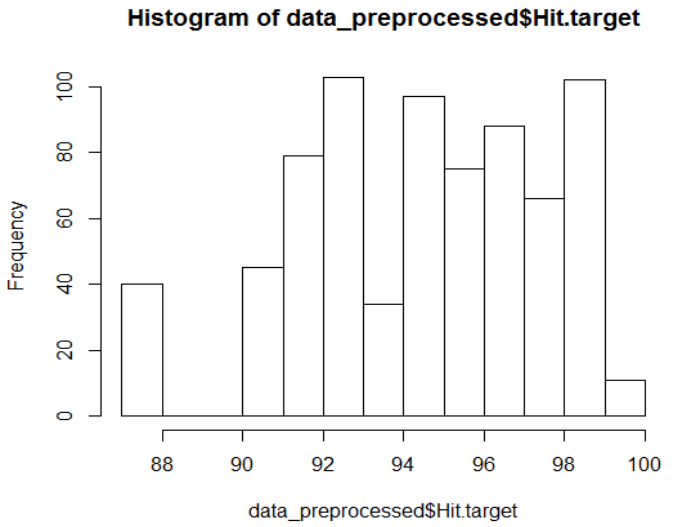
In order to avoid this situation, the data is being sampled. In this project, **normalization** has been used as the sampling technique, wherein the continuous variables have been normalized in a range of minimum 0 to maximum 1.

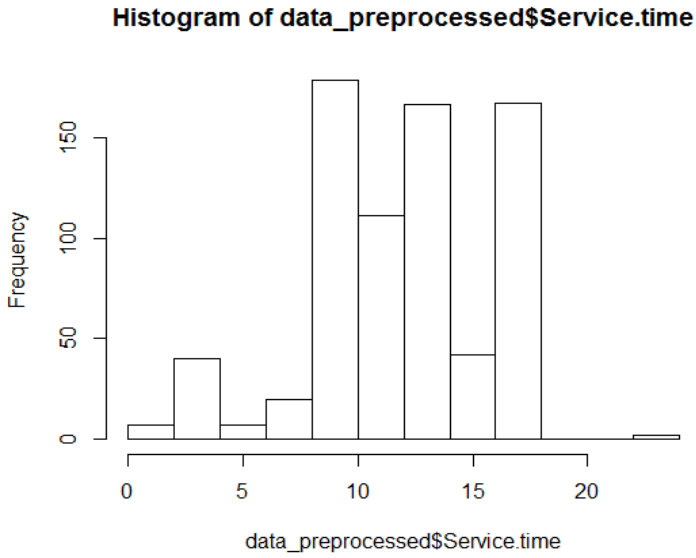
Following are the distribution plots(histogram) of the continuous variables before normalization:

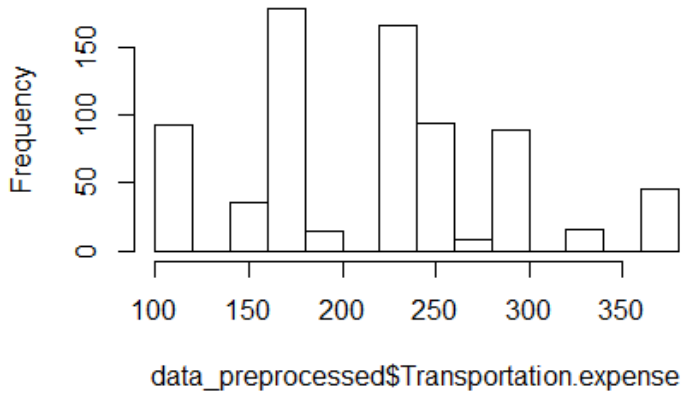
Distributions before normalization:

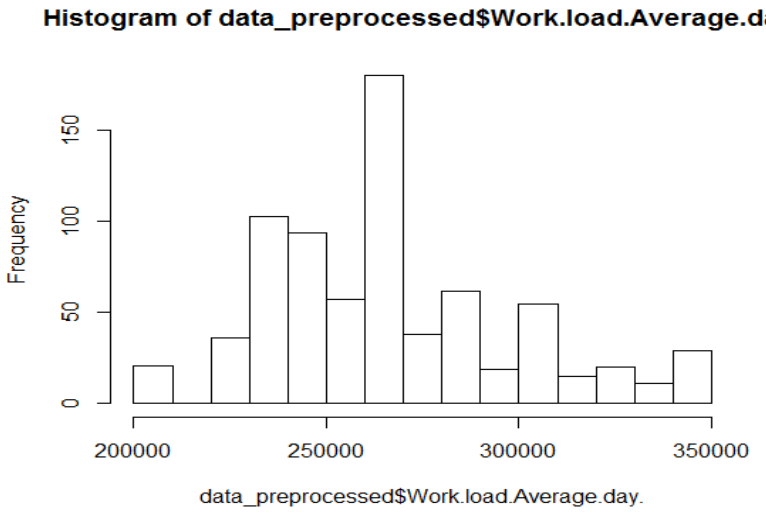





Histogram of data_preprocessed$Hit.target
Frequency
0
20
40
60
80
100
88
90
92
94
96
98
100
data_preprocessed$Hit.target

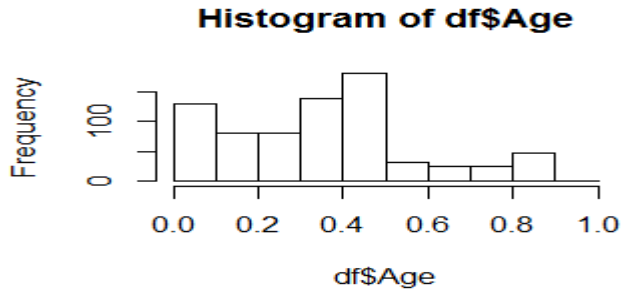

Histogram of data_preprocessed$Service.time
Frequency
0
50
100
150
0
5
10
15
20
data_preprocessed$Service.time

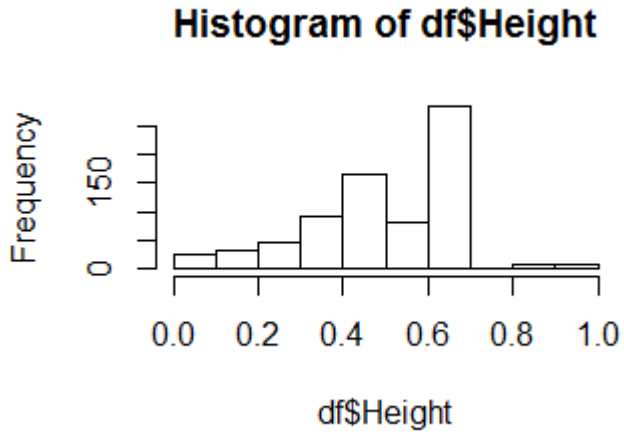

Frequency
0
50
100
150
100
150
200
250
300
350
data_preprocessed$Transportation.expense

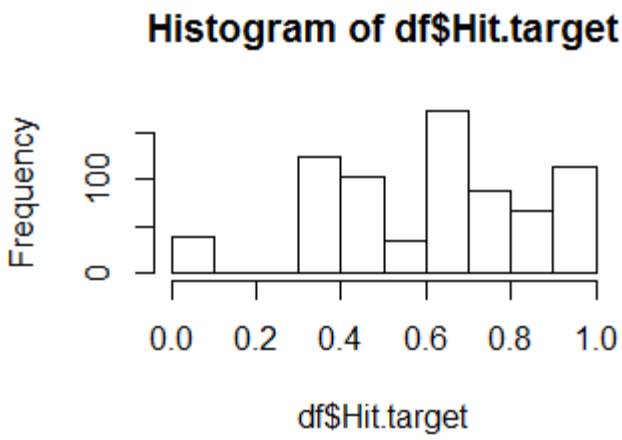


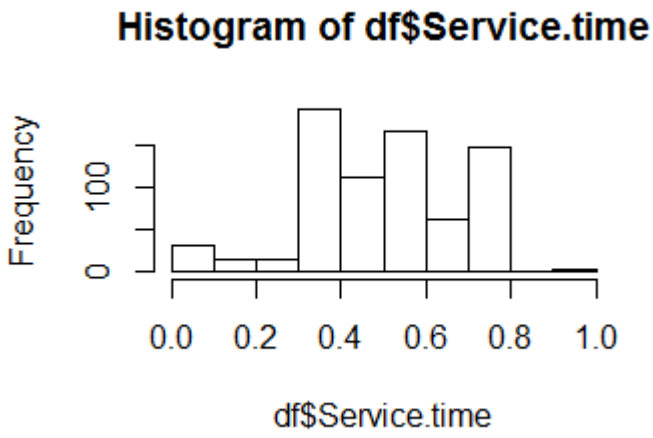
From the diagrams, it is quite evident that none of the variables are normally distributed. Post normalization, following are the histogram plots:

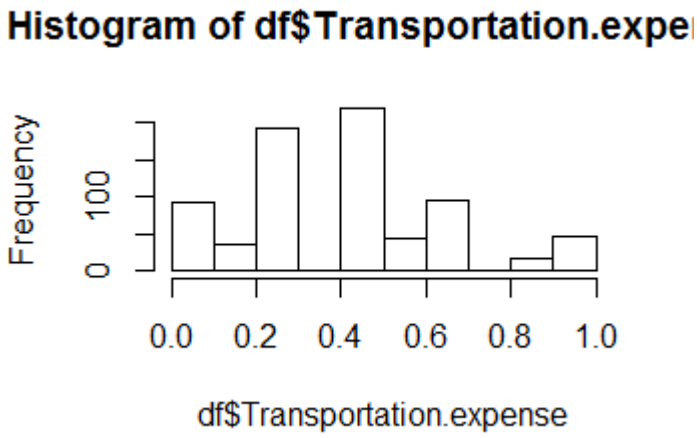
Distributions after normalization:

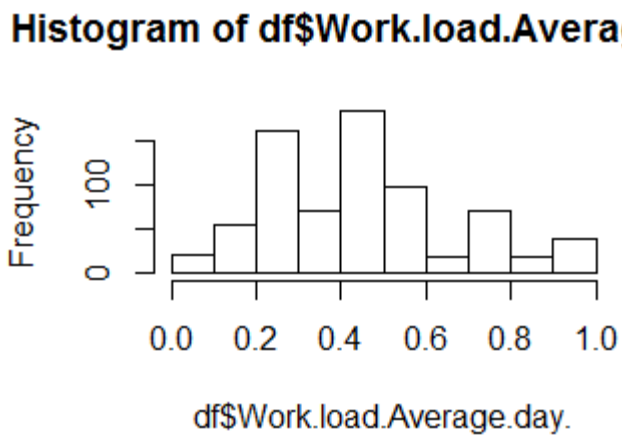





Histogram of df$Hit.target
Frequency
0
100
0.0
0.2
0.4
0.6
0.8
1.0
df$Hit.target


Histogram of df$Service.time
Frequency
0
100
0.0
0.2
0.4
0.6
0.8
1.0
df$Service.time


Histogram of df$Transportation.expe
Frequency
0
100
0.0
0.2
0.4
0.6
0.8
1.0
df$Transportation.expense



Following is the **summary** of the data post applying the three data pre-processing techniques, as explained above:

```
> summary(df)
       ID                                                                  Reason.for.absence Transportation.expense
 3      :113   medical consultation                                              :195       Min.   :0.0000
 28     : 76   dental consultation                                               :110       1st Qu.:0.2346
 34     : 55   physiotherapy                                                     : 69       Median :0.4115
 22     : 46   Diseases of the musculoskeletal system and connective tissue      : 55       Mean   :0.3944
 20     : 42   Injury, poisoning and certain other consequences of external causes: 40      3rd Qu.:0.5462
 11     : 40   patient follow-up                                                 : 37       Max.   :1.0000
 (Other):368   (Other)                                                           :234
  Service.time        Age        Work.load.Average.day.   Hit.target        Height       Absenteeism.time.in.hours
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000        Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
 1st Qu.:0.3478   1st Qu.:0.1538   1st Qu.:0.2801        1st Qu.:0.4615   1st Qu.:0.4000   1st Qu.:0.06667
 Median :0.5217   Median :0.3846   Median :0.4247        Median :0.6154   Median :0.6000   Median :0.13333
 Mean   :0.4987   Mean   :0.3548   Mean   :0.4398        Mean   :0.6115   Mean   :0.5427   Mean   :0.22243
 3rd Qu.:0.6522   3rd Qu.:0.5000   3rd Qu.:0.5688        3rd Qu.:0.7692   3rd Qu.:0.7000   3rd Qu.:0.46667
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000        Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
```

2.2 Modeling

2.2.1 Model Selection

In this case, we see that the dependent variable absenteeism is continuous in nature and moreover, we are supposed to predict how the absenteeism may be reduced. Thus, it falls under a **regression** problem wherein we need to select a model suitable for regression.

We will be checking out with the Decision Tree regression model and also Linear Regression model. However for linear regression model, all the input variables must be of numeric nature.

2.2.2 Decision Tree Regression Model

Following is the model implementation:

```
# Decision Tree regression model design

library(rpart)
#df = data_preprocessed
#df = data

#df = df[,-1]

# Dividing data into train and test

train_index_norm = sample(1:nrow(df), 0.95*nrow(df))
train_norm = df[train_index_norm,]
test_norm = df[-train_index_norm,]

fit = rpart(Absenteeism.time.in.hours ~. , data = train_norm, method = "anova")
predictions_DT = predict(fit,test_norm[,-9])

# Calculating model accuracy

library(DMwR)
regr.eval(test_norm[,9], predictions_DT, stats = c('mae','rmse','mse','mape'))
```

We have fed 95% of the observations in the training dataset and 5% observations in the testing dataset.

2.2.3 Decision Tree Regression Model Performance

The error metrics calculated from this model are as follows:

```
      mae       rmse        mse       mape
0.10252619 0.14606159 0.02133399        Inf
>
```

We see that the MAPE is infinite, which is due to the presence of 0s in the target variable absenteeism, inserted due to normalization.

However, the other error metrics are quite acceptable. If we consider the mean absolute error (mae), it is just about 10.25%, meaning the model is **89.75% accurate**.

Moreover, realizing that this is a time-series problem, we may also consider **mean square error (mse)** and **root mean square error (rmse)**, both of which are just about **2.13%** and **14.60%** respectively. Again, these metrics are quite acceptable.

2.2.4 Linear Regression Model

The linear regression model allows only numeric variables as input and hence, we have included only numeric variables, by excluding the categorical variables ID and Reason.for.absence.

Following is the implementation of the model:

```
# Linear Regression model design

library(usdm)
df2 = df
df2 = df2[-1]
df2 = df2[-1]
vif(df2[,-7])
vifcor(df2[,-7], th=0.9)

train_index_LM = sample(1:nrow(df2), 0.95*nrow(df2))
train_LM = df2[train_index_LM,]
test_LM = df2[-train_index_LM,]

lm_model = lm(Absenteeism.time.in.hours~., data = train_LM)

summary(lm_model)

predictions_LR=predict(lm_model,test_LM[,1:6])

regr.eval(test_LM[,7], predictions_LR, stats = c('mae','rmse','mse','mape'))
```

2.2.5 Linear Regression Model Performance

Following is the statistic of the linear model summary:

```
> summary(lm_model)

Call:
lm(formula = Absenteeism.time.in.hours ~ ., data = train_LM)

Residuals:
     Min       1Q   Median       3Q      Max
-0.32338 -0.13822 -0.07394  0.16464  0.81563

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          0.08463    0.05078   1.667   0.0960 .
Transportation.expense  0.16718    0.03388   4.934 1.01e-06 ***
Service.time         0.02495    0.06110   0.408   0.6831
Age                 -0.02510    0.04567  -0.550   0.5827
Work.load.Average.day.  0.06855    0.03457   1.983   0.0478 *
Hit.target           0.01384    0.03363   0.412   0.6807
Height               0.05587    0.04311   1.296   0.1954
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2079 on 696 degrees of freedom
Multiple R-squared:  0.04522,   Adjusted R-squared:  0.03699
F-statistic: 5.494 on 6 and 696 DF,  p-value: 1.431e-05
```

We see that the adjusted R-square is just about 3.699%, which is not impressive. It implies that the input data is able to define the target variable by only 3.699%.

However, by looking at the F-statistic and p-value (1.431e-05), we can at least reject the null hypothesis by saying that there is at least some dependency between the target variable(absenteeism) and the input variables.

The error metrics calculated from this model are as follows:

```
      mae       rmse        mse       mape
0.16904360 0.20993244 0.04407163        Inf
>
```

Like in decision trees, the MAPE value is infinity due to the presence of zeroes in the target variable. The mae is about 16.90%, rmse about 20.99% and mse about 4.40%.

Chapter 3

Conclusion

3.1 Model Evaluation

Now that we have two 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 data, 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 the predictions of the models with real values of the target variable and calculating some average error measure.

3.1.1 Mean Absolute Error (MAE) and Mean Square Error (MSE)

MAE and MSE are the important error measures used to calculate the predictive performance of the model. We will apply these measures to our models that we have generated in the previous section.

MAE for **Decision Trees**:

```
# Calculating model accuracy

library(DMwR)
regr.eval(test_norm[,9], predictions_DT, stats = c('mae','rmse','mse','mape'))
      mae       rmse          mse                mape
0.10252619 0.14606159 0.02133399                 Inf
>
```

MAE for **Linear Regression**:

```
# Calculating model accuracy

library(DMwR)
regr.eval(test_LM[,7], predictions_LR, stats = c('mae','rmse','mse','mape'))
      mae       rmse          mse                mape
0.16904360 0.20993244 0.04407163                 Inf
>
```

3.2 Model Selection

We can see that in Decision Tree, the **MAE** and **MSE** are approximately **10.25%** and **2.13%** respectively. In Linear Regression, these are increased to **16.90%** and **4.40%** respectively.

In other words, **Decision Tree Regression model is fetching more accuracy** (almost 89.75%) for us as compared to that by Linear Regression (83.10%).

We will therefore select Decision Tree as the suitable model for predicting the employee absenteeism more accurately.

3.3 Answers to Questions

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

   As we can see from the model that employee absenteeism depends on the following variables:

   - Transportation.expense (correlation coefficient : 0.19 positive)
   - Service.time (correlation coefficient : -0.08 negative)
   - Age (correlation coefficient : -0.07 negative)
   - Work.load.Average.day (correlation coefficient : 0.06 positive)
   - Hit.target (correlation coefficient : 0.02 positive)
   - Height (correlation coefficient : 0.04 positive)
   - Reason.for.absence (mode value : medical consultation)

   From the above details, we can assume that in order to reduce the number of absenteeism, the company needs to:

   - Engage employees with relatively lower transportation expense
   - Engage employees with higher service time (i.e., more experienced employees)
   - Engage employees with higher age as compared to younger employees
   - Reduce the workload on the employees to a certain extent
   - Reduce the hit target of employees marginally to a certain extent
   - Engage employees with relatively shorter heights more than the taller ones
   - Verify the medical history of employees and if they have a history of medical consultation, should keep away from those employees

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

   Loss may be projected as a calculated field by using the variables **Work.load.Average.day**, **Hit.target** and **Absenteeism.time.in.hours**. We understand that workload corresponds to the expected output from the employees and hit target is the desired target to be attained by the employees.

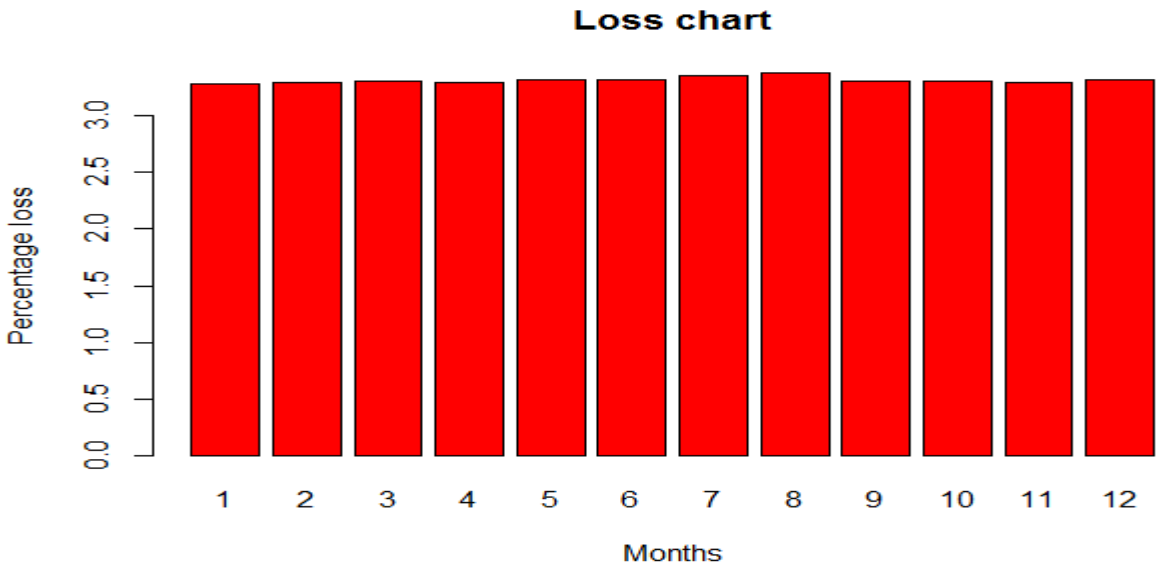
   Thus, if we calculate the total workload, desired hit target and absenteeism for a month and then calculate the actual hit target attained, that will give an impression of the loss per month.

Below is the code for calculating the loss per month:

```
# Logic for calculating loss per month. We will calculate loss from the original dataset "data".

df1 = data
list1 = c("Work.load.Average.day.","Hit.target","Absenteeism.time.in.hours")
for (i in list1){
  df1[,i] = (df1[,i] - min(df1[,i]))/((max(df1[,i])) - (min(df1[,i])))
}
a = aggregate(df1$Work.load.Average.day., by=list(Month.of.absence=df1$Month.of.absence), FUN=sum)
b = aggregate(df1$Hit.target, by=list(Month.of.absence=df1$Month.of.absence), FUN=sum)
c = aggregate(df1$Absenteeism.time.in.hours, by=list(Month.of.absence=df1$Month.of.absence), FUN=sum)
d = aggregate(df1$ID, by=list(Month.of.absence=df1$Month.of.absence), FUN=unique)

# Assumption is that every month has 30 days

# We are comparing the workload (in hours) for every month (expected output) and comparing it with the hit target.
# Also, we are subtracting the absenteeism (in hours) to get the actual output and calculate loss.

for (i in c(1,2,3,4,5,6,7,8,9,10,11,12)) {
  # Calculating loss
  #print(i)
  loss = (b[2][i,]-(b[2][i,]/(a[2][i,]*31*length(d[2][i,][[1]])))*((a[2][i,]*30*length(d[2][i,][[1]]))-c[2][i,]))/b[2][i,]*100
  print(loss)
}

x = c(3.279278,3.293305,3.301384,3.294327,3.321255,3.316309,3.356788,3.377586,3.301268,3.310392,3.289171,3.324995)
y = c("1","2","3","4","5","6","7","8","9","10","11","12")
barplot(x,names.arg=y,xlab="Months",ylab="Percentage loss",col="red",
        main="Loss chart",border="black")
```

Here, we have grouped all the required data by month and then calculated the loss for each month by applying simple unitary method technique. We see that the loss projected by this technique is approximately lesser than 4% for each month.

We have also plotted the bar chart to visualize the loss per month, as follows:



Appendix A - R Code

```
#remove all the objects stored
rm(list=ls())

#set current working directory
setwd("C:/Users/SAYAN/Desktop/Data Science/Projects/R Folder")

#Current working directory
getwd()

#instaling packages
install.packages(c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "C50",
"dummies", "e1071", "Information",
     "MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine',
'inTrees','Hmisc','corrplot','usdm'))

#reading Excel sheet
library(xlsx)
data = read.xlsx("Absenteeism_at_work_Project.xls", sheetIndex = 1, header = T)
#data

#Getting the column names of the dataset
colnames(data)

#Getting the structure of the dataset
str(data)

#Getting the number of variables and obervation in the datasets
#dim(data)

#data type
#class(data)

#Summary of a varaible
#summary(data)

df_summary = as.data.frame(summary(data))

df_summary

as.character(data$Reason.for.absence)

data

# Replacing values in "Reason for absence"

# With ICD
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="1",
"Certain infectious and parasitic diseases")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="2",
"Neoplasms")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="3",
"Diseases of the blood and blood-forming organs and certain disorders involving the immune
mechanism")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="4",
"Endocrine, nutritional and metabolic diseases")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="5",
"Mental and behavioural disorders")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="6",
"Diseases of the nervous system")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="7",
"Diseases of the eye and adnexa")
```

```
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="8",
"Diseases of the ear and mastoid process")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="9",
"Diseases of the circulatory system")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="10",
"Diseases of the respiratory system")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="11",
"Diseases of the digestive system")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="12",
"Diseases of the skin and subcutaneous tissue")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="13",
"Diseases of the musculoskeletal system and connective tissue")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="14",
"Diseases of the genitourinary system")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="15",
"Pregnancy, childbirth and the puerperium")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="16",
"Certain conditions originating in the perinatal period")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="17",
"Congenital malformations, deformations and chromosomal abnormalities")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="18",
"Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="19",
"Injury, poisoning and certain other consequences of external causes")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="20",
"External causes of morbidity and mortality")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="21",
"Factors influencing health status and contact with health services")
# Without ICD
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="22",
"patient follow-up")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="23",
"medical consultation")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="24",
"blood donation")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="25",
"laboratory examination")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="26",
"unjustified absence")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="27",
"physiotherapy")
data$Reason.for.absence = replace(data$Reason.for.absence,data$Reason.for.absence=="28",
"dental consultation")

# Replacing values in "Day of the week"

data$Day.of.the.week = replace(data$Day.of.the.week,data$Day.of.the.week==3, "Tuesday")
data$Day.of.the.week = replace(data$Day.of.the.week,data$Day.of.the.week==2, "Monday")
data$Day.of.the.week = replace(data$Day.of.the.week,data$Day.of.the.week==4, "Wednesday")
data$Day.of.the.week = replace(data$Day.of.the.week,data$Day.of.the.week==5, "Thursday")
data$Day.of.the.week = replace(data$Day.of.the.week,data$Day.of.the.week==6, "Friday")

# Replacing "Seasons"

data$Seasons = replace(data$Seasons,data$Seasons==1, "summer")
data$Seasons = replace(data$Seasons,data$Seasons==2, "autumn")
data$Seasons = replace(data$Seasons,data$Seasons==3, "winter")
data$Seasons = replace(data$Seasons,data$Seasons==4, "spring")

# Replacing "Disciplinary failure"

data$Disciplinary.failure = replace(data$Disciplinary.failure,data$Disciplinary.failure==1,"yes")
data$Disciplinary.failure = replace(data$Disciplinary.failure,data$Disciplinary.failure==0,"no")

# Replacing "Education"

data$Education = replace(data$Education,data$Education==1,"high school")
data$Education = replace(data$Education,data$Education==2,"graduate")
```

```
data$Education = replace(data$Education,data$Education==3,"postgraduate")
data$Education = replace(data$Education,data$Education==4,"master and doctor")

# Replacing "Social smoker"

data$Social.smoker = replace(data$Social.smoker,data$Social.smoker==1,"yes")
data$Social.smoker = replace(data$Social.smoker,data$Social.smoker==0,"no")

# Replacing "Social drinker"

data$Social.drinker = replace(data$Social.drinker,data$Social.drinker==1,"yes")
data$Social.drinker = replace(data$Social.drinker,data$Social.drinker==0,"no")

#str(data)
View(data)

#missing_val = as.data.frame(data[NA])

missing_val = as.data.frame(apply(data,2,function(x){sum(is.na(x))}))
View(missing_val)
missing_val$Columns = row.names(missing_val)
row.names(missing_val) = NULL
names(missing_val)[1] = 'Missing percentage'
missing_val = missing_val[,c(2,1)]
missing_val$`Missing percentage` = (missing_val$`Missing percentage`/nrow(data))*100

# Replacing missing values with 0

data[is.na(data)] = 0

# Replacing 0s with mode in categorical variables

getmode = function(x) {
  uniqv = unique(x)
  uniqv[which.max(tabulate(match(x, uniqv)))]
}

data$Reason.for.absence =
replace(data$Reason.for.absence,data$Reason.for.absence=='0',getmode(data$Reason.for.absenc
e))
data$Month.of.absence =
replace(data$Month.of.absence,data$Month.of.absence=='0',getmode(data$Month.of.absence))
data$Day.of.the.week =
replace(data$Day.of.the.week,data$Day.of.the.week=='0',getmode(data$Day.of.the.week))
data$Seasons = replace(data$Seasons,data$Seasons=='0',getmode(data$Seasons))
data$Disciplinary.failure =
replace(data$Disciplinary.failure,data$Disciplinary.failure=='0',getmode(data$Disciplinary.failure))
data$Education = replace(data$Education,data$Education=='0',getmode(data$Education))
data$Social.drinker =
replace(data$Social.drinker,data$Social.drinker=='0',getmode(data$Social.drinker))
data$Social.smoker =
replace(data$Social.smoker,data$Social.smoker=='0',getmode(data$Social.smoker))

#write.(data,'No_Missing_or_Zeroes')

#write.xlsx(data,'NoMissingorZeroes.xls',sheetName = 'Sheet1',row.names = T,col.names = T)

# Replacing 0s with median in continuous variables

data$Work.load.Average.day. =
replace(data$Work.load.Average.day.,data$Work.load.Average.day.==0,median(data$Work.load.Av
erage.day.))
data$Hit.target = replace(data$Hit.target,data$Hit.target==0,median(data$Hit.target))
data$Absenteeism.time.in.hours =
replace(data$Absenteeism.time.in.hours,data$Absenteeism.time.in.hours==0,median(data$Absent
eeism.time.in.hours))
```

```
# Sorting data with ID

data = data[order(data$ID),]
# View(data)

#Converting ID and Month of Absence as factor

data$ID = as.factor(data$ID)


unique(data$ID)
length(unique(data$ID))
for (i in range(1,length(unique(data$ID)))) {
   data$Transportation.expense =
replace(data$Transportation.expense,data$Transportation.expense==0,aggregate(data$Transporta
tion.expense, by=list(ID=data$ID), FUN=median)[2][i,])
   data$Distance.from.Residence.to.Work =
replace(data$Distance.from.Residence.to.Work,data$Distance.from.Residence.to.Work==0,aggreg
ate(data$Distance.from.Residence.to.Work, by=list(ID=data$ID), FUN=median)[2][i,])
   data$Service.time =
replace(data$Service.time,data$Service.time==0,aggregate(data$Service.time, by=list(ID=data$ID),
FUN=median)[2][i,])
   data$Age = replace(data$Age,data$Age==0,aggregate(data$Age, by=list(ID=data$ID),
FUN=median)[2][i,])
   data$Son = replace(data$Son,data$Son==0,aggregate(data$Son, by=list(ID=data$ID),
FUN=median)[2][i,])
   data$Pet = replace(data$Pet,data$Pet==0,aggregate(data$Pet, by=list(ID=data$ID),
FUN=median)[2][i,])
   data$Weight = replace(data$Weight,data$Weight==0,aggregate(data$Weight,
by=list(ID=data$ID), FUN=median)[2][i,])
   data$Height = replace(data$Height,data$Height==0,aggregate(data$Height, by=list(ID=data$ID),
FUN=median)[2][i,])
   data$Body.mass.index =
replace(data$Body.mass.index,data$Body.mass.index==0,aggregate(data$Body.mass.index,
by=list(ID=data$ID), FUN=median)[2][i,])

}

#write.xlsx(data,'NoMissingorZeroes_2.xls',sheetName = 'Sheet1',row.names = T,col.names = T)

# Outlier analysis

library(ggplot2)
numerics = sapply(data, is.numeric)
numeric_cols = data[,numerics]
cnames = colnames(numeric_cols)

for (i in 1:length(cnames))
  {
   assign(paste0("oa",i), ggplot(aes_string(y = (cnames[i]), x = "Absenteeism.time.in.hours"),
   data = subset(data))+
   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],x="Absenteeism.time.in.hours")+
   ggtitle(paste("Box plot",i)))
}

# Plotting box plots to show outlier values

gridExtra::grid.arrange(oa1,oa2,ncol=2)
gridExtra::grid.arrange(oa3,oa4,ncol=2)
gridExtra::grid.arrange(oa5,oa6,ncol=2)
gridExtra::grid.arrange(oa7,oa8,ncol=2)
gridExtra::grid.arrange(oa9,oa10,ncol=2)
gridExtra::grid.arrange(oa11,ncol=1)
```

```
#Remove outliers using boxplot method

df = data
data = df

#for(i in cnames){
  #print(i)
  #val = data[,i][data[,i] %in% boxplot.stats(data[,i])$out]
  #print(length(val))
  #data = data[which(!data[,i] %in% val),]
#}

#Replace all outliers with 0 and impute

for(i in cnames){
  #print(i)
  val = data[,i][data[,i] %in% boxplot.stats(data[,i])$out]
  print(length(val))
  data[,i][data[,i] %in% val] = 0
}

#write.xlsx(data,'Zeroes.xls',sheetName = 'Sheet1',row.names = T,col.names = T)

# Replacing 0s with median in continuous variables

data$Work.load.Average.day. =
replace(data$Work.load.Average.day.,data$Work.load.Average.day.==0,median(data$Work.load.Av
erage.day.))
data$Hit.target = replace(data$Hit.target,data$Hit.target==0,median(data$Hit.target))
data$Absenteeism.time.in.hours =
replace(data$Absenteeism.time.in.hours,data$Absenteeism.time.in.hours==0,median(data$Absent
eeism.time.in.hours))

# View(data)

unique(data$ID)
length(unique(data$ID))
for (i in range(1,length(unique(data$ID)))) {
  data$Transportation.expense =
replace(data$Transportation.expense,data$Transportation.expense==0,aggregate(data$Transporta
tion.expense, by=list(ID=data$ID), FUN=median)[2][i,])
  data$Distance.from.Residence.to.Work =
replace(data$Distance.from.Residence.to.Work,data$Distance.from.Residence.to.Work==0,aggreg
ate(data$Distance.from.Residence.to.Work, by=list(ID=data$ID), FUN=median)[2][i,])
  data$Service.time =
replace(data$Service.time,data$Service.time==0,aggregate(data$Service.time, by=list(ID=data$ID),
FUN=median)[2][i,])
  data$Age = replace(data$Age,data$Age==0,aggregate(data$Age, by=list(ID=data$ID),
FUN=median)[2][i,])
  data$Son = replace(data$Son,data$Son==0,aggregate(data$Son, by=list(ID=data$ID),
FUN=median)[2][i,])
  data$Pet = replace(data$Pet,data$Pet==0,aggregate(data$Pet, by=list(ID=data$ID),
FUN=median)[2][i,])
  data$Weight = replace(data$Weight,data$Weight==0,aggregate(data$Weight, by=list(ID=data$ID),
FUN=median)[2][i,])
  data$Height = replace(data$Height,data$Height==0,aggregate(data$Height, by=list(ID=data$ID),
FUN=median)[2][i,])
  data$Body.mass.index =
replace(data$Body.mass.index,data$Body.mass.index==0,aggregate(data$Body.mass.index,
by=list(ID=data$ID), FUN=median)[2][i,])

}

#write.xlsx(data,'Final1.xls',sheetName = 'Sheet1',row.names = T,col.names = T)
```

```
# Feature selection

# Correlation calculation and plot for numeric variables

value = round(cor(as.matrix(data[,numerics])),2)

value

library(corrgram)
corrgram(data[,numerics], order=FALSE, lower.panel=panel.shade,
        upper.panel=NULL, text.panel=panel.txt,
        main="Correlation plot")

# Anova test for categorical independent and continuous dependent variable

data$Month.of.absence = as.factor(data$Month.of.absence)
data$Reason.for.absence = as.factor(data$Reason.for.absence)
data$Day.of.the.week = as.factor(data$Day.of.the.week)
data$Seasons = as.factor(data$Seasons)
data$Disciplinary.failure = as.factor(data$Disciplinary.failure)
data$Education = as.factor(data$Education)
data$Social.drinker = as.factor(data$Social.drinker)
data$Social.smoker = as.factor(data$Social.smoker)

facts = sapply(data, is.factor)
facts_cols = data[,facts]
cnames_facts = colnames(facts_cols)

anova_result <-
aov(Absenteeism.time.in.hours~Reason.for.absence*Month.of.absence*Day.of.the.week
                *Seasons*Disciplinary.failure*Education*Social.drinker*Social.smoker , data = data)
summary(anova_result)

# Data after pre-processing

data_preprocessed = subset(data, select = -c(Month.of.absence, Day.of.the.week, Seasons,
Disciplinary.failure,
                                Education, Social.drinker,
Social.smoker,Distance.from.Residence.to.Work,
                                Son,Pet,Weight,Body.mass.index))

# Writing pre-processed data in the disc

#write.xlsx(data_preprocessed,'Preprocessed_data.xls',sheetName = 'Sheet1',row.names =
T,col.names = T)

# Checking distribution for continuous variables in pre-processed data

qqnorm(data_preprocessed$Transportation.expense)
hist(data_preprocessed$Transportation.expense)

qqnorm(data_preprocessed$Service.time)
hist(data_preprocessed$Service.time)

qqnorm(data_preprocessed$Age)
hist(data_preprocessed$Age)

qqnorm(data_preprocessed$Work.load.Average.day.)
hist(data_preprocessed$Work.load.Average.day.)

qqnorm(data_preprocessed$Hit.target)
hist(data_preprocessed$Hit.target)

qqnorm(data_preprocessed$Son)
hist(data_preprocessed$Son)
```

```
qqnorm(data_preprocessed$Height)
hist(data_preprocessed$Height)

qqnorm(data_preprocessed$Absenteeism.time.in.hours)
hist(data_preprocessed$Absenteeism.time.in.hours)

df = data_preprocessed
list_norm =
c("Transportation.expense","Service.time","Age","Hit.target","Absenteeism.time.in.hours",
          "Work.load.Average.day.","Height")
for (i in list_norm){
  df[,i] = (df[,i] - min(df[,i]))/((max(df[,i])) - (min(df[,i])))
}

# Check distribution post normalization

hist(df$Transportation.expense)
hist(df$Service.time)
hist(df$Age)
hist(df$Work.load.Average.day.)
hist(df$Hit.target)
hist(df$Height)
hist(df$Absenteeism.time.in.hours)


# Decision Tree regression model design

library(rpart)
#df = data_preprocessed
#df = data

#df = df[,-1]

# Dividing data into train and test

train_index_norm = sample(1:nrow(df), 0.95*nrow(df))
train_norm = df[train_index_norm,]
test_norm = df[-train_index_norm,]

fit = rpart(Absenteeism.time.in.hours ~. , data = train_norm, method = "anova")
predictions_DT = predict(fit,test_norm[,-9])

# Calculating model accuracy

library(DMwR)
regr.eval(test_norm[,9], predictions_DT, stats = c('mae','rmse','mse','mape'))

mape = function(y,yhat){
  mean(abs((y-yhat)/y))*100
}

mape(test[,9],predictions_DT)


# Linear Regression model design

library(usdm)
df2 = df
df2 = df2[-1]
df2 = df2[-1]
vif(df2[,-7])
vifcor(df2[,-7], th=0.9)

train_index_LM = sample(1:nrow(df2), 0.95*nrow(df2))
train_LM = df2[train_index_LM,]
test_LM = df2[-train_index_LM,]
```

```
lm_model = lm(Absenteeism.time.in.hours~., data = train_LM)

summary(lm_model)

predictions_LR=predict(lm_model,test_LM[,1:6])

regr.eval(test_LM[,7], predictions_LR, stats = c('mae','rmse','mse','mape'))


# Logic for calculating loss per month. We will calculate loss from the original dataset "data".

df1 = data
list1 = c("Work.load.Average.day.","Hit.target","Absenteeism.time.in.hours")
for (i in list1){
  df1[,i] = (df1[,i] - min(df1[,i]))/((max(df1[,i])) - (min(df1[,i])))
}
a = aggregate(df1$Work.load.Average.day., by=list(Month.of.absence=df1$Month.of.absence),
FUN=sum)
b = aggregate(df1$Hit.target, by=list(Month.of.absence=df1$Month.of.absence), FUN=sum)
c = aggregate(df1$Absenteeism.time.in.hours, by=list(Month.of.absence=df1$Month.of.absence),
FUN=sum)
d = aggregate(df1$ID, by=list(Month.of.absence=df1$Month.of.absence), FUN=unique)

# Assumption is that every month has 30 days

# We are comparing the workload (in hours) for every month (expected output) and comparing it
with the hit target.
# Also, we are subtracting the absenteeism (in hours) to get the actual output and calculate loss.

for (i in c(1,2,3,4,5,6,7,8,9,10,11,12)) {
  # Calculating loss
  #print(i)
  loss = (b[2][i,]-(b[2][i,]/(a[2][i,]*31*length(d[2][i,][[1]])))*((a[2][i,]*30*length(d[2][i,][[1]]))-
c[2][i,]))/b[2][i,]*100
  print(loss)
}

x =
c(3.279278,3.293305,3.301384,3.294327,3.321255,3.316309,3.356788,3.377586,3.301268,3.310392,3.2
89171,3.324995)
y = c("1","2","3","4","5","6","7","8","9","10","11","12")
barplot(x,names.arg=y,xlab="Months",ylab="Percentage loss",col="red",
        main="Loss chart",border="black")
```

5. References

- Algina, J., & Olejnik, S. (2003). Conducting power analyses for ANOVA and ANCOVA in between-subjects designs. *Evaluation & the Health Professions*
- https://www.r-bloggers.com/
- Wikipedia - Mean absolute percentage error (MAPE)
- http://www.tutorialspoint.com/
- https://study.com/academy/lesson/scatter-plot-and-correlation-definition-example-analysis.html
- https://www.datasciencecentral.com/profiles/blogs/top-20-python-libraries-for-data-science-in-2018
- https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052
- https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86
- https://www.marsja.se/three-ways-to-carry-out-2-way-anova-with-python/