R Notebook

Code

#——————————**Case_Study_Cars_Preference**——————————————————

#Car_Case_Study #This project requires you to understand what mode of transport employees prefers to #commute to their office. #The dataset “Cars-dataset” includes employee information about their mode of transport #as well as their personal and professional #details like age, salary, work exp. We need to predict whether or not an employee #will use Car as a mode of transport. #Also, which variables are a significant predictor behind this decision.

#Following is expected out of the candidate in this assessment.

#EDA (15 Marks)

#Perform an EDA on the data - (7 marks) #Illustrate the insights based on EDA (5 marks) #What is the most challenging aspect of this problem? #What method will you use to deal with this? Comment (3 marks) #Data Preparation (10 marks)

#Modeling (30 Marks) #Create multiple models and explore how each model perform using appropriate model performance metrics (15 marks) #KNN #Naive Bayes (is it applicable here? comment and if it is not applicable, how can you build an NB model in this case?) #Logistic Regression #Apply both bagging and boosting modeling procedures to create 2 models and compare its accuracy with the #best model of the above step. (15 marks) #Actionable Insights & Recommendations (5 Marks)

#Summarize your findings from the exercise in a concise yet actionable note #Library usage: library(lattice) library(readxl) library(rpart) library(caret) library(e1071) library(dplyr) library(DMwR) library(ggplot2) library(corrplot) library(caTools) library(class) library(usdm) library(naivebayes) library(gbm) library(ggplot2) library(rlang) library(caret) library(gbm)

#This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

#Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Hide

```
car_data <-read.csv("Cars-dataset.csv", header = TRUE)
```

Hide

```
str(car_data)
```

```
'data.frame':	417 obs. of  9 variables:
 $ Age      : int  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : chr  "Male" "Male" "Female" "Male" ...
 $ Engineer : int  1 1 1 0 0 0 1 0 1 1 ...
 $ MBA      : int  0 0 0 0 0 0 1 0 0 0 ...
 $ Work.Exp : int  5 6 9 1 3 3 3 0 4 6 ...
 $ Salary   : num  14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...
 $ Distance : num  5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...
 $ license  : int  0 0 0 0 0 0 0 0 0 1 ...
 $ Transport: chr  "2Wheeler" "2Wheeler" "2Wheeler" "2Wheeler" ...
 - attr(*, "na.action")= 'omit' Named int 243
  ..- attr(*, "names")= chr "243"
```

Hide

```
View(car_data)
summary(car_data)
```

```
      Age           Gender             Engineer          MBA        
 Min.   :18.00   Length:417         Min.   :0.0000   Min.   :0.0000  
 1st Qu.:25.00   Class :character   1st Qu.:1.0000   1st Qu.:0.0000  
 Median :27.00   Mode  :character   Median :1.0000   Median :0.0000  
 Mean   :27.33                      Mean   :0.7506   Mean   :0.2614  
 3rd Qu.:29.00                      3rd Qu.:1.0000   3rd Qu.:1.0000  
 Max.   :43.00                      Max.   :1.0000   Max.   :1.0000  
    Work.Exp          Salary         Distance       license         Transport        
 Min.   : 0.000   Min.   : 6.50   Min.   : 3.2   Min.   :0.0000   Length:417        
 1st Qu.: 3.000   1st Qu.: 9.60   1st Qu.: 8.6   1st Qu.:0.0000   Class :character  
 Median : 5.000   Median :13.00   Median :10.9   Median :0.0000   Mode  :character  
 Mean   : 5.873   Mean   :15.42   Mean   :11.3   Mean   :0.2038                     
 3rd Qu.: 8.000   3rd Qu.:14.90   3rd Qu.:13.6   3rd Qu.:0.0000                     
 Max.   :24.000   Max.   :57.00   Max.   :23.4   Max.   :1.0000                     
```

Hide

```
#devtools::install_github('r-lib/later#96')
#pkgbuild::with_build_tools(install.packages("r-lib", repos = NULL, type = "source"))
# We notice that MBA has 1 NA value, just to be sure:
sum(is.na(car_data))
```

```
[1] 0
```

Hide

```
# Lets remove it right away:
car_data<-na.omit(car_data)
car_data<-knnImputation(car_data)
```

```
Error in colMeans(x, na.rm = TRUE) : 'x' deve ser numérico
```

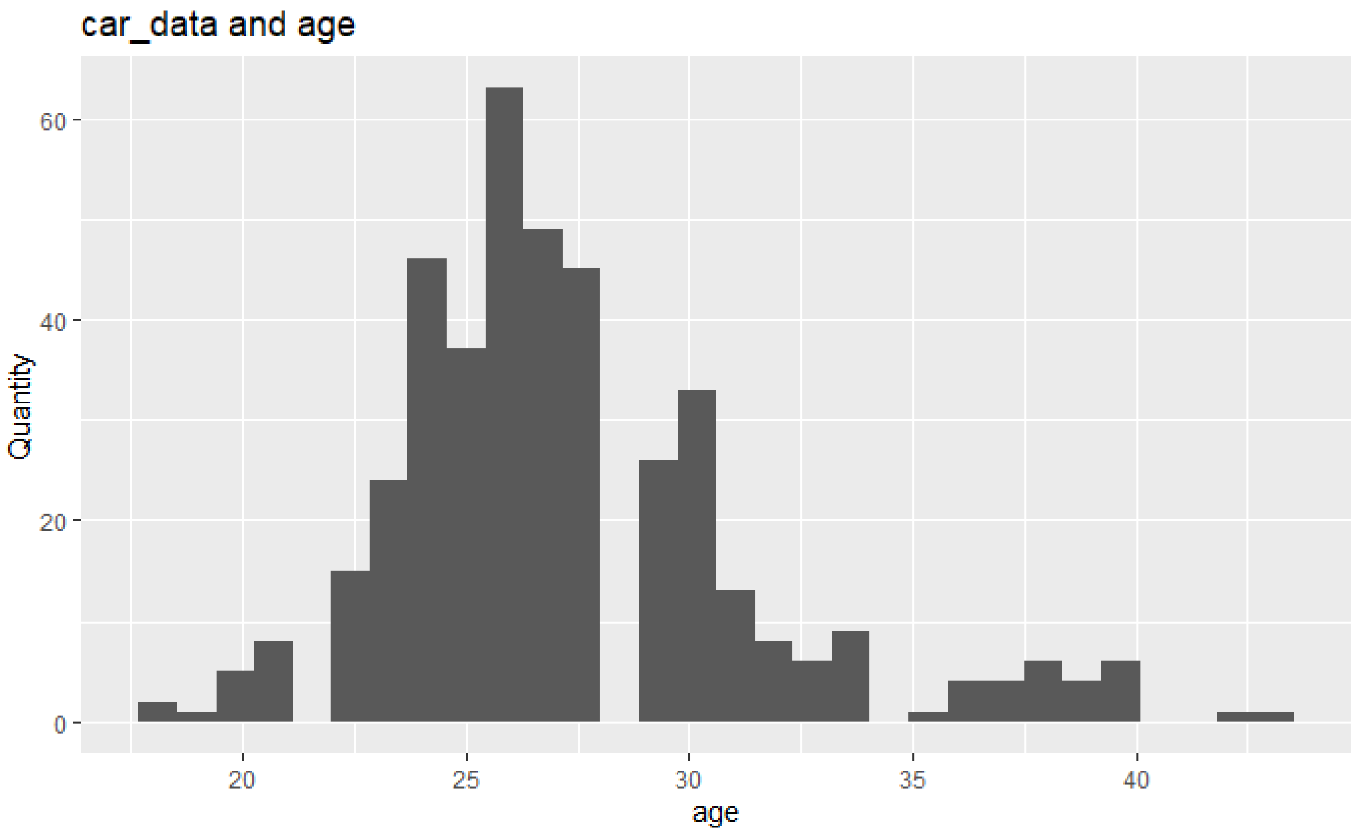
Hide

```
# Checking variables we need to turn into factor the following variables listed below:

#engineer
car_data$Engineer <- as.factor(car_data$Engineer)

#MBA
car_data$MBA <- as.factor(car_data$MBA)

#license - Drivers License is a pre requisite in order to drive a car
car_data$license <- as.factor(car_data$license)

# Turn into binary gender column
car_data$Gender = factor(car_data$Gender,
                         levels = c("Male", "Female"),
                         labels = c(0,1))

View(car_data)
```

Now we have the car users percentage in this scenario

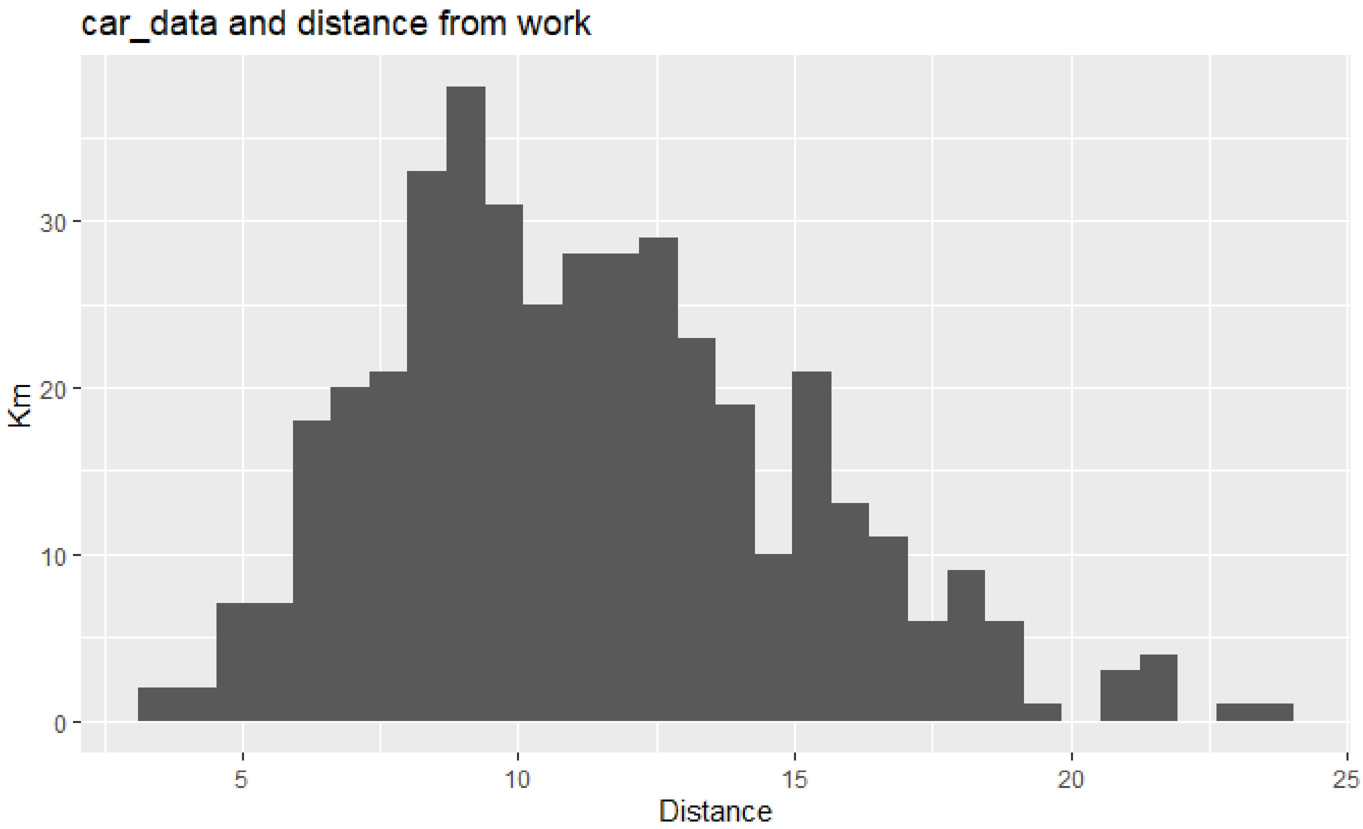
Hide

```
#------------------------Univariate Analysis----------------------------------

qplot(Age, data = car_data,
      main = "car_data and age",
      xlab = "age",
      ylab = "Quantity")
```



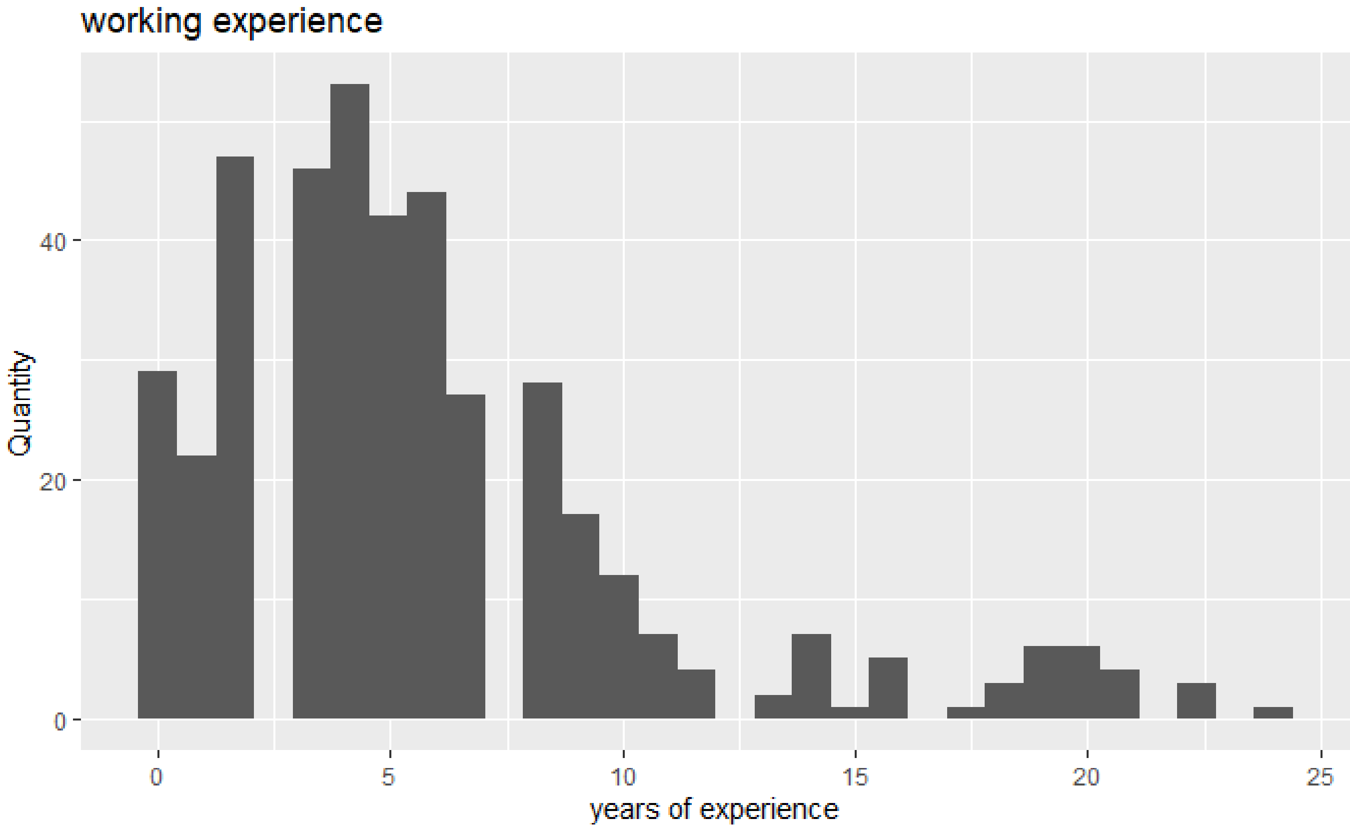
Hide

```
#Age is right skewed tailing at the end. We notice there are, probably as most companies, mor
e juniors than seniors

qplot( Distance, data = car_data,
      main = "car_data and distance from work",
      xlab = "Distance",
      ylab = "Km")
```



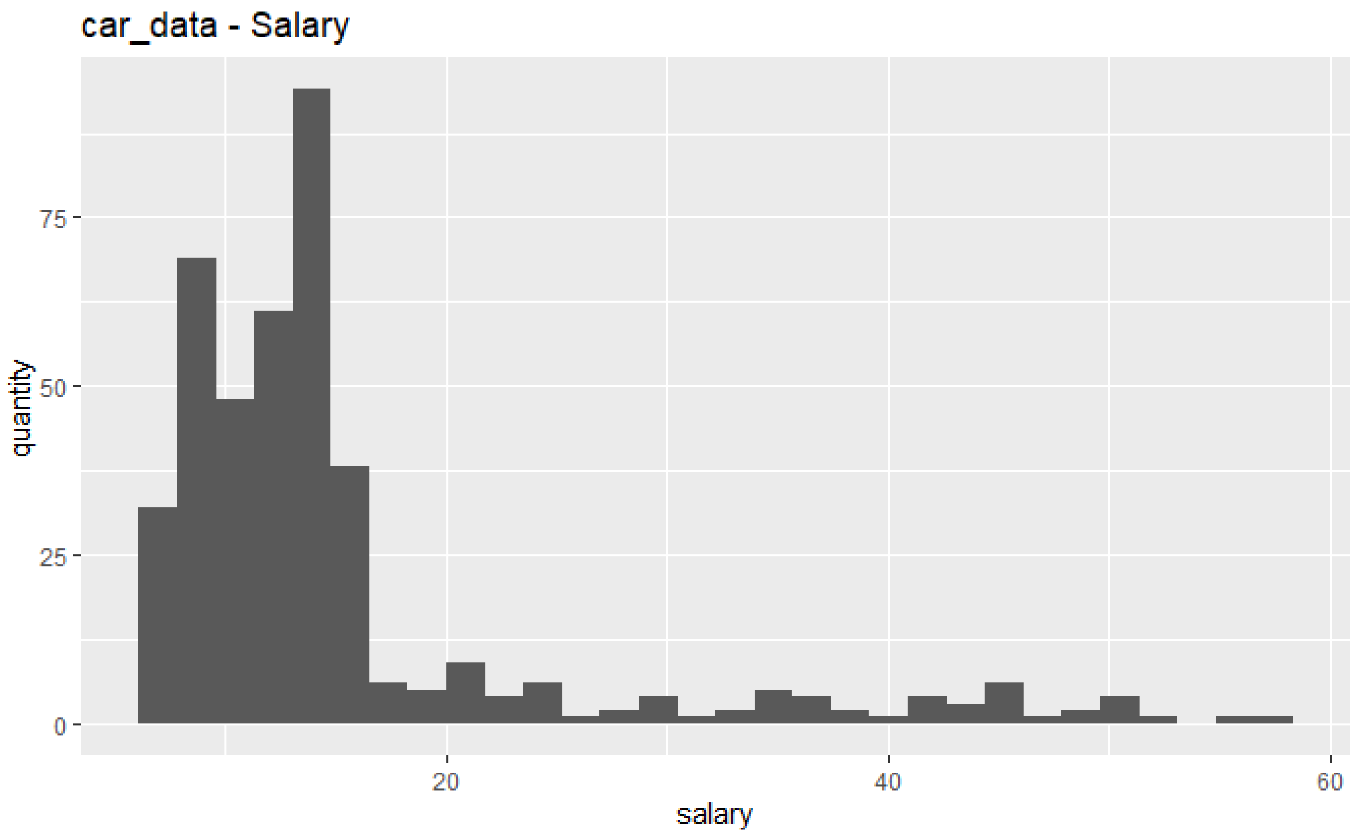
Hide

```
#Left skewed Most people lives in the 8-14 (u.m.) away from work, would likely pay attention
 to car option correlation

qplot(Work.Exp, data = car_data,
      main = "working experience",
      xlab = "years of experience",
      ylab = "Quantity")
```



Hide

```
#Left skewed, what confirms the junior hypothesis

qplot(Salary, data = car_data,
      main = "car_data - Salary",
      xlab = "salary",
      ylab = "quantity")
```

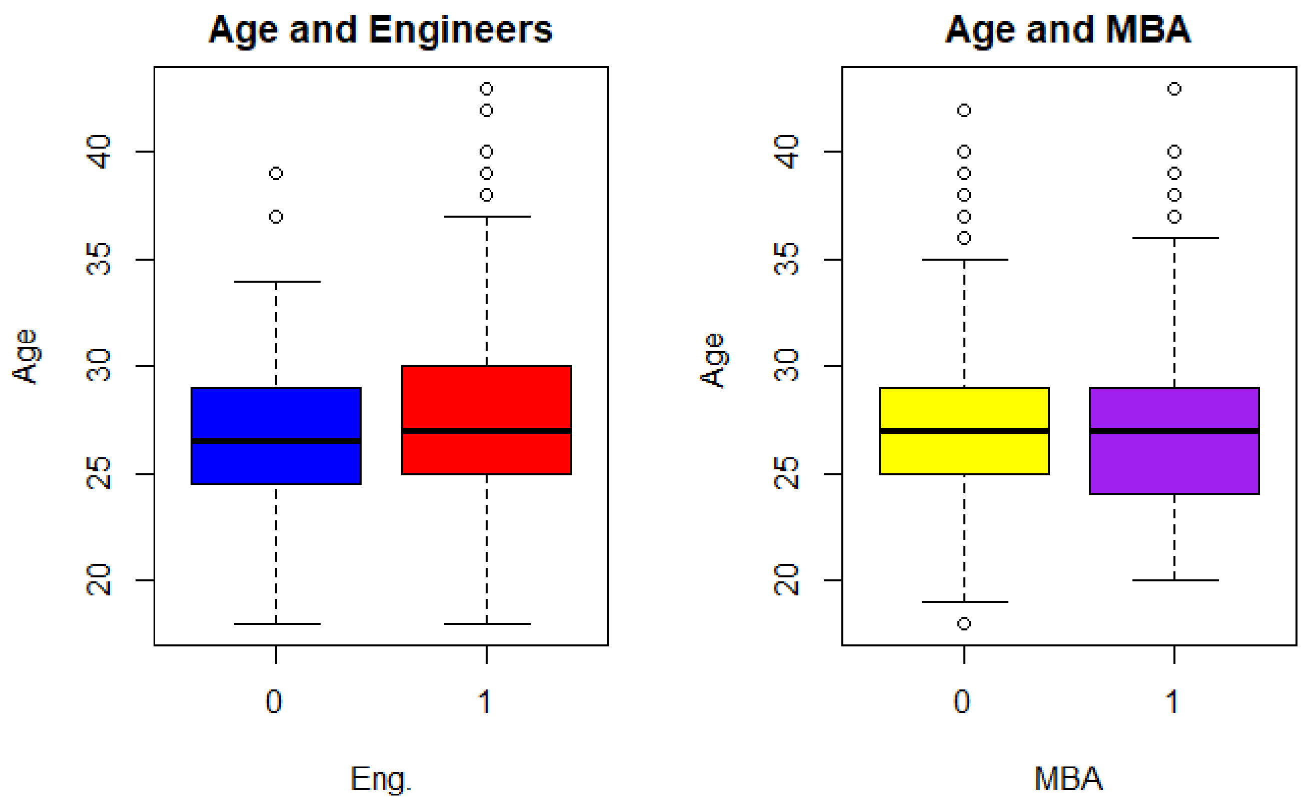


Hide

```
# salary is unbalanced probably concentrated amongst the senior and persons with more years o
f experience
```

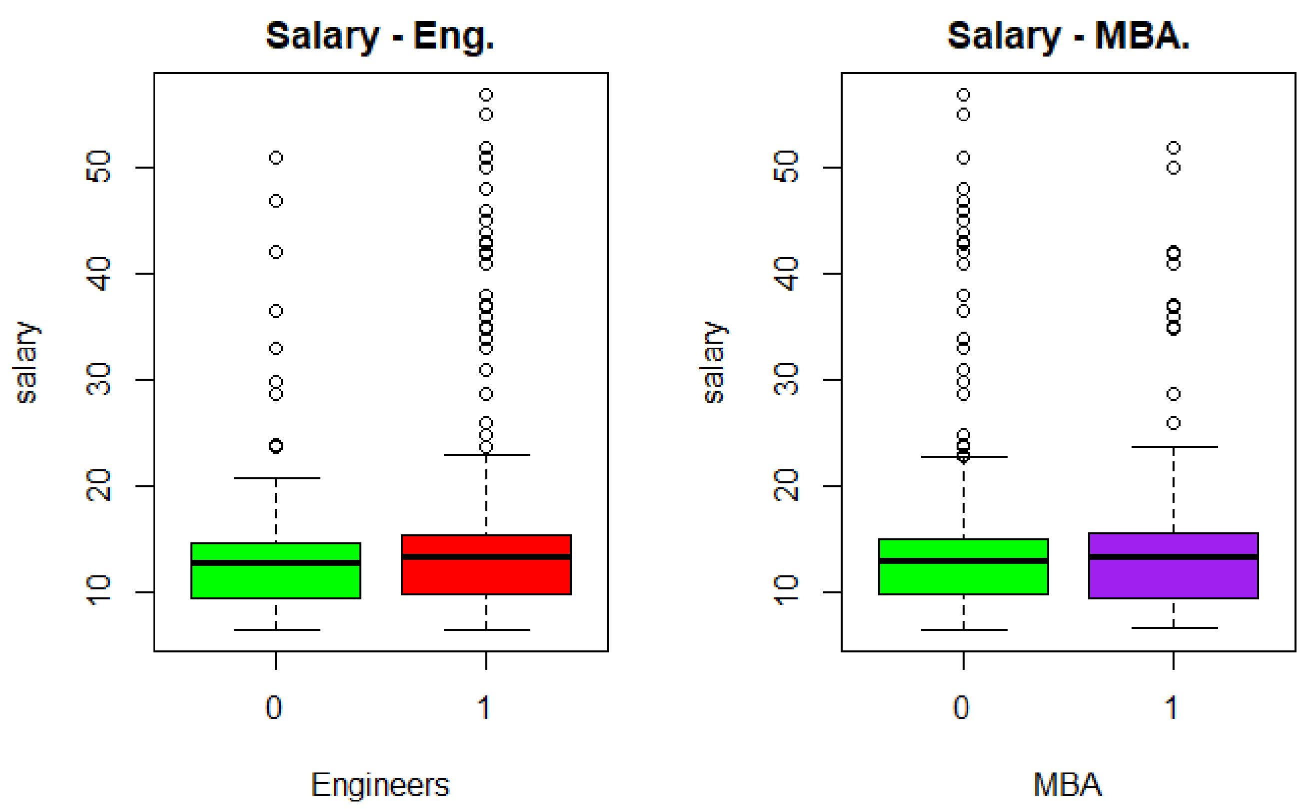
Hide

```
#----------------------------BiVariate Analysis------------------------------------

par(mfrow= c(1,2))
boxplot(car_data$Age~car_data$Engineer, vertical = TRUE,
        col = c("blue", "red"), main = "Age and Engineers",
        ylab = "Age",
        xlab = "Eng.")

boxplot(car_data$Age~car_data$MBA, vertical = TRUE,
        col = c("yellow", "purple"), main = "Age and MBA",
        ylab = "Age ",
        xlab = "MBA")
```



Hide

```
#As expected not much of difference here, people for all qualifications and all work exp woul
d be employed in firm.

boxplot(car_data$Salary~car_data$Engineer, vertical = TRUE,
      col = c("green", "red"), main = "Salary - Eng.",
      xlab = "Engineers",
      ylab = "salary")

boxplot(car_data$Salary~car_data$MBA, vertical = TRUE,
        col = c("green", "purple"), main = "Salary - MBA.",
        xlab = "MBA",
        ylab = "salary")
```



Hide

```
mean(car_data$Salary)
```

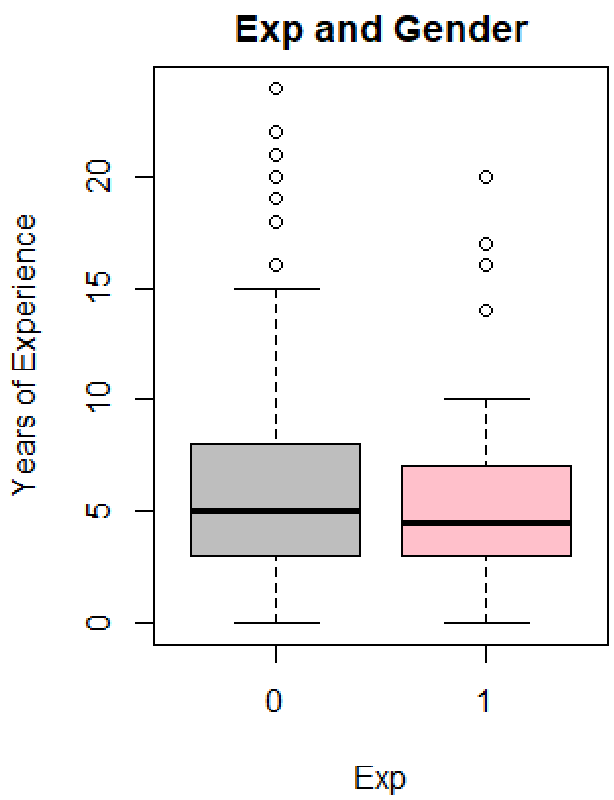
```
[1] 15.42254
```

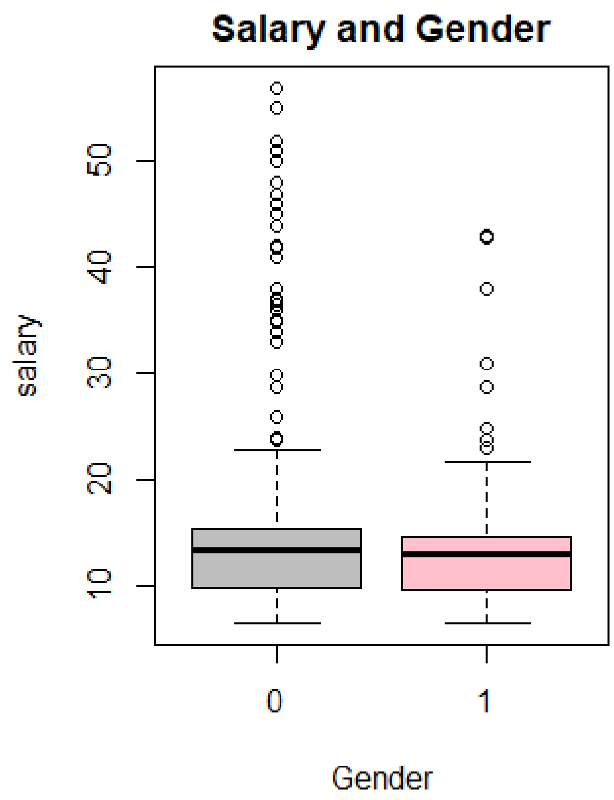
Hide

```
#We do not see any appreciable difference in salary of Eng Vs Non-Eng or MBA vs Non-MBA's
#Also, mean salary for both MBA's and Eng is around 16.

boxplot(car_data$Work.Exp~car_data$Gender, vertical = TRUE,
        col = c("grey","pink"),
        main="Exp and Gender",
        xlab = "Exp",
        ylab = "Years of Experience")

boxplot(car_data$Salary~car_data$Gender, vertical = TRUE,
        col = c("grey","pink"),
        main="Salary and Gender",
        xlab = "Gender",
        ylab = "salary")
```

Hide

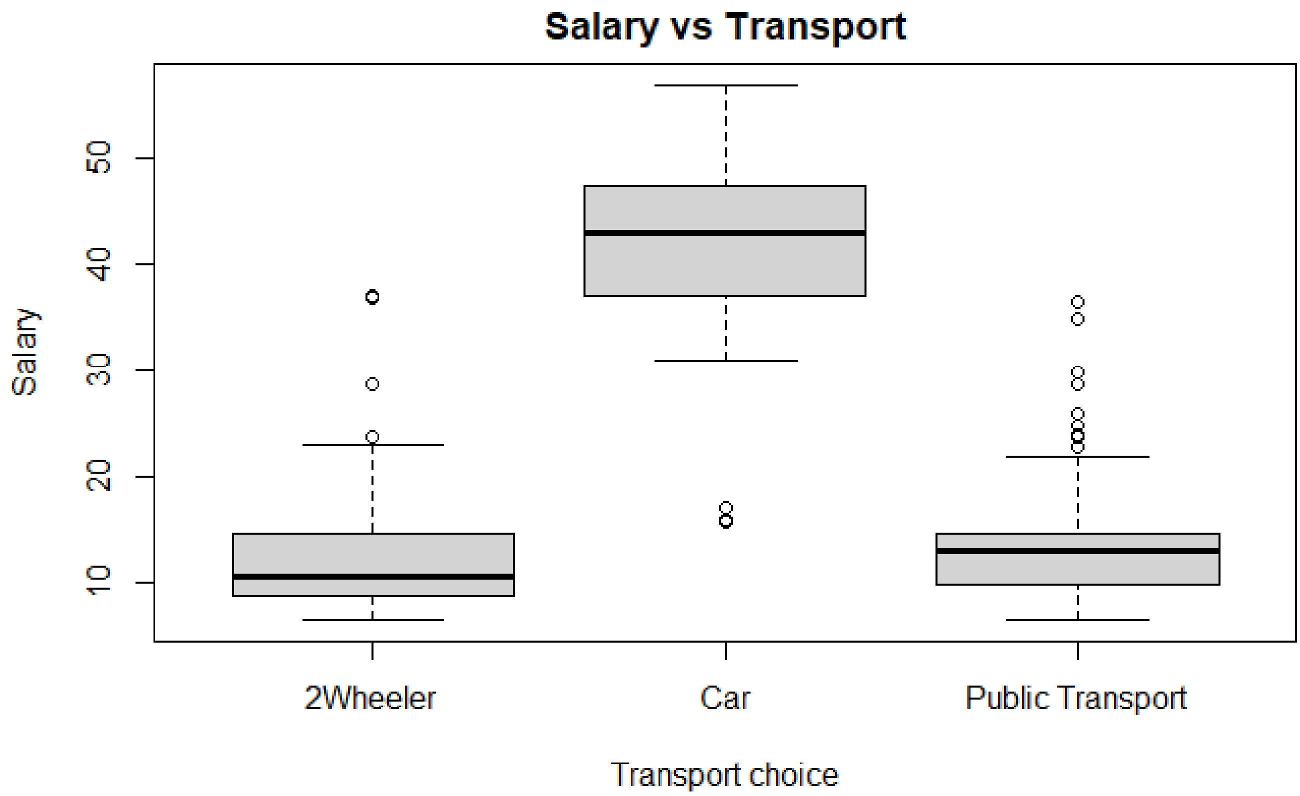
```
#Not much of difference between mean work experience in two genders.
#However the highest salaries (ouliers) are clearly concentrated amongst male workers

par(mfrow=c(1,1))
```





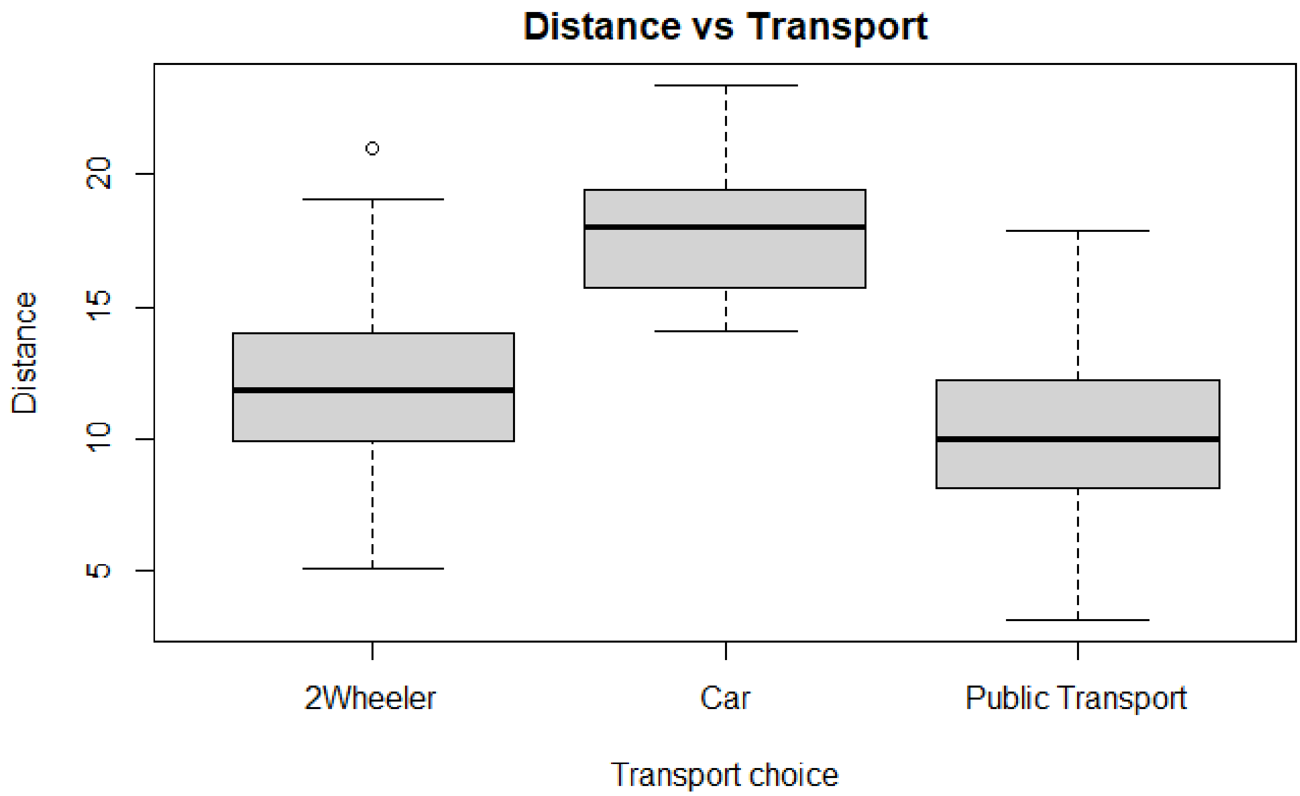
Hide

```
boxplot(car_data$Salary~car_data$Transport, vertical = TRUE, main= "Salary vs Transport",
        xlab = "Transport choice",
        ylab = "Salary")
```



Hide

```
# higher the salary, greater the probability of going by car


#Predilection for public transport and motorcycles by the younger workers
#Age and Car must be related

boxplot(car_data$Distance~car_data$Transport, vertical = TRUE, main= "Distance vs Transport",
        xlab = "Transport choice",
        ylab = "Distance")
```



Hide

```
# Public Transport is commonly chosen with lesser distances, by thr other hand, with greater
 distances, car is chosen
table(car_data$Gender, car_data$Transport)
```

```

    2Wheeler Car Public Transport
  0       45  29              223
  1       38   6               76
```

Hide

```
# 0 as female
# 1 as male

# We can see that around 25 % of females use private transport and 37% of males uses private
 transport a sensible difference here
#Thus, even though percentage of car usage is high females also  shows high % on public trans
port.
# females showed low interest in motorcycles

#Correlation Plot:
#collumn "Transport" Must be numeric,

car_data$Transport = factor(car_data$Transport,
                         levels = c("2Wheeler","Car", "Public Transport"),
                         labels = c(0,1,0))

levels(car_data$Transport)
```

```
[1] "0" "1"
```

Hide

```
View(car_data)

str(car_data)
```

```
'data.frame':	417 obs. of  9 variables:
 $ Age      : int  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : Factor w/ 2 levels "0","1": 1 1 2 1 2 1 1 1 1 1 ...
 $ Engineer : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 2 1 2 2 ...
 $ MBA      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...
 $ Work.Exp : int  5 6 9 1 3 3 3 0 4 6 ...
 $ Salary   : num  14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...
 $ Distance : num  5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...
 $ license  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
 $ Transport: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, "na.action")= 'omit' Named int 243
  ..- attr(*, "names")= chr "243"
```

Hide

```
summary(car_data)
```

```
      Age        Gender  Engineer MBA       Work.Exp         Salary     
 Min.   :18.00   0:297   0:104    0:308   Min.   : 0.000   Min.   : 6.50  
 1st Qu.:25.00   1:120   1:313    1:109   1st Qu.: 3.000   1st Qu.: 9.60  
 Median :27.00                            Median : 5.000   Median :13.00  
 Mean   :27.33                            Mean   : 5.873   Mean   :15.42  
 3rd Qu.:29.00                            3rd Qu.: 8.000   3rd Qu.:14.90  
 Max.   :43.00                            Max.   :24.000   Max.   :57.00  
    Distance     license Transport
 Min.   : 3.2   0:332   0:382    
 1st Qu.: 8.6   1: 85   1: 35    
 Median :10.9                    
 Mean   :11.3                    
 3rd Qu.:13.6                    
 Max.   :23.4                    
```

Hide

```
#---------------------------------------------------
# List numeric features in this dataset
#nums = unlist(lapply(car_data, is.numeric))
#nums = lapply(cleandata, is.numeric)
#print(nums)

# Age       Gender      Engineer        MBA
#TRUE       FALSE     FALSE            FALSE

# We have to treat this variables into numeric in order to run the correlaction plot matrix

#---------------------------------------------------

car_data$Age<-as.numeric(car_data$Age)
car_data$Gender<-as.numeric(car_data$Gender)
car_data$Engineer<-as.numeric(car_data$Engineer)
car_data$MBA<-as.numeric(car_data$MBA)
car_data$license<-as.numeric(car_data$license)

str(car_data)
```

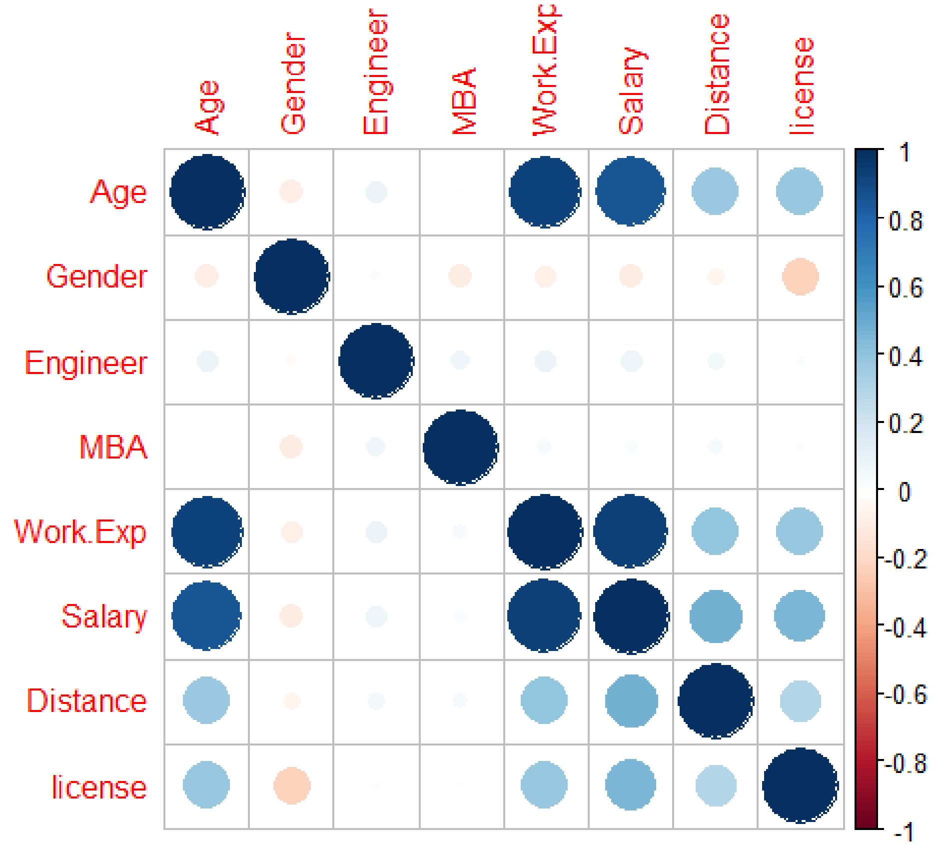
```
'data.frame':   417 obs. of  9 variables:
 $ Age      : num  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : num  1 1 2 1 2 1 1 1 1 1 ...
 $ Engineer : num  2 2 2 1 1 1 2 1 2 2 ...
 $ MBA      : num  1 1 1 1 1 1 2 1 1 1 ...
 $ Work.Exp : int  5 6 9 1 3 3 3 0 4 6 ...
 $ Salary   : num  14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...
 $ Distance : num  5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...
 $ license  : num  1 1 1 1 1 1 1 1 1 2 ...
 $ Transport: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, "na.action")= 'omit' Named int 243
  ..- attr(*, "names")= chr "243"
```

Hide

```
View(car_data)
```

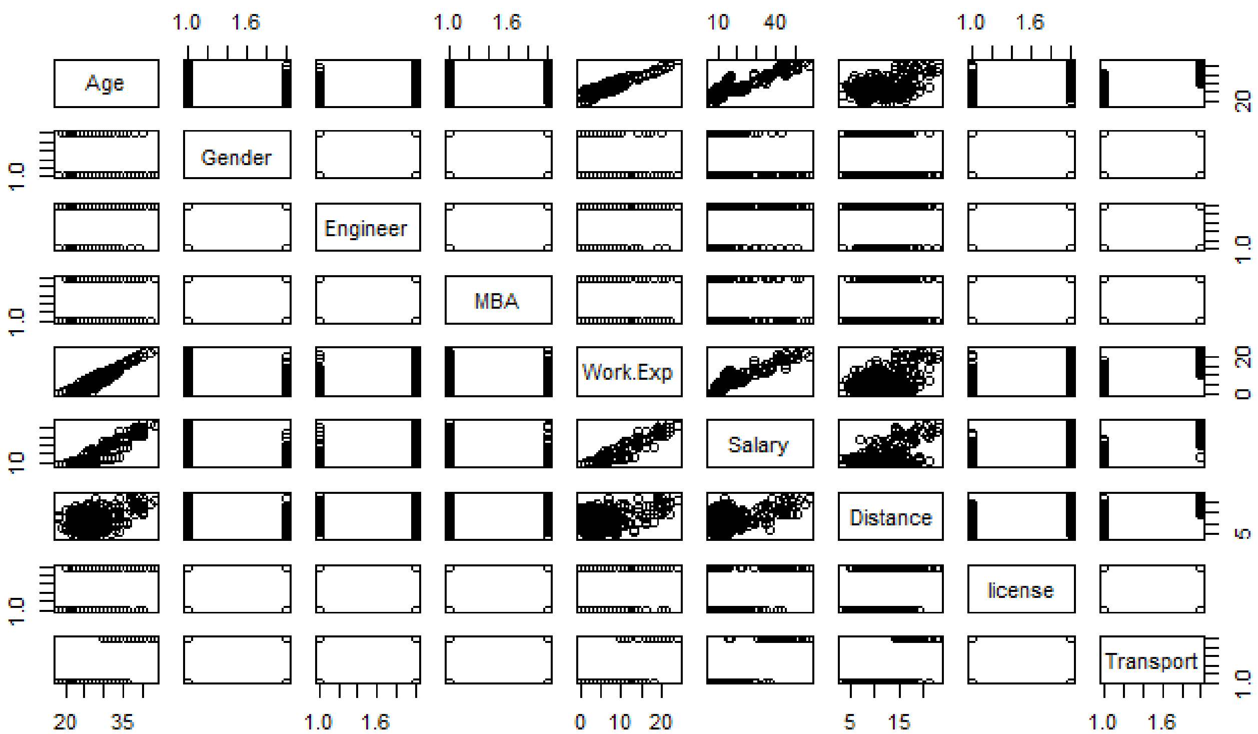
Hide

```
#---------------------------------------------------
#Correlation Plot

corrplot(cor(car_data[-9]))
```



Hide

```
#----------Multicollinearity------------------------

#We will treat outliers and We are using vifcor function to remove highly correlated variable
s from the dataset.

plot(car_data)
```



Hide

```
vifcor(car_data[-9])
```

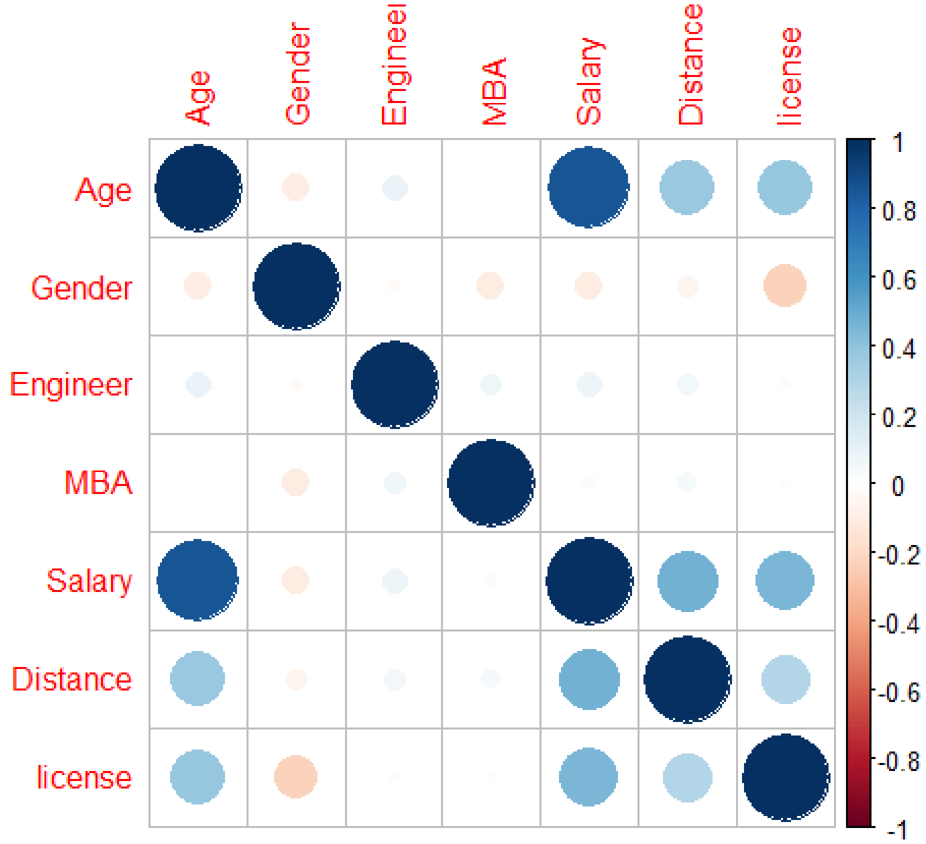
```
1 variables from the 8 input variables have collinearity problem:

Work.Exp

After excluding the collinear variables, the linear correlation coefficients ranges between:
min correlation ( MBA ~ Age ):  -0.001752158
max correlation ( Salary ~ Age ):  0.8579114

---------- VIFs of the remained variables --------
```

Variables <chr>	VIF <dbl>
Age	3.827422
Gender	1.067936
Engineer	1.012862
MBA	1.019179
Salary	4.482439
Distance	1.320710
license	1.339501

7 rows

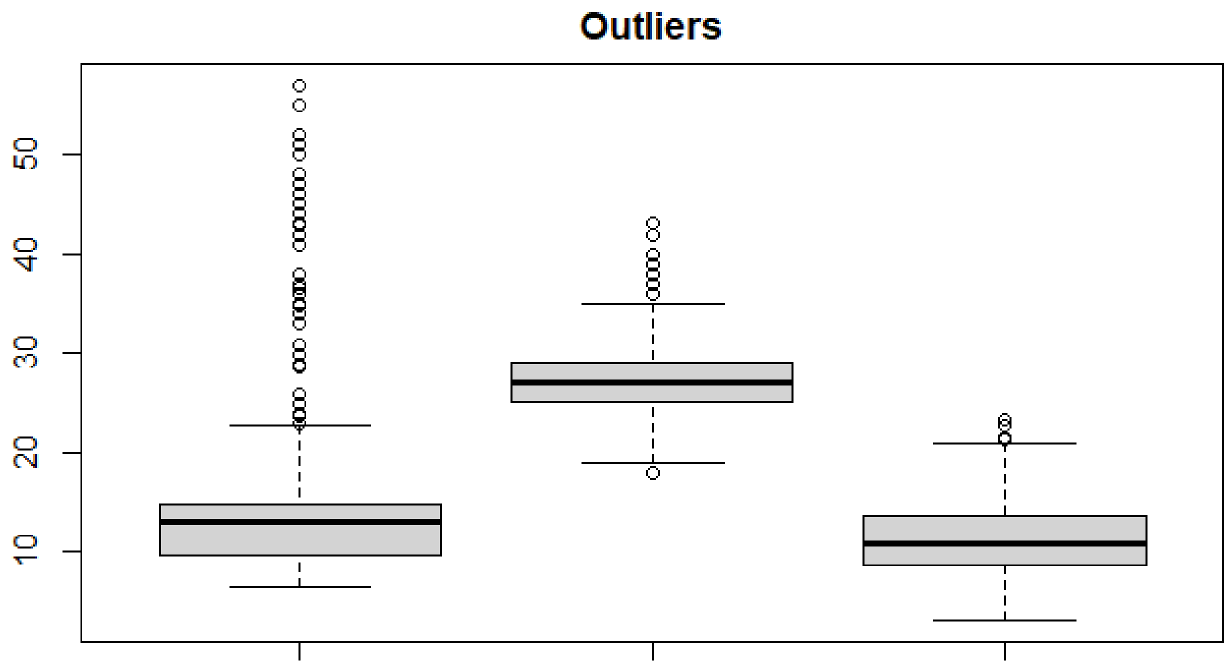
Hide

```
#Exclude Work_Exp column from the dataset for Multicollinearity treatment.

car_data <- car_data[-5]

View(car_data)

corrplot(cor(car_data[-8]))
```



Hide

```
#1. lets search for outliers:
#We will be checking outliers in Age, Salary and Distance as the rest variables are Binary in nature.

boxplot(car_data$Salary, car_data$Age, car_data$Distance, main = "Outliers")
```



Hide

```
#-------------Treating Outliers-------------------------

# All three variables has outliers, lets work with some percentage (95%) of the dataset in or
der to exclude those outliers

#Age
quantile(car_data$Age, c(0.95))
```

```
95%
 37
```

Hide

```
car_data$Age[which(car_data$Age>37)] <-37

#Salary
quantile(car_data$Salary, c(0.95))
```

```
  95%
41.92
```

Hide

```
car_data$Salary[which(car_data$Salary>41.92)] <-41.92

#Distance
quantile(car_data$Distance, c(0.95))
```

```
  95%
17.92
```

Hide

```
car_data$Distance[which(car_data$Distance>17.92)]<-17.92

#We will be generating Synthetic data using SMOTE.First we need change the target variable (T
ransport) into factor variable.

car_data$Transport <- as.factor(car_data$Transport)

View(car_data)

str(car_data$Transport)
```

```
 Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

Hide

```
#----------------------SMOTE----------------------

# car_data2 will be our balanced dataset

set.seed(42)
car_data_smote = SMOTE(Transport ~., car_data)
summary(car_data_smote)
```

```
      Age            Gender          Engineer          MBA            Salary     
 Min.   :20.00   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   : 6.80  
 1st Qu.:26.00   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:12.70  
 Median :30.00   Median :1.000   Median :2.000   Median :1.000   Median :15.80  
 Mean   :30.68   Mean   :1.240   Mean   :1.815   Mean   :1.264   Mean   :24.17  
 3rd Qu.:37.00   3rd Qu.:1.197   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:41.82  
 Max.   :37.00   Max.   :2.000   Max.   :2.000   Max.   :2.000   Max.   :41.92  
    Distance        license        Transport
 Min.   : 3.20   Min.   :1.000   0:140    
 1st Qu.: 9.40   1st Qu.:1.000   1:105    
 Median :13.70   Median :1.000            
 Mean   :13.03   Mean   :1.425            
 3rd Qu.:17.21   3rd Qu.:2.000            
 Max.   :17.92   Max.   :2.000            
```

Hide

```
# As we can see, now we more balance in the dataset increasing the object of prediction of 35
to 105 cars option
```

Hide

```
#---------------------Naive Bayes--------------------
#Naive Bayes:
#Naive Bayes classifier presumes that the presence of the feature in a class is unrelated to
#the presence of any other feature in the same class, so let's build the model and see how
#good our model is as per this classification model.
set.seed(231)
split = sample.split(car_data_smote$Transport, SplitRatio = 0.70)
training_set = subset(car_data_smote, split == TRUE)
test_set = subset(car_data_smote, split == FALSE)


NBModel <- naive_bayes(Transport~., data = training_set)
NB_Predict <- predict(NBModel, test_set)
```

```
predict.naive_bayes(): more features in the newdata are provided as there are probability tab
les in the object. Calculation is performed based on features to be found in the tables.
```

Hide

```
#table(NB_Predict, test_set$Transport)

# Making the Confusion Matrix
NB_car_cm <- table(NB_Predict,test_set[, 8])
print(NB_car_cm)
```

```
NB_Predict  0  1
         0 42  4
         1  0 27
```

Hide

```
# calculating the accuracy - We are defining the accuracy function
#to show the NB performance output
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x))))*100}
accuracy(NB_car_cm)
```

```
[1] 94.52055
```

Hide

```
#From confusion matrix we can clearly see that test data has 94,52% accurate in
#predicting.

#-----------------------------------------------------
#Modelling - KNN - K Nearest Neighbours
# Fitting K-NN to the Training set and Predicting the Test set results
Knn_car_pred = knn(training_set[ , -8],
               test=test_set[ , -8], cl=training_set[ , 8],
               k=3)

print(Knn_car_pred)
```

```
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[42] 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Levels: 0 1
```

Hide

```
# Making the Confusion Matrix
Knn_car_cm <- table(Knn_car_pred,test_set[, 8])
print(Knn_car_cm)
```

```
Knn_car_pred  0  1
           0 42  3
           1  0 28
```

Hide

```
# calculating the accuracy - We are defining the accuracy function to show the KNN performanc
e output
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x))))*100}
accuracy(Knn_car_cm)
```

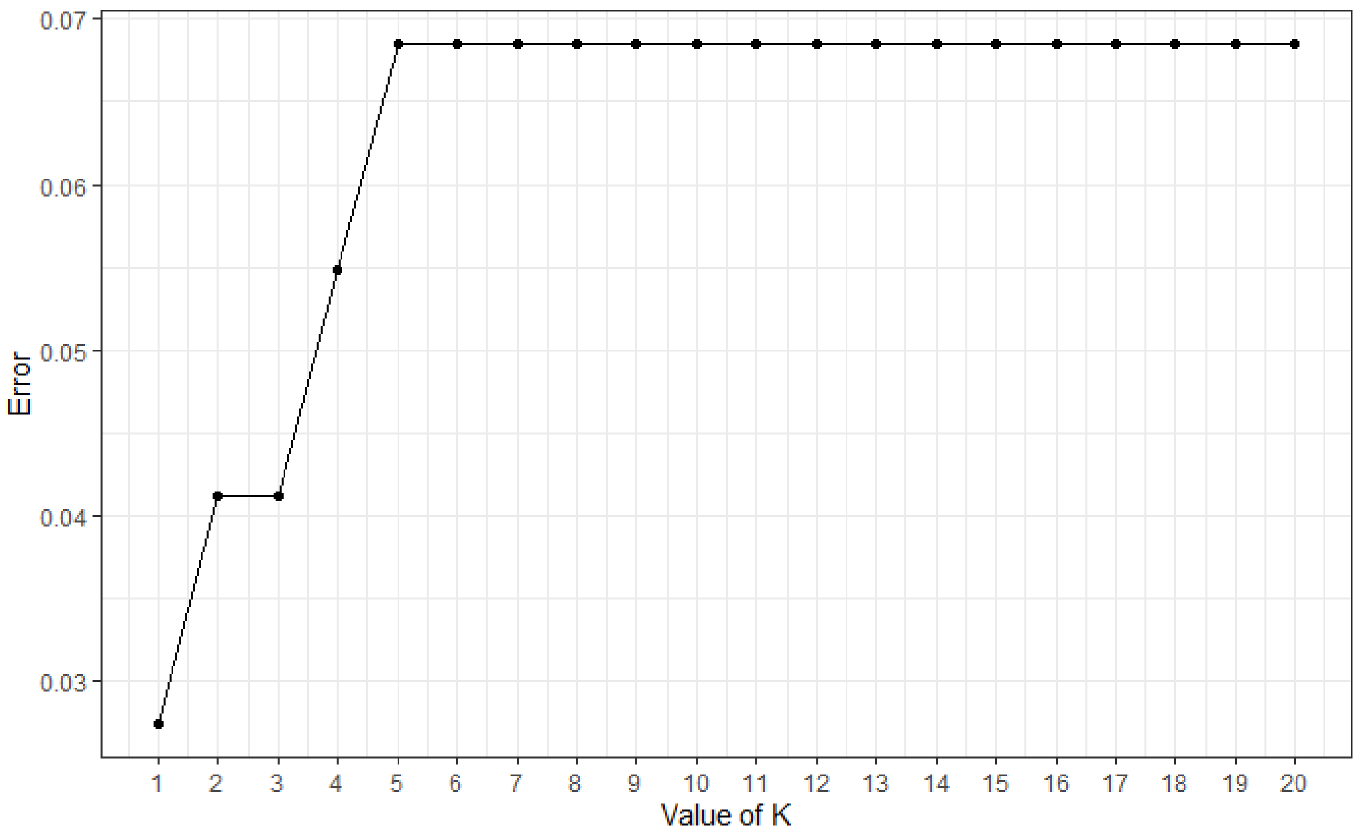
```
[1] 95.89041
```

Hide

```
predicted.type <- NULL
error.rate <- NULL
for (i in 1:20) {
  predicted.type <- knn(training_set[ , -9],test_set[ , -9], training_set$Transport,k=i)
  error.rate[i] <- mean(predicted.type!=test_set$Transport)}

knn.error <- as.data.frame(cbind(k=1:20,error.type =error.rate))
```

Hide

```
# Visualising the Training set results

ggplot(knn.error,aes(k,error.type))+
  geom_point()+
  geom_line() +
  scale_x_continuous(breaks=1:20)+
  theme_bw() +
  xlab("Value of K") +
  ylab('Error')
```



Hide

```
#KNN proved to be a real good model for this dataset
```

Hide

```
#---------------------------------------------------
#Baggin and BOOST!!!
set.seed(231)
split = sample.split(car_data$Transport, SplitRatio = 0.70)
training_set = subset(car_data, split == TRUE)
test_set = subset(car_data, split == FALSE)

m_boost = gbm(Transport~., data= training_set, verbose=TRUE, distribution='gaussian',n.trees=
5000,cv=10,interaction.depth=4,shrinkage = 0.01)
```

```
Iter   TrainDeviance   ValidDeviance   StepSize   Improve
     1        0.0747             nan     0.0100    0.0011
     2        0.0737             nan     0.0100    0.0009
     3        0.0726             nan     0.0100    0.0010
     4        0.0716             nan     0.0100    0.0007
     5        0.0704             nan     0.0100    0.0009
     6        0.0698             nan     0.0100    0.0009
     7        0.0689             nan     0.0100    0.0005
     8        0.0682             nan     0.0100    0.0009
     9        0.0675             nan     0.0100    0.0009
    10        0.0666             nan     0.0100    0.0008
    20        0.0585             nan     0.0100    0.0008
    40        0.0456             nan     0.0100    0.0006
    60        0.0362             nan     0.0100    0.0003
    80        0.0299             nan     0.0100    0.0003
   100        0.0247             nan     0.0100    0.0001
   120        0.0207             nan     0.0100    0.0001
   140        0.0178             nan     0.0100    0.0000
   160        0.0155             nan     0.0100    0.0000
   180        0.0135             nan     0.0100    0.0001
   200        0.0122             nan     0.0100   -0.0000
   220        0.0112             nan     0.0100   -0.0000
   240        0.0104             nan     0.0100   -0.0000
   260        0.0098             nan     0.0100   -0.0000
   280        0.0094             nan     0.0100   -0.0000
   300        0.0091             nan     0.0100   -0.0000
   320        0.0087             nan     0.0100    0.0000
   340        0.0084             nan     0.0100    0.0000
   360        0.0083             nan     0.0100    0.0000
   380        0.0080             nan     0.0100   -0.0000
   400        0.0077             nan     0.0100   -0.0000
   420        0.0075             nan     0.0100   -0.0000
   440        0.0073             nan     0.0100    0.0000
   460        0.0072             nan     0.0100   -0.0000
   480        0.0069             nan     0.0100    0.0000
   500        0.0067             nan     0.0100   -0.0000
   520        0.0066             nan     0.0100   -0.0000
   540        0.0064             nan     0.0100   -0.0000
   560        0.0062             nan     0.0100   -0.0000
   580        0.0060             nan     0.0100   -0.0000
   600        0.0059             nan     0.0100   -0.0000
   620        0.0058             nan     0.0100   -0.0000
   640        0.0057             nan     0.0100   -0.0000
   660        0.0055             nan     0.0100   -0.0000
   680        0.0054             nan     0.0100   -0.0000
   700        0.0053             nan     0.0100   -0.0000
   720        0.0052             nan     0.0100   -0.0000
   740        0.0051             nan     0.0100   -0.0000
   760        0.0050             nan     0.0100   -0.0000
   780        0.0049             nan     0.0100   -0.0000
   800        0.0049             nan     0.0100   -0.0000
   820        0.0048             nan     0.0100   -0.0000
   840        0.0047             nan     0.0100   -0.0000
   860        0.0046             nan     0.0100   -0.0000
   880        0.0046             nan     0.0100   -0.0000
   900        0.0045             nan     0.0100   -0.0000
   920        0.0044             nan     0.0100   -0.0000
```

```
     940        0.0043              nan     0.0100   -0.0000
     960        0.0043              nan     0.0100   -0.0000
     980        0.0042              nan     0.0100   -0.0000
    1000        0.0041              nan     0.0100   -0.0000
    1020        0.0041              nan     0.0100    0.0000
    1040        0.0040              nan     0.0100   -0.0000
    1060        0.0040              nan     0.0100   -0.0000
    1080        0.0039              nan     0.0100   -0.0000
    1100        0.0039              nan     0.0100   -0.0000
    1120        0.0038              nan     0.0100   -0.0000
    1140        0.0038              nan     0.0100   -0.0000
    1160        0.0037              nan     0.0100   -0.0000
    1180        0.0037              nan     0.0100   -0.0000
    1200        0.0036              nan     0.0100   -0.0000
    1220        0.0036              nan     0.0100   -0.0000
    1240        0.0036              nan     0.0100   -0.0000
    1260        0.0035              nan     0.0100   -0.0000
    1280        0.0035              nan     0.0100   -0.0000
    1300        0.0034              nan     0.0100   -0.0000
    1320        0.0034              nan     0.0100   -0.0000
    1340        0.0033              nan     0.0100   -0.0000
    1360        0.0033              nan     0.0100   -0.0000
    1380        0.0033              nan     0.0100   -0.0000
    1400        0.0032              nan     0.0100   -0.0000
    1420        0.0032              nan     0.0100   -0.0000
    1440        0.0031              nan     0.0100   -0.0000
    1460        0.0031              nan     0.0100   -0.0000
    1480        0.0031              nan     0.0100   -0.0000
    1500        0.0031              nan     0.0100   -0.0000
    1520        0.0030              nan     0.0100   -0.0000
    1540        0.0030              nan     0.0100   -0.0000
    1560        0.0030              nan     0.0100   -0.0000
    1580        0.0029              nan     0.0100   -0.0000
    1600        0.0029              nan     0.0100   -0.0000
    1620        0.0029              nan     0.0100    0.0000
    1640        0.0028              nan     0.0100   -0.0000
    1660        0.0028              nan     0.0100   -0.0000
    1680        0.0028              nan     0.0100   -0.0000
    1700        0.0027              nan     0.0100   -0.0000
    1720        0.0027              nan     0.0100   -0.0000
    1740        0.0027              nan     0.0100   -0.0000
    1760        0.0026              nan     0.0100   -0.0000
    1780        0.0026              nan     0.0100   -0.0000
    1800        0.0026              nan     0.0100   -0.0000
    1820        0.0026              nan     0.0100   -0.0000
    1840        0.0025              nan     0.0100   -0.0000
    1860        0.0025              nan     0.0100   -0.0000
    1880        0.0025              nan     0.0100   -0.0000
    1900        0.0025              nan     0.0100   -0.0000
    1920        0.0024              nan     0.0100   -0.0000
    1940        0.0024              nan     0.0100   -0.0000
    1960        0.0024              nan     0.0100   -0.0000
    1980        0.0024              nan     0.0100   -0.0000
    2000        0.0023              nan     0.0100   -0.0000
    2020        0.0023              nan     0.0100   -0.0000
    2040        0.0023              nan     0.0100   -0.0000
    2060        0.0023              nan     0.0100   -0.0000
    2080        0.0023              nan     0.0100   -0.0000
```

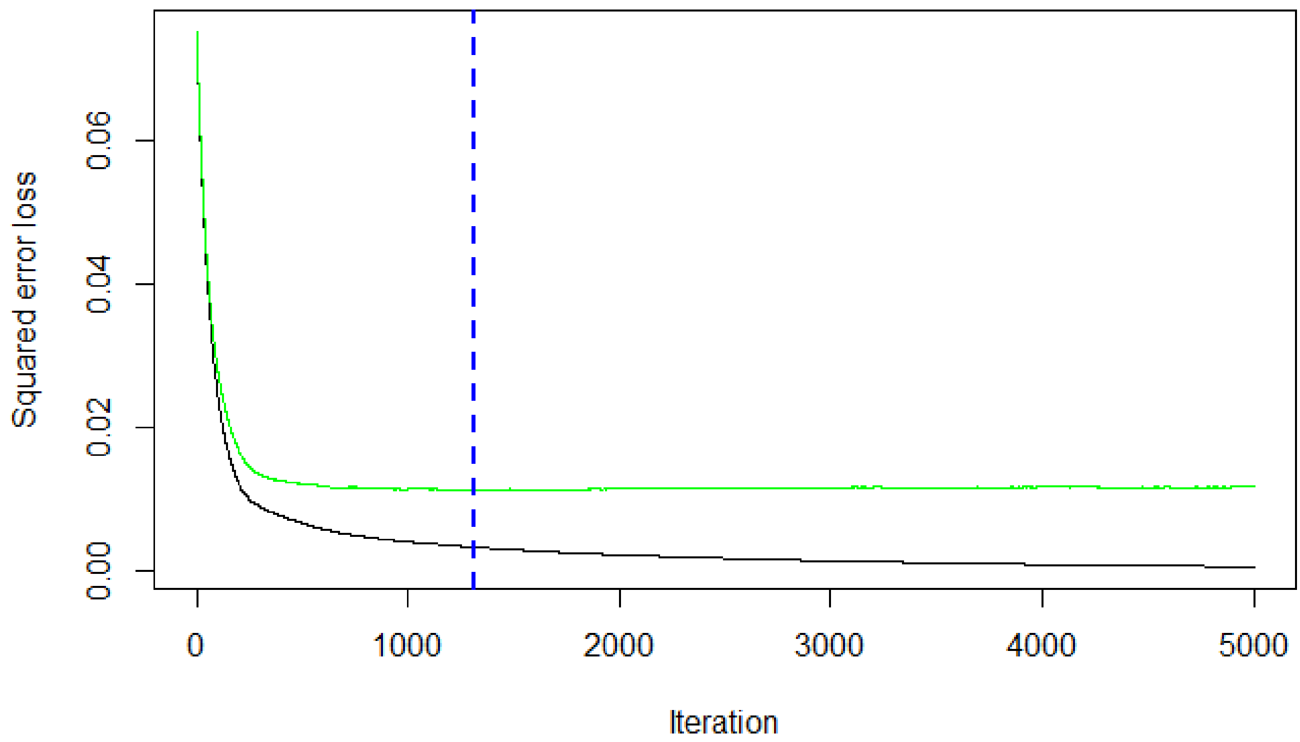
```
2100        0.0022          nan     0.0100   -0.0000
2120        0.0022          nan     0.0100    0.0000
2140        0.0022          nan     0.0100   -0.0000
2160        0.0022          nan     0.0100   -0.0000
2180        0.0022          nan     0.0100    0.0000
2200        0.0021          nan     0.0100   -0.0000
2220        0.0021          nan     0.0100   -0.0000
2240        0.0021          nan     0.0100   -0.0000
2260        0.0021          nan     0.0100   -0.0000
2280        0.0021          nan     0.0100   -0.0000
2300        0.0020          nan     0.0100   -0.0000
2320        0.0020          nan     0.0100   -0.0000
2340        0.0020          nan     0.0100   -0.0000
2360        0.0020          nan     0.0100   -0.0000
2380        0.0020          nan     0.0100   -0.0000
2400        0.0020          nan     0.0100   -0.0000
2420        0.0019          nan     0.0100   -0.0000
2440        0.0019          nan     0.0100   -0.0000
2460        0.0019          nan     0.0100   -0.0000
2480        0.0019          nan     0.0100   -0.0000
2500        0.0019          nan     0.0100   -0.0000
2520        0.0019          nan     0.0100   -0.0000
2540        0.0018          nan     0.0100   -0.0000
2560        0.0018          nan     0.0100   -0.0000
2580        0.0018          nan     0.0100   -0.0000
2600        0.0018          nan     0.0100   -0.0000
2620        0.0018          nan     0.0100   -0.0000
2640        0.0018          nan     0.0100   -0.0000
2660        0.0018          nan     0.0100   -0.0000
2680        0.0017          nan     0.0100   -0.0000
2700        0.0017          nan     0.0100   -0.0000
2720        0.0017          nan     0.0100   -0.0000
2740        0.0017          nan     0.0100   -0.0000
2760        0.0017          nan     0.0100   -0.0000
2780        0.0017          nan     0.0100   -0.0000
2800        0.0016          nan     0.0100   -0.0000
2820        0.0016          nan     0.0100   -0.0000
2840        0.0016          nan     0.0100   -0.0000
2860        0.0016          nan     0.0100   -0.0000
2880        0.0016          nan     0.0100   -0.0000
2900        0.0016          nan     0.0100   -0.0000
2920        0.0016          nan     0.0100   -0.0000
2940        0.0016          nan     0.0100   -0.0000
2960        0.0015          nan     0.0100   -0.0000
2980        0.0015          nan     0.0100   -0.0000
3000        0.0015          nan     0.0100   -0.0000
3020        0.0015          nan     0.0100   -0.0000
3040        0.0015          nan     0.0100   -0.0000
3060        0.0015          nan     0.0100   -0.0000
3080        0.0015          nan     0.0100   -0.0000
3100        0.0015          nan     0.0100   -0.0000
3120        0.0015          nan     0.0100    0.0000
3140        0.0014          nan     0.0100   -0.0000
3160        0.0014          nan     0.0100   -0.0000
3180        0.0014          nan     0.0100   -0.0000
3200        0.0014          nan     0.0100   -0.0000
3220        0.0014          nan     0.0100   -0.0000
3240        0.0014          nan     0.0100   -0.0000
```

```
  3260         0.0014               nan      0.0100   -0.0000
  3280         0.0014               nan      0.0100   -0.0000
  3300         0.0014               nan      0.0100   -0.0000
  3320         0.0013               nan      0.0100   -0.0000
  3340         0.0013               nan      0.0100   -0.0000
  3360         0.0013               nan      0.0100   -0.0000
  3380         0.0013               nan      0.0100   -0.0000
  3400         0.0013               nan      0.0100   -0.0000
  3420         0.0013               nan      0.0100   -0.0000
  3440         0.0013               nan      0.0100   -0.0000
  3460         0.0013               nan      0.0100   -0.0000
  3480         0.0013               nan      0.0100   -0.0000
  3500         0.0012               nan      0.0100   -0.0000
  3520         0.0012               nan      0.0100   -0.0000
  3540         0.0012               nan      0.0100   -0.0000
  3560         0.0012               nan      0.0100   -0.0000
  3580         0.0012               nan      0.0100   -0.0000
  3600         0.0012               nan      0.0100   -0.0000
  3620         0.0012               nan      0.0100   -0.0000
  3640         0.0012               nan      0.0100   -0.0000
  3660         0.0012               nan      0.0100   -0.0000
  3680         0.0012               nan      0.0100   -0.0000
  3700         0.0012               nan      0.0100   -0.0000
  3720         0.0011               nan      0.0100   -0.0000
  3740         0.0011               nan      0.0100   -0.0000
  3760         0.0011               nan      0.0100   -0.0000
  3780         0.0011               nan      0.0100   -0.0000
  3800         0.0011               nan      0.0100   -0.0000
  3820         0.0011               nan      0.0100   -0.0000
  3840         0.0011               nan      0.0100   -0.0000
  3860         0.0011               nan      0.0100   -0.0000
  3880         0.0011               nan      0.0100   -0.0000
  3900         0.0011               nan      0.0100   -0.0000
  3920         0.0011               nan      0.0100   -0.0000
  3940         0.0011               nan      0.0100   -0.0000
  3960         0.0010               nan      0.0100   -0.0000
  3980         0.0010               nan      0.0100   -0.0000
  4000         0.0010               nan      0.0100   -0.0000
  4020         0.0010               nan      0.0100   -0.0000
  4040         0.0010               nan      0.0100   -0.0000
  4060         0.0010               nan      0.0100   -0.0000
  4080         0.0010               nan      0.0100   -0.0000
  4100         0.0010               nan      0.0100   -0.0000
  4120         0.0010               nan      0.0100   -0.0000
  4140         0.0010               nan      0.0100   -0.0000
  4160         0.0010               nan      0.0100   -0.0000
  4180         0.0010               nan      0.0100   -0.0000
  4200         0.0010               nan      0.0100   -0.0000
  4220         0.0010               nan      0.0100   -0.0000
  4240         0.0010               nan      0.0100   -0.0000
  4260         0.0009               nan      0.0100   -0.0000
  4280         0.0009               nan      0.0100   -0.0000
  4300         0.0009               nan      0.0100   -0.0000
  4320         0.0009               nan      0.0100   -0.0000
  4340         0.0009               nan      0.0100   -0.0000
  4360         0.0009               nan      0.0100   -0.0000
  4380         0.0009               nan      0.0100   -0.0000
  4400         0.0009               nan      0.0100   -0.0000
```

```
  4420        0.0009             nan     0.0100   -0.0000
  4440        0.0009             nan     0.0100   -0.0000
  4460        0.0009             nan     0.0100   -0.0000
  4480        0.0009             nan     0.0100   -0.0000
  4500        0.0009             nan     0.0100   -0.0000
  4520        0.0009             nan     0.0100   -0.0000
  4540        0.0009             nan     0.0100   -0.0000
  4560        0.0008             nan     0.0100   -0.0000
  4580        0.0008             nan     0.0100   -0.0000
  4600        0.0008             nan     0.0100   -0.0000
  4620        0.0008             nan     0.0100   -0.0000
  4640        0.0008             nan     0.0100   -0.0000
  4660        0.0008             nan     0.0100   -0.0000
  4680        0.0008             nan     0.0100   -0.0000
  4700        0.0008             nan     0.0100   -0.0000
  4720        0.0008             nan     0.0100   -0.0000
  4740        0.0008             nan     0.0100   -0.0000
  4760        0.0008             nan     0.0100   -0.0000
  4780        0.0008             nan     0.0100   -0.0000
  4800        0.0008             nan     0.0100   -0.0000
  4820        0.0008             nan     0.0100   -0.0000
  4840        0.0008             nan     0.0100   -0.0000
  4860        0.0008             nan     0.0100   -0.0000
  4880        0.0008             nan     0.0100   -0.0000
  4900        0.0008             nan     0.0100   -0.0000
  4920        0.0007             nan     0.0100   -0.0000
  4940        0.0007             nan     0.0100   -0.0000
  4960        0.0007             nan     0.0100   -0.0000
  4980        0.0007             nan     0.0100   -0.0000
  5000        0.0007             nan     0.0100   -0.0000
```

Hide

```
m_boost_perf = gbm.perf(m_boost, method = "cv")
```



Hide

```
BB_prob_pred = predict(m_boost,test_set, n.trees=m_boost_perf)
BB_pred = ifelse(BB_prob_pred > 0.5, 1, 0)

# Making the Confusion Matrix
BB_car_cm <- table(BB_pred,test_set[, 8])
print(BB_car_cm)
```

```
BB_pred   0   1
      1 115  11
```

Hide

```
# calculating the accuracy - We are defining the accuracy function to show the Boosting perfo
rmance output
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x))))*100}
accuracy(BB_car_cm)
```

```
[1] 91.26984
```