Capstone Project - Prediction of Default In Insurance Coy

Chinedu H Obetta

11/8/2020

Load all libraries/Packages

```
library(readxl)
library(ggplot2)
library(gridExtra)
library(DataExplorer)
library(mice)   # To treat missing values using k-Nearest Neighbour(KNN)
library(caTools) # Split Data into Test and Train Set
library(lmtest) # To confirm the validity of the logistics models
library(plyr) # To rename variable
library(usdm) # for VIF
library(caTools) # Split Data into Test and Train Set
library(caret) # for confusion matrix function
library(randomForest) # to build a random forest model
library(rpart) # to build a decision model
library(rpart.plot) # to plot decision tree model
library(rattle)
library(xgboost) # to build a XGboost model
library(ROCR)
```

Environment Set up and Data Import

Set Working Directory

```
setwd("C:/Users/Chinedu/Documents/GREAT LEARNING-UNIVERSITY OF
TEXAS/TABLEAU/Capstone Project")
dip <- read_xlsx("premium.xlsx")
```

Take 5% of the whole data to speed-up the model building & iterations

```
prop.table(table(dip$renewal))*100

##
##         0         1
##  6.259001 93.740999

split <- sample.split(dip$renewal, SplitRatio = 0.95)
db <- subset(dip, split == FALSE)
```

Dropping the huge dip file from the encironment as it is no longer required

```
rm(dip)
rm(split)
```

Renaming of variables

```
db <- rename(db, c("perc_premium_paid_by_cash_credit" = "cash.credit",
"Count_3-6_months_late"="late.pmt.3_6", "Count_6-
12_months_late"="late.pmt.6_12", "Count_more_than_12_months_late"=
"late.pmt.More_12Mnth", "Marital Status" = "Marital.Status", "Veh_Owned" =
"Vehicle", "No_of_dep" ="Dependents", "no_of_premiums_paid" = "No_premium",
"sourcing_channel" ="Sources", "residence_area_type" = "Residence",
"age_in_days" ="Age", "renewal"="Default" ))
```

Dropping ID

```
db$id<- NULL
```

#2b Creation of new variables " Late Payment"

```
db <- as.data.frame(db)

db$late.pmt <- rowSums(subset(db, select= late.pmt.3_6:
late.pmt.More_12Mnth))
```

#Dropping

```
db$late.pmt.3_6<- NULL
db$late.pmt.6_12 <- NULL
db$late.pmt.More_12Mnth <- NULL
db$Default<-as.factor(db$Default)
```

Ensure that the target variable Renamed the levels & Relevel

```
levels(db$Default) <- c("Default", "NotDefault")
db$Default <- relevel(db$Default, ref = "Default") # Reference Default :
Default
levels(db$Default)

## [1] "Default"    "NotDefault"
```

Summary of the data

```
summary(db)

##   cash.credit          Age             Income          Marital.Status
##  Min.   :0.0000   Min.   : 7676   Min.   :  24030   Min.   :0.0000
##  1st Qu.:0.0330   1st Qu.:14977   1st Qu.: 108140   1st Qu.:0.0000
##  Median :0.1700   Median :18629   Median : 168080   Median :0.0000
```

```
##  Mean   :0.3169   Mean   :18948   Mean   : 204516   Mean   :0.4949
##  3rd Qu.:0.5400   3rd Qu.:22640   3rd Qu.: 250720   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :33950   Max.   :6560280   Max.   :1.0000
##     Vehicle     Dependents      Accomodation      risk_score      No_premium
##  Min.   :1   Min.   :1.000   Min.   :0.0000   Min.   :92.59   Min.   :
2.00
##  1st Qu.:1   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:98.84   1st Qu.:
7.00
##  Median :2   Median :2.000   Median :1.0000   Median :99.18   Median
:10.00
##  Mean   :2   Mean   :2.471   Mean   :0.5016   Mean   :99.10   Mean
:10.83
##  3rd Qu.:3   3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:99.54   3rd
Qu.:13.00
##  Max.   :3   Max.   :4.000   Max.   :1.0000   Max.   :99.89   Max.
:58.00
##    Sources            Residence            premium            Default
##  Length:3993        Length:3993        Min.   : 1200   Default   : 250
##  Class :character   Class :character   1st Qu.: 5400   NotDefault:3743
##  Mode  :character   Mode  :character   Median : 7500
##                                        Mean   :10926
##                                        3rd Qu.:13800
##                                        Max.   :60000
##     late.pmt
##  Min.   : 0.0000
##  1st Qu.: 0.0000
##  Median : 0.0000
##  Mean   : 0.3722
##  3rd Qu.: 0.0000
##  Max.   :13.0000
```

Observations:

- A glimpse of the median and the maximum values shows the existence of potential outliers in some of the variables such as the ratio of cash payment, the age of the policy holders, the income of the policy holders, the total number of premium paid by policy holders and the value of the premium paid

COnfirmation of missing variables

```
sum(is.na(db ))
```

```
## [1] 0
```

```
str(db)
```

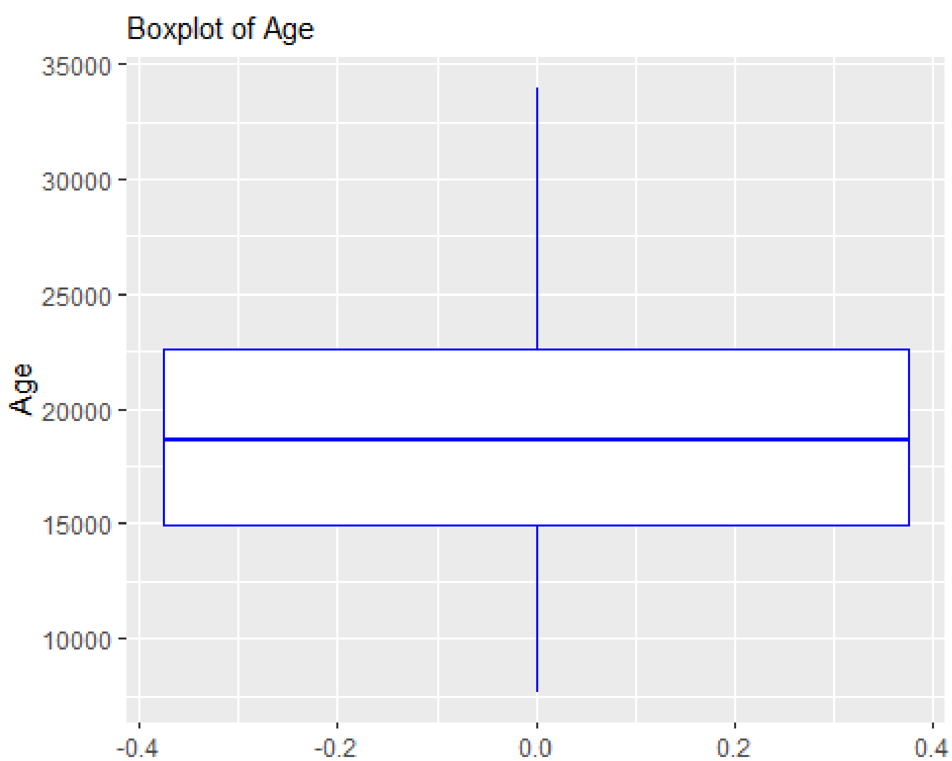
```
## 'data.frame':    3993 obs. of  14 variables:
##  $ cash.credit   : num  0.467 0.035 0.679 0.256 0.169 0.233 1 1 0.791
0.023 ...
##  $ Age           : num  20813 12781 23001 10956 16805 ...
```

```
## $ Income : num 174160 187570 378130 129680 150140 ...
## $ Marital.Status: num 1 1 0 1 1 0 1 1 0 0 ...
## $ Vehicle : num 1 3 3 1 1 1 3 2 3 2 ...
## $ Dependents : num 2 4 3 2 2 4 3 1 3 3 ...
## $ Accomodation : num 0 0 0 1 0 0 0 1 1 0 ...
## $ risk_score : num 99.1 98.8 98.2 99.3 99 ...
## $ No_premium : num 19 12 17 7 9 13 7 3 6 13 ...
## $ Sources : chr "A" "A" "B" "C" ...
## $ Residence : chr "Rural" "Urban" "Urban" "Urban" ...
## $ premium : num 11700 13800 20100 5400 13800 11700 5400 5700 11700
13800 ...
## $ Default : Factor w/ 2 levels "Default","NotDefault": 2 2 2 2 2 2
2 2 2 2 ...
## $ late.pmt : num 0 0 1 0 1 0 5 0 1 0 ...
```

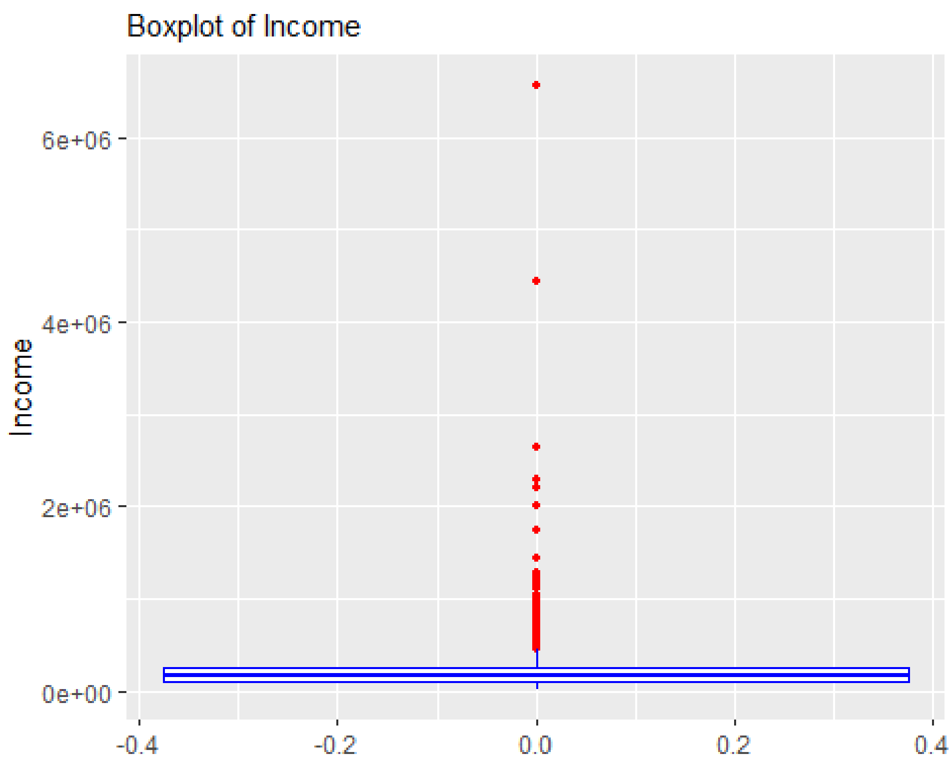
Observations: * The data does not have a missing value

#Checking for outliers on continous variables

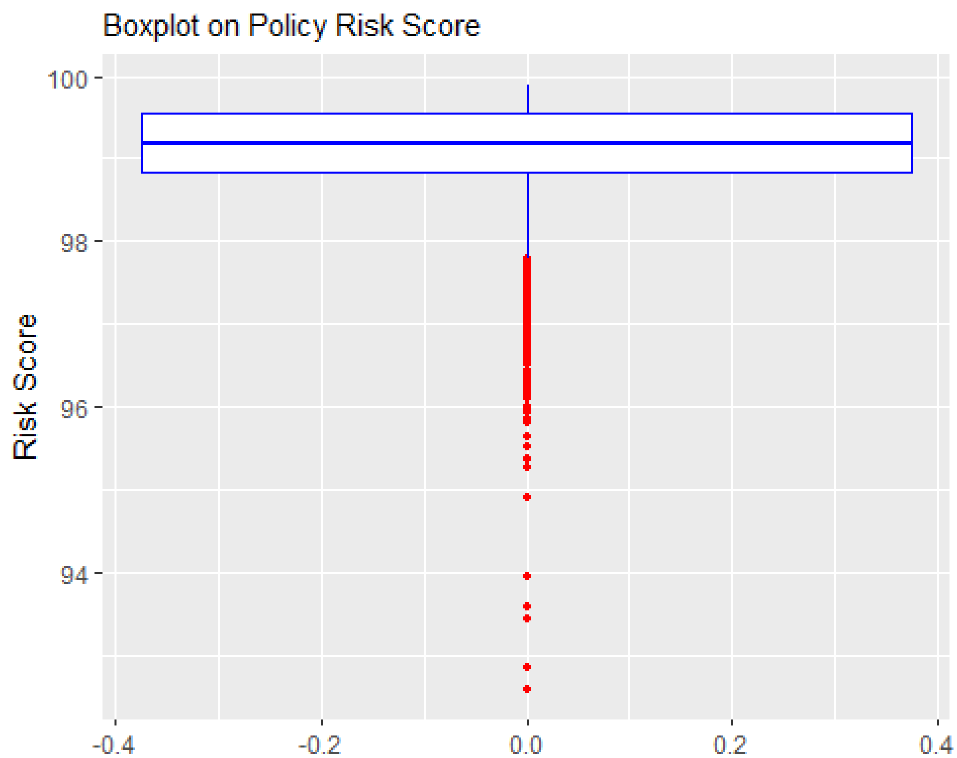
```
outlier_dip <- db

ggplot(outlier_dip, aes( y = Age)) +
    geom_boxplot(outlier.colour = "red", outlier.size = 1, col= "blue") +
labs( y = "Age", subtitle = "Boxplot of Age")
```



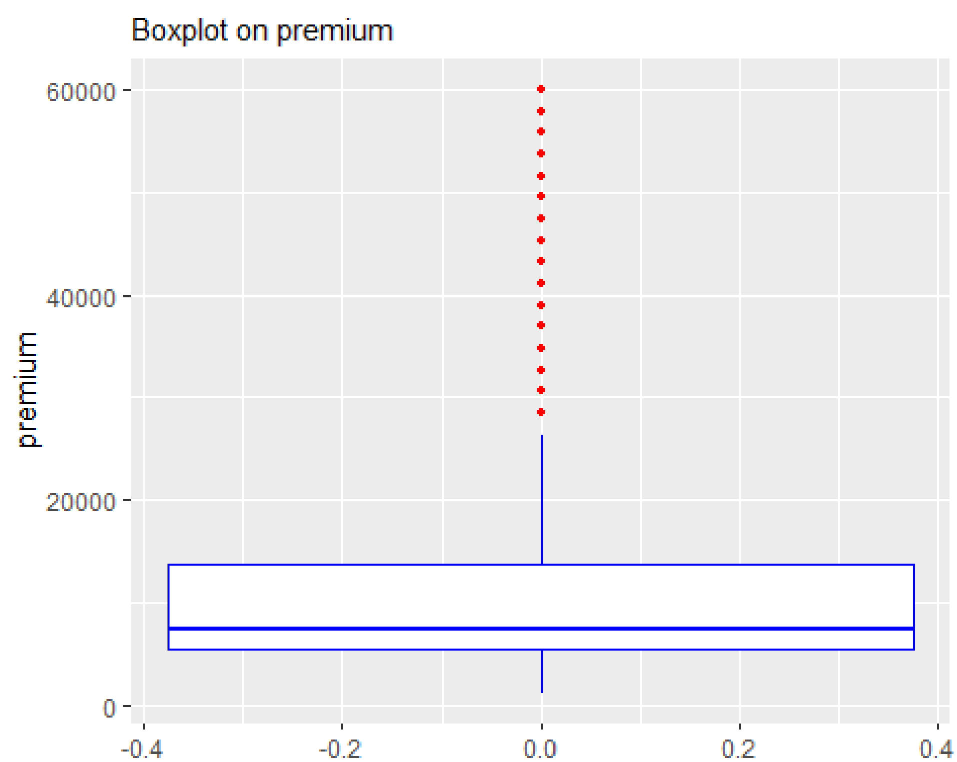
```
ggplot(outlier_dip, aes( y = Income)) +
    geom_boxplot(outlier.colour = "red", outlier.size = 1, col= "blue") +
labs( y = "Income", subtitle = "Boxplot of Income")
```




```
ggplot(outlier_dip, aes( y = risk_score)) +
    geom_boxplot(outlier.colour = "red", outlier.size = 1, col= "blue") +
labs( y = "Risk Score", subtitle = "Boxplot on Policy Risk Score")
```




```r
ggplot(outlier_dip, aes( y = premium)) +
    geom_boxplot(outlier.colour = "red", outlier.size = 1, col= "blue") +
labs( y = "premium", subtitle = "Boxplot on premium")
```



Observation * The boxplots for the continous variables confirms the exisitence of outliers in the variables. These identified outliers will be treated later.

##Treatment of outliers

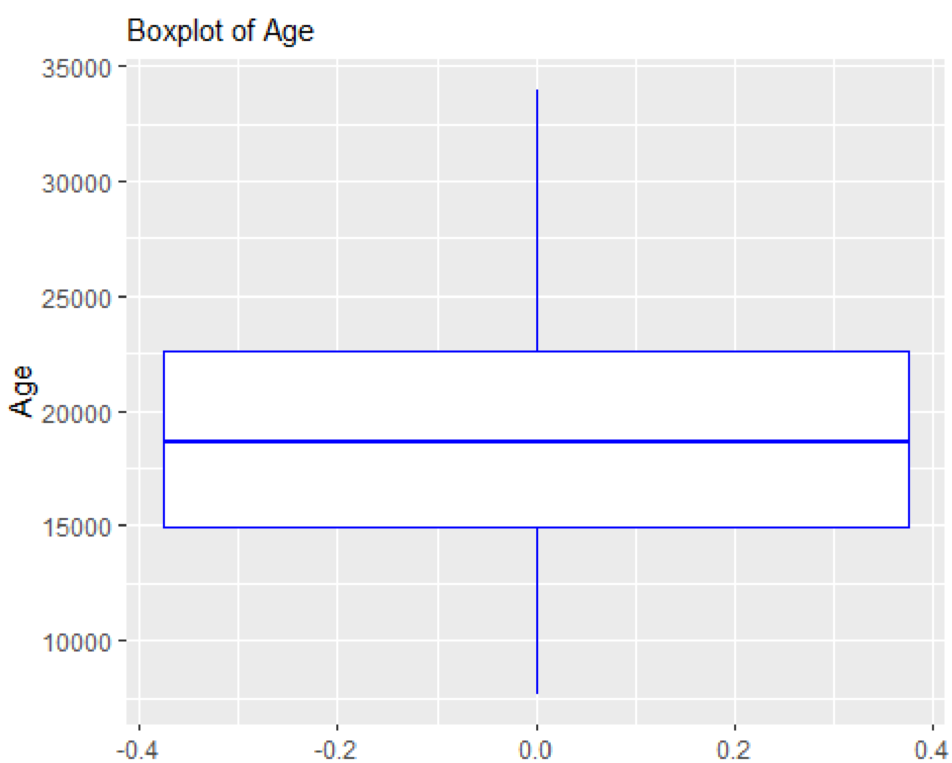
```
outfun <- function(x){
    qntile <- quantile(x, probs = c(.25, 0.75))
    caps <- quantile(x, probs = c(0.05, 0.95))
    H <- 1.5 *IQR(x, na.rm = T)
    x[x< (qntile[1]-H)] <- caps[1]
    x[x> (qntile[2])+ H] <- caps[2]
    return(x)
}
```

Treatment by applying the custom function for outliers as defined

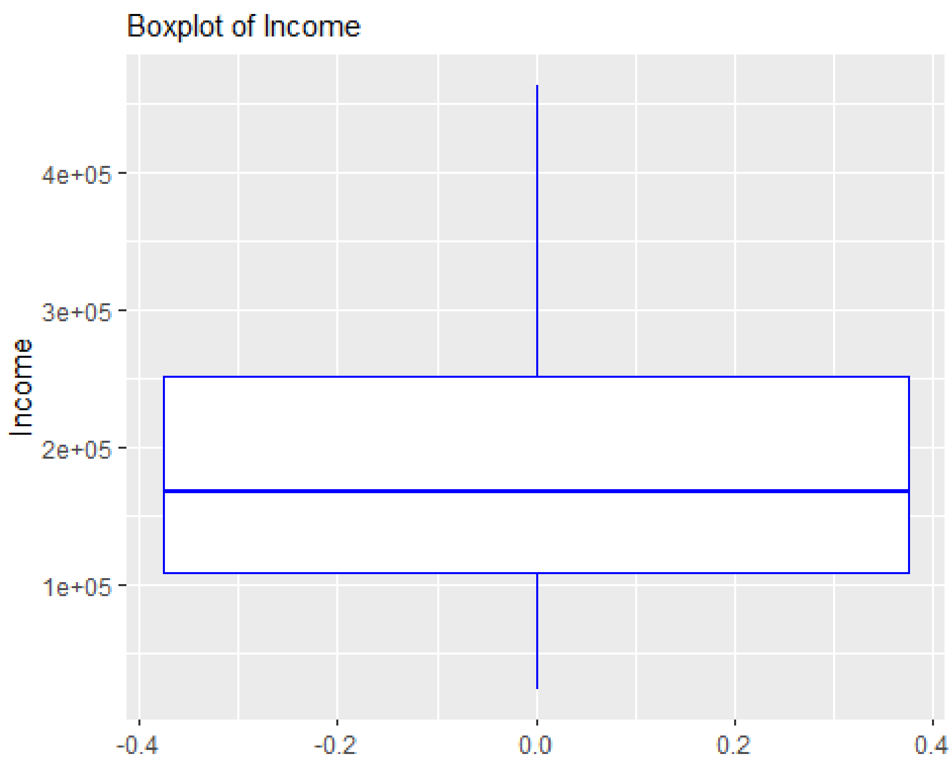
```
outlier_dip$Age <- outfun(outlier_dip$Age)
outlier_dip$premium <-outfun(outlier_dip$premium)
outlier_dip$risk_score <- outfun(outlier_dip$risk_score)
outlier_dip$Income <- outfun(outlier_dip$Income)
```

Confirmation of the treatment

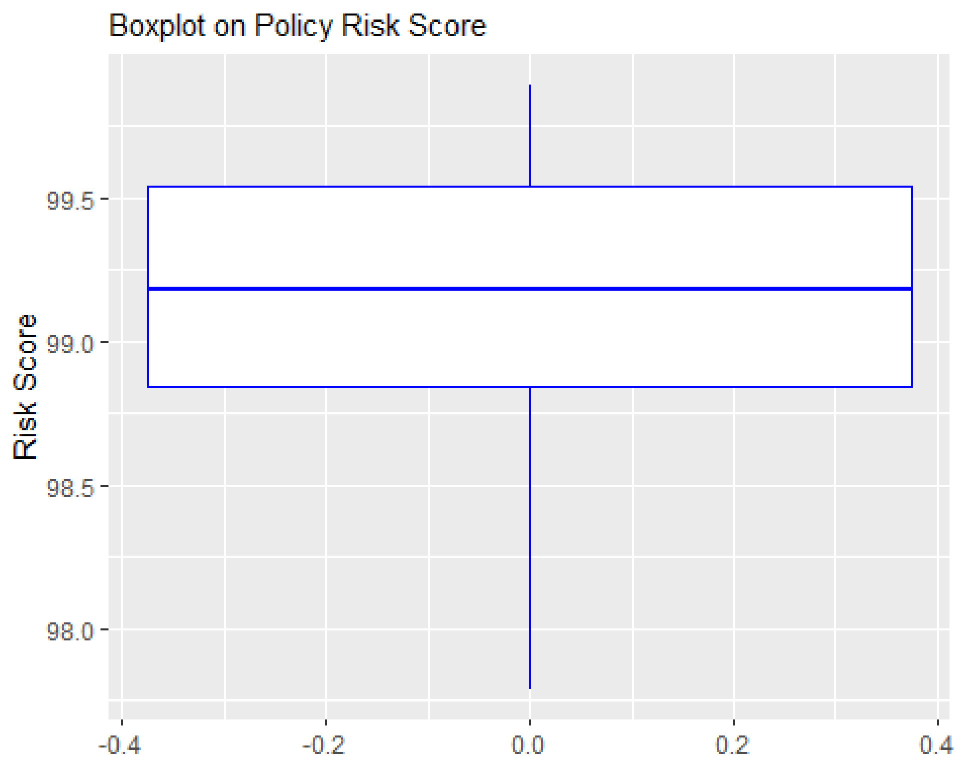
```
ggplot(outlier_dip, aes( y = Age)) +
    geom_boxplot(outlier.colour = "red", outlier.size = 1, col= "blue") +
labs( y = "Age", subtitle = "Boxplot of Age")
```



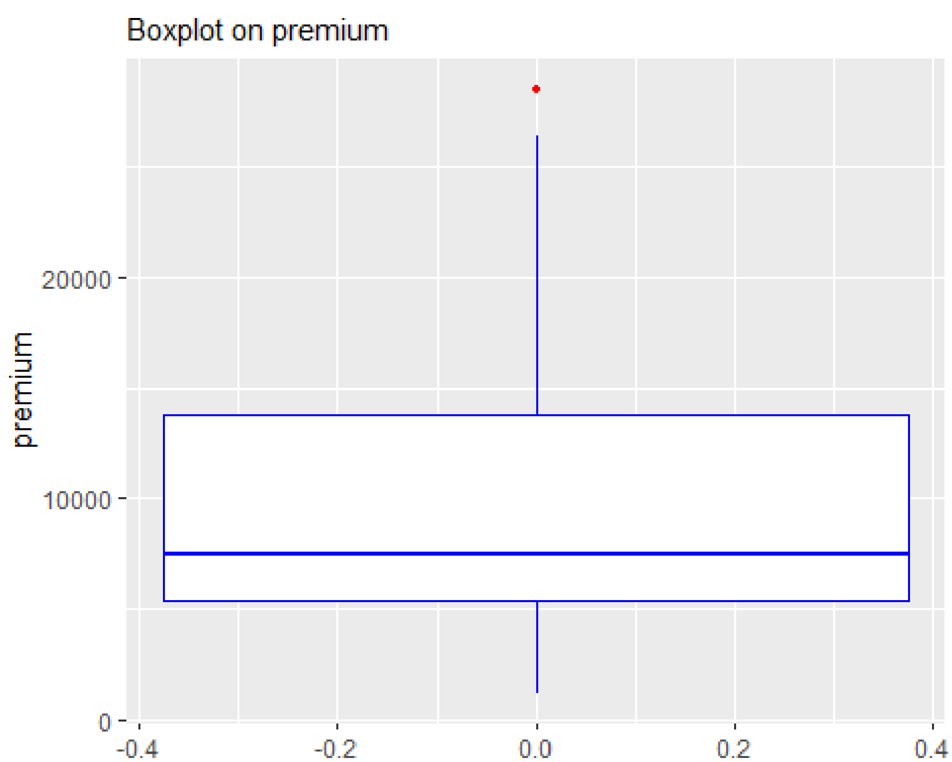
```r
ggplot(outlier_dip, aes( y = Income)) +
    geom_boxplot(outlier.colour = "red", outlier.size = 1, col= "blue") +
labs( y = "Income", subtitle = "Boxplot of Income")
```




```r
ggplot(outlier_dip, aes( y = risk_score)) +
    geom_boxplot(outlier.colour = "red", outlier.size = 1, col= "blue") +
labs( y = "Risk Score", subtitle = "Boxplot on Policy Risk Score")
```




```
ggplot(outlier_dip, aes( y = premium)) +
    geom_boxplot(outlier.colour = "red", outlier.size = 1, col= "blue") +
labs( y = "premium", subtitle = "Boxplot on premium")
```



#Variable transformation otherwise known as the feature Engineering

Here, we will modify existing features to get a better insights into the dependent variable"Default"

#1 Variable: Age

```
summary(outlier_dip$Income)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24030  108140  168080  191386  250720  462600
```

Observation: * The age of the policy holders were recorded in days in stead of years. Thus, the variable "Age" will be transformed to be in years instead of days. This will give us more useful insight about the age of the policyholders and its relevant on the dependent variable

#1 Conversion of age in days to age in years

```
outlier_dip$Age <- round((outlier_dip$Age)/365)
summary(outlier_dip$Income)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24030  108140  168080  191386  250720  462600
```

##Default Rate Across Income Group

```
# Between $20,000 & $44,999 =>  Low_income(20k-45k)
# Between $45,000 & $119,999 => Middle_class(46k-119k)
# Between $120,000 & $149,999    => Upper_middle_class(120k-150k)
# Between $150,000 & $199,999   =>  High_income(150k-200k)
# More than $200,000 =>  Super_Rich(>200k)
d <- c(20000, 45000,120000, 150000, 200000, 500000)
groups <- c("Low_income(20k-45k)", "Middle_class(46k-119k)",
"Upper_middle_class(120k-150k)", "High_income(150k-200k)", "Super_Rich(>200)"
)
```

#Addition of new variables

```
outlier_dip$Income_Group <- outlier_dip$Income
outlier_dip$Income_Group <- cut(outlier_dip$Income_Group, breaks = d, labels
= groups)
round(prop.table(table(outlier_dip$Income_Group))*100)

##
##           Low_income(20k-45k)        Middle_class(46k-119k)
##                             3                            26
## Upper_middle_class(120k-150k)        High_income(150k-200k)
##                            12                            20
##              Super_Rich(>200)
##                            39
```

Observations: * 39% of the customer's under review are super rich, that is they earn over $200,000 annually. While 29% are middle class customers.

- Less than 5% of the policy holders earn less than $45, 000 annually

##Default Rate Across Generation

```
# [Age]>=20 AND [Age]<=40 THEN " Millennials-'32-40'"
# [Age]>=41 AND [Age]<=55 THEN " Generation X-'41-55'"
# [Age]>=56 AND [Age]<=74 THEN " Baby Boomer-'56-74'"
# [Age]>= 75 AND [Age]<=92 THEN " Silent Gen-'75-95'"
b <- c(20,40,55,75,95)
names <- c("Millennials(32-40)", "Generation_X(41-55)", "Baby Boomer(56-74)",
"Silent_Gen(75-95)")
```

#Addition of new variables

```
outlier_dip$Generation <- outlier_dip$Age
outlier_dip$Generation <- cut(outlier_dip$Generation, breaks = b, labels =
names)
round(prop.table(table(outlier_dip$Generation))*100)

##
##  Millennials(32-40) Generation_X(41-55)  Baby Boomer(56-74)
Silent_Gen(75-95)
##                  24                  36                  35
6
```

Observation:

- Over 70% of the policy holders are between 41 and 74 years old, 25% are Millennials while the balance of 6% are over 75 years old.

```
# 0 Number ->  "Zero"
# between 1-5# of late payment -> "Between 1 & 5"
# More than 5# of late payment ->  "greater than 5"
e <- c(-5,0,5,20)
parts <- c("Zero", "Between 1 & 5", " greater than 5")
```

#Addition of new variables

```
outlier_dip$late.pmt.type <- outlier_dip$late.pmt
outlier_dip$late.pmt.type <- cut(outlier_dip$late.pmt.type, breaks= e, labels
= parts)
prop.table(table(outlier_dip$late.pmt.type))*100

##
##            Zero   Between 1 & 5  greater than 5
##      79.8898072      19.4340095       0.6761833
```

##Treatment of factor variables

```
names(outlier_dip)
```

```
##  [1] "cash.credit"    "Age"            "Income"         "Marital.Status"
##  [5] "Vehicle"        "Dependents"     "Accomodation"   "risk_score"
##  [9] "No_premium"     "Sources"        "Residence"      "premium"
## [13] "Default"        "late.pmt"       "Income_Group"   "Generation"
## [17] "late.pmt.type"

rfac.names <- c(4,10,11, 13, 15, 16, 17)
outlier_dip[, rfac.names] <- lapply(outlier_dip[, rfac.names], factor)
```

FInal Review of preprocessed dataset

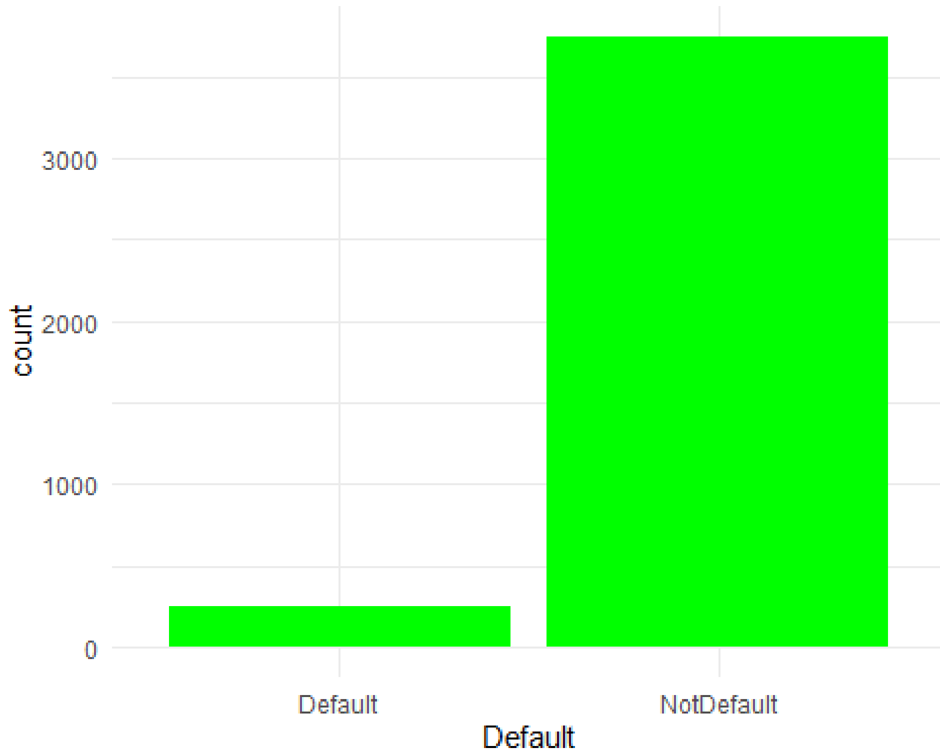
```
treated_dip <- outlier_dip
str(treated_dip)

## 'data.frame':    3993 obs. of  17 variables:
##  $ cash.credit   : num  0.467 0.035 0.679 0.256 0.169 0.233 1 1 0.791 
0.023 ...
##  $ Age           : num  57 35 63 30 46 58 45 22 46 51 ...
##  $ Income        : num  174160 187570 378130 129680 150140 ...
##  $ Marital.Status: Factor w/ 2 levels "0","1": 2 2 1 2 2 1 2 2 1 1 ...
##  $ Vehicle       : num  1 3 3 1 1 1 3 2 3 2 ...
##  $ Dependents    : num  2 4 3 2 2 4 3 1 3 3 ...
##  $ Accomodation  : num  0 0 0 1 0 0 0 1 1 0 ...
##  $ risk_score    : num  99.1 98.8 98.2 99.3 99 ...
##  $ No_premium    : num  19 12 17 7 9 13 7 3 6 13 ...
##  $ Sources       : Factor w/ 5 levels "A","B","C","D",..: 1 1 2 3 3 4 3 1 
1 3 ...
##  $ Residence     : Factor w/ 2 levels "Rural","Urban": 1 2 2 2 2 1 2 2 2 2 
...
##  $ premium       : num  11700 13800 20100 5400 13800 11700 5400 5700 11700 
13800 ...
##  $ Default       : Factor w/ 2 levels "Default","NotDefault": 2 2 2 2 2 2 
2 2 2 2 ...
##  $ late.pmt      : num  0 0 1 0 1 0 5 0 1 0 ...
##  $ Income_Group  : Factor w/ 5 levels "Low_income(20k-45k)",..: 4 4 5 3 4 
5 2 1 5 5 ...
##  $ Generation    : Factor w/ 4 levels "Millennials(32-40)",..: 3 1 3 1 2 3 
2 1 2 2 ...
##  $ late.pmt.type : Factor w/ 3 levels "Zero","Between 1 & 5",..: 1 1 2 1 2 
1 2 1 2 1 ...
```

EDA

#Distribution of the dependent variable

```
prop.table(table(treated_dip$Default))*100

## 
##    Default NotDefault 
##   6.260957  93.739043
```

```r
ggplot(treated_dip) +
 aes(x = Default) +
 geom_bar(fill = "green") +
 theme_minimal()
```



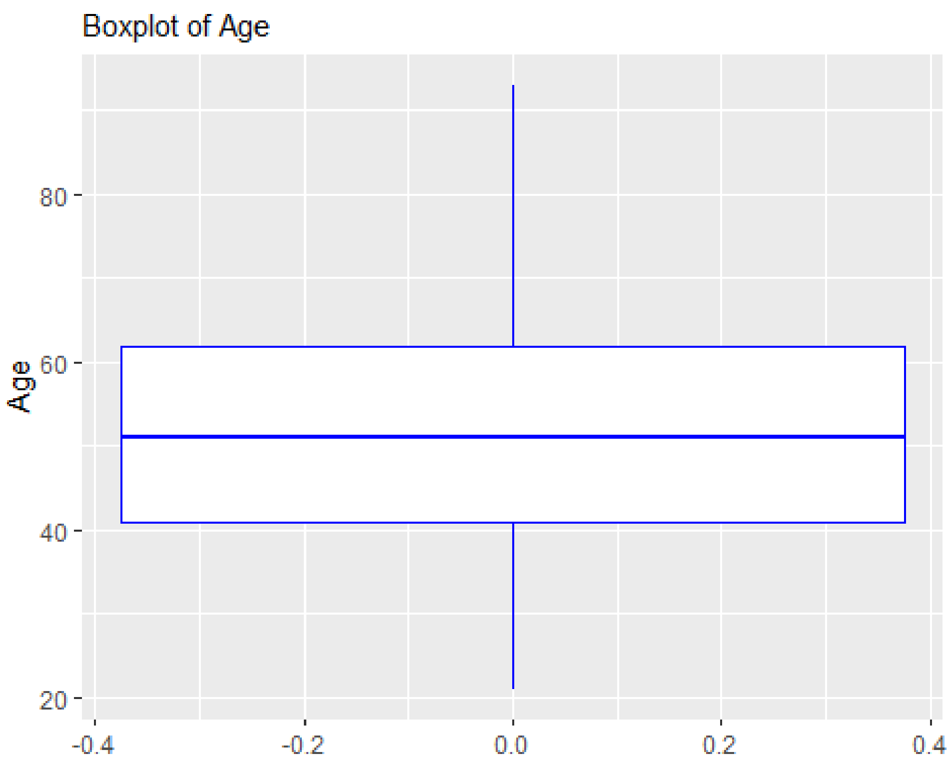

```r
prop.table(table(treated_dip$Marital.Status))

##
##        0        1
## 0.505134 0.494866

names(treated_dip)

##  [1] "cash.credit"    "Age"            "Income"         "Marital.Status"
##  [5] "Vehicle"        "Dependents"     "Accomodation"   "risk_score"
##  [9] "No_premium"     "Sources"        "Residence"      "premium"
## [13] "Default"        "late.pmt"       "Income_Group"   "Generation"
## [17] "late.pmt.type"
```

Observations;

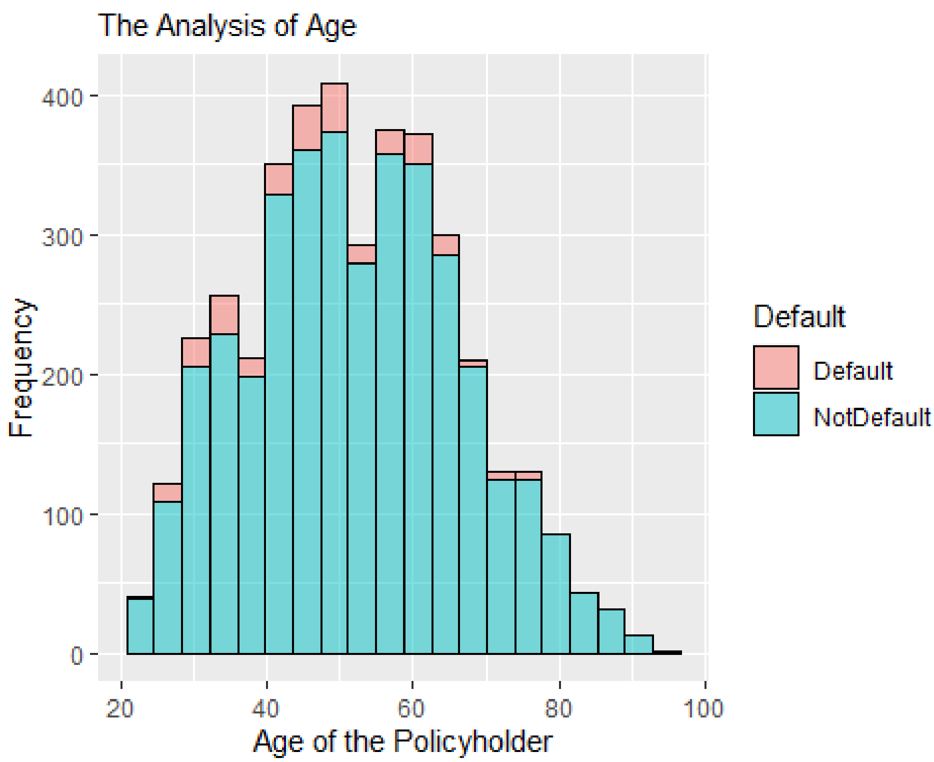
- The observation shows that 6% of the dataset defaulted in the renewal of the policy while 94% did not default.
- The dataset is imbalanced as it is skewed to non defaulters. It is therefore important to balance the dataset using smote during the model building

Distribution on Age

```r
ggplot(treated_dip, aes( y = Age)) +
    geom_boxplot(outlier.colour = "red", outlier.size = 1, col= "blue") +
labs( y = "Age", subtitle = "Boxplot of Age")
```




```r
ggplot(treated_dip, aes_string(x=treated_dip$Age, fill="Default")) +
geom_histogram(bins=20,alpha=0.5,colour='black') + labs(x = " Age of the
Policyholder ", y = "Frequency", subtitle = "The Analysis of Age")
```

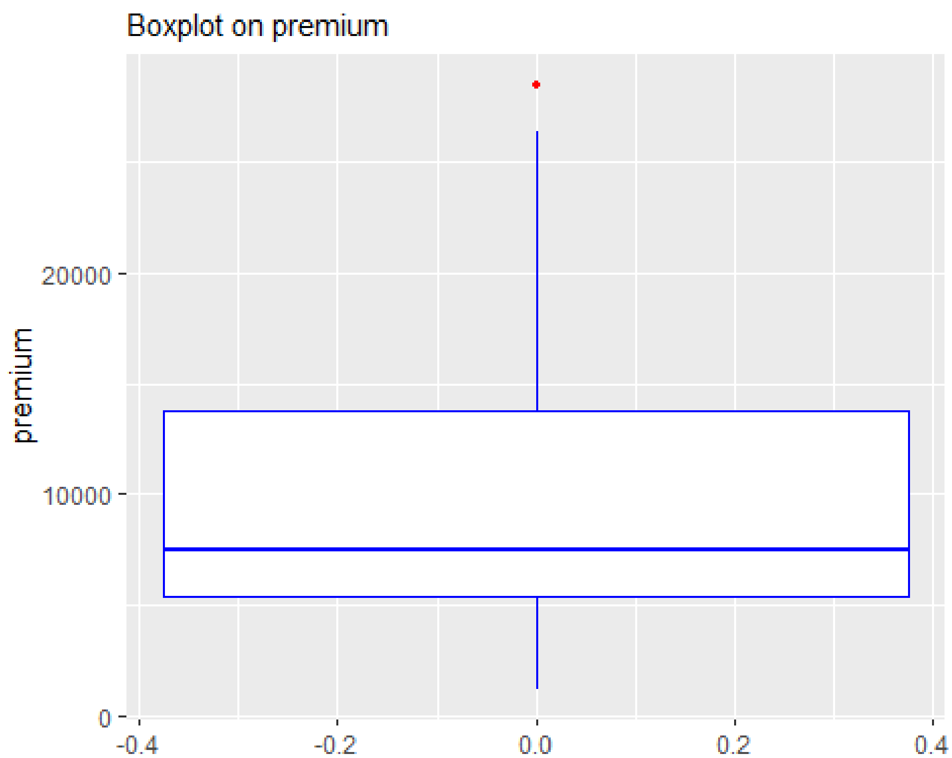


Observations; ** The average of all the policyholders is 52 years old, the youngest and oldest policyholder is 21 and 103 years old respectively. ** There is no much difference in the age range of policyholder that renew their policy and those that do not renew theirs. ** Most of the policy holders are within the working age as 75% of all the policy holders are below 62 years old .

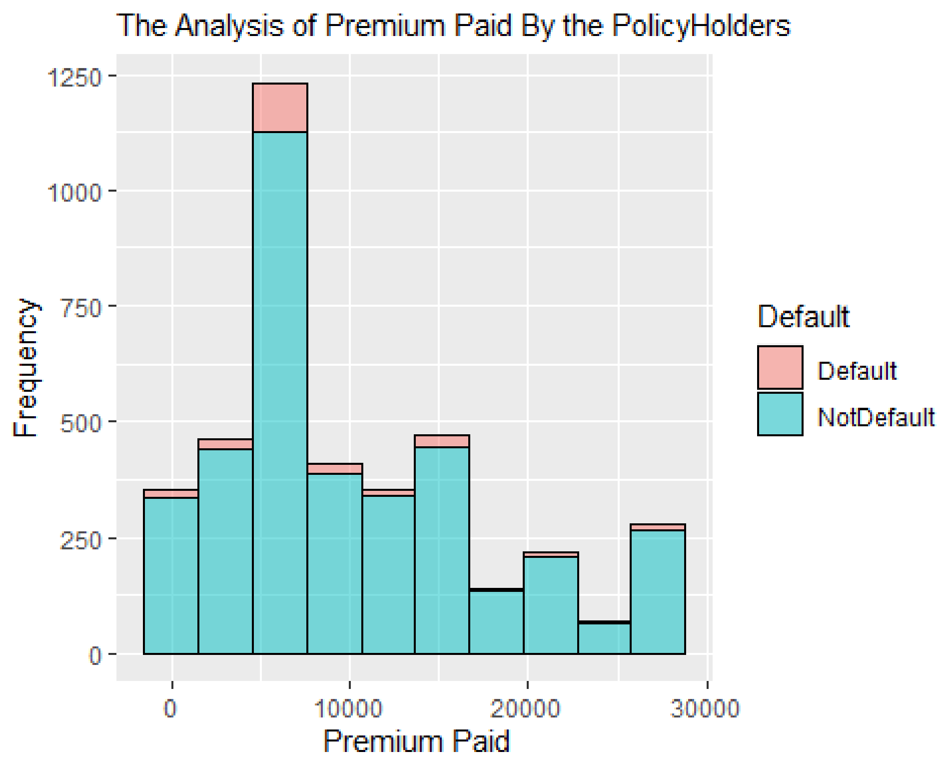
- The boxplot does show some number of potential outliers as the difference between the mean age and the oldest person is very high. Thus, there will be need for outliers treatment.
- The P-value is very low, thus, the distribution of age follows normal distribution and not due to chance

Observations on Premium

```
ggplot(treated_dip, aes( y = premium)) +
    geom_boxplot(outlier.colour = "red", outlier.size = 1, col= "blue") +
labs( y = "premium", subtitle = "Boxplot on premium")
```




```
ggplot(treated_dip, aes_string(x=treated_dip$premium, fill="Default")) +
geom_histogram(bins=10,alpha=0.5,colour='black') + labs(x = " Premium Paid",
y = "Frequency", subtitle = "The Analysis of Premium Paid By the
PolicyHolders")
```

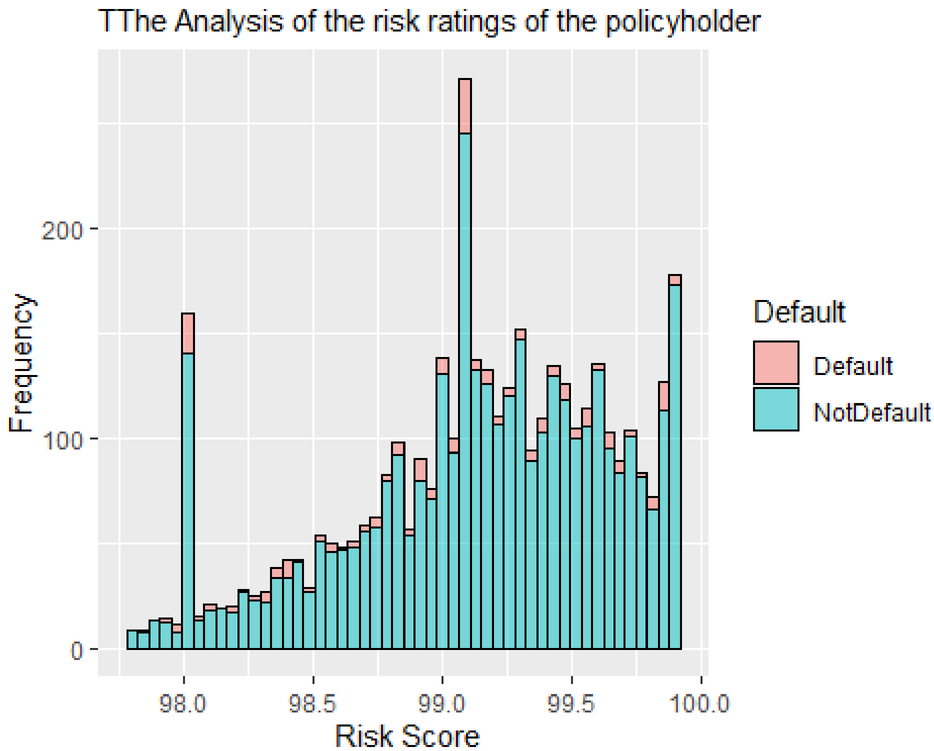


Observations Premiums:

- There seems to be no difference in premium paid amongst the policyholders that renew their policy and those that do not
- The P-value is very low, thus, the distribution of premium paid by the policy holders follows normal distribution and not due to chance.
- The average premium paid by policyholders is USd10,988 and 75% of them pay less than USD13,800
- The Boxplot shows that the observations contains few outliers and this has been treated before building a model.

The Analysis of the risk ratings of the policyholder.

```
ggplot(treated_dip, aes_string(x=treated_dip$risk_score, fill="Default")) +
geom_histogram(bins=50,alpha=0.5,colour='black') + labs(x = " Risk Score", y
= "Frequency", subtitle = "TThe Analysis of the risk ratings of the
policyholder")
```

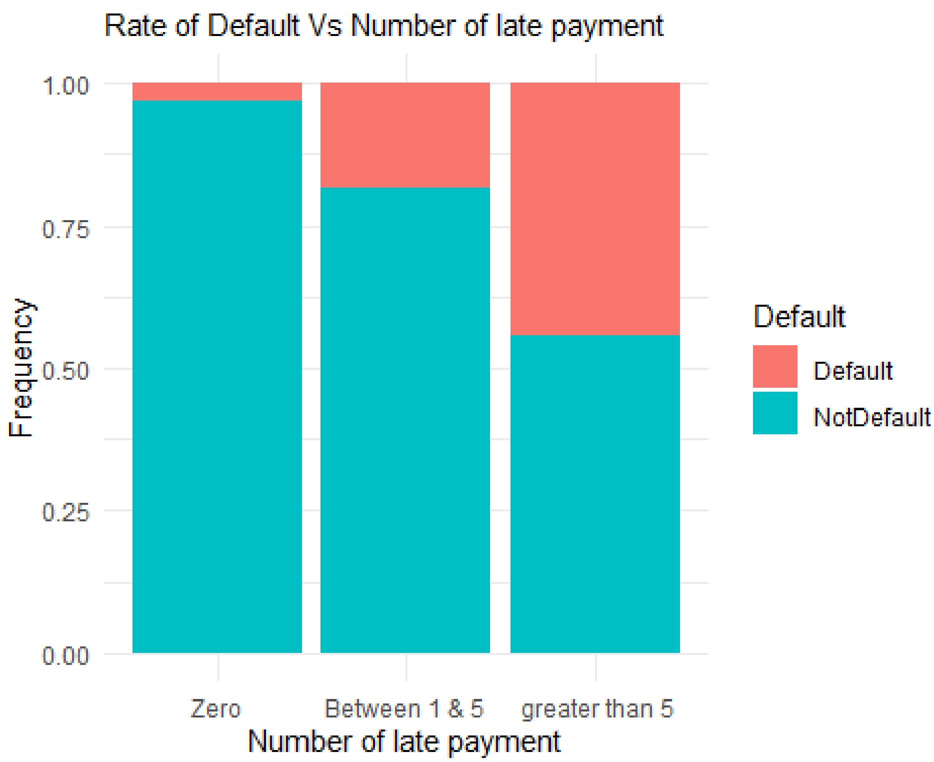


Observations on the Risk Rating of the Policyholders:

- The risk score is skewed to the right with an average risk rating of 99.08. The minimum and maxium risk score is 92.76 and 99.89 respectively.
- There seems to be an effect of the risk rating of the policyholders on the status of the Defaults. The average risk rating of those that meets that premium payment seems slightly higher than those that fails to make the payment. The insight is a bit strange as one had expected the impact to be the other way round

Rate of Default Vs Number of late payment

```
ggplot(treated_dip) +
 aes(x = late.pmt.type, fill = Default) +
 geom_bar(position = "fill") +
 scale_fill_hue() +
 labs(x = " Number of late payment ", y = "Frequency", subtitle = "Rate of
Default Vs Number of late payment") +
 theme_minimal()
```

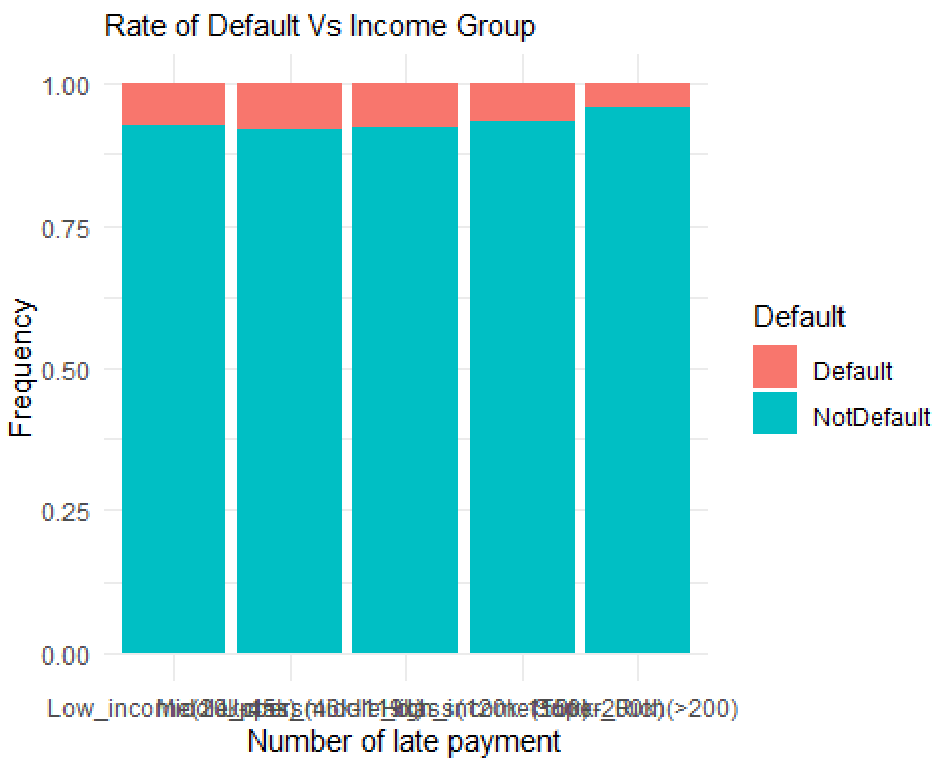


Observations;

- More than 50% of the policyholders that has a record of more than 5 late payment default in the premium payment.
- It is also observed that most of the policyholders that do not have any record of late payment hardly miss their payment.
- It is very believed that the rate of default increases as the number of late payment increases

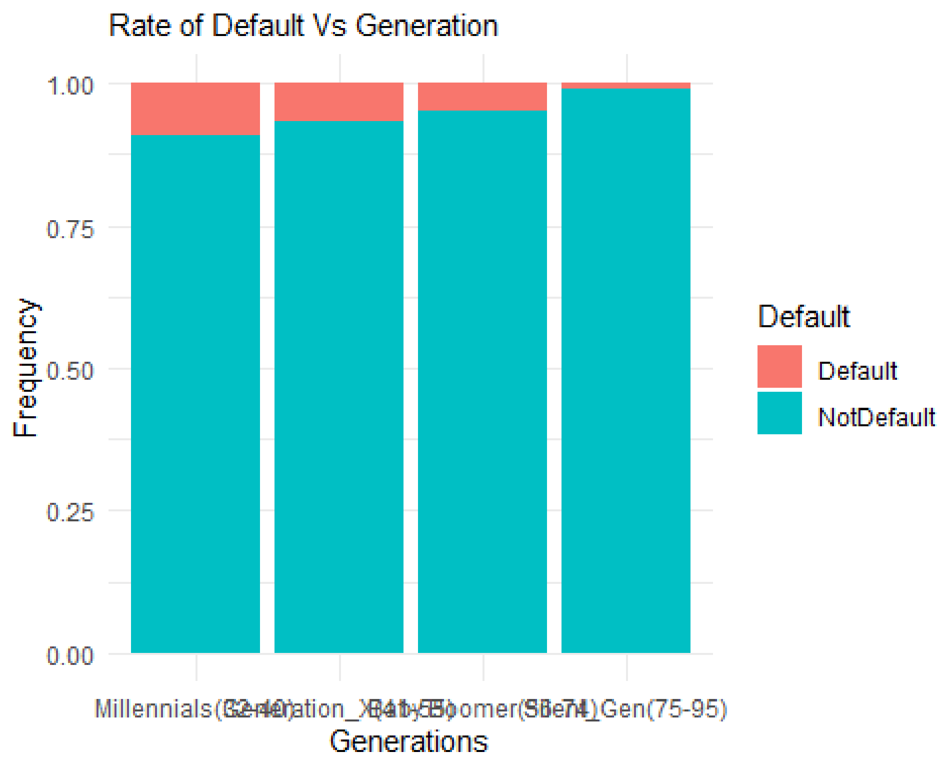
Rate of Default Vs Rate of Default Vs Income Group

```
ggplot(treated_dip) +
 aes(x = Income_Group, fill = Default) +
 geom_bar(position = "fill") +
 scale_fill_hue() +
 labs(x = " Number of late payment ", y = "Frequency", subtitle = "Rate of
Default Vs Income Group") +
 theme_minimal()
```



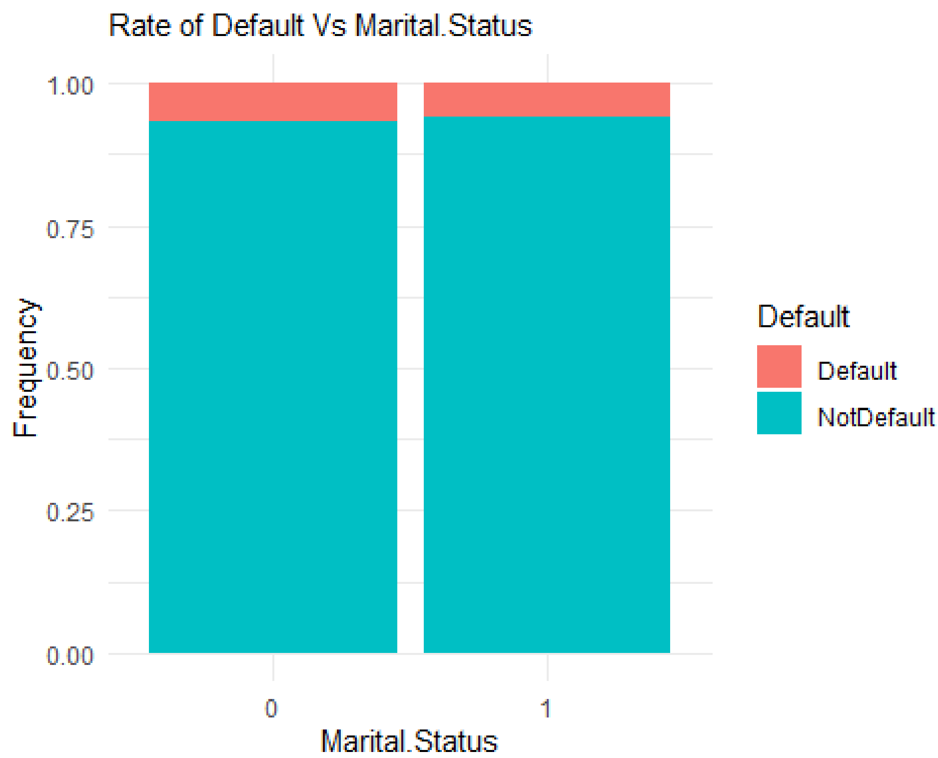
Rate of Default Vs Generation

```
ggplot(treated_dip) +
 aes(x = Generation, fill = Default) +
 geom_bar(position = "fill") +
 scale_fill_hue() +
 labs(x = " Generations ", y = "Frequency", subtitle = "Rate of Default Vs
Generation") +
 theme_minimal()
```



#Rate of Default Vs Marital.Status

```
ggplot(treated_dip) +
 aes(x = Marital.Status, fill = Default) +
 geom_bar(position = "fill") +
 scale_fill_hue() +
 labs(x = " Marital.Status ", y = "Frequency", subtitle = "Rate of Default Vs
Marital.Status") +
 theme_minimal()
```

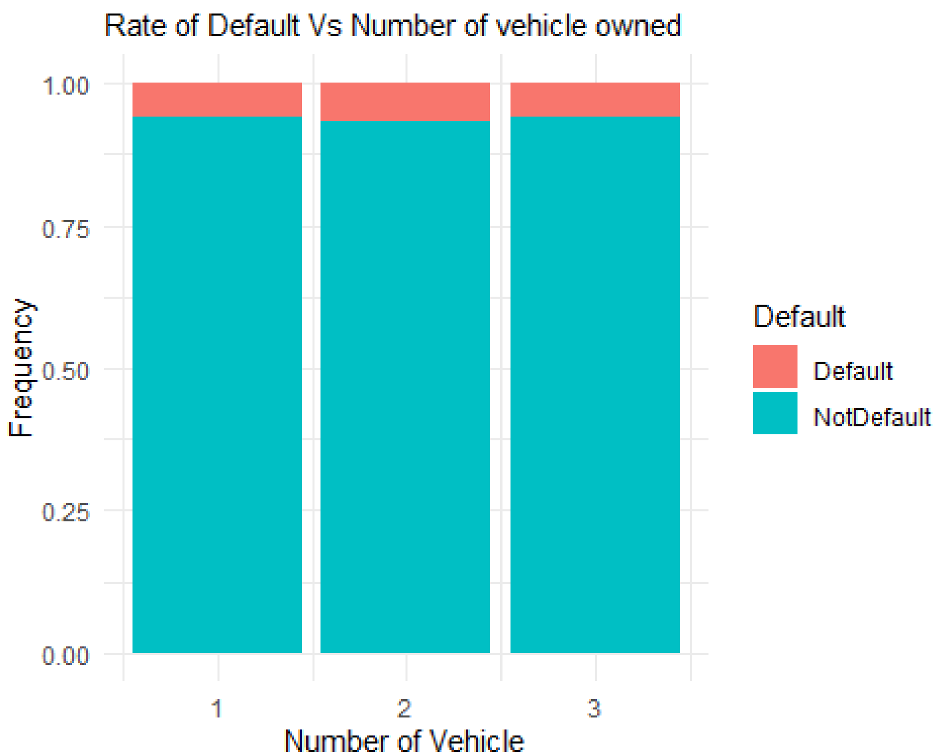


Observations:

- The Default of the insurance policy does not seem to be dependent on the marital status of the policyholder
- The test statisitic also confirms this as p-value is more than 0.05.

Rate of Default Vs Number of vehicle owned

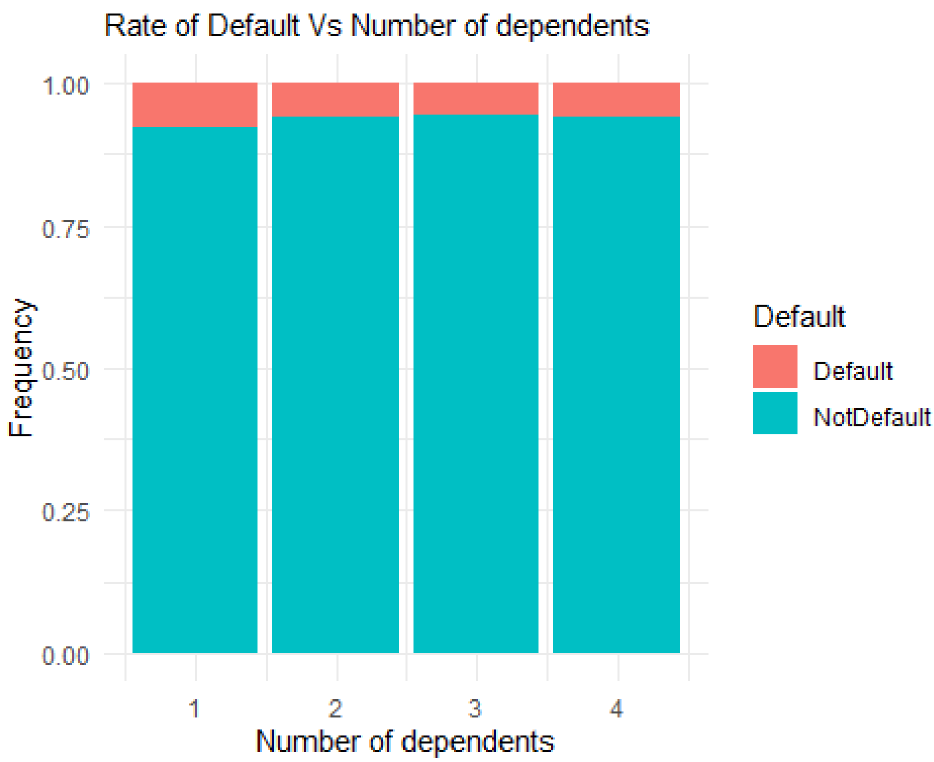
```
ggplot(treated_dip) +
 aes(x = Vehicle, fill = Default) +
 geom_bar(position = "fill") +
 scale_fill_hue() +
 labs(x = " Number of Vehicle ", y = "Frequency", subtitle = "Rate of Default
Vs Number of vehicle owned") +
 theme_minimal()
```



Observations; * The Default of the insurance policy does not seem to be dependent on the number of vehcles owned by the policyholder. * The test statisitic also confirms this as p-value is more than 0.05.

Rate of Default Vs Number of dependents

```
ggplot(treated_dip) +
 aes(x = Dependents, fill = Default) +
 geom_bar(position = "fill") +
 scale_fill_hue() +
 labs(x = " Number of dependents ", y = "Frequency", subtitle = "Rate of
Default Vs Number of dependents") +
 theme_minimal()
```



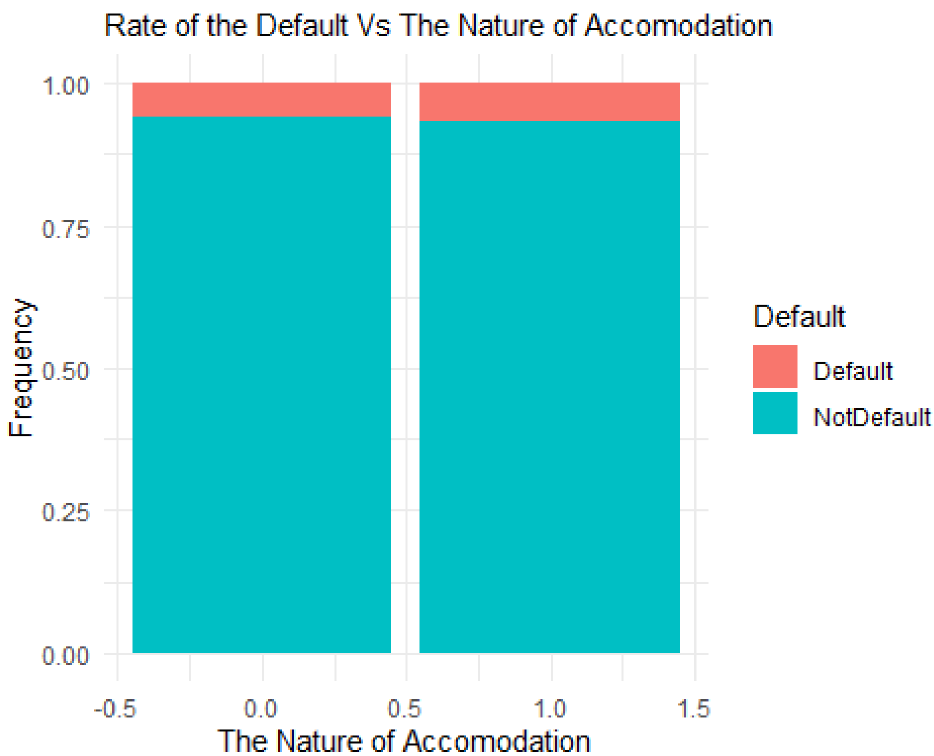

```
chisq.test(treated_dip$Default, treated_dip$Dependents)

##
##  Pearson's Chi-squared test
##
## data:  treated_dip$Default and treated_dip$Dependents
## X-squared = 4.4977, df = 3, p-value = 0.2125
```

Observations;

- The Default of the insurance policy does not seem to be dependent on the number of dependent on thepolicyholder.
- The test statisitic also confirms this as p-value is more than 0.05.

Rate of the Default Vs The Nature of Accomodation

```
ggplot(treated_dip) +
 aes(x = Accomodation, fill = Default) +
 geom_bar(position = "fill") +
 scale_fill_hue() +
 labs(x = " The Nature of Accomodation ", y = "Frequency", subtitle = "Rate
of the Default Vs The Nature of Accomodation") +
 theme_minimal()
```

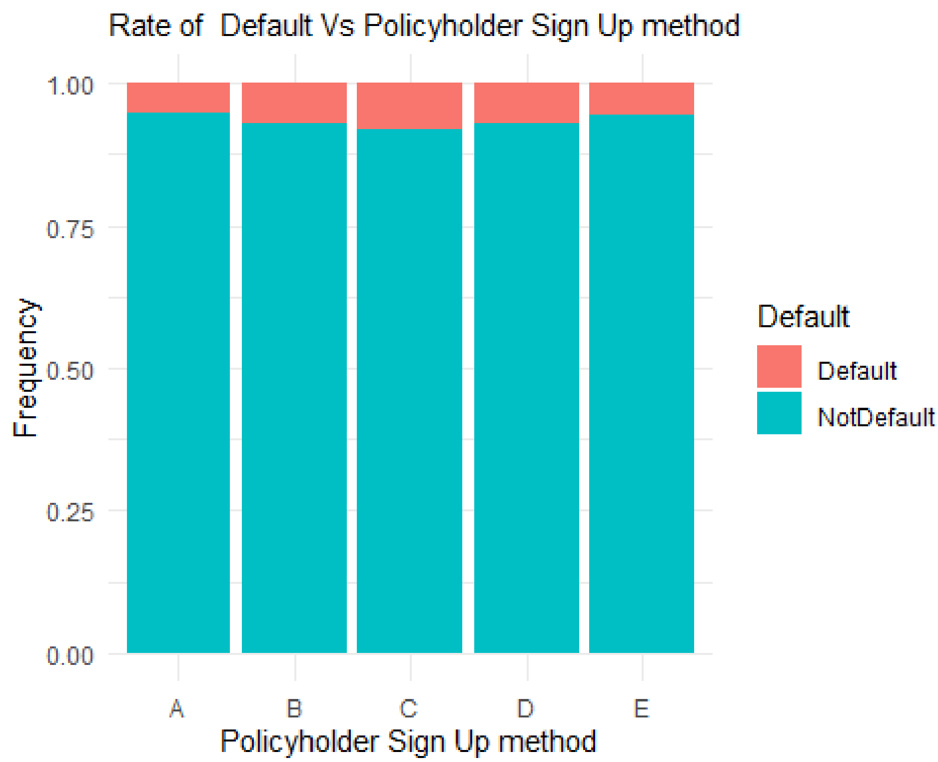


Observations;

- The Default of the insurance policy does not depend on whether the policyholder resides in a owned or rented appartment.
- The test statistic also confirms this as p-value is more than 0.05.

Rate of Default Vs Policyholder Sign Up method

```
ggplot(treated_dip) +
 aes(x = Sources, fill = Default) +
 geom_bar(position = "fill") +
 scale_fill_hue() +
 labs(x = " Policyholder Sign Up method ", y = "Frequency", subtitle = "Rate
of  Default Vs Policyholder Sign Up method") +
 theme_minimal()
```

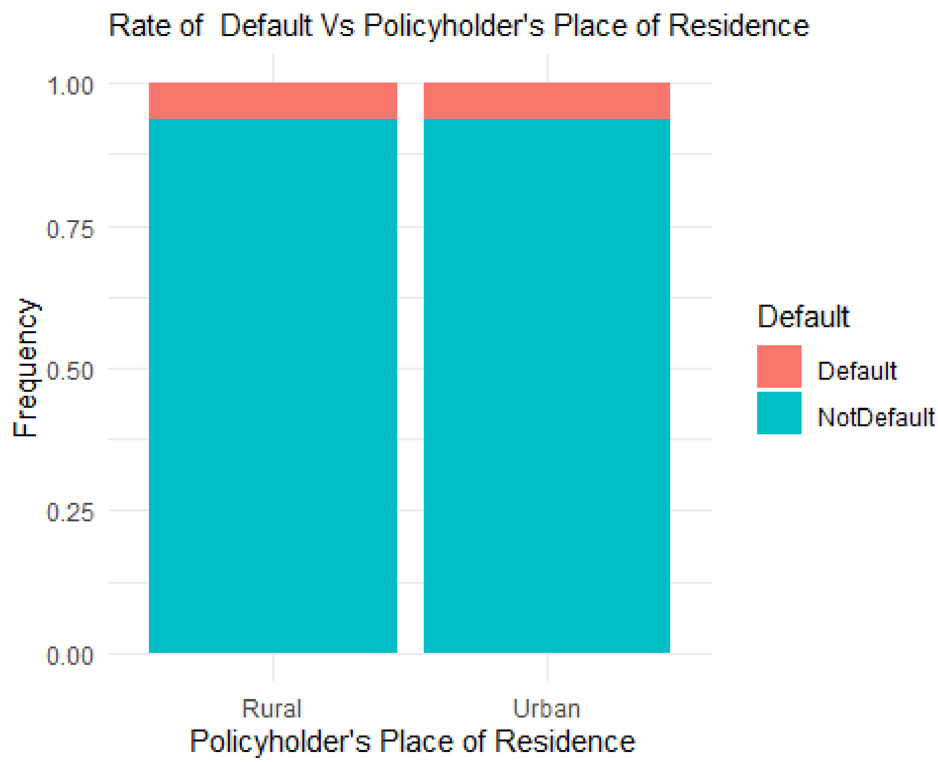


Observation:

- The method through which policyholders are sourced does not have any sigificant effect on whether the policy will be renewed or not as shown in the bar plot. The test statisitic also confirms this as p-value is more than 0.05.

Rate of Default Vs Policyholder's Place of Residence

```
ggplot(treated_dip) +
 aes(x = Residence, fill = Default) +
 geom_bar(position = "fill") +
 scale_fill_hue() +
 labs(x = " Policyholder's Place of Residence", y = "Frequency", subtitle =
"Rate of  Default Vs Policyholder's Place of Residence") +
 theme_minimal()
```



Observations;

- The tendency to renew the policy does not depend on whether the policy holder resides in urban or rural as shown in the bar plot. The test statistic also confirms the barplot

Split the 10% data-subset into Train & Test (70-30 split)

```
set.seed(1234)

trainIndex <- createDataPartition(db$Default, p = .70, list = FALSE)

db_Train <- db[ trainIndex,]
db_Test  <- db[-trainIndex,]

prop.table(table(db_Train$Default))*100

##
##    Default NotDefault
##   6.258941  93.741059

prop.table(table(db_Test$Default))*100

##
##    Default NotDefault
##   6.265664  93.734336
```

Setting up the general parameters for training multiple models

```
fitControl <- trainControl(
              method = 'repeatedcv',          # k-fold cross validation
              number = 3,                    # number of folds or k
              repeats = 1,                    # repeated k-fold cross-
validation
              allowParallel = TRUE,
              classProbs = TRUE,
              sampling = "up",
              summaryFunction=twoClassSummary# should class probabilities be
returned
    )
```

Model _1 : GLM : Simple Logistic Regression Model

```
lr_model <- train(Default ~ ., data = db_Train,
                  method = "glm",
                  family = "binomial",
                  preProcess = c("scale"),
                  trControl = fitControl)

## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy"
was not
## in the result set. ROC will be used instead.

summary(lr_model)

##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2603  -0.8391   0.0009   0.7883   4.0110
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.020930   4.686003  -0.858 0.390853
## cash.credit    -0.715832   0.038646 -18.523  < 2e-16 ***
## Age             0.154219   0.037586   4.103 4.08e-05 ***
## Income          0.243154   0.076728   3.169 0.001529 **
## Marital.Status  0.006040   0.034086   0.177 0.859346
## Vehicle        -0.014730   0.034056  -0.433 0.665355
## Dependents      0.083472   0.034411   2.426 0.015279 *
## Accomodation   -0.074730   0.033889  -2.205 0.027443 *
## risk_score      0.040420   0.036073   1.121 0.262491
## No_premium     -0.231805   0.040190  -5.768 8.04e-09 ***
## SourcesB       -0.175870   0.035994  -4.886 1.03e-06 ***
## SourcesC       -0.022638   0.036495  -0.620 0.535065
## SourcesD        0.137230   0.037241   3.685 0.000229 ***
```

```
## SourcesE       -0.029237   0.031150  -0.939 0.347939
## ResidenceUrban -0.004624   0.034329  -0.135 0.892858
## premium        -0.035065   0.055320  -0.634 0.526181
## late.pmt       -1.511998   0.070488 -21.450  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7267.0  on 5241  degrees of freedom
## Residual deviance: 5317.5  on 5225  degrees of freedom
## AIC: 5351.5
##
## Number of Fisher Scoring iterations: 5
```

```
varImp(lr_model)
```

```
## glm variable importance
##
##                 Overall
## late.pmt       100.0000
## cash.credit     86.2653
## No_premium      26.4265
## SourcesB        22.2908
## Age             18.6175
## SourcesD        16.6552
## Income          14.2353
## Dependents      10.7480
## Accomodation     9.7133
## risk_score       4.6249
## SourcesE         3.7714
## premium          2.3417
## SourcesC         2.2781
## Vehicle          1.3973
## Marital.Status   0.1995
## ResidenceUrban   0.0000
```

Predict using the trained model & check performance on test set

```
lr_predictions_test <- predict(lr_model, newdata = db_Train, type = "raw")
confusionMatrix(lr_predictions_test, db_Train$Default)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Default NotDefault
##   Default        127        546
##   NotDefault      48       2075
##
##                Accuracy : 0.7876
```

```
##                  95% CI : (0.7719, 0.8026)
##     No Information Rate : 0.9374
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2223
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.72571
##             Specificity : 0.79168
##          Pos Pred Value : 0.18871
##          Neg Pred Value : 0.97739
##              Prevalence : 0.06259
##          Detection Rate : 0.04542
##    Detection Prevalence : 0.24070
##       Balanced Accuracy : 0.75870
##
##        'Positive' Class : Default
##
```

```
lr_predictions_test <- predict(lr_model, newdata = db_Train, type = "raw")
confusionMatrix(lr_predictions_test, db_Train$Default)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Default NotDefault
##   Default        127        546
##   NotDefault      48       2075
##
##                Accuracy : 0.7876
##                  95% CI : (0.7719, 0.8026)
##     No Information Rate : 0.9374
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2223
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.72571
##             Specificity : 0.79168
##          Pos Pred Value : 0.18871
##          Neg Pred Value : 0.97739
##              Prevalence : 0.06259
##          Detection Rate : 0.04542
##    Detection Prevalence : 0.24070
##       Balanced Accuracy : 0.75870
##
##        'Positive' Class : Default
##
```

```
# se"N"sitivity : True "P"ositive rate
# s"P"ecificity : True "N"egative rate
```

Model _2 : Step-Wise AIC

```
lr_stepAIC_model <- train(Default ~ ., data = db_Train,
                  method = "glmStepAIC",
                  family = "binomial",
                  preProcess = c("scale"),
                  trControl = fitControl)

## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy"
was not
## in the result set. ROC will be used instead.

## Start:  AIC=3490.53
## .outcome ~ cash.credit + Age + Income + Marital.Status + Vehicle +
##     Dependents + Accomodation + risk_score + No_premium + SourcesB +
##     SourcesC + SourcesD + SourcesE + ResidenceUrban + premium +
##     late.pmt
##
##                  Df Deviance    AIC
## - Marital.Status  1   3456.7 3488.7
## - SourcesC        1   3456.9 3488.9
## - Vehicle         1   3457.0 3489.0
## - SourcesD        1   3457.3 3489.3
## - risk_score      1   3457.6 3489.6
## - premium         1   3457.6 3489.6
## - SourcesE        1   3457.7 3489.7
## <none>                3456.5 3490.5
## - Dependents      1   3458.9 3490.9
## - SourcesB        1   3459.4 3491.4
## - Accomodation    1   3460.6 3492.6
## - Age             1   3463.4 3495.4
## - Income          1   3464.9 3496.9
## - ResidenceUrban  1   3465.6 3497.6
## - No_premium      1   3505.8 3537.8
## - cash.credit     1   3725.1 3757.1
## - late.pmt        1   3999.7 4031.7
##
## Step:  AIC=3488.73
## .outcome ~ cash.credit + Age + Income + Vehicle + Dependents +
##     Accomodation + risk_score + No_premium + SourcesB + SourcesC +
##     SourcesD + SourcesE + ResidenceUrban + premium + late.pmt
##
##                  Df Deviance    AIC
## - SourcesC        1   3457.1 3487.1
## - Vehicle         1   3457.2 3487.2
## - SourcesD        1   3457.5 3487.5
## - risk_score      1   3457.8 3487.8
```

```
## - premium         1   3457.8 3487.8
## - SourcesE        1   3457.9 3487.9
## <none>                3456.7 3488.7
## - Dependents      1   3459.1 3489.1
## - SourcesB        1   3459.7 3489.7
## - Accomodation    1   3460.8 3490.8
## - Age             1   3463.6 3493.6
## - Income          1   3465.1 3495.1
## - ResidenceUrban  1   3465.7 3495.7
## - No_premium      1   3506.1 3536.1
## - cash.credit     1   3728.9 3758.9
## - late.pmt        1   4000.2 4030.2
##
## Step:  AIC=3487.11
## .outcome ~ cash.credit + Age + Income + Vehicle + Dependents +
##     Accomodation + risk_score + No_premium + SourcesB + SourcesD +
##     SourcesE + ResidenceUrban + premium + late.pmt
##
##                  Df Deviance    AIC
## - Vehicle         1   3457.6 3485.6
## - risk_score      1   3458.1 3486.1
## - SourcesE        1   3458.2 3486.2
## - SourcesD        1   3458.2 3486.2
## - premium         1   3458.2 3486.2
## <none>                3457.1 3487.1
## - Dependents      1   3459.6 3487.6
## - SourcesB        1   3459.7 3487.7
## - Accomodation    1   3461.3 3489.3
## - Income          1   3465.2 3493.2
## - Age             1   3465.2 3493.2
## - ResidenceUrban  1   3466.2 3494.2
## - No_premium      1   3506.7 3534.7
## - cash.credit     1   3729.3 3757.3
## - late.pmt        1   4004.0 4032.0
##
## Step:  AIC=3485.59
## .outcome ~ cash.credit + Age + Income + Dependents + Accomodation +
##     risk_score + No_premium + SourcesB + SourcesD + SourcesE +
##     ResidenceUrban + premium + late.pmt
##
##                  Df Deviance    AIC
## - risk_score      1   3458.5 3484.5
## - SourcesE        1   3458.7 3484.7
## - premium         1   3458.7 3484.7
## - SourcesD        1   3458.7 3484.7
## <none>                3457.6 3485.6
## - Dependents      1   3460.1 3486.1
## - SourcesB        1   3460.1 3486.1
## - Accomodation    1   3461.7 3487.7
## - Income          1   3465.7 3491.7
```

```
## - Age             1   3466.0 3492.0
## - ResidenceUrban  1   3467.1 3493.1
## - No_premium      1   3507.0 3533.0
## - cash.credit     1   3731.1 3757.1
## - late.pmt        1   4008.6 4034.6
##
## Step:  AIC=3484.53
## .outcome ~ cash.credit + Age + Income + Dependents + Accomodation +
##     No_premium + SourcesB + SourcesD + SourcesE + ResidenceUrban +
##     premium + late.pmt
##
##                  Df Deviance    AIC
## - SourcesE        1   3459.6 3483.6
## - SourcesD        1   3459.7 3483.7
## - premium         1   3459.8 3483.8
## <none>                3458.5 3484.5
## - Dependents      1   3460.8 3484.8
## - SourcesB        1   3460.9 3484.9
## - Accomodation    1   3462.3 3486.3
## - Income          1   3465.9 3489.9
## - Age             1   3467.2 3491.2
## - ResidenceUrban  1   3467.7 3491.7
## - No_premium      1   3508.5 3532.5
## - cash.credit     1   3740.4 3764.4
## - late.pmt        1   4008.6 4032.6
##
## Step:  AIC=3483.57
## .outcome ~ cash.credit + Age + Income + Dependents + Accomodation +
##     No_premium + SourcesB + SourcesD + ResidenceUrban + premium +
##     late.pmt
##
##                  Df Deviance    AIC
## - premium         1   3460.6 3482.6
## - SourcesD        1   3460.9 3482.9
## <none>                3459.6 3483.6
## - SourcesB        1   3461.7 3483.7
## - Dependents      1   3462.0 3484.0
## - Accomodation    1   3463.0 3485.0
## - Income          1   3467.1 3489.1
## - Age             1   3468.3 3490.3
## - ResidenceUrban  1   3468.9 3490.9
## - No_premium      1   3508.8 3530.8
## - cash.credit     1   3741.1 3763.1
## - late.pmt        1   4010.8 4032.8
##
## Step:  AIC=3482.6
## .outcome ~ cash.credit + Age + Income + Dependents + Accomodation +
##     No_premium + SourcesB + SourcesD + ResidenceUrban + late.pmt
##
##                  Df Deviance    AIC
```

```
## - SourcesD        1   3462.0 3482.0
## <none>                3460.6 3482.6
## - SourcesB        1   3462.8 3482.8
## - Dependents      1   3463.0 3483.0
## - Accomodation    1   3464.0 3484.0
## - Age             1   3469.2 3489.2
## - ResidenceUrban  1   3471.1 3491.1
## - Income          1   3488.8 3508.8
## - No_premium      1   3510.6 3530.6
## - cash.credit     1   3745.3 3765.3
## - late.pmt        1   4011.6 4031.6
##
## Step:  AIC=3481.95
## .outcome ~ cash.credit + Age + Income + Dependents + Accomodation +
##     No_premium + SourcesB + ResidenceUrban + late.pmt
##
##                  Df Deviance    AIC
## <none>                3462.0 3482.0
## - Dependents      1   3464.3 3482.3
## - SourcesB        1   3464.8 3482.8
## - Accomodation    1   3465.4 3483.4
## - Age             1   3469.8 3487.8
## - ResidenceUrban  1   3472.4 3490.4
## - Income          1   3493.8 3511.8
## - No_premium      1   3511.2 3529.2
## - cash.credit     1   3745.4 3763.4
## - late.pmt        1   4012.8 4030.8
## Start:  AIC=3635.47
## .outcome ~ cash.credit + Age + Income + Marital.Status + Vehicle +
##     Dependents + Accomodation + risk_score + No_premium + SourcesB +
##     SourcesC + SourcesD + SourcesE + ResidenceUrban + premium +
##     late.pmt
##
##                  Df Deviance    AIC
## - SourcesC        1   3601.5 3633.5
## - No_premium      1   3601.6 3633.6
## - premium         1   3601.7 3633.7
## - Income          1   3601.9 3633.9
## - ResidenceUrban  1   3602.3 3634.3
## - Accomodation    1   3602.5 3634.5
## <none>                3601.5 3635.5
## - Vehicle         1   3604.7 3636.7
## - Marital.Status  1   3606.6 3638.6
## - Age             1   3611.0 3643.0
## - Dependents      1   3612.2 3644.2
## - risk_score      1   3616.9 3648.9
## - SourcesE        1   3617.1 3649.1
## - SourcesB        1   3626.5 3658.5
## - SourcesD        1   3632.9 3664.9
## - cash.credit     1   3794.1 3826.1
```

```
## - late.pmt        1   4005.4 4037.4
##
## Step:  AIC=3633.49
## .outcome ~ cash.credit + Age + Income + Marital.Status + Vehicle +
##     Dependents + Accomodation + risk_score + No_premium + SourcesB +
##     SourcesD + SourcesE + ResidenceUrban + premium + late.pmt
##
##                  Df Deviance    AIC
## - No_premium      1   3601.6 3631.6
## - premium         1   3601.7 3631.7
## - Income          1   3601.9 3631.9
## - ResidenceUrban  1   3602.3 3632.3
## - Accomodation    1   3602.5 3632.5
## <none>                3601.5 3633.5
## - Vehicle         1   3604.7 3634.7
## - Marital.Status  1   3606.6 3636.6
## - Age             1   3611.1 3641.1
## - Dependents      1   3612.2 3642.2
## - risk_score      1   3617.0 3647.0
## - SourcesE        1   3617.2 3647.2
## - SourcesB        1   3630.0 3660.0
## - SourcesD        1   3633.8 3663.8
## - cash.credit     1   3797.1 3827.1
## - late.pmt        1   4006.1 4036.1
##
## Step:  AIC=3631.61
## .outcome ~ cash.credit + Age + Income + Marital.Status + Vehicle +
##     Dependents + Accomodation + risk_score + SourcesB + SourcesD +
##     SourcesE + ResidenceUrban + premium + late.pmt
##
##                  Df Deviance    AIC
## - premium         1   3601.9 3629.9
## - Income          1   3602.0 3630.0
## - ResidenceUrban  1   3602.4 3630.4
## - Accomodation    1   3602.7 3630.7
## <none>                3601.6 3631.6
## - Vehicle         1   3604.8 3632.8
## - Marital.Status  1   3606.8 3634.8
## - Age             1   3611.1 3639.1
## - Dependents      1   3612.4 3640.4
## - SourcesE        1   3617.3 3645.3
## - risk_score      1   3620.2 3648.2
## - SourcesB        1   3630.2 3658.2
## - SourcesD        1   3634.1 3662.1
## - cash.credit     1   3810.3 3838.3
## - late.pmt        1   4008.5 4036.5
##
## Step:  AIC=3629.85
## .outcome ~ cash.credit + Age + Income + Marital.Status + Vehicle +
##     Dependents + Accomodation + risk_score + SourcesB + SourcesD +
```

```
##     SourcesE + ResidenceUrban + late.pmt
##
##                   Df Deviance    AIC
## - ResidenceUrban  1   3602.8 3628.8
## - Accomodation    1   3602.9 3628.9
## <none>                3601.9 3629.9
## - Income          1   3604.4 3630.4
## - Vehicle         1   3605.1 3631.1
## - Marital.Status  1   3607.0 3633.0
## - Age             1   3611.2 3637.2
## - Dependents      1   3612.6 3638.6
## - SourcesE        1   3617.6 3643.6
## - risk_score      1   3620.4 3646.4
## - SourcesB        1   3630.2 3656.2
## - SourcesD        1   3634.4 3660.4
## - cash.credit     1   3811.0 3837.0
## - late.pmt        1   4008.5 4034.5
##
## Step:  AIC=3628.78
## .outcome ~ cash.credit + Age + Income + Marital.Status + Vehicle +
##     Dependents + Accomodation + risk_score + SourcesB + SourcesD +
##     SourcesE + late.pmt
##
##                   Df Deviance    AIC
## - Accomodation    1   3603.6 3627.6
## <none>                3602.8 3628.8
## - Income          1   3605.1 3629.1
## - Vehicle         1   3605.8 3629.8
## - Marital.Status  1   3608.0 3632.0
## - Age             1   3612.3 3636.3
## - Dependents      1   3613.6 3637.6
## - SourcesE        1   3618.5 3642.5
## - risk_score      1   3621.2 3645.2
## - SourcesB        1   3630.8 3654.8
## - SourcesD        1   3635.6 3659.6
## - cash.credit     1   3813.8 3837.8
## - late.pmt        1   4008.6 4032.6
##
## Step:  AIC=3627.65
## .outcome ~ cash.credit + Age + Income + Marital.Status + Vehicle +
##     Dependents + risk_score + SourcesB + SourcesD + SourcesE +
##     late.pmt
##
##                   Df Deviance    AIC
## <none>                3603.6 3627.6
## - Income          1   3606.0 3628.0
## - Vehicle         1   3606.4 3628.4
## - Marital.Status  1   3609.0 3631.0
## - Age             1   3613.2 3635.2
## - Dependents      1   3614.4 3636.4
```

```
## - SourcesE        1   3619.3 3641.3
## - risk_score      1   3621.7 3643.7
## - SourcesB        1   3631.8 3653.8
## - SourcesD        1   3637.0 3659.0
## - cash.credit     1   3813.8 3835.8
## - late.pmt        1   4012.9 4034.9
## Start:  AIC=3448.76
## .outcome ~ cash.credit + Age + Income + Marital.Status + Vehicle +
##     Dependents + Accomodation + risk_score + No_premium + SourcesB +
##     SourcesC + SourcesD + SourcesE + ResidenceUrban + premium +
##     late.pmt
##
##                  Df Deviance    AIC
## - SourcesE        1   3414.9 3446.9
## - Vehicle         1   3415.4 3447.4
## - Marital.Status  1   3415.5 3447.5
## - risk_score      1   3415.9 3447.9
## - SourcesB        1   3416.5 3448.5
## <none>                3414.8 3448.8
## - ResidenceUrban  1   3417.2 3449.2
## - Dependents      1   3418.3 3450.3
## - Accomodation    1   3418.5 3450.5
## - premium         1   3418.8 3450.8
## - SourcesC        1   3420.4 3452.4
## - Income          1   3425.7 3457.7
## - SourcesD        1   3428.8 3460.8
## - Age             1   3430.3 3462.3
## - No_premium      1   3433.1 3465.1
## - cash.credit     1   3659.2 3691.2
## - late.pmt        1   4028.4 4060.4
##
## Step:  AIC=3446.92
## .outcome ~ cash.credit + Age + Income + Marital.Status + Vehicle +
##     Dependents + Accomodation + risk_score + No_premium + SourcesB +
##     SourcesC + SourcesD + ResidenceUrban + premium + late.pmt
##
##                  Df Deviance    AIC
## - Vehicle         1   3415.5 3445.5
## - Marital.Status  1   3415.7 3445.7
## - risk_score      1   3416.0 3446.0
## - SourcesB        1   3416.6 3446.6
## <none>                3414.9 3446.9
## - ResidenceUrban  1   3417.3 3447.3
## - Dependents      1   3418.5 3448.5
## - Accomodation    1   3418.5 3448.5
## - premium         1   3419.1 3449.1
## - SourcesC        1   3420.4 3450.4
## - Income          1   3425.9 3455.9
## - SourcesD        1   3429.2 3459.2
## - Age             1   3430.4 3460.4
```

```
## - No_premium      1   3433.1 3463.1
## - cash.credit     1   3659.6 3689.6
## - late.pmt        1   4029.5 4059.5
##
## Step:  AIC=3445.49
## .outcome ~ cash.credit + Age + Income + Marital.Status + Dependents +
##     Accomodation + risk_score + No_premium + SourcesB + SourcesC +
##     SourcesD + ResidenceUrban + premium + late.pmt
##
##                  Df Deviance    AIC
## - Marital.Status  1   3416.2 3444.2
## - risk_score      1   3416.7 3444.7
## - SourcesB        1   3417.2 3445.2
## <none>                3415.5 3445.5
## - ResidenceUrban  1   3417.9 3445.9
## - Accomodation    1   3419.0 3447.0
## - Dependents      1   3419.1 3447.1
## - premium         1   3419.6 3447.6
## - SourcesC        1   3420.9 3448.9
## - Income          1   3426.3 3454.3
## - SourcesD        1   3429.7 3457.7
## - Age             1   3430.9 3458.9
## - No_premium      1   3433.7 3461.7
## - cash.credit     1   3662.8 3690.8
## - late.pmt        1   4030.5 4058.5
##
## Step:  AIC=3444.22
## .outcome ~ cash.credit + Age + Income + Dependents + Accomodation +
##     risk_score + No_premium + SourcesB + SourcesC + SourcesD +
##     ResidenceUrban + premium + late.pmt
##
##                  Df Deviance    AIC
## - risk_score      1   3417.5 3443.5
## - SourcesB        1   3417.9 3443.9
## <none>                3416.2 3444.2
## - ResidenceUrban  1   3418.6 3444.6
## - Dependents      1   3419.6 3445.6
## - Accomodation    1   3419.7 3445.7
## - premium         1   3420.4 3446.4
## - SourcesC        1   3421.4 3447.4
## - Income          1   3427.1 3453.1
## - SourcesD        1   3430.2 3456.2
## - Age             1   3431.2 3457.2
## - No_premium      1   3434.2 3460.2
## - cash.credit     1   3662.9 3688.9
## - late.pmt        1   4030.5 4056.5
##
## Step:  AIC=3443.51
## .outcome ~ cash.credit + Age + Income + Dependents + Accomodation +
##     No_premium + SourcesB + SourcesC + SourcesD + ResidenceUrban +
```

```
##     premium + late.pmt
##
##                  Df Deviance    AIC
## - SourcesB        1   3419.1 3443.1
## <none>                3417.5 3443.5
## - ResidenceUrban  1   3420.0 3444.0
## - Dependents      1   3420.8 3444.8
## - Accomodation    1   3420.9 3444.9
## - premium         1   3421.4 3445.4
## - SourcesC        1   3422.3 3446.3
## - Income          1   3427.4 3451.4
## - SourcesD        1   3431.3 3455.3
## - Age             1   3433.1 3457.1
## - No_premium      1   3434.4 3458.4
## - cash.credit     1   3669.7 3693.7
## - late.pmt        1   4030.8 4054.8
##
## Step:  AIC=3443.1
## .outcome ~ cash.credit + Age + Income + Dependents + Accomodation +
##     No_premium + SourcesC + SourcesD + ResidenceUrban + premium +
##     late.pmt
##
##                  Df Deviance    AIC
## <none>                3419.1 3443.1
## - ResidenceUrban  1   3421.9 3443.9
## - Accomodation    1   3422.4 3444.4
## - Dependents      1   3422.5 3444.5
## - SourcesC        1   3422.7 3444.7
## - premium         1   3423.3 3445.3
## - Income          1   3428.8 3450.8
## - Age             1   3435.0 3457.0
## - SourcesD        1   3435.8 3457.8
## - No_premium      1   3437.1 3459.1
## - cash.credit     1   3675.8 3697.8
## - late.pmt        1   4030.9 4052.9
## Start:  AIC=5386.21
## .outcome ~ cash.credit + Age + Income + Marital.Status + Vehicle +
##     Dependents + Accomodation + risk_score + No_premium + SourcesB +
##     SourcesC + SourcesD + SourcesE + ResidenceUrban + premium +
##     late.pmt
##
##                  Df Deviance    AIC
## - ResidenceUrban  1   5352.2 5384.2
## - Vehicle         1   5352.2 5384.2
## - premium         1   5352.5 5384.5
## - Marital.Status  1   5353.2 5385.2
## - SourcesE        1   5353.9 5385.9
## <none>                5352.2 5386.2
## - risk_score      1   5355.3 5387.3
## - Accomodation    1   5355.5 5387.5
```

```
## - Dependents      1   5356.3 5388.3
## - Age             1   5359.2 5391.2
## - SourcesC        1   5359.5 5391.5
## - No_premium      1   5361.5 5393.5
## - Income          1   5361.5 5393.5
## - SourcesD        1   5362.4 5394.4
## - SourcesB        1   5379.8 5411.8
## - cash.credit     1   5675.4 5707.4
## - late.pmt        1   6122.1 6154.1
##
## Step:  AIC=5384.22
## .outcome ~ cash.credit + Age + Income + Marital.Status + Vehicle +
##     Dependents + Accomodation + risk_score + No_premium + SourcesB +
##     SourcesC + SourcesD + SourcesE + premium + late.pmt
##
##                  Df Deviance    AIC
## - Vehicle         1   5352.2 5382.2
## - premium         1   5352.5 5382.5
## - Marital.Status  1   5353.2 5383.2
## - SourcesE        1   5353.9 5383.9
## <none>                5352.2 5384.2
## - risk_score      1   5355.3 5385.3
## - Accomodation    1   5355.5 5385.5
## - Dependents      1   5356.3 5386.3
## - Age             1   5359.2 5389.2
## - SourcesC        1   5359.5 5389.5
## - No_premium      1   5361.5 5391.5
## - Income          1   5361.7 5391.7
## - SourcesD        1   5362.4 5392.4
## - SourcesB        1   5379.8 5409.8
## - cash.credit     1   5676.2 5706.2
## - late.pmt        1   6124.5 6154.5
##
## Step:  AIC=5382.23
## .outcome ~ cash.credit + Age + Income + Marital.Status + Dependents +
##     Accomodation + risk_score + No_premium + SourcesB + SourcesC +
##     SourcesD + SourcesE + premium + late.pmt
##
##                  Df Deviance    AIC
## - premium         1   5352.5 5380.5
## - Marital.Status  1   5353.2 5381.2
## - SourcesE        1   5353.9 5381.9
## <none>                5352.2 5382.2
## - risk_score      1   5355.3 5383.3
## - Accomodation    1   5355.5 5383.5
## - Dependents      1   5356.4 5384.4
## - Age             1   5359.2 5387.2
## - SourcesC        1   5359.5 5387.5
## - No_premium      1   5361.5 5389.5
## - Income          1   5361.7 5389.7
```

```
## - SourcesD        1   5362.4 5390.4
## - SourcesB        1   5380.0 5408.0
## - cash.credit     1   5676.3 5704.3
## - late.pmt        1   6124.9 6152.9
##
## Step:  AIC=5380.52
## .outcome ~ cash.credit + Age + Income + Marital.Status + Dependents +
##     Accomodation + risk_score + No_premium + SourcesB + SourcesC +
##     SourcesD + SourcesE + late.pmt
##
##                  Df Deviance    AIC
## - Marital.Status  1   5353.5 5379.5
## - SourcesE        1   5354.4 5380.4
## <none>                5352.5 5380.5
## - risk_score      1   5355.7 5381.7
## - Accomodation    1   5355.9 5381.9
## - Dependents      1   5356.7 5382.7
## - Age             1   5359.7 5385.7
## - SourcesC        1   5359.8 5385.8
## - No_premium      1   5361.8 5387.8
## - SourcesD        1   5362.7 5388.7
## - Income          1   5367.9 5393.9
## - SourcesB        1   5380.6 5406.6
## - cash.credit     1   5676.4 5702.4
## - late.pmt        1   6126.8 6152.8
##
## Step:  AIC=5379.47
## .outcome ~ cash.credit + Age + Income + Dependents + Accomodation +
##     risk_score + No_premium + SourcesB + SourcesC + SourcesD +
##     SourcesE + late.pmt
##
##                Df Deviance    AIC
## - SourcesE      1   5355.3 5379.3
## <none>              5353.5 5379.5
## - risk_score    1   5356.6 5380.6
## - Accomodation  1   5356.9 5380.9
## - Dependents    1   5357.5 5381.5
## - Age           1   5360.3 5384.3
## - SourcesC      1   5360.6 5384.6
## - No_premium    1   5362.5 5386.5
## - SourcesD      1   5363.4 5387.4
## - Income        1   5368.7 5392.7
## - SourcesB      1   5381.2 5405.2
## - cash.credit   1   5676.7 5700.7
## - late.pmt      1   6127.2 6151.2
##
## Step:  AIC=5379.34
## .outcome ~ cash.credit + Age + Income + Dependents + Accomodation +
##     risk_score + No_premium + SourcesB + SourcesC + SourcesD +
##     late.pmt
```

```
##
##                 Df Deviance    AIC
## <none>              5355.3 5379.3
## - Accomodation  1   5358.4 5380.4
## - risk_score    1   5358.7 5380.7
## - Dependents    1   5359.7 5381.7
## - SourcesC      1   5361.8 5383.8
## - Age           1   5362.3 5384.3
## - No_premium    1   5363.7 5385.7
## - SourcesD      1   5365.9 5387.9
## - Income        1   5369.7 5391.7
## - SourcesB      1   5381.9 5403.9
## - cash.credit   1   5677.9 5699.9
## - late.pmt      1   6133.7 6155.7
```

```r
summary(lr_stepAIC_model)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1546  -0.8410   0.0123   0.7943   4.0180
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -7.64747    4.79003  -1.597 0.110368
## cash.credit  -0.67319    0.03839 -17.535  < 2e-16 ***
## Age           0.09732    0.03696   2.633 0.008453 **
## Income        0.16616    0.04713   3.526 0.000423 ***
## Dependents    0.07095    0.03416   2.077 0.037815 *
## Accomodation -0.05831    0.03354  -1.738 0.082140 .
## risk_score    0.06687    0.03612   1.851 0.064125 .
## No_premium   -0.11585    0.04000  -2.896 0.003779 **
## SourcesB     -0.18247    0.03544  -5.149 2.62e-07 ***
## SourcesC     -0.09211    0.03616  -2.547 0.010850 *
## SourcesD      0.12148    0.03775   3.218 0.001292 **
## late.pmt     -1.53976    0.07084 -21.736  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7267.0  on 5241  degrees of freedom
## Residual deviance: 5355.3  on 5230  degrees of freedom
## AIC: 5379.3
##
## Number of Fisher Scoring iterations: 5
```

Predict using the trained model & check performance on test set

```
lr_StepAIc_predictions_test <- predict(lr_stepAIC_model, newdata = db_Test,
type = "raw")
confusionMatrix(lr_StepAIc_predictions_test, db_Test$Default)
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction    Default NotDefault
##   Default          45        209
##   NotDefault       30        913
##
##                Accuracy : 0.8003
##                  95% CI : (0.7765, 0.8226)
##     No Information Rate : 0.9373
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1957
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.60000
##             Specificity : 0.81373
##          Pos Pred Value : 0.17717
##          Neg Pred Value : 0.96819
##              Prevalence : 0.06266
##          Detection Rate : 0.03759
##    Detection Prevalence : 0.21220
##       Balanced Accuracy : 0.70686
##
##        'Positive' Class : Default
##
```

```
# se"N"sitivity : True "P"ositive rate
# s"P"ecificity : True "N"egative rate
```

Model_3 : Naive-Bayes

```
nb_model <- train(Default ~ ., data = db_Train,
                 method = "naive_bayes",
                 preProcess = c("center"),
                 trControl = fitControl)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy"
was not
## in the result set. ROC will be used instead.
```

```
summary(nb_model)
```

```
##
## ================================== Naive Bayes
==================================
##
## - Call: naive_bayes.default(x = x, y = y, laplace = param$laplace,
usekernel = FALSE)
## - Laplace: 0
## - Classes: 2
## - Samples: 5242
## - Features: 16
## - Conditional distributions:
##     - Gaussian: 16
## - Prior probabilities:
##     - Default: 0.5
##     - NotDefault: 0.5
##
## ---------------------------------------------------------------------------
-------
```

```
nb_model$finalModel
```

```
##
## ================================== Naive Bayes
==================================
##
##  Call:
## naive_bayes.default(x = x, y = y, laplace = param$laplace, usekernel =
FALSE)
##
## ---------------------------------------------------------------------------
-------
##
## Laplace smoothing: 0
##
## ---------------------------------------------------------------------------
-------
##
##  A priori probabilities:
##
##    Default NotDefault
##        0.5        0.5
##
## ---------------------------------------------------------------------------
-------
##
##  Tables:
##
## ---------------------------------------------------------------------------
-------
##  ::: cash.credit (Gaussian)
```

```
## ---------------------------------------------------------------------------
-------
##
## cash.credit    Default NotDefault
##        mean  0.1637535 -0.1637535
##        sd    0.3622351  0.3279006
##
## ---------------------------------------------------------------------------
-------
##  ::: Age (Gaussian)
## ---------------------------------------------------------------------------
-------
##
## Age      Default NotDefault
##   mean -1031.750   1031.750
##   sd    4768.572   5248.784
##
## ---------------------------------------------------------------------------
-------
##  ::: Income (Gaussian)
## ---------------------------------------------------------------------------
-------
##
## Income   Default NotDefault
##   mean -19796.51   19796.51
##   sd   139118.07  221175.68
##
## ---------------------------------------------------------------------------
-------
##  ::: Marital.Status (Gaussian)
## ---------------------------------------------------------------------------
-------
##
## Marital.Status     Default  NotDefault
##           mean -0.01812285  0.01812285
##           sd    0.49841003  0.50007266
##
## ---------------------------------------------------------------------------
-------
##  ::: Vehicle (Gaussian)
## ---------------------------------------------------------------------------
-------
##
## Vehicle     Default  NotDefault
##    mean  0.01297215 -0.01297215
##    sd    0.80821983  0.81538761
##
## ---------------------------------------------------------------------------
-------
##
```

```
## # ... and 11 more tables
##
## -----------------------------------------------------------------------------
-------
```

Predict using the trained model & check performance on test set

```
nb_predictions_train <- predict(nb_model, newdata = db_Train, type = "raw")
confusionMatrix(nb_predictions_train, db_Train$Default)

## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Default NotDefault
##   Default        117        462
##   NotDefault      58       2159
##
##                Accuracy : 0.814
##                  95% CI : (0.7991, 0.8283)
##     No Information Rate : 0.9374
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.237
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.66857
##             Specificity : 0.82373
##          Pos Pred Value : 0.20207
##          Neg Pred Value : 0.97384
##              Prevalence : 0.06259
##          Detection Rate : 0.04185
##    Detection Prevalence : 0.20708
##       Balanced Accuracy : 0.74615
##
##        'Positive' Class : Default
##

nb_predictions_test <- predict(nb_model, newdata = db_Test, type = "raw")
confusionMatrix(nb_predictions_test, db_Test$Default)

## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Default NotDefault
##   Default         43        184
##   NotDefault      32        938
##
##                Accuracy : 0.8195
##                  95% CI : (0.7966, 0.8409)
```

```
##     No Information Rate : 0.9373
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2104
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.57333
##             Specificity : 0.83601
##          Pos Pred Value : 0.18943
##          Neg Pred Value : 0.96701
##              Prevalence : 0.06266
##          Detection Rate : 0.03592
##    Detection Prevalence : 0.18964
##       Balanced Accuracy : 0.70467
##
##        'Positive' Class : Default
##
```

Model_4 : KNN

```
knn_model <- train(Default ~ ., data = db_Train,
                   preProcess = c("center", "scale"),
                   method = "knn",
                   tuneLength = 49,
                   trControl = fitControl)

## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy"
was not
## in the result set. ROC will be used instead.
```

Predict using the trained model & check performance on test set

```
knn_predictions_test <- predict(knn_model, newdata = db_Train, type = "raw")
confusionMatrix(knn_predictions_test, db_Train$Default)

## Confusion Matrix and Statistics
##
##                Reference
## Prediction    Default NotDefault
##   Default         127        602
##   NotDefault       48       2019
##
##                Accuracy : 0.7675
##                  95% CI : (0.7514, 0.7831)
##     No Information Rate : 0.9374
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2002
```

```
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.72571
##             Specificity : 0.77032
##          Pos Pred Value : 0.17421
##          Neg Pred Value : 0.97678
##              Prevalence : 0.06259
##          Detection Rate : 0.04542
##    Detection Prevalence : 0.26073
##       Balanced Accuracy : 0.74802
##
##        'Positive' Class : Default
##
```

```
knn_predictions_test <- predict(knn_model, newdata = db_Test, type = "raw")
confusionMatrix(knn_predictions_test, db_Test$Default)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Default NotDefault
##   Default         39        265
##   NotDefault      36        857
##
##                Accuracy : 0.7485
##                  95% CI : (0.723, 0.7729)
##     No Information Rate : 0.9373
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1171
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.52000
##             Specificity : 0.76381
##          Pos Pred Value : 0.12829
##          Neg Pred Value : 0.95969
##              Prevalence : 0.06266
##          Detection Rate : 0.03258
##    Detection Prevalence : 0.25397
##       Balanced Accuracy : 0.64191
##
##        'Positive' Class : Default
##
```

Model_5 : Random Forest

```
rf_model <- train(Default ~ ., data = db_Train,
                     method = "rf",
```

```
                     ntree = 30,
                     maxdepth = 5,
                     tuneLength = 10,
                     trControl = fitControl)

## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy"
was not
## in the result set. ROC will be used instead.

rf_model

## Random Forest
##
## 2796 samples
##   13 predictor
##    2 classes: 'Default', 'NotDefault'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold, repeated 1 times)
## Summary of sample sizes: 1864, 1865, 1863
## Addtional sampling using up-sampling
##
## Resampling results across tuning parameters:
##
##   mtry  ROC        Sens        Spec
##    2    0.7717847  0.12010520  0.9835933
##    3    0.7811079  0.09175921  0.9916055
##    5    0.7688504  0.14289889  0.9835942
##    6    0.7663510  0.14299630  0.9832132
##    8    0.7627116  0.18312877  0.9786344
##    9    0.7673637  0.16588740  0.9759634
##   11    0.7642931  0.18283655  0.9782526
##   12    0.7508843  0.19452562  0.9736729
##   14    0.7611661  0.17153711  0.9729092
##   16    0.7499942  0.20027274  0.9664230
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 3.
```

Predict using the trained model & check performance on test set

```
rf_predictions_test <- predict(rf_model, newdata = db_Train, type = "raw")
confusionMatrix(rf_predictions_test, db_Train$Default)

## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Default NotDefault
##   Default        175          0
##   NotDefault       0       2621
```

```
##
##                Accuracy : 1
##                  95% CI : (0.9987, 1)
##     No Information Rate : 0.9374
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.00000
##             Specificity : 1.00000
##          Pos Pred Value : 1.00000
##          Neg Pred Value : 1.00000
##              Prevalence : 0.06259
##          Detection Rate : 0.06259
##    Detection Prevalence : 0.06259
##       Balanced Accuracy : 1.00000
##
##        'Positive' Class : Default
##
```

```
rf_predictions_test <- predict(rf_model, newdata = db_Test, type = "raw")
confusionMatrix(rf_predictions_test, db_Test$Default)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Default NotDefault
##   Default          7          5
##   NotDefault      68       1117
##
##                Accuracy : 0.939
##                  95% CI : (0.9239, 0.9519)
##     No Information Rate : 0.9373
##     P-Value [Acc > NIR] : 0.4357
##
##                   Kappa : 0.1462
##
##  Mcnemar's Test P-Value : 3.971e-13
##
##             Sensitivity : 0.093333
##             Specificity : 0.995544
##          Pos Pred Value : 0.583333
##          Neg Pred Value : 0.942616
##              Prevalence : 0.062657
##          Detection Rate : 0.005848
##    Detection Prevalence : 0.010025
##       Balanced Accuracy : 0.544439
##
```

```
##        'Positive' Class : Default
##
```

Model_6 : Xtreme Gradient boosting Machines

```
cv.ctrl <- trainControl(method = "repeatedcv", repeats = 1,number = 3,
                        summaryFunction = twoClassSummary,
                        classProbs = TRUE,
                        sampling = "up",
                        allowParallel=T)

    xgb.grid <- expand.grid(nrounds = 100,
                            eta = c(0.01),
                            max_depth = c(2,4),
                            gamma = 0,                #default=0
                            colsample_bytree = 1,     #default=1
                            min_child_weight = 1,     #default=1
                            subsample = 1             #default=1
    )

    xgb_model <-train(Default~.,
                     data=db_Train,
                     method="xgbTree",
                     preProcess = c("scale"),
                     trControl=cv.ctrl,
                     tuneGrid=xgb.grid,
                     verbose=T,
                     nthread = 2
        )

## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy"
was not
## in the result set. ROC will be used instead.
```

Predict using the trained model & check performance on test set

```
xgb_predictions_train <- predict(xgb_model, newdata = db_Train, type = "raw")
confusionMatrix(xgb_predictions_train, db_Train$Default)

## Confusion Matrix and Statistics
##
##               Reference
## Prediction    Default NotDefault
##   Default         139        674
##   NotDefault       36       1947
##
##                Accuracy : 0.7461
##                  95% CI : (0.7295, 0.7621)
##     No Information Rate : 0.9374
```

```
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1989
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.79429
##             Specificity : 0.74285
##          Pos Pred Value : 0.17097
##          Neg Pred Value : 0.98185
##              Prevalence : 0.06259
##          Detection Rate : 0.04971
##    Detection Prevalence : 0.29077
##       Balanced Accuracy : 0.76857
##
##        'Positive' Class : Default
##
```

```
xgb_predictions_test <- predict(xgb_model, newdata = db_Test, type = "raw")
confusionMatrix(xgb_predictions_test, db_Test$Default)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Default NotDefault
##   Default         52        269
##   NotDefault      23        853
##
##                Accuracy : 0.7561
##                  95% CI : (0.7307, 0.7802)
##     No Information Rate : 0.9373
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1793
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.69333
##             Specificity : 0.76025
##          Pos Pred Value : 0.16199
##          Neg Pred Value : 0.97374
##              Prevalence : 0.06266
##          Detection Rate : 0.04344
##    Detection Prevalence : 0.26817
##       Balanced Accuracy : 0.72679
##
##        'Positive' Class : Default
##
```

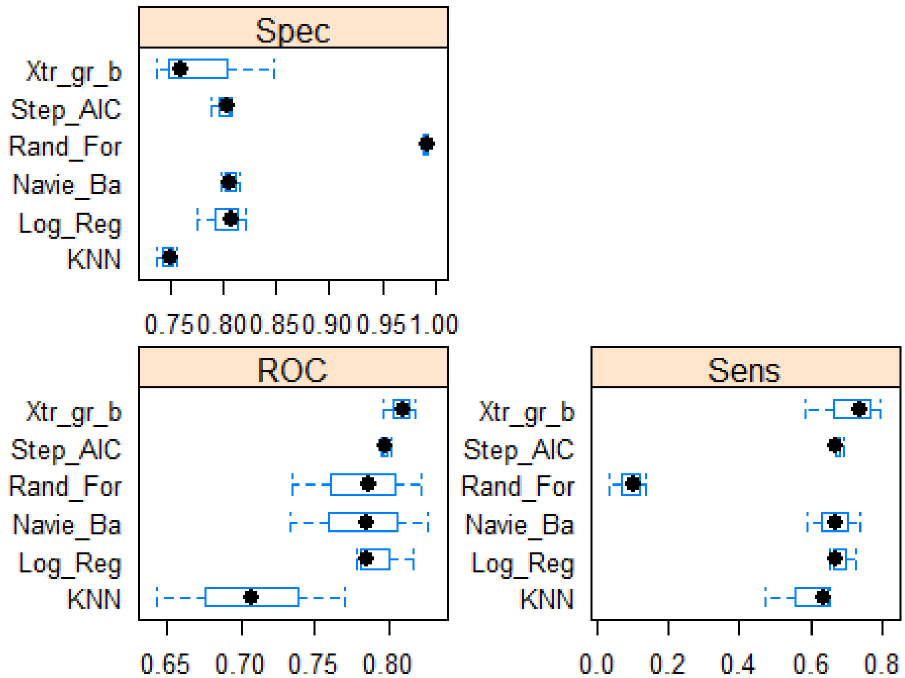
——————————— COMPARING MODELS —————————

```r
# Compare model performances using resample()
models_to_compare <- resamples(list(Log_Reg = lr_model, Step_AIC
=lr_stepAIC_model,
                                    Navie_Ba = nb_model,
                                    KNN = knn_model,
                                    Rand_For = rf_model,
                                    Xtr_gr_b = xgb_model
                                                                  ))

# Summary of the models performances
summary(models_to_compare)

##
## Call:
## summary.resamples(object = models_to_compare)
##
## Models: Log_Reg, Step_AIC, Navie_Ba, KNN, Rand_For, Xtr_gr_b
## Number of resamples: 3
##
## ROC
##                Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## Log_Reg  0.7782293 0.7817658 0.7853024 0.7934432 0.8010501 0.8167979    0
## Step_AIC 0.7950764 0.7963289 0.7975815 0.7979301 0.7993569 0.8011323    0
## Navie_Ba 0.7337255 0.7595529 0.7853803 0.7819019 0.8059901 0.8265999    0
## KNN      0.6427767 0.6750043 0.7072319 0.7065851 0.7384893 0.7697467    0
## Rand_For 0.7345163 0.7607128 0.7869092 0.7811079 0.8044037 0.8218981    0
## Xtr_gr_b 0.7965892 0.8031761 0.8097629 0.8083146 0.8141773 0.8185917    0
##
## Sens
##                 Min.    1st Qu.    Median       Mean   3rd Qu.      Max.
NA's
## Log_Reg  0.65517241 0.66379310 0.6724138 0.68546659 0.7006137 0.7288136
0
## Step_AIC 0.67241379 0.67241379 0.6724138 0.67991428 0.6836645 0.6949153
0
## Navie_Ba 0.59322034 0.63281707 0.6724138 0.66900448 0.7068966 0.7413793
0
## KNN      0.47457627 0.55625365 0.6379310 0.58922657 0.6465517 0.6551724
0
## Rand_For 0.03389831 0.06867329 0.1034483 0.09175921 0.1206897 0.1379310
0
## Xtr_gr_b 0.58620690 0.66379310 0.7413793 0.70806546 0.7689947 0.7966102
0
##
## Spec
##                Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## Log_Reg  0.7757437 0.7910792 0.8064147 0.8008415 0.8133904 0.8203661    0
## Step_AIC 0.7880871 0.7956454 0.8032037 0.7996903 0.8054920 0.8077803    0
## Navie_Ba 0.7974828 0.8013760 0.8052692 0.8061805 0.8105293 0.8157895    0
## KNN      0.7365407 0.7429843 0.7494279 0.7470391 0.7522883 0.7551487    0
```

```
## Rand_For 0.9896907 0.9908408 0.9919908 0.9916055 0.9925629 0.9931350    0
## Xtr_gr_b 0.7365407 0.7481330 0.7597254 0.7809827 0.8032037 0.8466819    0
```

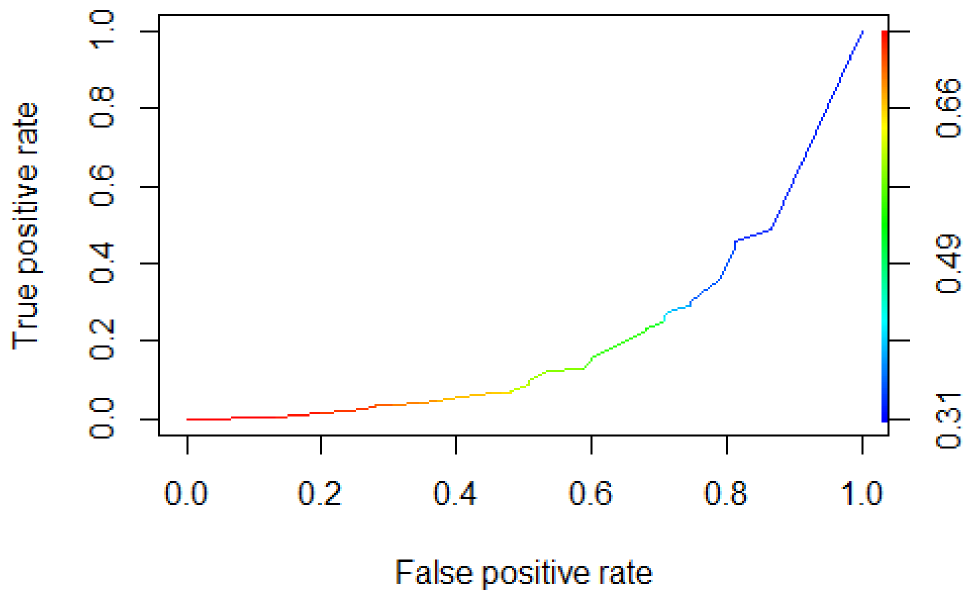
Draw box plots to compare models

```
scales <- list(x=list(relation="free"), y=list(relation="free"))
bwplot(models_to_compare, scales=scales)
```



#Improving Extreme Gradient Boosting

```
xgb_predictions_test <- predict(xgb_model, newdata = db_Test, type = "prob")

table(xgb_predictions_test$Default > 0.44, db_Test$Default)

##
##          Default NotDefault
##   FALSE       22        839
##   TRUE        53        283

library(ROCR)
p_test <- prediction(xgb_predictions_test$Default, db_Test$Default)
perf <- performance(p_test, "tpr", "fpr")
plot(perf, colorize= TRUE)
```



True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.31
0.49
0.66