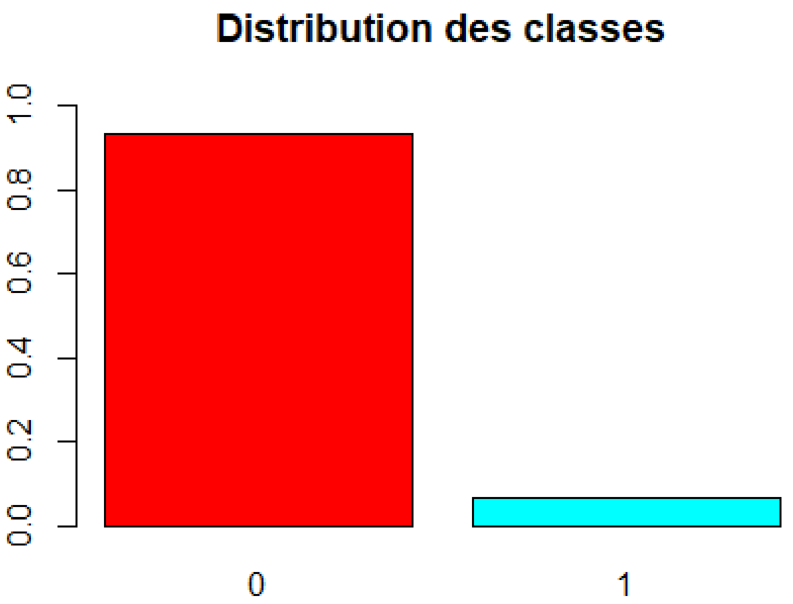
Projet : Modèle de prévision d'état financière pour des clients d'une banque

Auteur : *Ziate Ayoub*

03/11/2019

```
#------ Chargement des données et déclaration des librairies ------#

source("AFD_procedures.r")
trainData<-read.csv("ScoringTraining.csv",header = TRUE, sep = ",")[,2:12]

require(FactoMineR)

## Loading required package: FactoMineR

library(Amelia)

## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2019 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##

library(rpart)
library(caTools)
library(lattice)
library(ggplot2)
library(caret)
library(sqldf)

## Loading required package: gsubfn

## Loading required package: proto

## Loading required package: RSQLite

library(MASS)
library(ROCR)

## Loading required package: gplots

##
## Attaching package: 'gplots'
## The following object is masked from 'package:stats':
##
##     lowess
```

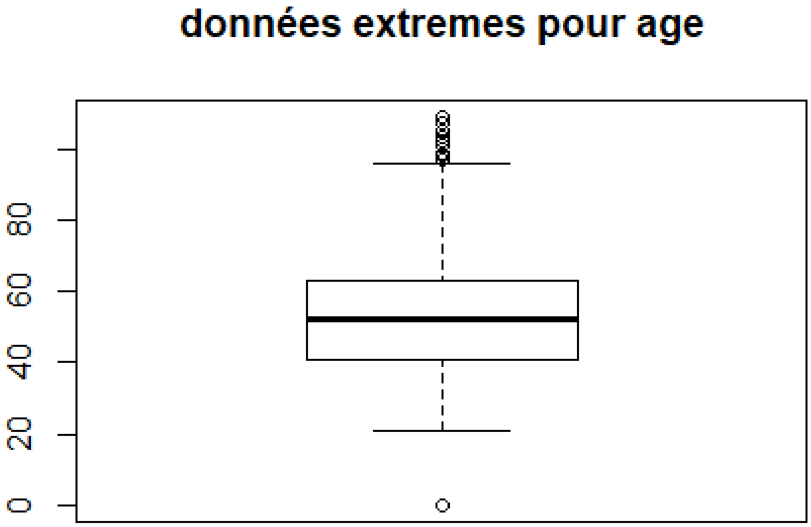
```
#########################
#Phase de prétraitement#
#########################

#----- Préparation de données -----#

####### Question 1 #######

barplot(prop.table(table(trainData$SeriousDlqin2yrs)),
        col = rainbow(2),
        ylim = c(0,1),
        main="Distribution des classes")
```



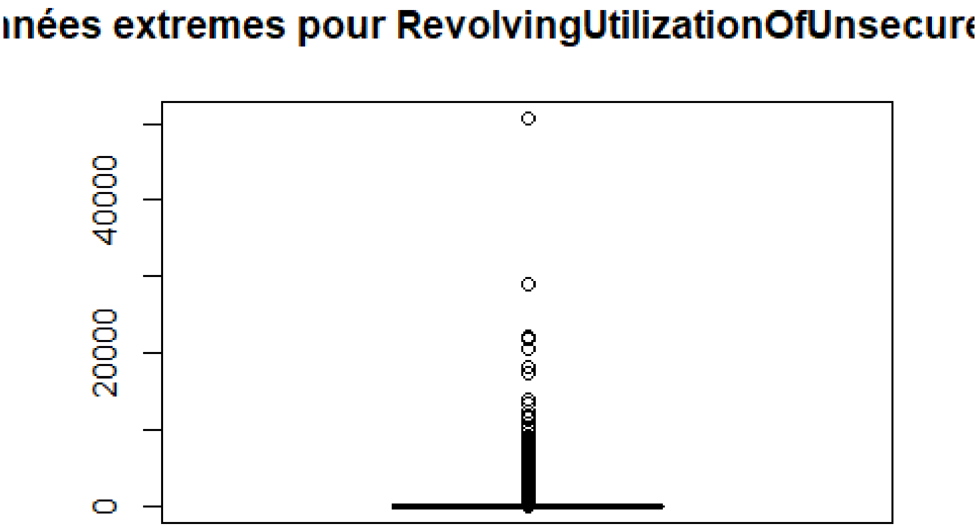
```
prop = prop.table(table(trainData$SeriousDlqin2yrs))*100
prop
```

```
##
##      0      1
## 93.316  6.684
```

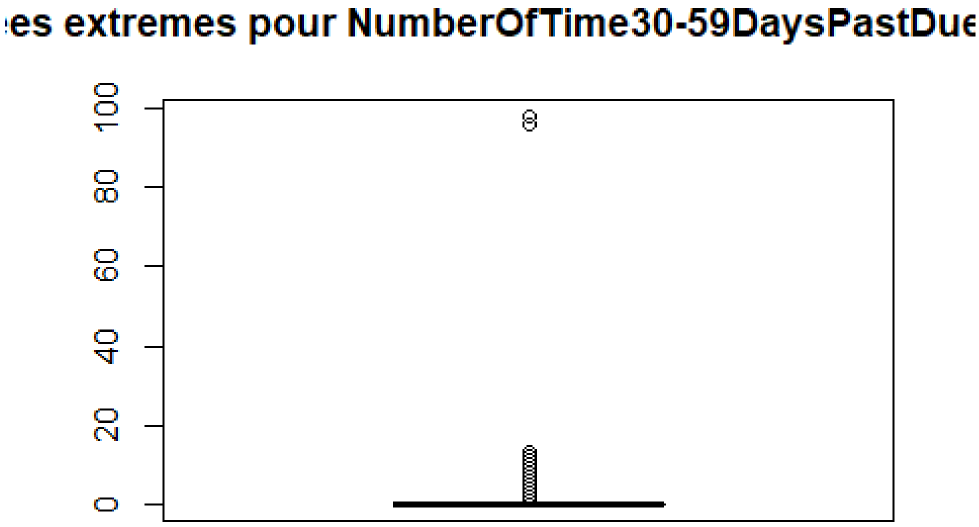
```
####### Question 2 #######

#par(mfrow = c(3,4))
boxplot(trainData$age, xlabel="age",main="données extremes pour age")
```



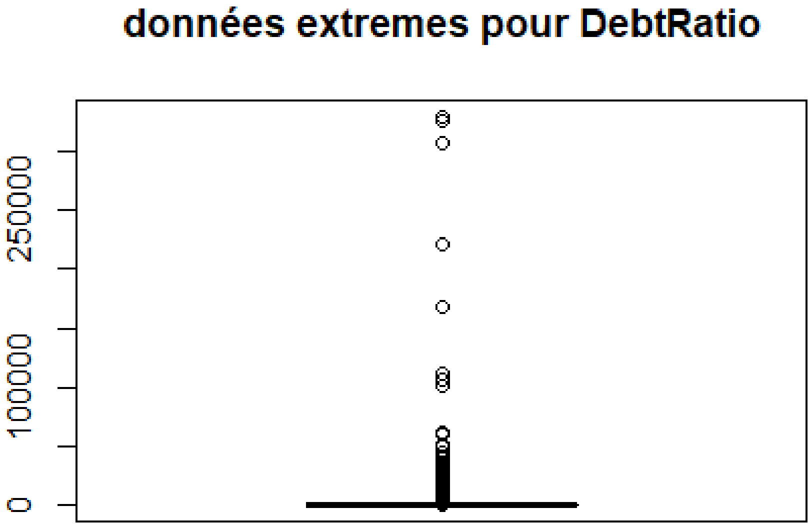
```
boxplot(trainData$RevolvingUtilizationOfUnsecuredLines,
xlabel="RevolvingUtilizationOfUnsecuredLines",main="données extremes pour
RevolvingUtilizationOfUnsecuredLines" )
```



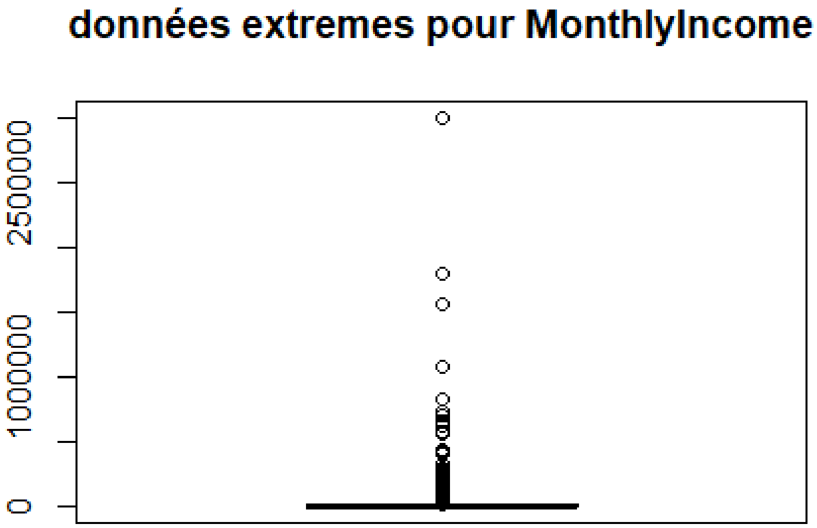
```
boxplot(trainData$`NumberOfTime30_59DaysPastDueNotWorse`,
xlabel="NumberOfTime30_59DaysPastDueNotWorse",main="données extremes pour
NumberOfTime30-59DaysPastDueNotWorse")
```



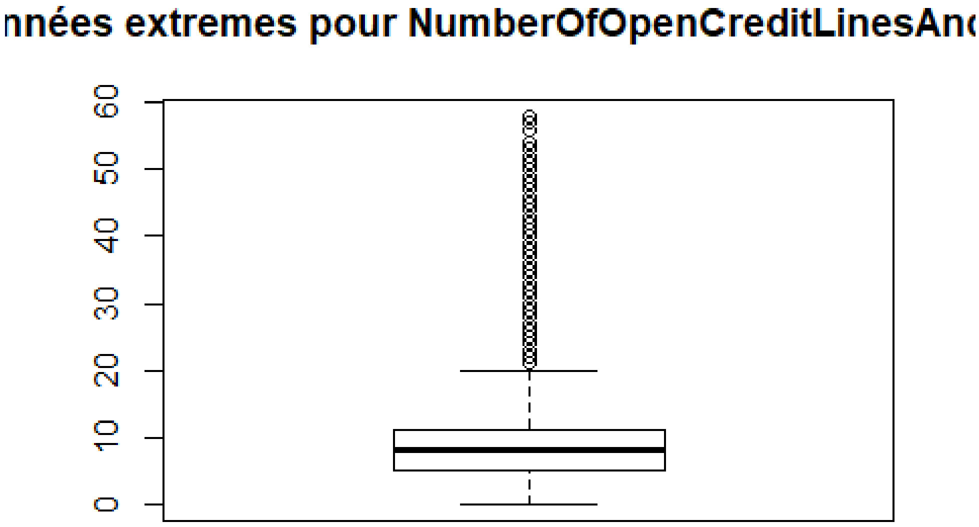
```
boxplot(trainData$DebtRatio, xlabel="DebtRatio",main="données extremes pour
DebtRatio")
```



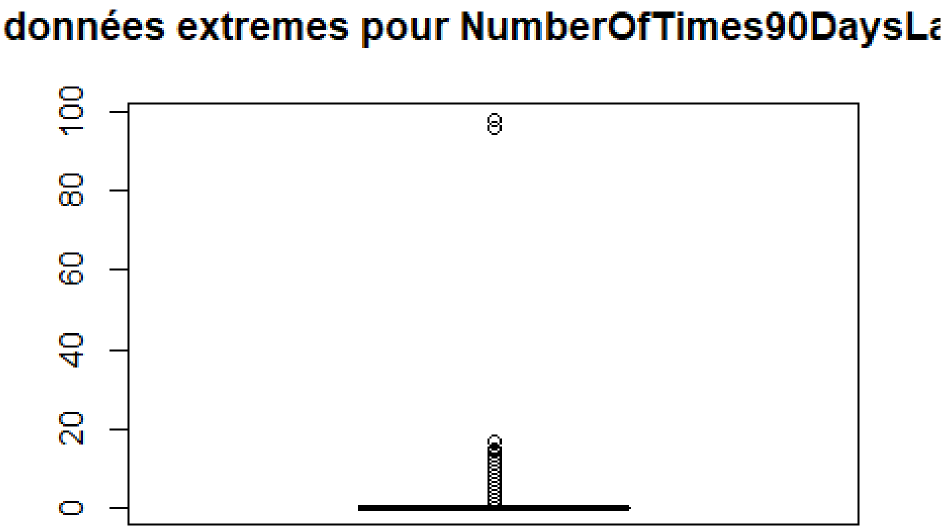
```
boxplot(trainData$MonthlyIncome, xlabel="MonthlyIncome",main="données
extremes pour MonthlyIncome")
```



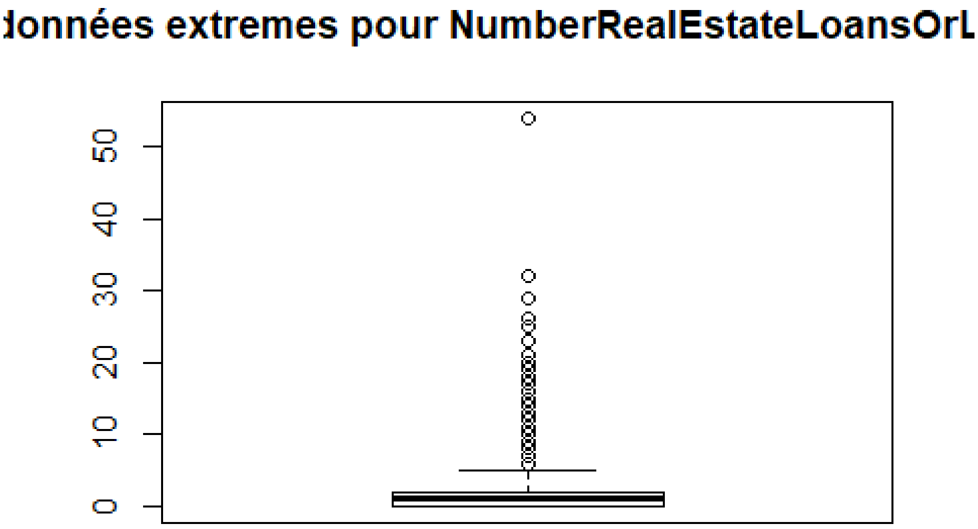
```
boxplot(trainData$NumberOfOpenCreditLinesAndLoans,
xlabel="NumberOfOpenCreditLinesAndLoans",main="données extremes pour
NumberOfOpenCreditLinesAndLoans")
```



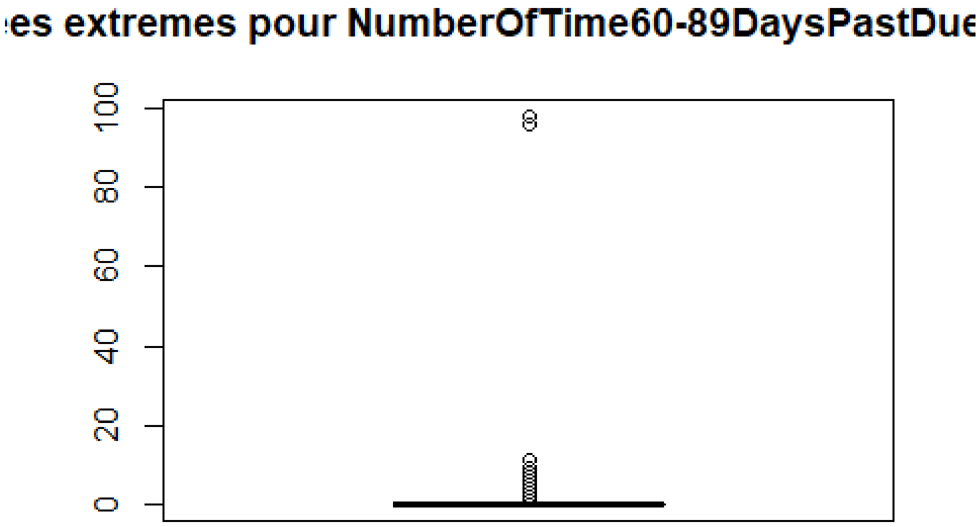
```
boxplot(trainData$NumberOfTimes90DaysLate,
xlabel="NumberOfTimes90DaysLate",main="données extremes pour
NumberOfTimes90DaysLate")
```



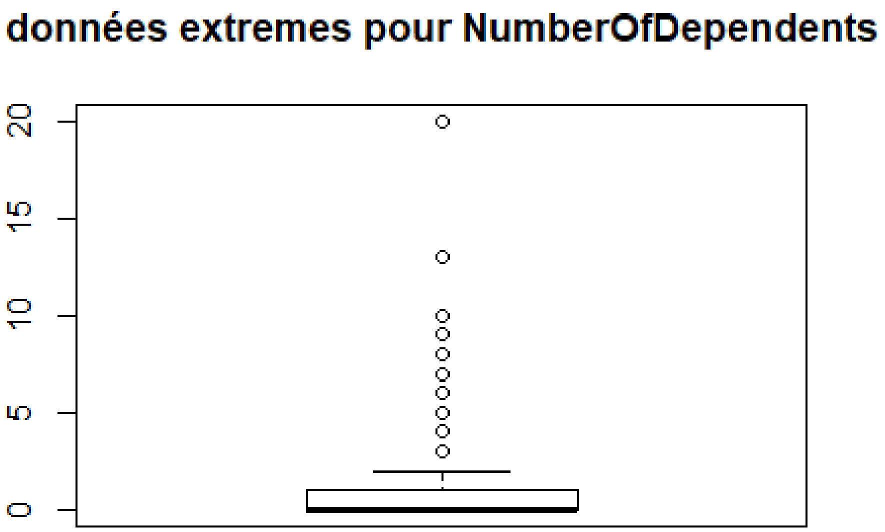
```
boxplot(trainData$NumberRealEstateLoansOrLines,
xlabel="NumberRealEstateLoansOrLines",main="données extremes pour
NumberRealEstateLoansOrLines")
```



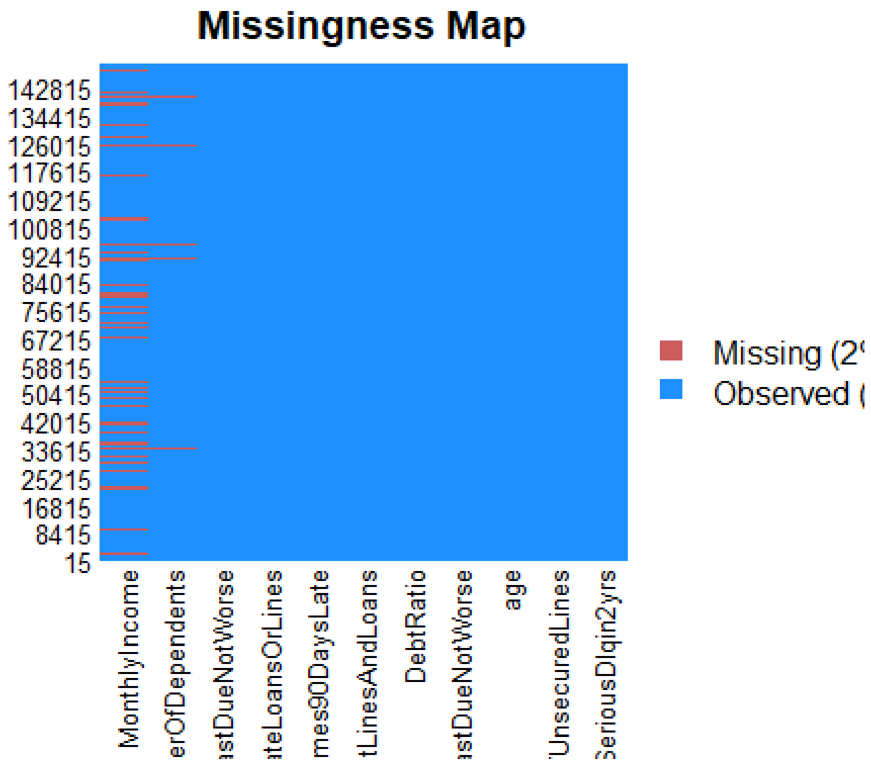
```
boxplot(trainData$`NumberOfTime60_89DaysPastDueNotWorse`,
xlabel="NumberOfTime60_89DaysPastDueNotWorse",main="données extremes pour
NumberOfTime60-89DaysPastDueNotWorse")
```



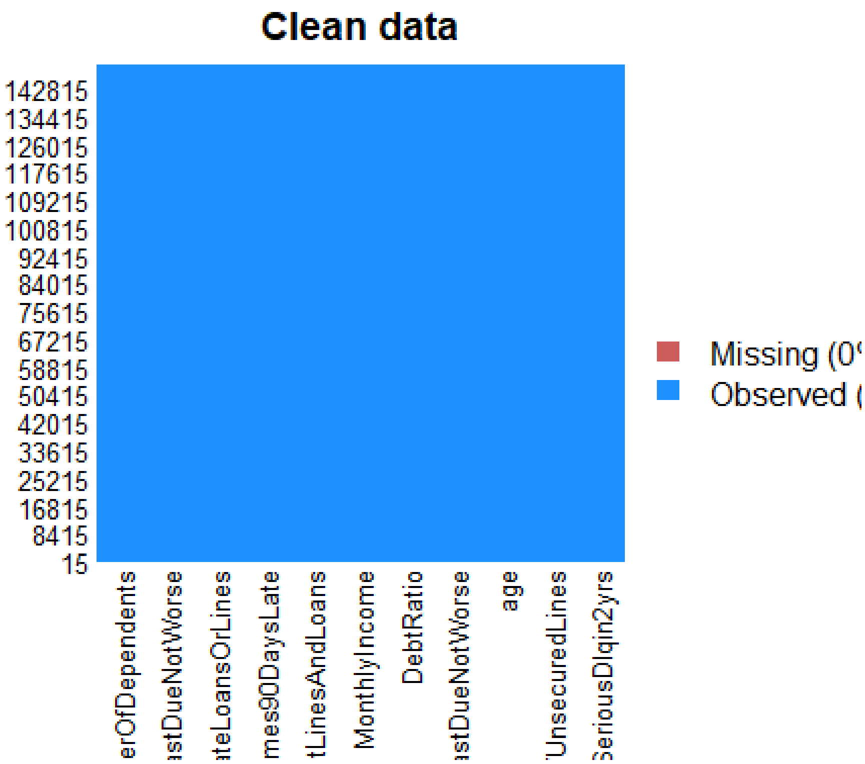
```
boxplot(trainData$NumberOfDependents,
xlabel="NumberOfDependents",main="données extremes pour NumberOfDependents")
```



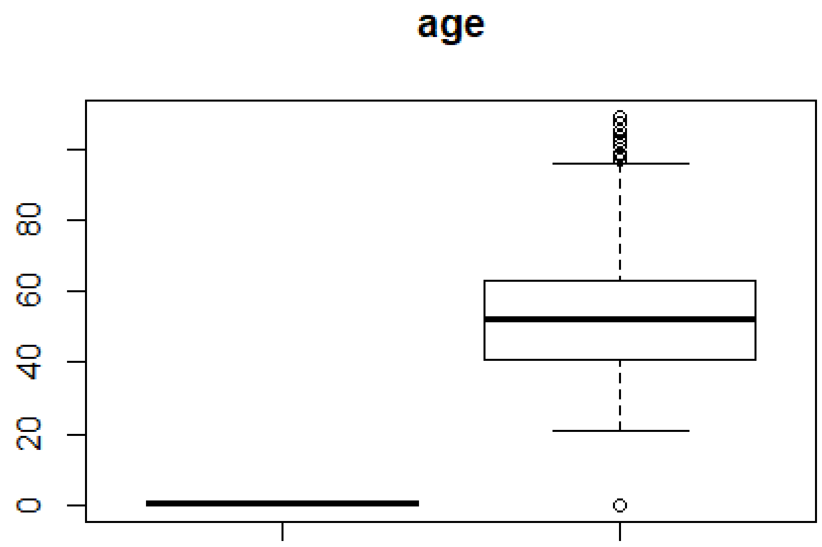
```
####### Question 3 #######

missmap(trainData)
```



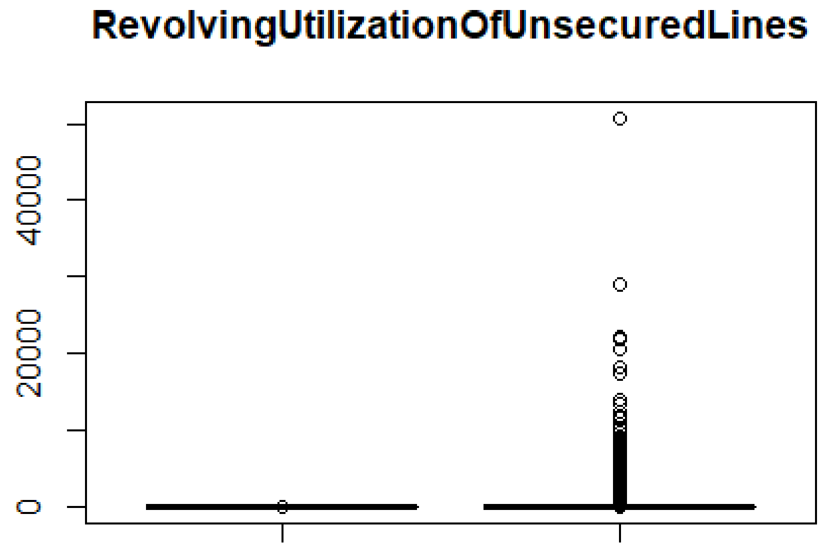
```
####### Question 4 #######

cleanData=trainData
sapply(cleanData, function(x) sum(is.na(x)))

##                     SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines
##                                    0                                    0
##                                  age NumberOfTime30_59DaysPastDueNotWorse
##                                    0                                    0
##                            DebtRatio                        MonthlyIncome
##                                    0                                29731
##      NumberOfOpenCreditLinesAndLoans              NumberOfTimes90DaysLate
##                                    0                                    0
##         NumberRealEstateLoansOrLines NumberOfTime60_89DaysPastDueNotWorse
##                                    0                                    0
##                   NumberOfDependents
##                                 3924

#which(is.na(cleanData$MonthlyIncome)) #Tells us the location of all NA
values

cleanData$MonthlyIncome[which(is.na(cleanData$MonthlyIncome))] <-
median(cleanData$MonthlyIncome, na.rm=TRUE) #Substitutes NA values for the
median in that column
cleanData$NumberOfDependents[which(is.na(cleanData$NumberOfDependents))] <-
median(cleanData$NumberOfDependents, na.rm=TRUE) #Substitutes NA values for
the median in that column

missmap(cleanData,main="Clean data") #Only missing values in file variable
```



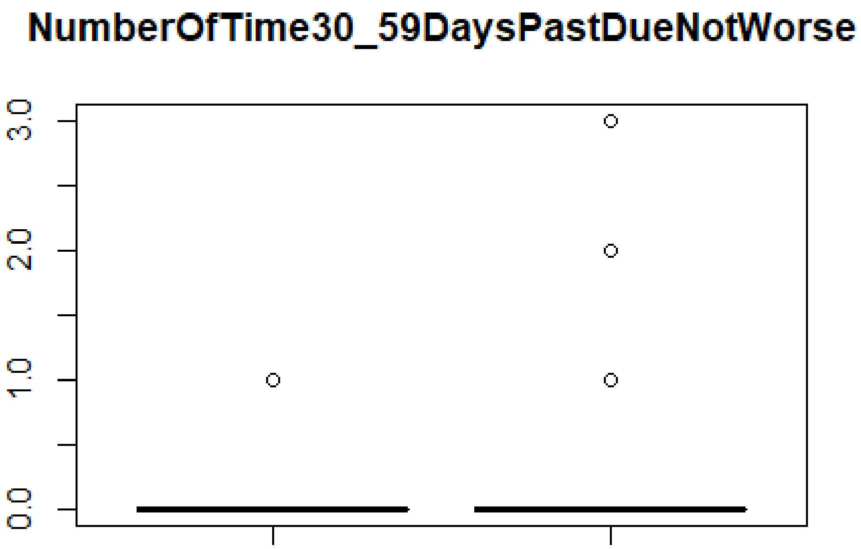
```
#----- Equilibrage des données d’apprentissage -----#

####### Question 5 #######

set.seed(123)
cleanData = downSample(x=cleanData[, -ncol(cleanData)],
y=factor(cleanData$SeriousDlqin2yrs))
prop.table(table(cleanData$SeriousDlqin2yrs))

##
##   0   1
## 0.5 0.5

split = sample.split(cleanData$SeriousDlqin2yrs, SplitRatio = 0.7)

TrainingData = subset(cleanData, split == TRUE)
TestData = subset(cleanData, split == TRUE)

#----- Identification des meilleurs prédicteurs parmi les variables -----#

####### Question 6 #######

#par(mfrow = c(3,4))
boxplot(cleanData$SeriousDlqin2yrs, trainData$age,main="age")
```



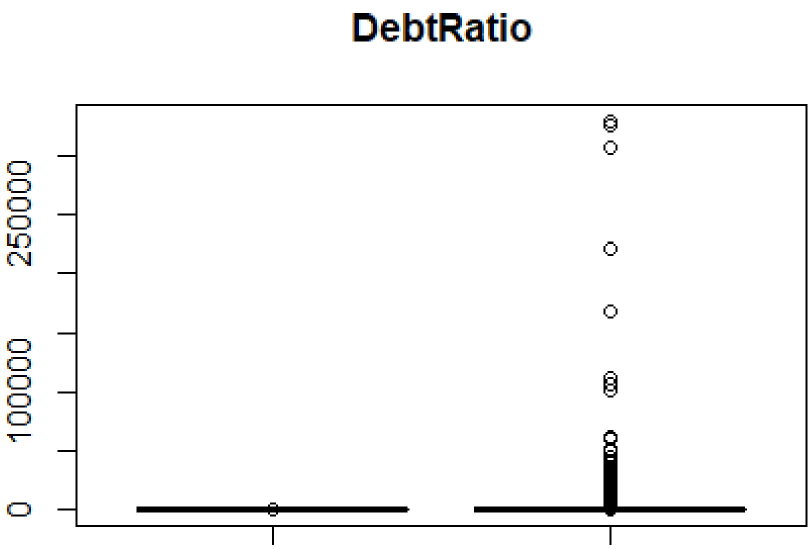
```
boxplot(trainData$SeriousDlqin2yrs,
trainData$RevolvingUtilizationOfUnsecuredLines,main="RevolvingUtilizationOfUn
securedLines")
```



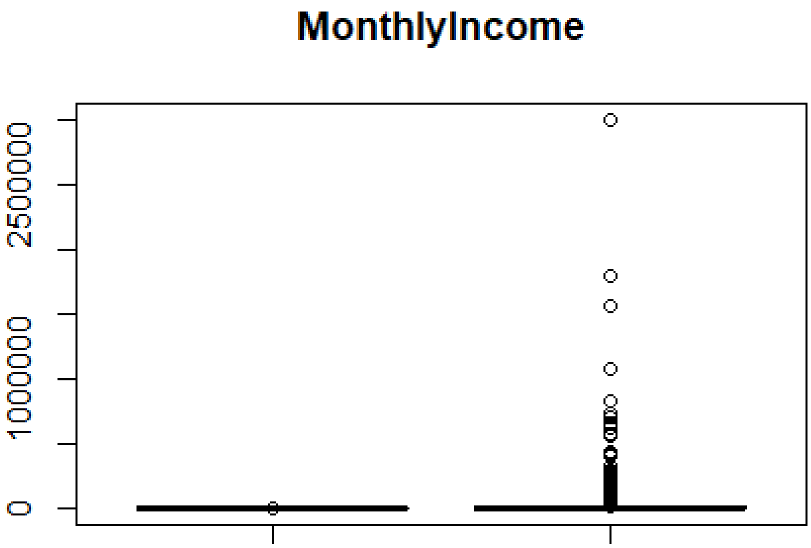
```
boxplot(trainData$SeriousDlqin2yrs,
trainData$NumberOfTime30_59DaysPastDueNotWorse,main="NumberOfTime30_59DaysPas
tDueNotWorse",ylim=c(0,3))
```



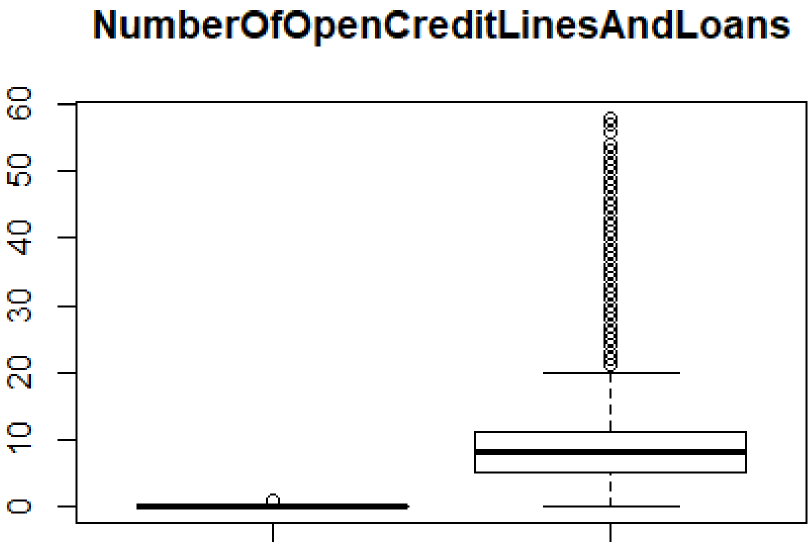
```
boxplot(trainData$SeriousDlqin2yrs, trainData$DebtRatio,main="DebtRatio")
```



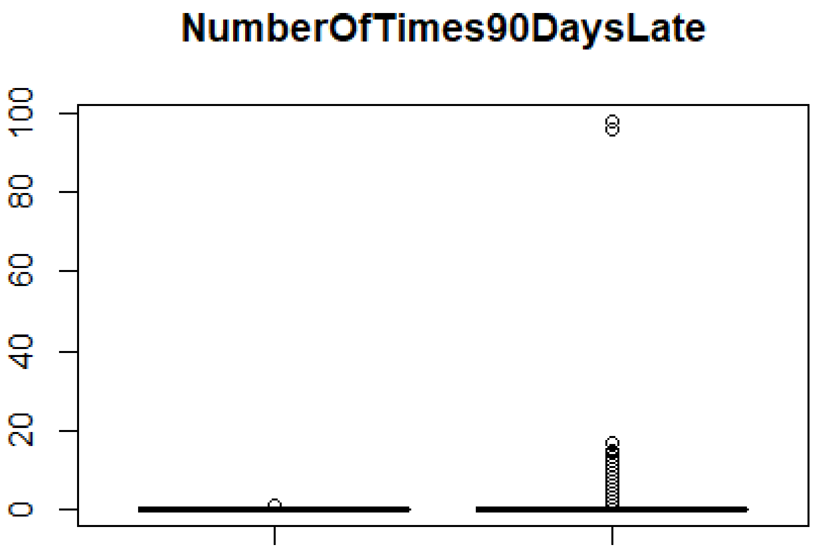
```
boxplot(trainData$SeriousDlqin2yrs,
trainData$MonthlyIncome,main="MonthlyIncome")
```



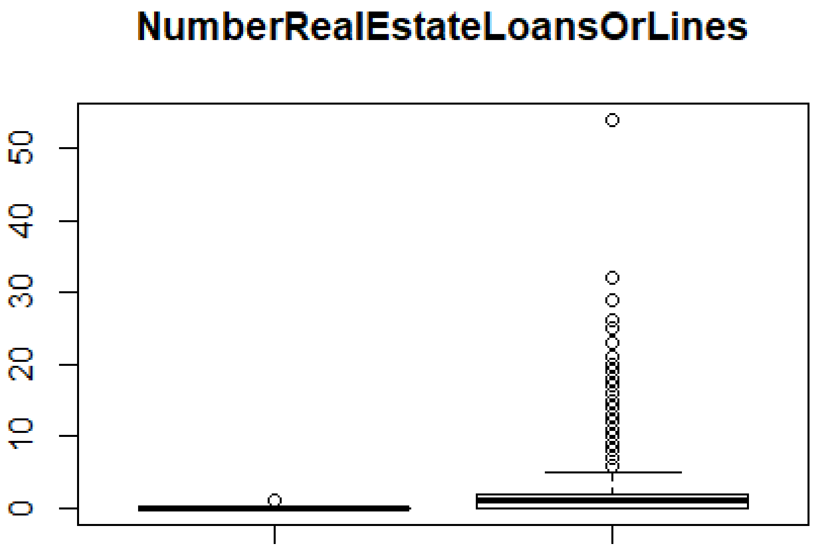
```
boxplot(trainData$SeriousDlqin2yrs,
trainData$NumberOfOpenCreditLinesAndLoans,main="NumberOfOpenCreditLinesAndLoa
ns")
```



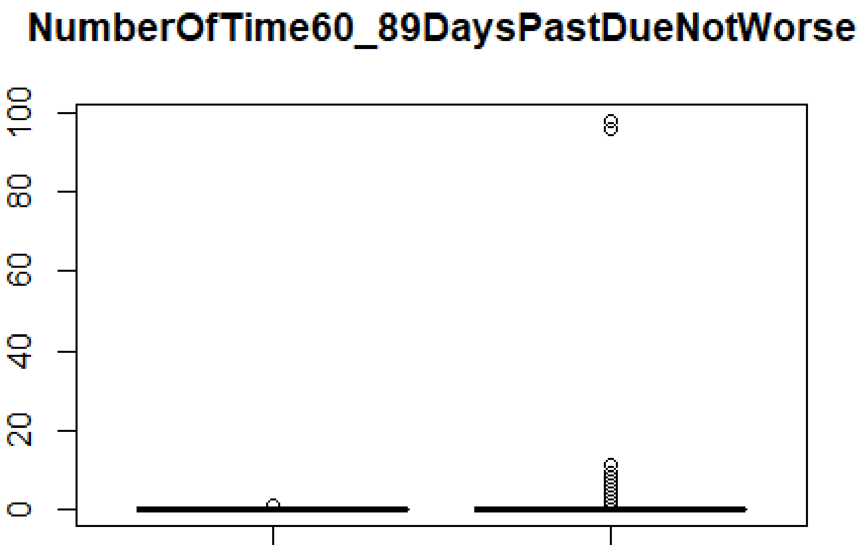
```
boxplot(trainData$SeriousDlqin2yrs,
trainData$NumberOfTimes90DaysLate,main="NumberOfTimes90DaysLate")
```



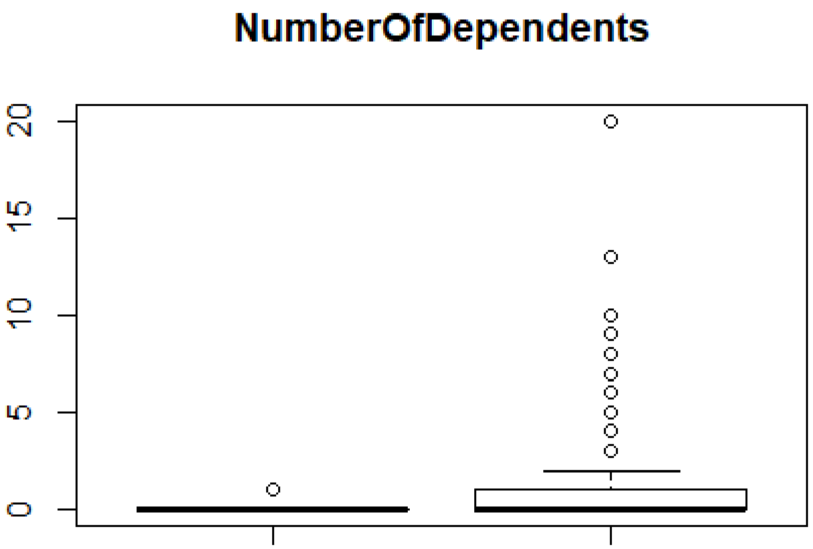
```r
boxplot(trainData$SeriousDlqin2yrs,
trainData$NumberRealEstateLoansOrLines,main="NumberRealEstateLoansOrLines")
```



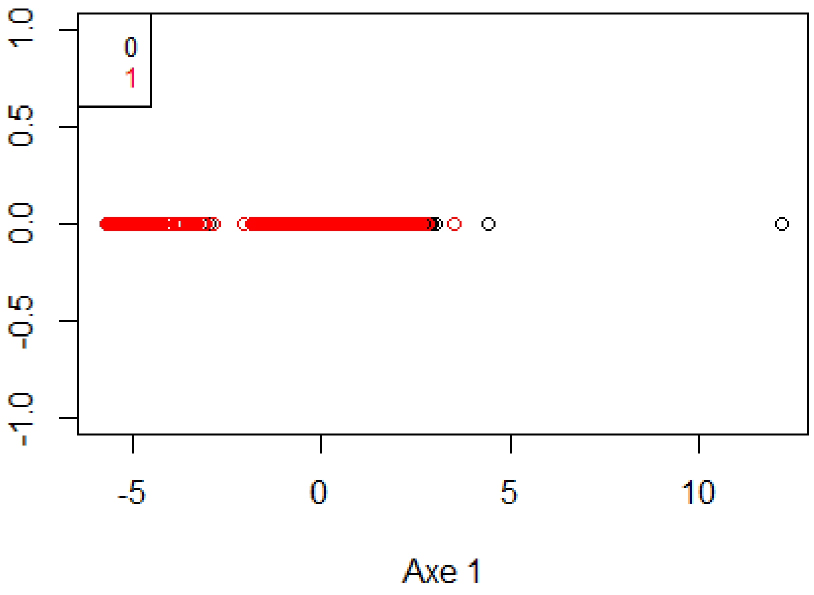
```
boxplot(trainData$SeriousDlqin2yrs,
trainData$NumberOfTime60_89DaysPastDueNotWorse,main="NumberOfTime60_89DaysPastDueNotWorse")
```



```
boxplot(trainData$SeriousDlqin2yrs,
trainData$NumberOfDependents,main="NumberOfDependents")
```



```
########################
#Preparation des données#
########################

####### Question 7 #######

DataChosen = sqldf("select age, DebtRatio, MonthlyIncome,
NumberOfOpenCreditLinesAndLoans, NumberOfTimes90DaysLate from TrainingData;")
ResAFD = AFD(DataChosen, TrainingData$SeriousDlqin2yrs)

plotAFD(ResAFD)
```



```
# axe1 ne donne pas une bonne descrimination car les deux groupes ne sont pas
assez séparé

####### Question 8 #######

#-------- LDA --------#

data.lda = lda(TrainingData$SeriousDlqin2yrs ~ .,data=TrainingData[,c("age",
"DebtRatio", "MonthlyIncome", "NumberOfOpenCreditLinesAndLoans",
"NumberOfTimes90DaysLate")])
# les deux graphes centrée sur 0 et ont le meme étendue

#data.lda$scaling                #facteur discriminant

PredictionLDA <- predict(data.lda)

#head(PredictionLDA$x)               #variable discriminante (canonique)

tab = table(Predicted=PredictionLDA$class, TrainingData$SeriousDlqin2yrs)
tab

##
## Predicted    0    1
##         0 4204 2643
##         1 2814 4375

print("Sensitivity")
```

```
## [1] "Sensitivity"
```

```r
sensitivity(tab)              # Sensitivity
```

```
## [1] 0.5990311
```

```r
print("Specificity")
```

```
## [1] "Specificity"
```

```r
specificity(tab)              # Specificity
```

```
## [1] 0.623397
```

```r
print("Accuracy")
```

```
## [1] "Accuracy"
```

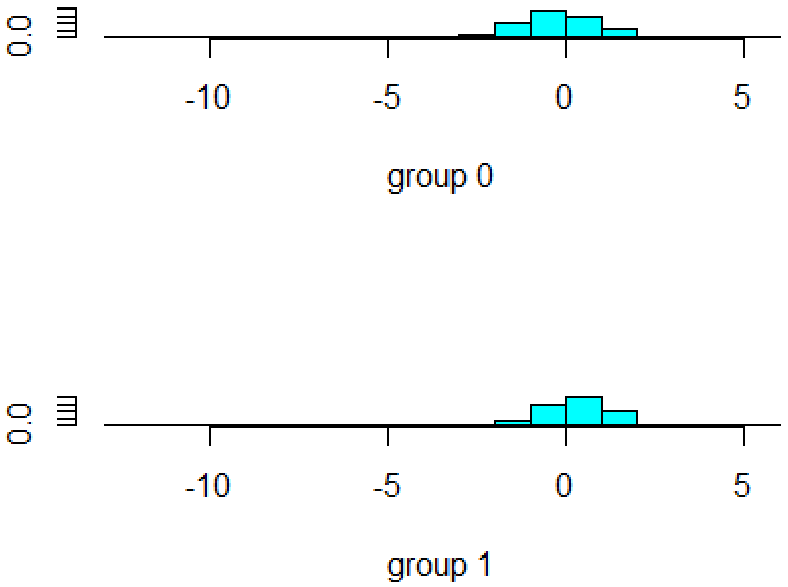
```r
sum(diag(tab))/sum(tab)       # Accuracy = 61%
```

```
## [1] 0.611214
```

```r
head(data.lda)
```

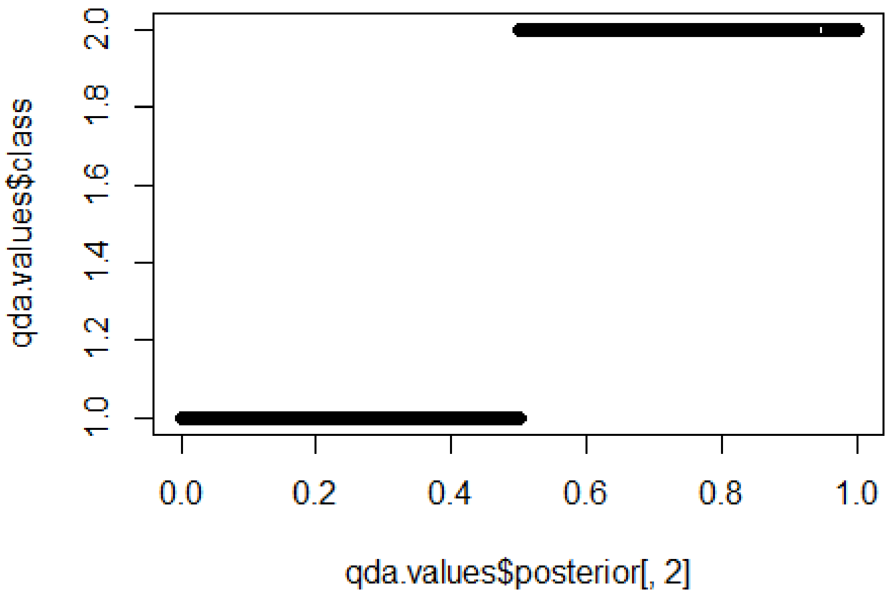
```
## $prior
##   0   1 
## 0.5 0.5 
## 
## $counts
##    0    1 
## 7018 7018 
## 
## $means
##        age DebtRatio MonthlyIncome NumberOfOpenCreditLinesAndLoans
## 0 53.03434  358.7557      6767.220                        8.511542
## 1 45.82331  291.4172      5558.461                        7.838843
##   NumberOfTimes90DaysLate
## 0               0.1202622
## 1               2.0347677
## 
## $scaling
##                                           LD1
## age                             -6.522101e-02
## DebtRatio                       -3.857482e-05
## MonthlyIncome                   -8.342219e-06
## NumberOfOpenCreditLinesAndLoans  4.835181e-03
## NumberOfTimes90DaysLate          4.190318e-02
## 
## $lev
## [1] "0" "1"
## 
## $svd
## [1] 33.17043
```

```
plot(data.lda)
```



```
#-------- QDA --------#

data.qda <- qda(TrainingData$SeriousDlqin2yrs ~.,data=TrainingData[,c("age",
"DebtRatio", "MonthlyIncome", "NumberOfOpenCreditLinesAndLoans",
"NumberOfTimes90DaysLate")])

qda.values <- predict(data.qda, data=TrainingData)
predQDA = predict(data.qda, data=TrainingData)
tab = table(qda.values$class, TrainingData$SeriousDlqin2yrs)
tab

##
##        0    1
##   0 4572 2842
##   1 2446 4176

print("Sensitivity")

## [1] "Sensitivity"

sensitivity(tab)             # Sensitivity

## [1] 0.6514677

print("Specificity")
```

```
## [1] "Specificity"

specificity(tab)              # Specificity

## [1] 0.5950413

print("Accuracy")

## [1] "Accuracy"

sum(diag(tab))/sum(tab)       # Accuracy = 62%

## [1] 0.6232545

plot(qda.values$posterior[,2], qda.values$class,
col=TrainingData$SeriousDlqin2yrs)
```



```
####### Question 9 #######

#-------- Régression Logistique --------#

ResRL<- glm(TrainingData$SeriousDlqin2yrs ~
TrainingData$age+TrainingData$NumberOfTimes90DaysLate,
data=TrainingData,family='binomial')

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(ResRL)
```

```
##
## Call:
## glm(formula = TrainingData$SeriousDlqin2yrs ~ TrainingData$age +
##     TrainingData$NumberOfTimes90DaysLate, family = "binomial",
##     data = TrainingData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.4904  -1.0470  -0.2743   1.1106   1.9258
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           1.42185    0.06797   20.92   <2e-16
## TrainingData$age                     -0.03340    0.00132  -25.30   <2e-16
## TrainingData$NumberOfTimes90DaysLate  0.77461    0.03270   23.69   <2e-16
##
## (Intercept)                          ***
## TrainingData$age                     ***
## TrainingData$NumberOfTimes90DaysLate ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 19458  on 14035  degrees of freedom
## Residual deviance: 17150  on 14033  degrees of freedom
## AIC: 17156
##
## Number of Fisher Scoring iterations: 8
```

```
PredictionRL<-predict(ResRL, TrainingData, type="response")
```

```
pred1 = ifelse(PredictionRL>0.5, 1, 0)
tab = table(Predicted = pred1, Actual = TrainingData$SeriousDlqin2yrs)
tab
```

```
##          Actual
## Predicted    0    1
##         0 5087 2814
##         1 1931 4204
```

```
print("Sensitivity")
```

```
## [1] "Sensitivity"
```

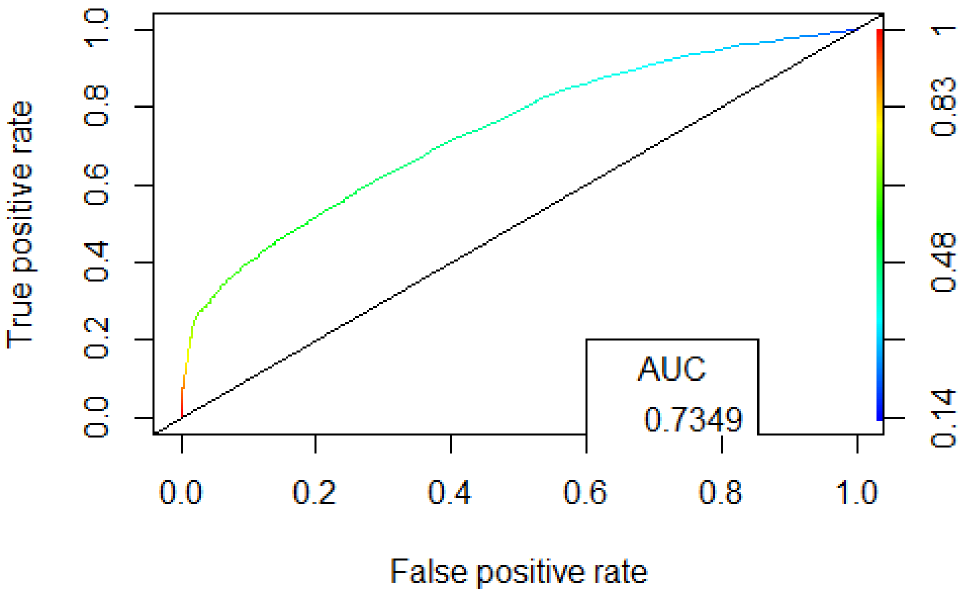
```
sensitivity(tab)              # Sensitivity
```

```
## [1] 0.7248504
```

```
print("Specificity")
```

```
## [1] "Specificity"
```

```r
specificity(tab)              # Specificity

## [1] 0.5990311

print("Accuracy")

## [1] "Accuracy"

sum(diag(tab))/sum(tab)       # Accuracy = 66%

## [1] 0.6619407

##################################################
#Phase d’évaluation et règle de décision retenue#
##################################################

####### Question 11 #######

#-------- Courbe ROC & AUC --------#
# courbe ROC construite à l'aide d'incrementer treshold et la matrice de
confusion
# Axe x (FP): 1 - specifity
# Axe y (TP): sensitivity

table(TrainingData$SeriousDlqin2yrs,PredictionRL>0.5)

##
##      FALSE TRUE
##   0  5087 1931
##   1  2814 4204

pred=prediction(PredictionRL,TrainingData$SeriousDlqin2yrs)
perf=performance(pred,"tpr", "fpr")
plot(perf,colorize = TRUE)

abline(a=0, b=1)

auc = performance(pred, "auc")
auc = unlist(slot(auc,"y.values"))
auc = round(auc, 4)
print("AUC")

## [1] "AUC"

auc

## [1] 0.7349

legend(.6, .2, auc, title = "AUC")
```


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC
0.7349
1
0.83
0.48
0.14