Cedit Risk Model

Anushree Tomar

8-1-2019

Import data

```r
Data <- read.delim("Credit Data.txt")
```

Observing Data

```r
head(Data)
```

```
##   OBS. CHK_ACCT DURATION HISTORY NEW_CAR USED_CAR FURNITURE RADIO.TV EDUCATION
## 1    1        0        6       4       0        0         0        1         0
## 2    2        1       48       2       0        0         0        1         0
## 3    3        3       12       4       0        0         0        0         1
## 4    4        0       42       2       0        0         1        0         0
## 5    5        0       24       3       1        0         0        0         0
## 6    6        3       36       2       0        0         0        0         1
##   RETRAINING AMOUNT SAV_ACCT EMPLOYMENT INSTALL_RATE MALE_DIV MALE_SINGLE
## 1          0   1169        4          4            4        0           1
## 2          0   5951        0          2            2        0           0
## 3          0   2096        0          3            2        0           1
## 4          0   7882        0          3            2        0           1
## 5          0   4870        0          2            3        0           1
## 6          0   9055        4          2            2        0           1
##   MALE_MAR_or_WID CO.APPLICANT GUARANTOR PRESENT_RESIDENT REAL_ESTATE
## 1               0            0         0                4           1
## 2               0            0         0                2           1
## 3               0            0         0                3           1
## 4               0            0         1                4           0
## 5               0            0         0                4           0
## 6               0            0         0                4           0
##   PROP_UNKN_NONE AGE OTHER_INSTALL RENT OWN_RES NUM_CREDITS JOB NUM_DEPENDENTS
## 1              0  67             0    0       1           2   2              1
## 2              0  22             0    0       1           1   2              1
## 3              0  49             0    0       1           1   1              2
## 4              0  45             0    0       0           1   2              2
## 5              1  53             0    0       0           2   2              2
## 6              1  35             0    0       0           1   1              2
##   TELEPHONE FOREIGN RESPONSE
```

```
## 1          1       0        1
## 2          0       0        0
## 3          0       0        1
## 4          0       0        1
## 5          0       0        0
## 6          1       0        1
```

```r
tail(Data)
```

```
##      OBS. CHK_ACCT DURATION HISTORY NEW_CAR USED_CAR FURNITURE RADIO.TV
## 997   997        0       30       2       0        1         0        0
## 998   998        3       12       2       0        0         0        1
## 999   999        0       45       2       0        0         0        1
## 1000 1000        1       45       4       0        1         0        0
## 1001   NA       NA       NA      NA      NA       NA        NA       NA
## 1002   NA       NA       NA      NA      NA       NA        NA       NA
##      EDUCATION RETRAINING AMOUNT SAV_ACCT EMPLOYMENT INSTALL_RATE MALE_DIV
## 997          0          0   3857        0          2            4        1
## 998          0          0    804        0          4            4        0
## 999          0          0   1845        0          2            4        0
## 1000         0          0   4576        1          0            3        0
## 1001        NA         NA     NA       NA         NA           NA       NA
## 1002        NA         NA     NA       NA         NA           NA       NA
##      MALE_SINGLE MALE_MAR_or_WID CO.APPLICANT GUARANTOR PRESENT_RESIDENT
## 997            0               0            0         0                4
## 998            1               0            0         0                4
## 999            1               0            0         0                4
## 1000           1               0            0         0                4
## 1001          NA              NA           NA        NA               NA
## 1002          NA              NA           NA        NA               NA
##      REAL_ESTATE PROP_UNKN_NONE AGE OTHER_INSTALL RENT OWN_RES NUM_CREDITS JOB
## 997            0              0  40             0    0       1           1   3
## 998            0              0  38             0    0       1           1   2
## 999            0              1  23             0    0       0           1   2
## 1000           0              0  27             0    0       1           1   2
## 1001          NA             NA  NA            NA   NA      NA          NA  NA
## 1002          NA             NA  NA            NA   NA      NA          NA  NA
##      NUM_DEPENDENTS TELEPHONE FOREIGN RESPONSE
## 997               1         1       0        1
## 998               1         0       0        1
## 999               1         1       0        0
## 1000              1         0       0        1
## 1001             NA        NA      NA       NA
## 1002             NA        NA      NA       NA
```

Removing Last 2 rows and 1 OBS. column

```r
Data<-Data[1:1000,-1]
```

Data Preprocessing

```r
dim(Data)
```

```
## [1] 1000   31
```

```r
colnames(Data)
```

```
##  [1] "CHK_ACCT"         "DURATION"         "HISTORY"          "NEW_CAR"
##  [5] "USED_CAR"         "FURNITURE"        "RADIO.TV"         "EDUCATION"
##  [9] "RETRAINING"       "AMOUNT"           "SAV_ACCT"         "EMPLOYMENT"
## [13] "INSTALL_RATE"     "MALE_DIV"         "MALE_SINGLE"      "MALE_MAR_or_WID"
## [17] "CO.APPLICANT"     "GUARANTOR"        "PRESENT_RESIDENT" "REAL_ESTATE"
## [21] "PROP_UNKN_NONE"   "AGE"              "OTHER_INSTALL"    "RENT"
## [25] "OWN_RES"          "NUM_CREDITS"      "JOB"              "NUM_DEPENDENTS"
## [29] "TELEPHONE"        "FOREIGN"          "RESPONSE"
```

Check Missing Values

```r
anyNA(Data)
```

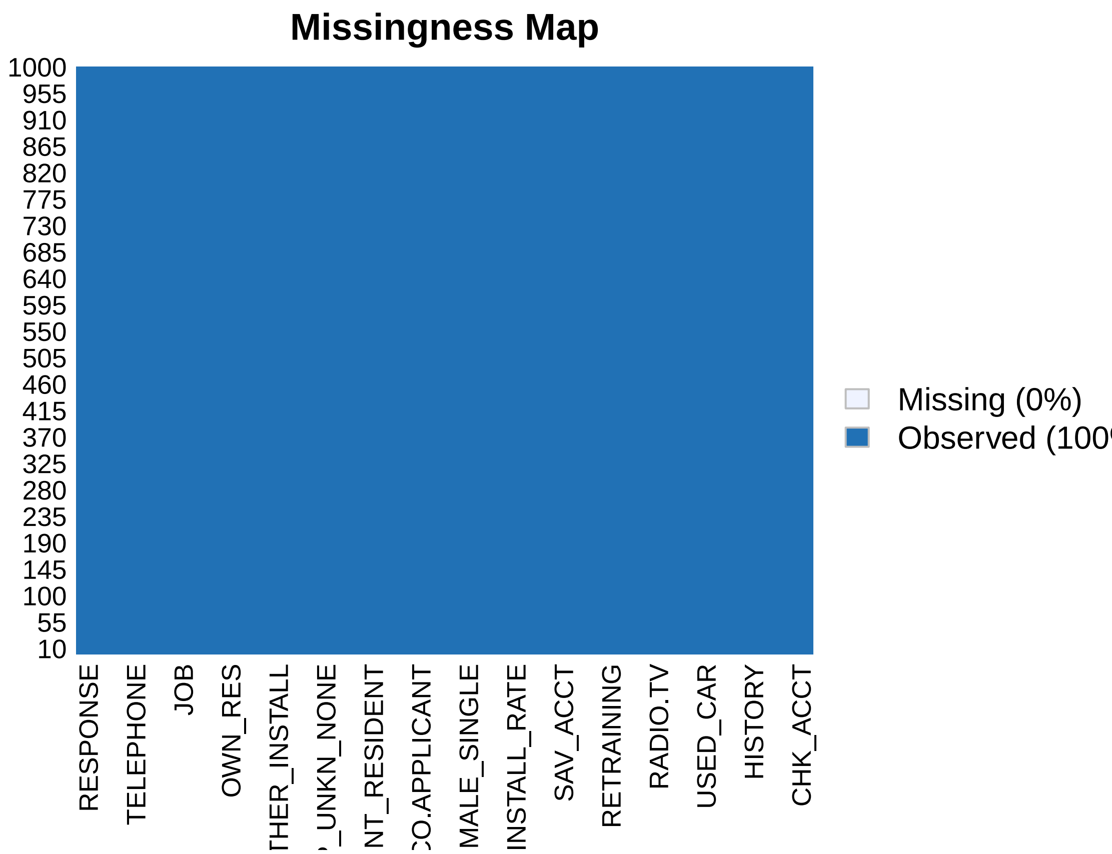
```
## [1] FALSE
```

```r
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.0, built: 2021-05-26)
## ## Copyright (C) 2005-2021 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```r
missmap(Data)
```



Checking for duplicate rows

```r
dim(Data)
```

```
## [1] 1000   31
```

```r
dim(unique(Data))
```

```
## [1] 1000   31
```

```r
dim(Data[!duplicated(Data),])
```

```
## [1] 1000   31
```

```r
dim(Data[duplicated(Data),])
```

```
## [1]   0 31
```

```r
str(Data)
```

```
## 'data.frame':    1000 obs. of  31 variables:
##  $ CHK_ACCT        : int  0 1 3 0 0 3 3 1 3 1 ...
##  $ DURATION        : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ HISTORY         : int  4 2 4 2 3 2 2 2 2 4 ...
##  $ NEW_CAR         : int  0 0 0 0 1 0 0 0 0 1 ...
##  $ USED_CAR        : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ FURNITURE       : int  0 0 0 1 0 0 1 0 0 0 ...
##  $ RADIO.TV        : int  1 1 0 0 0 0 0 0 1 0 ...
##  $ EDUCATION       : int  0 0 1 0 0 1 0 0 0 0 ...
##  $ RETRAINING      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMOUNT          : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ SAV_ACCT        : int  4 0 0 0 0 4 2 0 3 0 ...
##  $ EMPLOYMENT      : int  4 2 3 3 2 2 4 2 3 0 ...
##  $ INSTALL_RATE    : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ MALE_DIV        : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ MALE_SINGLE     : int  1 0 1 1 1 1 1 1 0 0 ...
##  $ MALE_MAR_or_WID : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ CO.APPLICANT    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GUARANTOR       : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ PRESENT_RESIDENT: int  4 2 3 4 4 4 4 2 4 2 ...
##  $ REAL_ESTATE     : int  1 1 1 0 0 0 0 0 1 0 ...
##  $ PROP_UNKN_NONE  : int  0 0 0 0 1 1 0 0 0 0 ...
##  $ AGE             : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ OTHER_INSTALL   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RENT            : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ OWN_RES         : int  1 1 1 0 0 0 1 0 1 1 ...
##  $ NUM_CREDITS     : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ JOB             : int  2 2 1 2 2 1 2 3 1 3 ...
##  $ NUM_DEPENDENTS  : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ TELEPHONE       : int  1 0 0 0 0 1 0 1 0 0 ...
##  $ FOREIGN         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RESPONSE        : int  1 0 1 1 0 1 1 1 1 0 ...
```

Change the data type of Data

```r
Data[,c(1,3:9,11,12,14:21,23:25,27,29:31)]<-lapply(Data[,c(1,3:9,11,12,14:21,23:25,27,29:31)],as.factor
Data[,c(2,10,13,22,26,28)]<-lapply(Data[,c(2,10,13,22,26,28)],as.numeric)

str(Data)
```

```
## 'data.frame':    1000 obs. of  31 variables:
##  $ CHK_ACCT        : Factor w/ 4 levels "0","1","2","3": 1 2 4 1 1 4 4 2 4 2 ...
##  $ DURATION        : num  6 48 12 42 24 36 24 36 12 30 ...
##  $ HISTORY         : Factor w/ 5 levels "0","1","2","3",..: 5 3 5 3 4 3 3 3 3 5 ...
##  $ NEW_CAR         : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 2 ...
##  $ USED_CAR        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ FURNITURE       : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 2 1 1 1 ...
##  $ RADIO.TV        : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 1 2 1 ...
```

```
##  $ EDUCATION       : Factor w/ 2 levels "0","1": 1 1 2 1 1 2 1 1 1 1 ...
##  $ RETRAINING      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ AMOUNT          : num  1169 5951 2096 7882 4870 ...
##  $ SAV_ACCT        : Factor w/ 5 levels "0","1","2","3",..: 5 1 1 1 1 5 3 1 4 1 ...
##  $ EMPLOYMENT      : Factor w/ 5 levels "0","1","2","3",..: 5 3 4 4 3 3 5 3 4 1 ...
##  $ INSTALL_RATE    : num  4 2 2 2 3 2 3 2 2 4 ...
##  $ MALE_DIV        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ MALE_SINGLE     : Factor w/ 2 levels "0","1": 2 1 2 2 2 2 2 2 1 1 ...
##  $ MALE_MAR_or_WID : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
##  $ CO.APPLICANT    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ GUARANTOR       : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
##  $ PRESENT_RESIDENT: Factor w/ 4 levels "1","2","3","4": 4 2 3 4 4 4 4 2 4 2 ...
##  $ REAL_ESTATE     : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 1 2 1 ...
##  $ PROP_UNKN_NONE  : Factor w/ 2 levels "0","1": 1 1 1 1 2 2 1 1 1 1 ...
##  $ AGE             : num  67 22 49 45 53 35 53 35 61 28 ...
##  $ OTHER_INSTALL   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ RENT            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ OWN_RES         : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 2 1 2 2 ...
##  $ NUM_CREDITS     : num  2 1 1 1 2 1 1 1 1 2 ...
##  $ JOB             : Factor w/ 4 levels "0","1","2","3": 3 3 2 3 3 2 3 4 2 4 ...
##  $ NUM_DEPENDENTS  : num  1 1 2 2 2 2 1 1 1 1 ...
##  $ TELEPHONE       : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 2 1 1 ...
##  $ FOREIGN         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ RESPONSE        : Factor w/ 2 levels "0","1": 2 1 2 2 1 2 2 2 2 1 ...
```

Exploratory Data Analysis

```r
library(psych)
#Numerical data
describe(Data[,c(2,10,13,22,26,28)], na.rm = TRUE, interp=FALSE,skew = TRUE, ranges = TRUE,trim=.1,
              type=3,check=TRUE,fast=NULL,quant=NULL,IQR=FALSE,omit=FALSE)
```

```
##                vars    n    mean      sd median trimmed     mad min   max range
## DURATION          1 1000   20.90   12.06   18.0   19.47    8.90   4    72    68
## AMOUNT            2 1000 3271.26 2822.74 2319.5 2754.57 1627.15 250 18424 18174
## INSTALL_RATE      3 1000    2.97    1.12    3.0    3.09    1.48   1     4     3
## AGE               4 1000   35.55   11.38   33.0   34.17   10.38  19    75    56
## NUM_CREDITS       5 1000    1.41    0.58    1.0    1.33    0.00   1     4     3
## NUM_DEPENDENTS    6 1000    1.16    0.36    1.0    1.07    0.00   1     2     1
##                 skew kurtosis    se
## DURATION        1.09     0.90  0.38
## AMOUNT          1.94     4.25 89.26
## INSTALL_RATE   -0.53    -1.21  0.04
## AGE             1.02     0.58  0.36
## NUM_CREDITS     1.27     1.58  0.02
## NUM_DEPENDENTS  1.90     1.63  0.01
```

```r
#categorical data
summary(Data[,-c(2,10,13,22,26,28)])
```

```
##  CHK_ACCT HISTORY NEW_CAR USED_CAR FURNITURE RADIO.TV EDUCATION RETRAINING
```

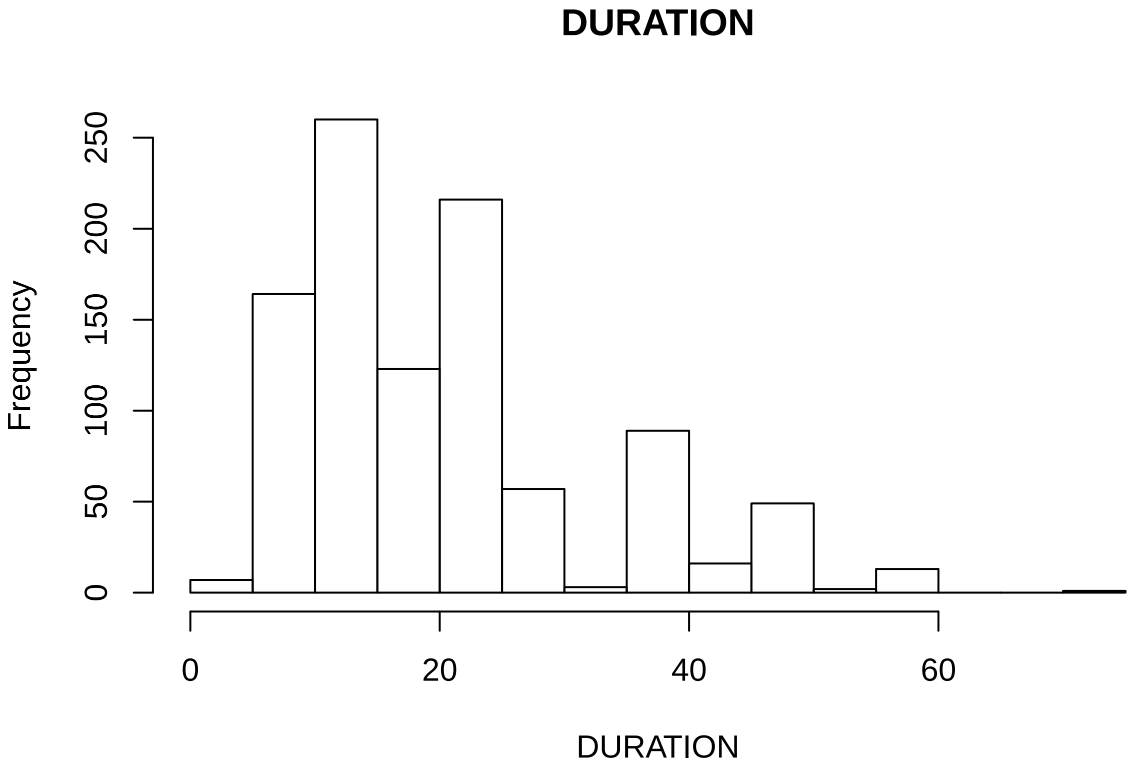
```
##  0:274    0: 40   0:766   0:897    0:819     0:720    0:950     0:903
##  1:269    1: 49   1:234   1:103    1:181     1:280    1: 50     1: 97
##  2: 63    2:530
##  3:394    3: 88
##           4:293
##  SAV_ACCT EMPLOYMENT MALE_DIV MALE_SINGLE MALE_MAR_or_WID CO.APPLICANT
##  0:603    0: 62      0:950    0:452       0:908           0:959
##  1:103    1:172      1: 50    1:548       1: 92           1: 41
##  2: 63    2:339
##  3: 48    3:174
##  4:183    4:253
##  GUARANTOR PRESENT_RESIDENT REAL_ESTATE PROP_UNKN_NONE OTHER_INSTALL RENT
##  0:948     1:130            0:718       0:846          0:814         0:821
##  1: 52     2:308            1:282       1:154          1:186         1:179
##            3:149
##            4:413
##
##  OWN_RES JOB     TELEPHONE FOREIGN RESPONSE
##  0:287   0: 22   0:596     0:963   0:300
##  1:713   1:200   1:404     1: 37   1:700
##          2:630
##          3:148
##
```

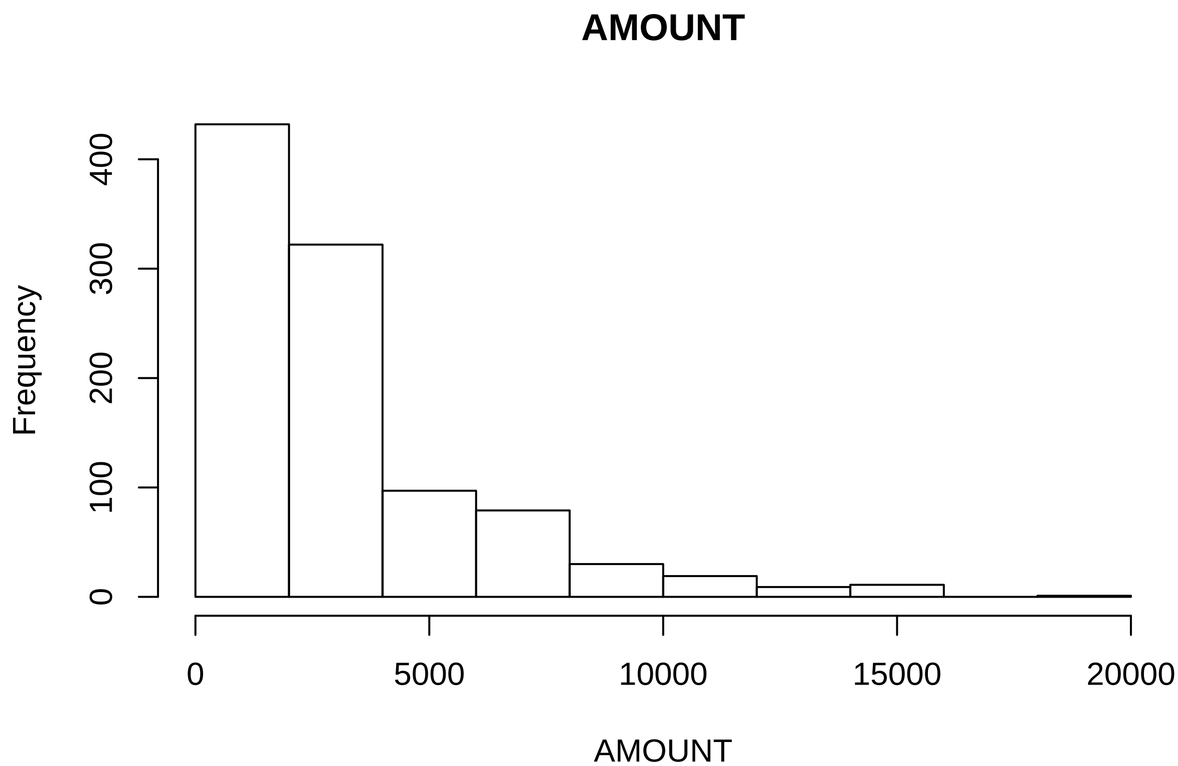
visualization

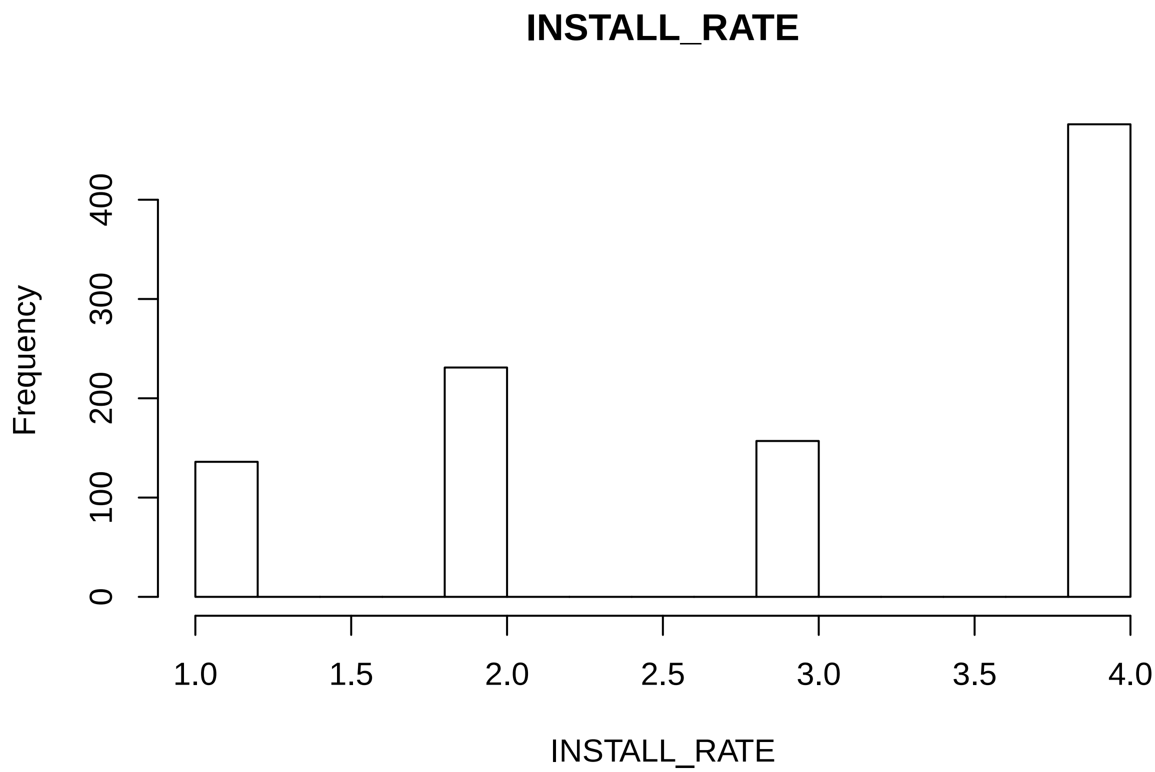
Univariate Analysis

Histogram For Distribution of num data

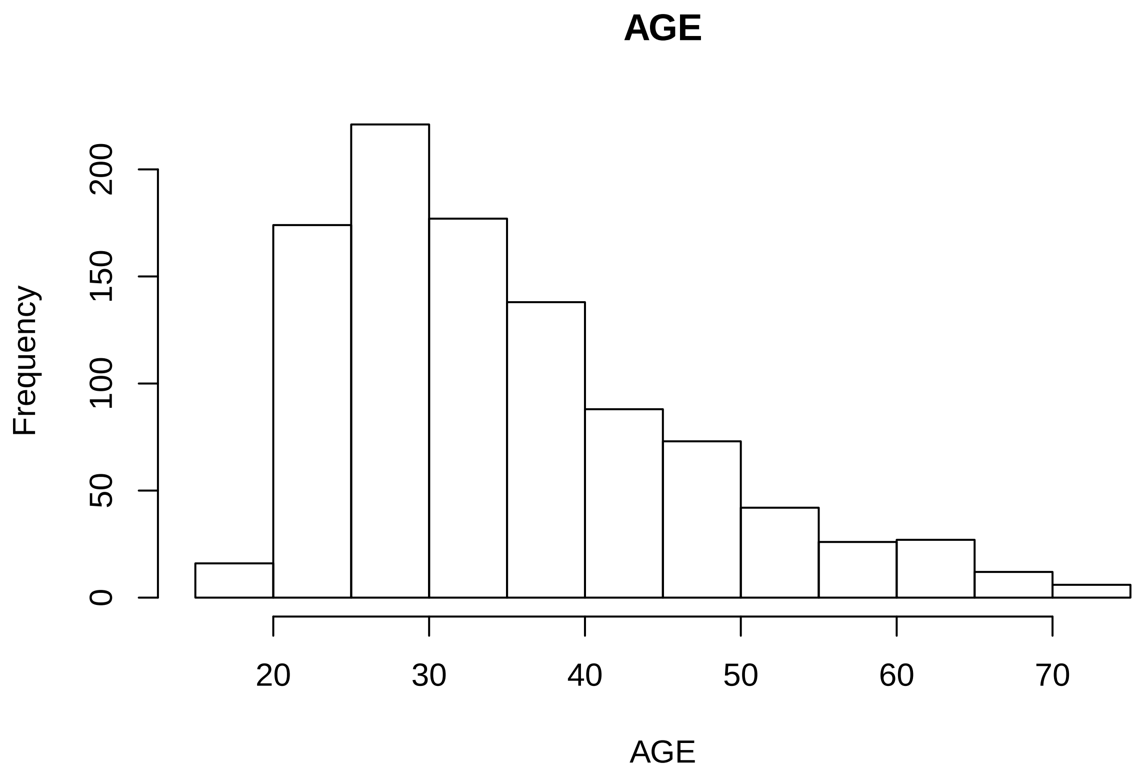
```
num_data<-Data[,c(2,10,13,22,26,28)]
for (i in 1:ncol(num_data)) {hist(num_data[[i]],main=colnames(num_data[i]),xlab = colnames(num_data[i]))
}
```

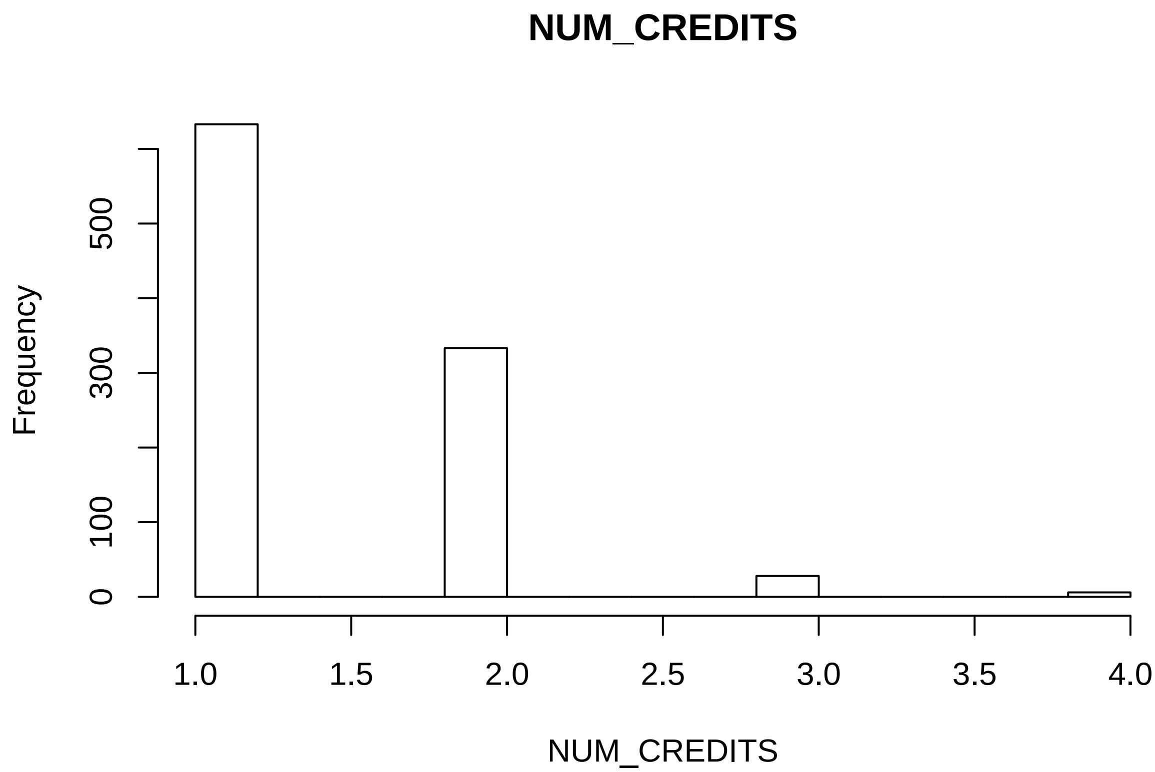

DURATION
Frequency
0
50
100
150
200
250
0
20
40
60
DURATION

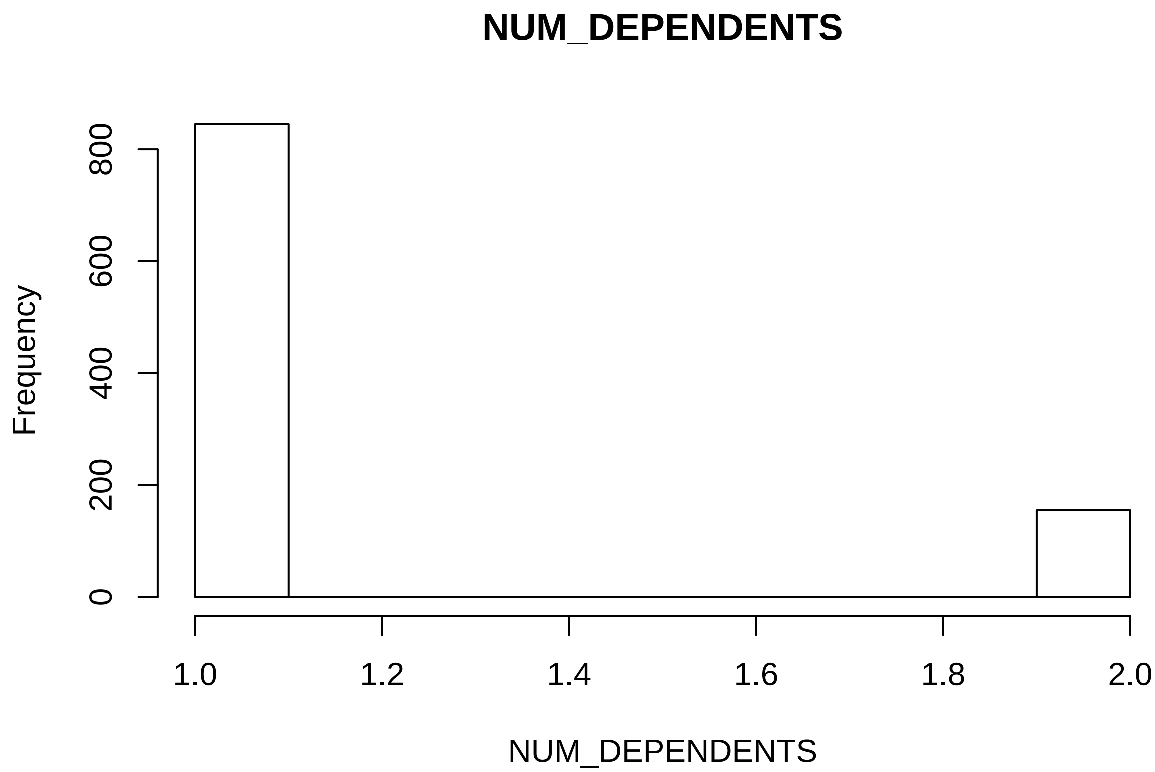

AMOUNT
Frequency
0
100
200
300
400
0
5000
10000
15000
20000
AMOUNT



INSTALL_RATE
Frequency
0
100
200
300
400
1.0
1.5
2.0
2.5
3.0
3.5
4.0
INSTALL_RATE


AGE
Frequency
0
50
100
150
200
20
30
40
50
60
70
AGE


NUM_CREDITS
Frequency
0
100
300
500
1.0
1.5
2.0
2.5
3.0
3.5
4.0
NUM_CREDITS

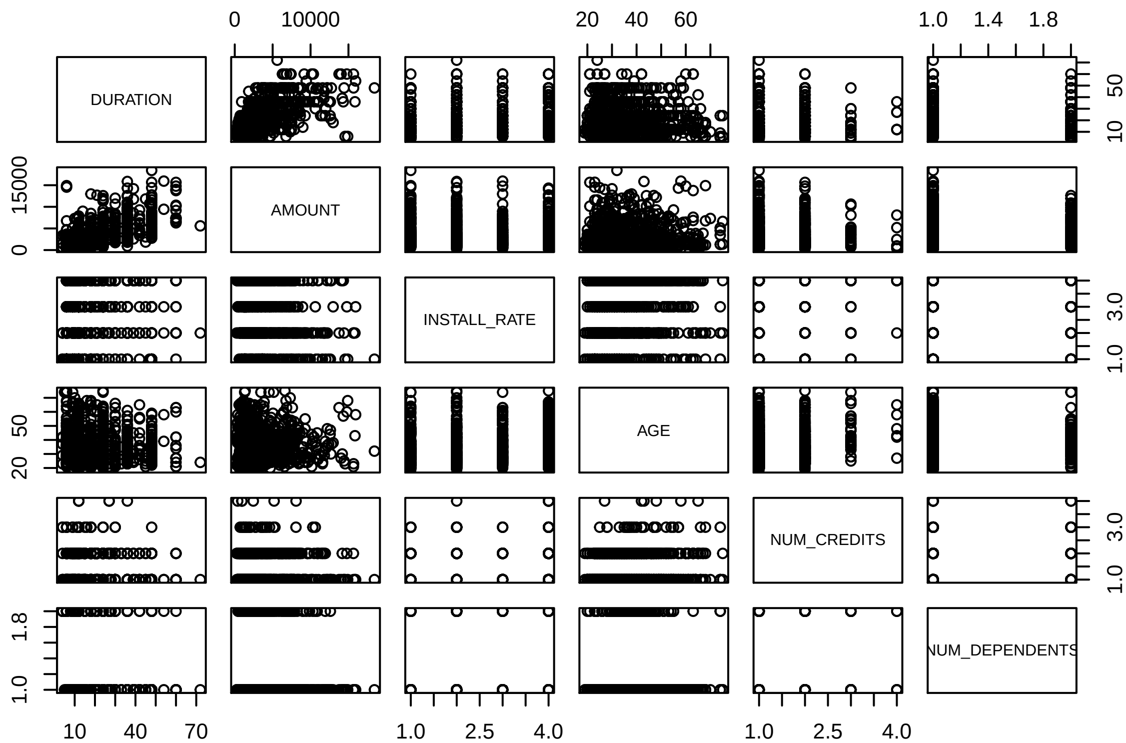


Correlation between num variables

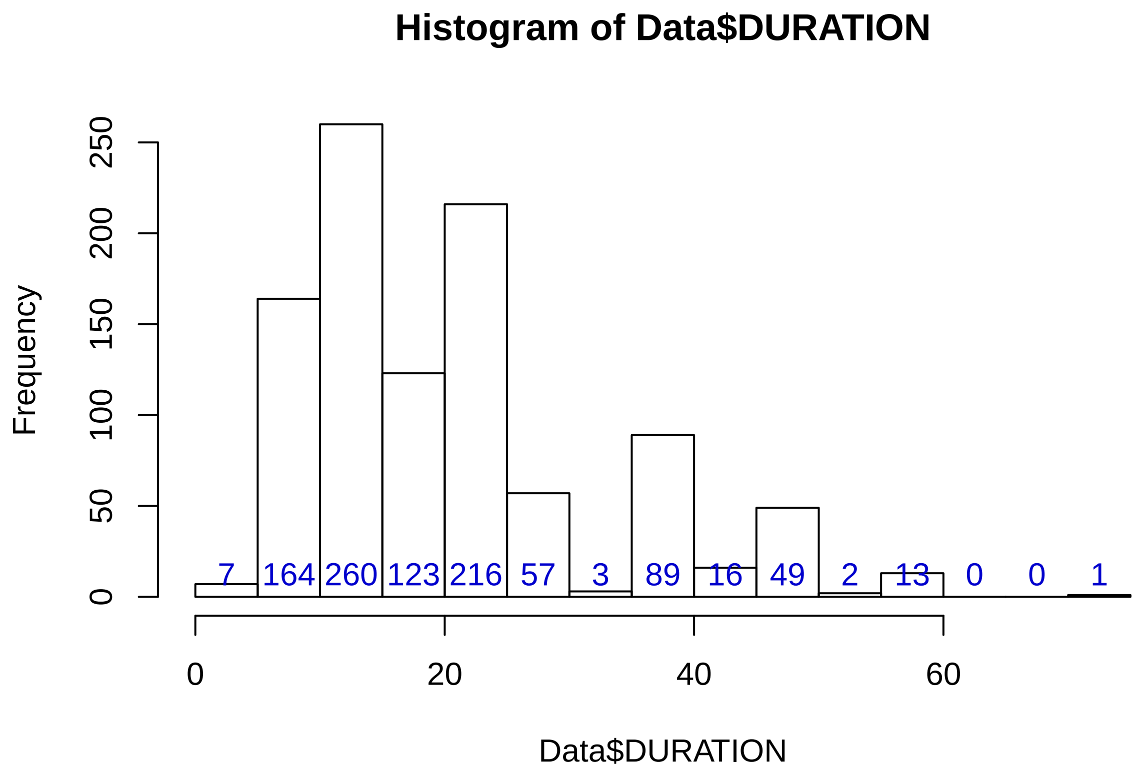
```
cor(num_data)
```

```
##                  DURATION      AMOUNT INSTALL_RATE         AGE NUM_CREDITS
## DURATION        1.00000000  0.62498420   0.07474882 -0.03613637 -0.01128360
## AMOUNT          0.62498420  1.00000000  -0.27131570  0.03271642  0.02079455
## INSTALL_RATE    0.07474882 -0.27131570   1.00000000  0.05826568  0.02166874
## AGE            -0.03613637  0.03271642   0.05826568  1.00000000  0.14925358
## NUM_CREDITS    -0.01128360  0.02079455   0.02166874  0.14925358  1.00000000
## NUM_DEPENDENTS -0.02383448  0.01714215  -0.07120694  0.11820083  0.10966670
##                NUM_DEPENDENTS
## DURATION          -0.02383448
## AMOUNT             0.01714215
## INSTALL_RATE      -0.07120694
## AGE                0.11820083
## NUM_CREDITS        0.10966670
## NUM_DEPENDENTS     1.00000000
```

```
pairs(num_data)
```




```r
#save histogram value
r<-hist(Data$DURATION)
text(r$mids, r$density, r$counts, adj = c(.5, -.5), col = "blue3")
```



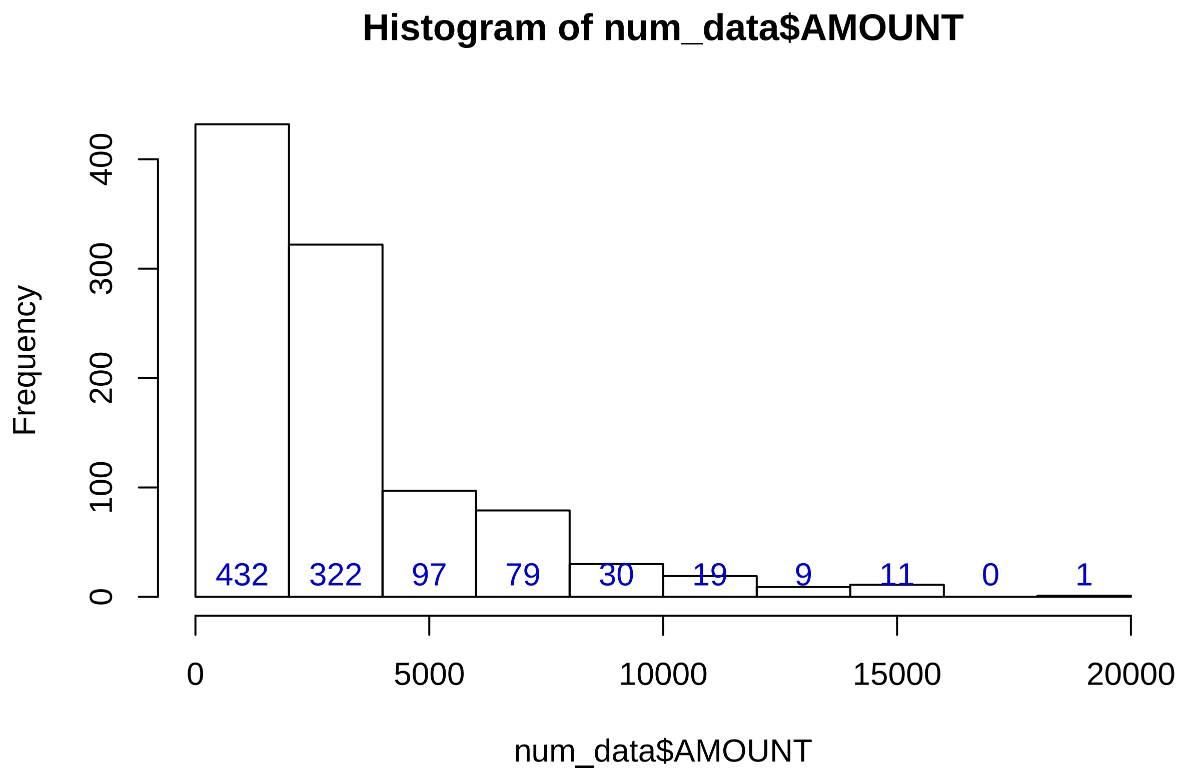

```r
sapply(r[2:3], sum)
```

```
##  counts density
##   1e+03   2e-01
```

```r
sum(r$density * diff(r$breaks)) # == 1
```

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

```r
r<-hist(num_data$AMOUNT)
text(r$mids, r$density, r$counts, adj = c(.5, -.5), col = "blue3")
```



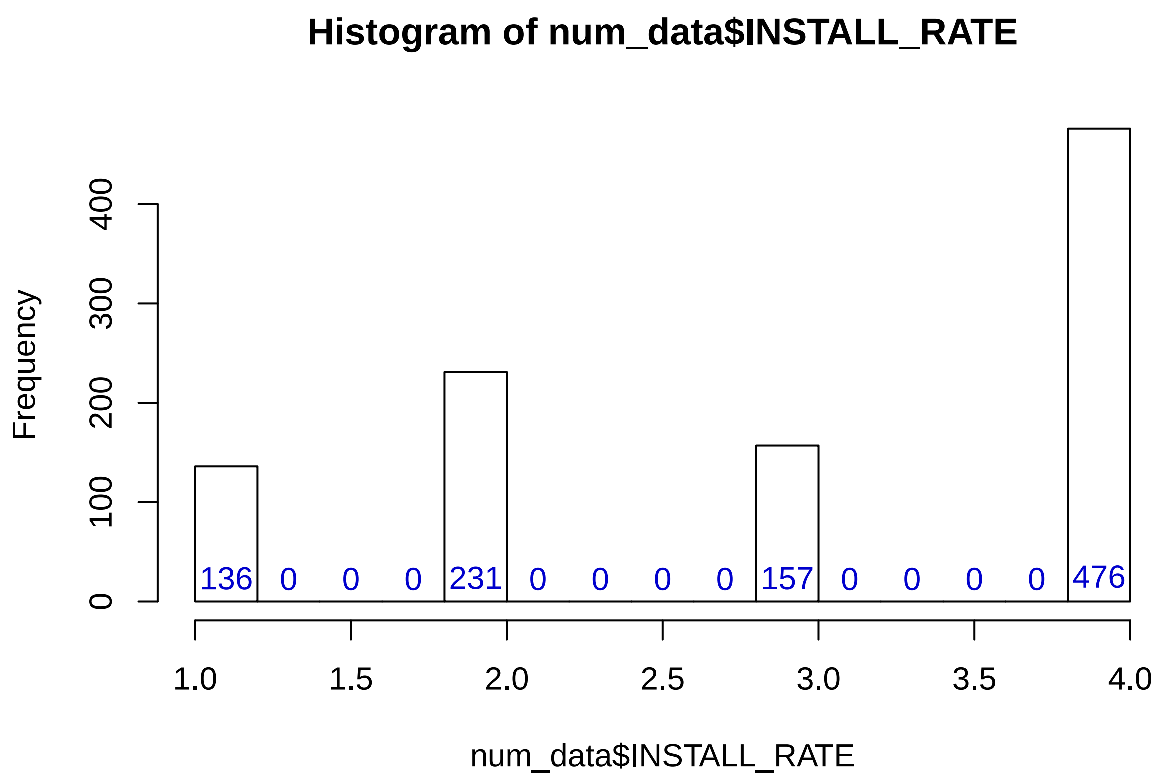

```r
sapply(r[2:3], sum)
```

```
##   counts density
##    1e+03   5e-04
```

```r
sum(r$density * diff(r$breaks)) # == 1
```

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

```r
r<-hist(num_data$INSTALL_RATE)
text(r$mids, r$density, r$counts, adj = c(.5, -.5), col = "blue3")
```



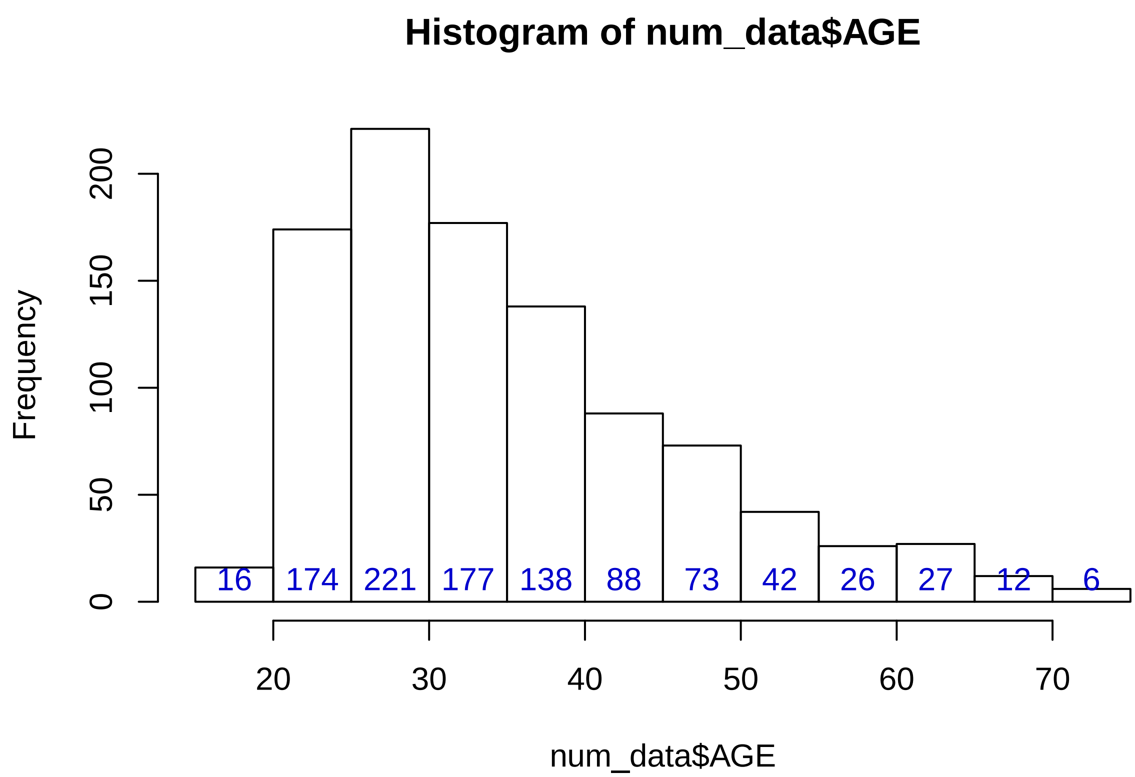

```r
sapply(r[2:3], sum)
```

```
##   counts density
##     1000       5
```

```r
sum(r$density * diff(r$breaks)) # == 1
```

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

```r
r<-hist(num_data$AGE)
text(r$mids, r$density, r$counts, adj = c(.5, -.5), col = "blue3")
```



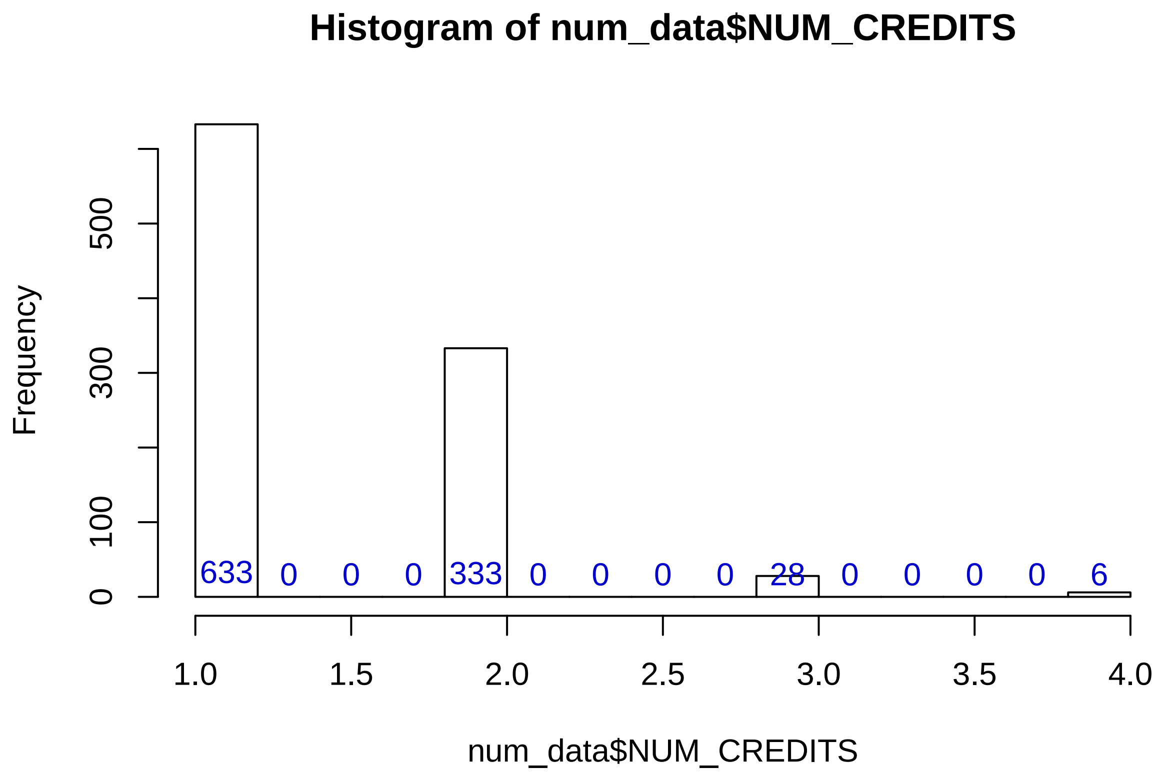

```r
sapply(r[2:3], sum)
```

```
##   counts density
##    1e+03   2e-01
```

```r
sum(r$density * diff(r$breaks)) # == 1
```

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

```r
r<-hist(num_data$NUM_CREDITS)
text(r$mids, r$density, r$counts, adj = c(.5, -.5), col = "blue3")
```



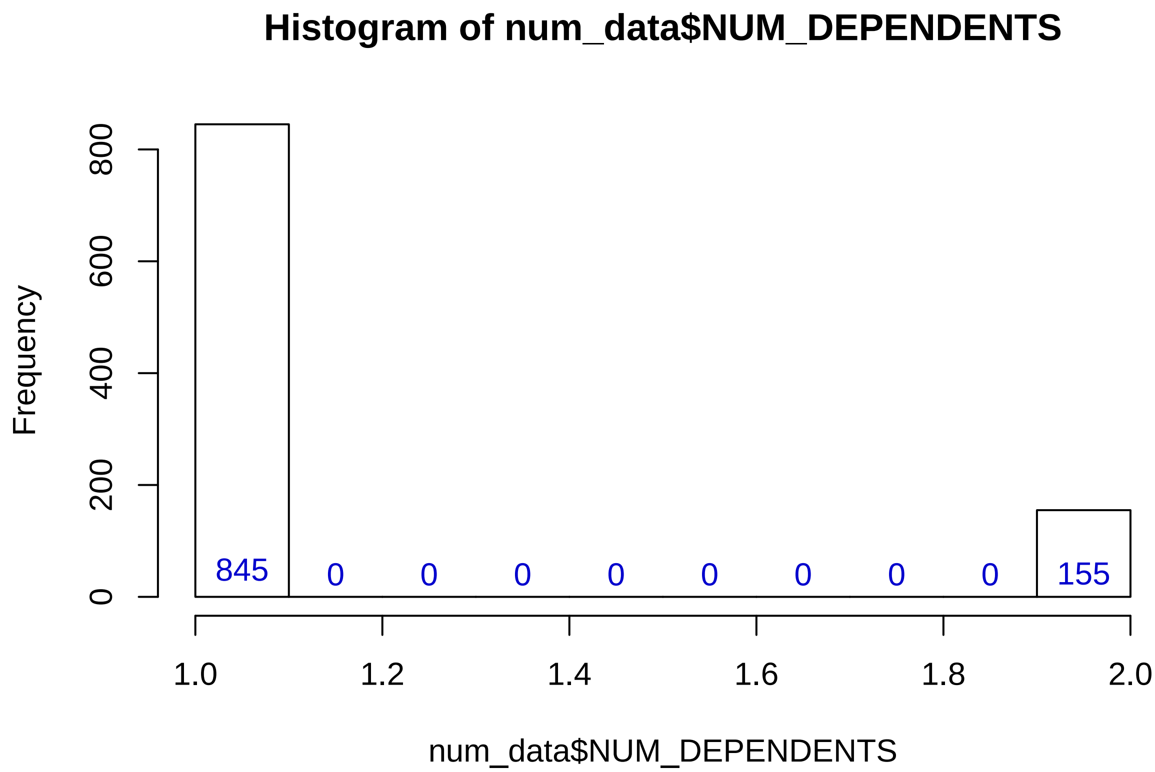

```
sapply(r[2:3], sum)
```

```
##  counts density
##    1000       5
```

```
sum(r$density * diff(r$breaks)) # == 1
```

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

```
r<-hist(num_data$NUM_DEPENDENTS)
text(r$mids, r$density, r$counts, adj = c(.5, -.5), col = "blue3")
```




```r
sapply(r[2:3], sum)
```

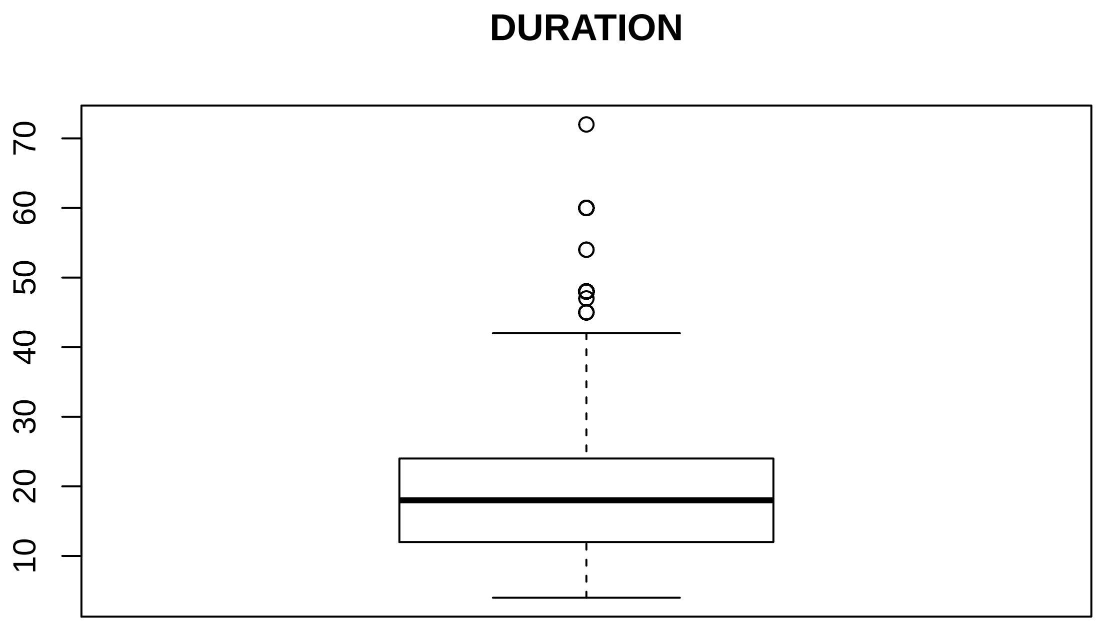
```
##  counts density
##    1000      10
```

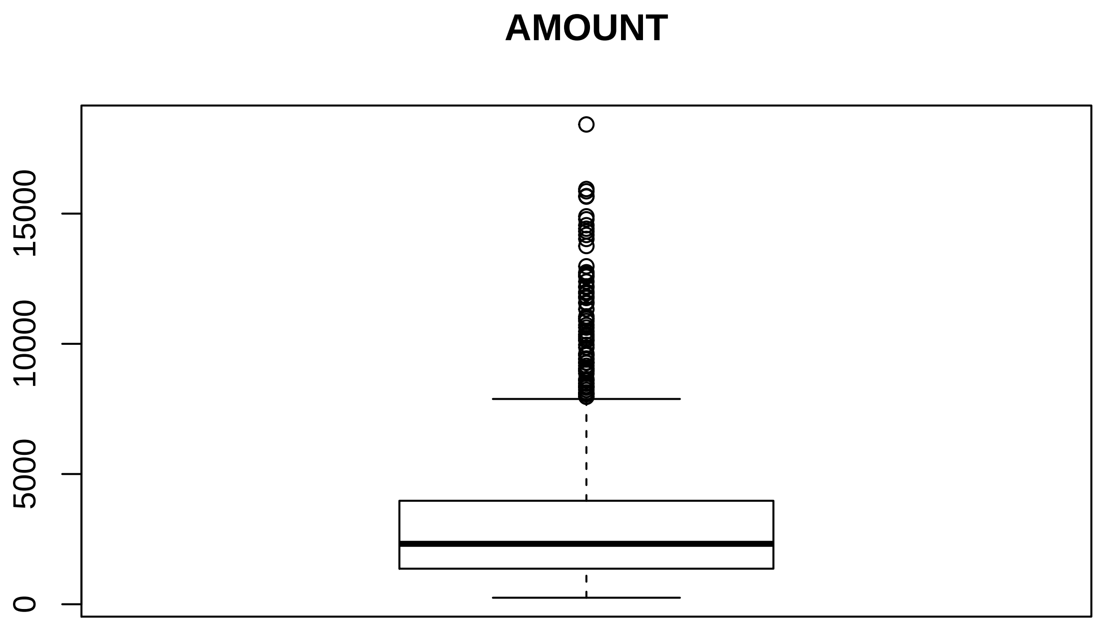
```r
sum(r$density * diff(r$breaks)) # == 1
```

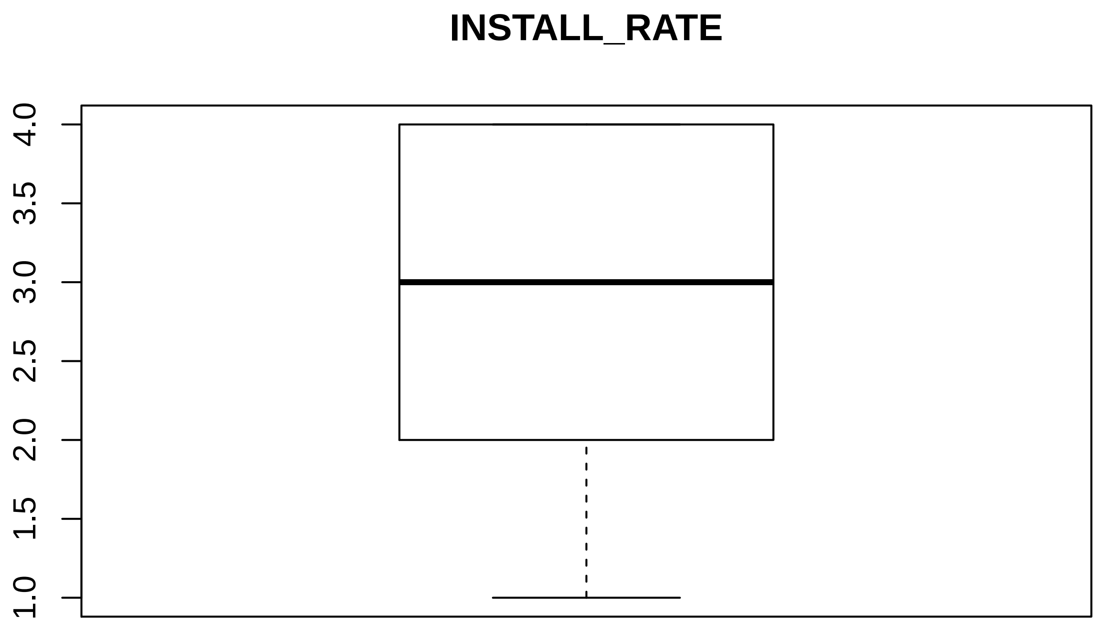
```
## [1] 1
```

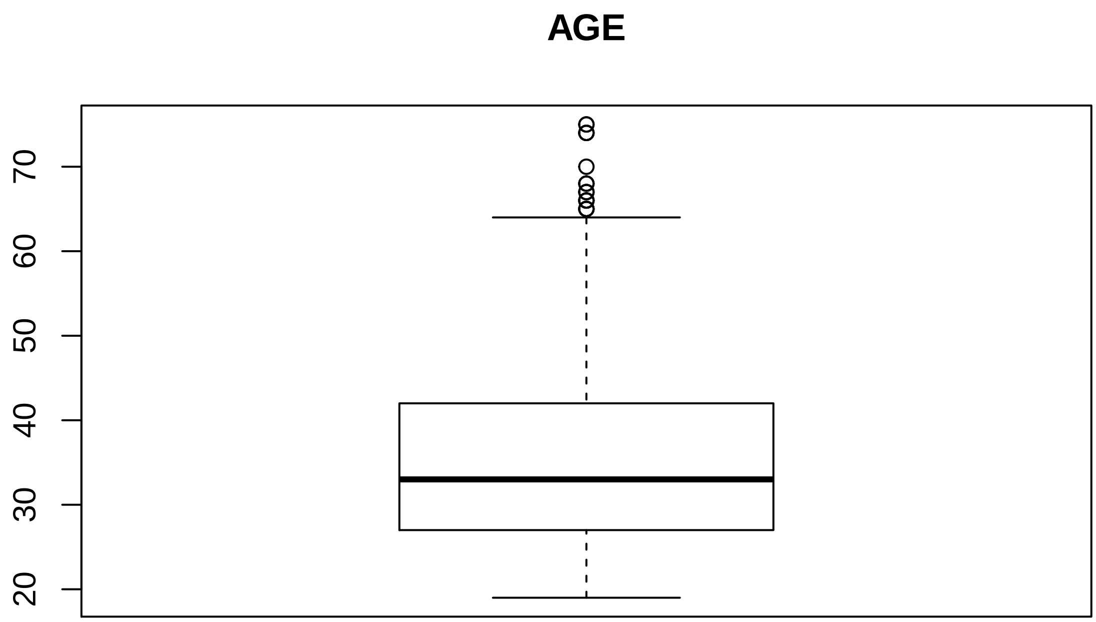
Boxplot for checking outliers

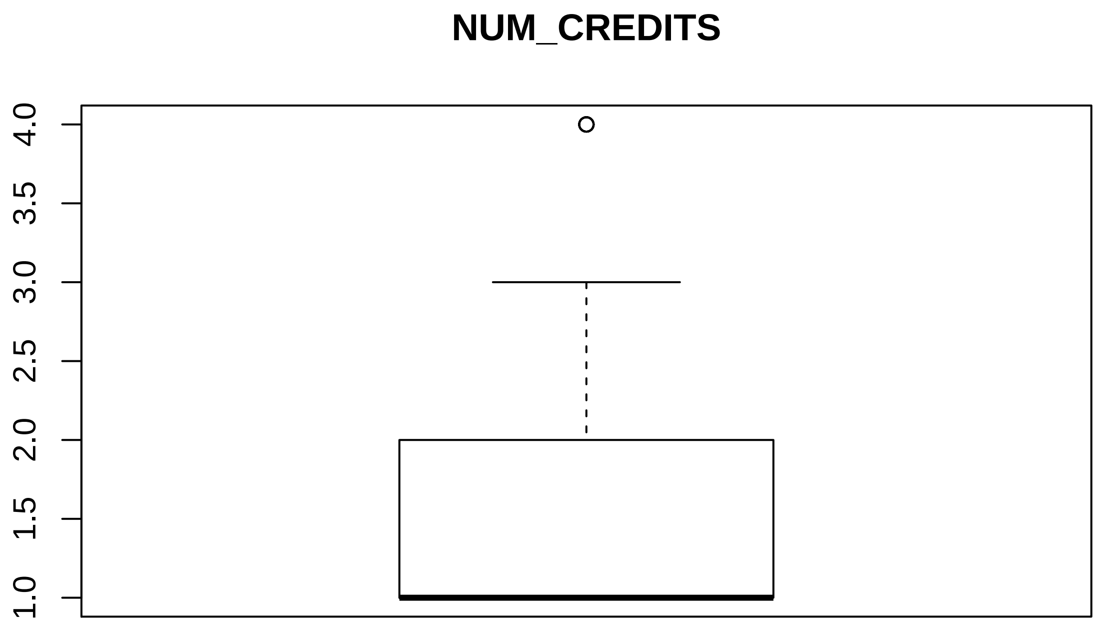
```r
for (i in 1:ncol(num_data)) {boxplot(num_data[[i]],main=colnames(num_data[i]))
}
```

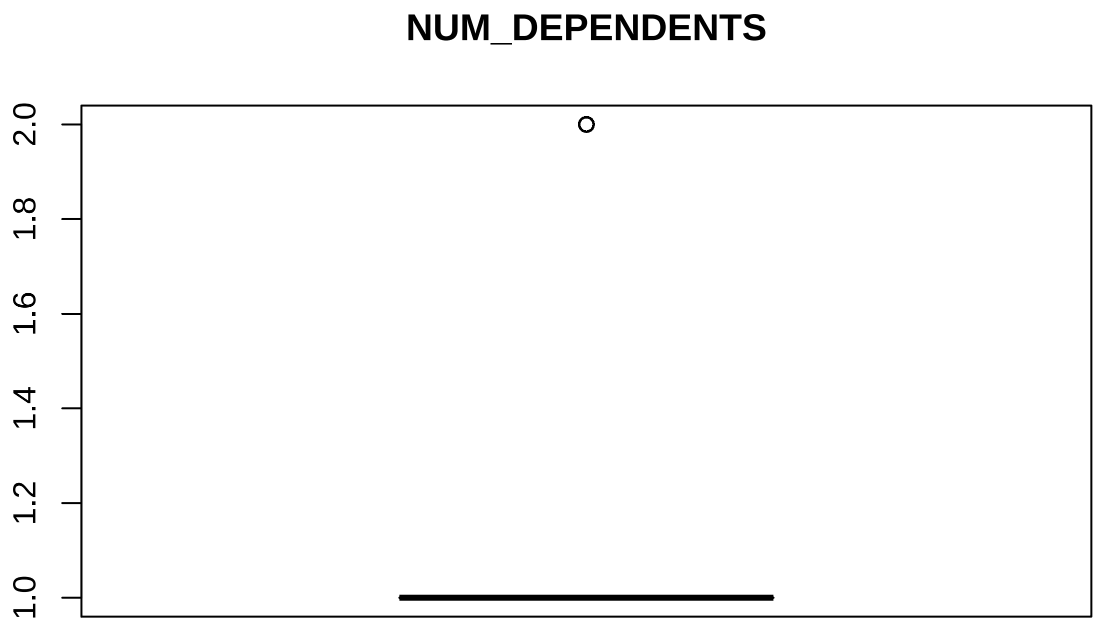

DURATION
70
60
50
40
30
20
10


AMOUNT
0
5000
10000
15000


INSTALL_RATE
1.0
1.5
2.0
2.5
3.0
3.5
4.0

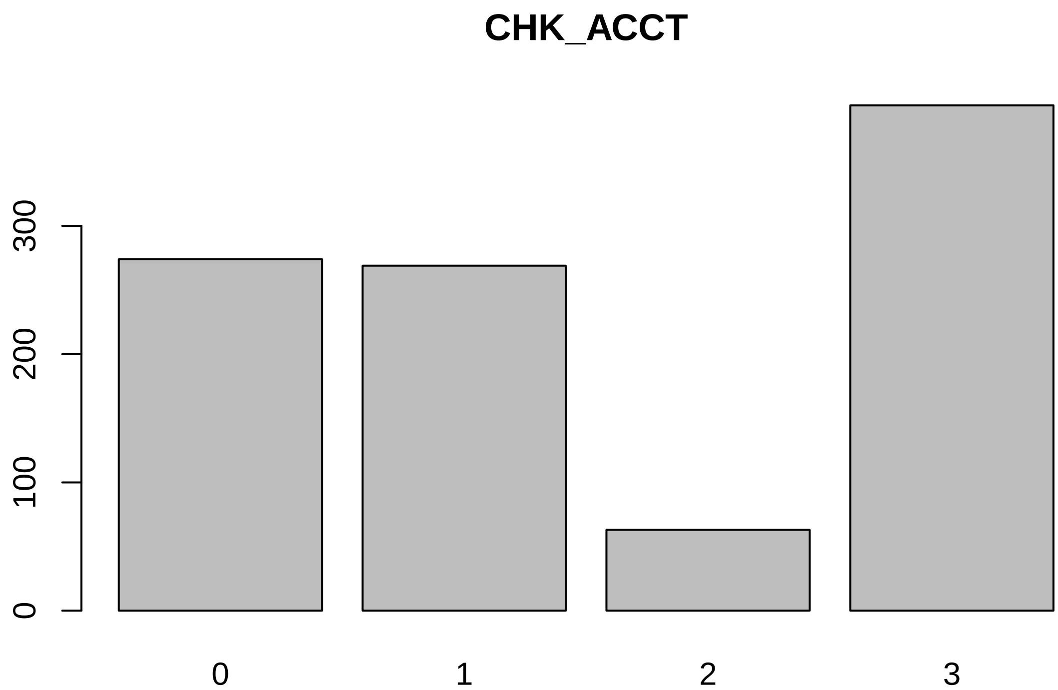

AGE
20
30
40
50
60
70

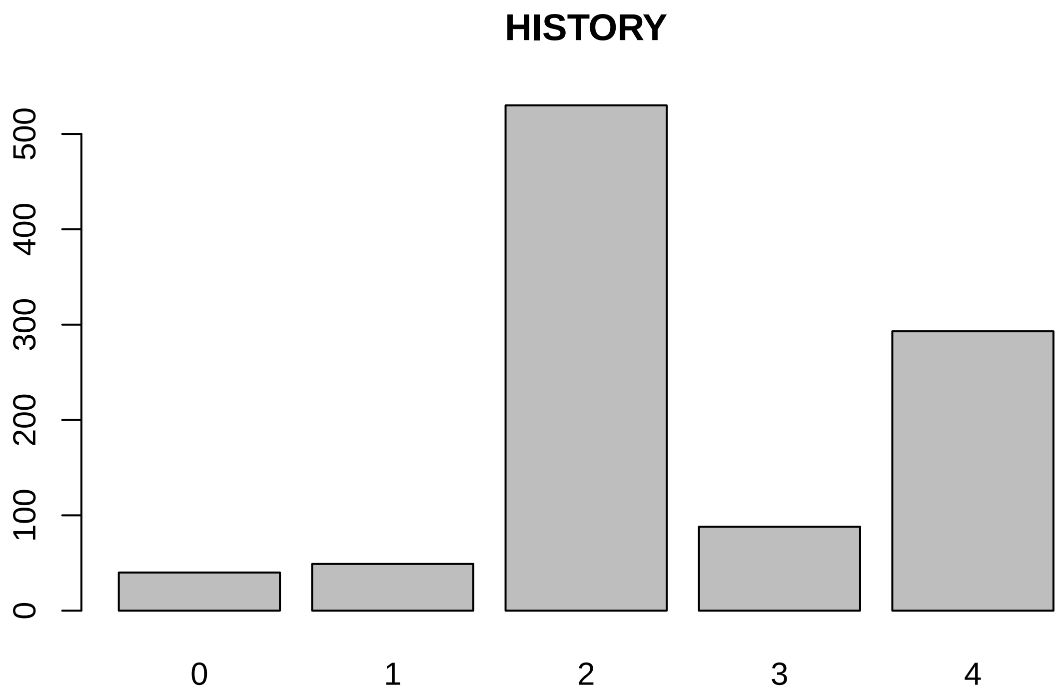

NUM_CREDITS
4.0
3.5
3.0
2.5
2.0
1.5
1.0

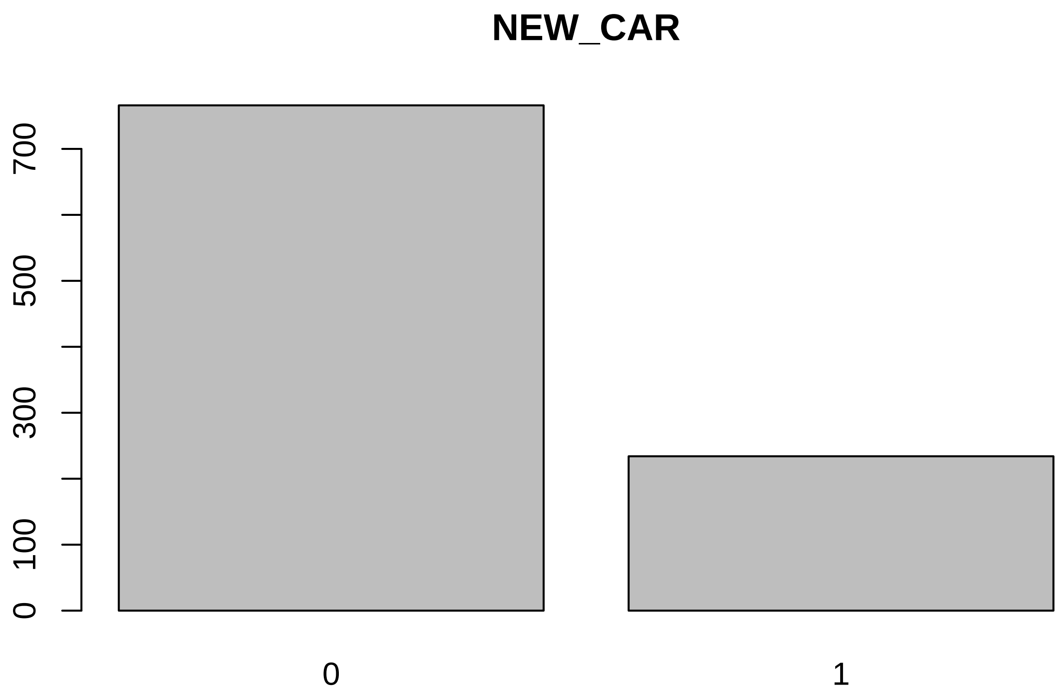


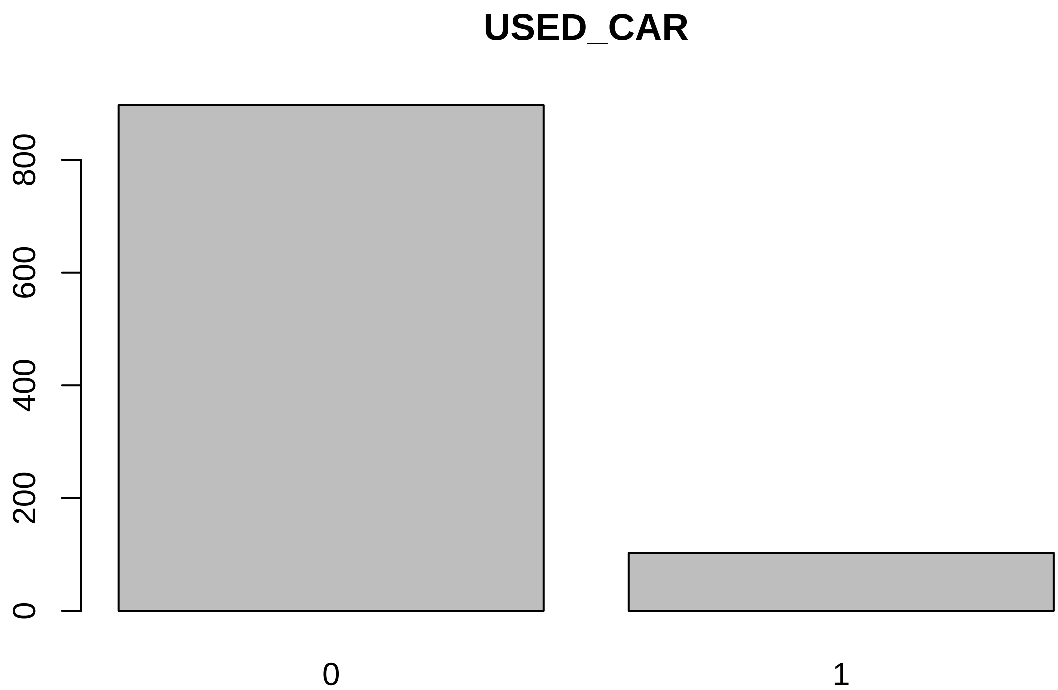
Barplot for categorical variable

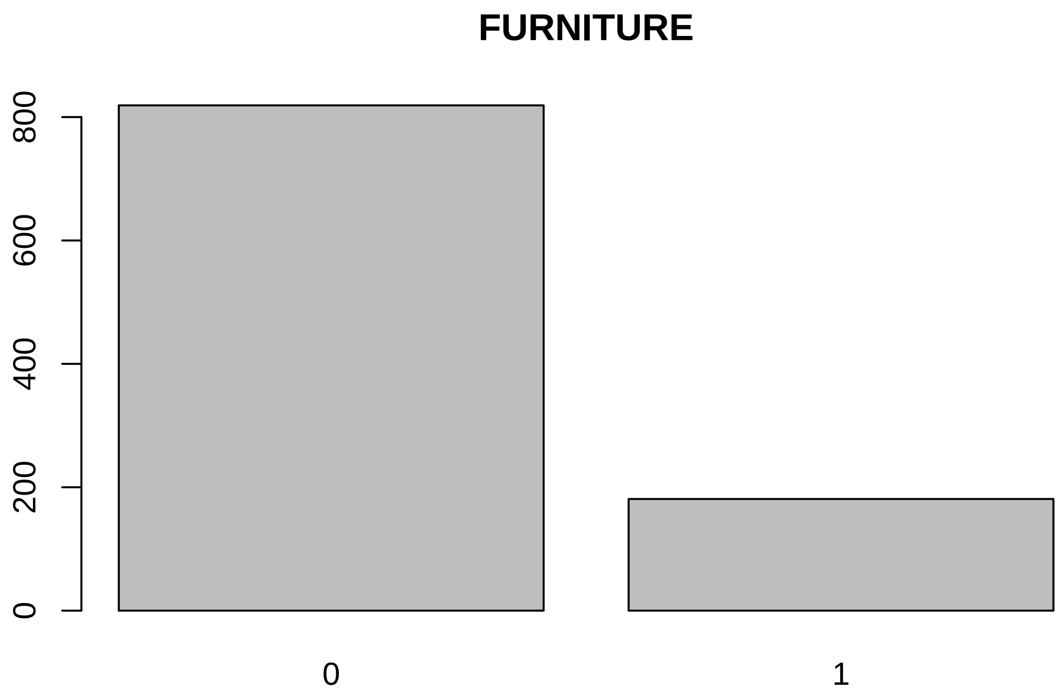
```
categorical<-Data[,-c(2,10,13,22,26,28)]
for (i in 1:ncol(categorical)) {barplot(table(categorical[[i]]),main=colnames(categorical[i]))
}
```

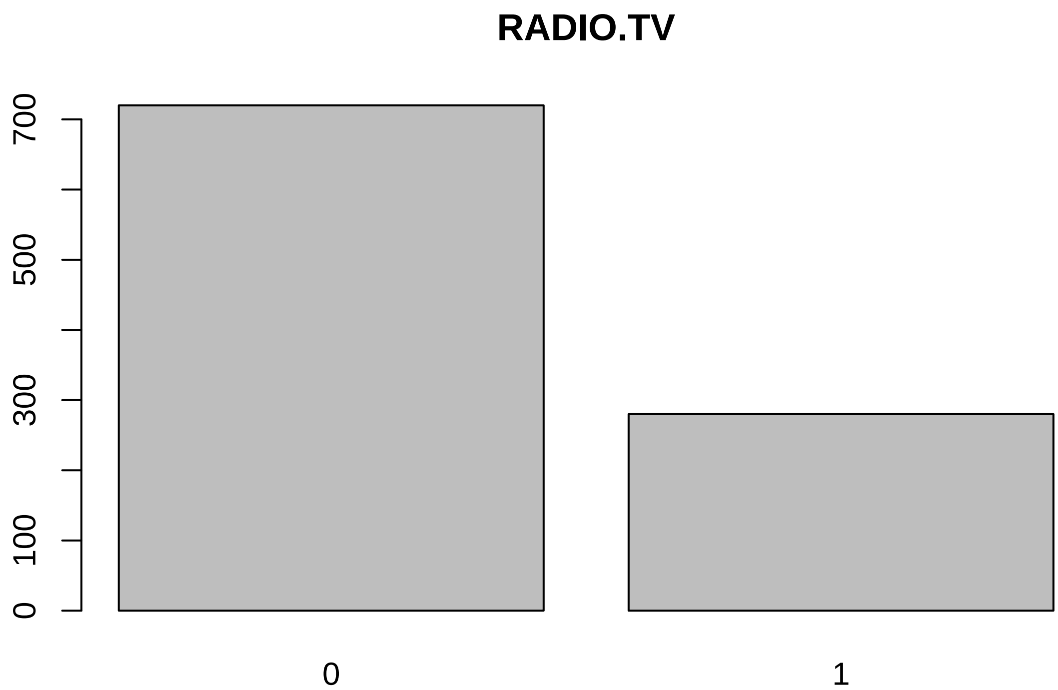

CHK_ACCT
0
100
200
300
0
1
2
3

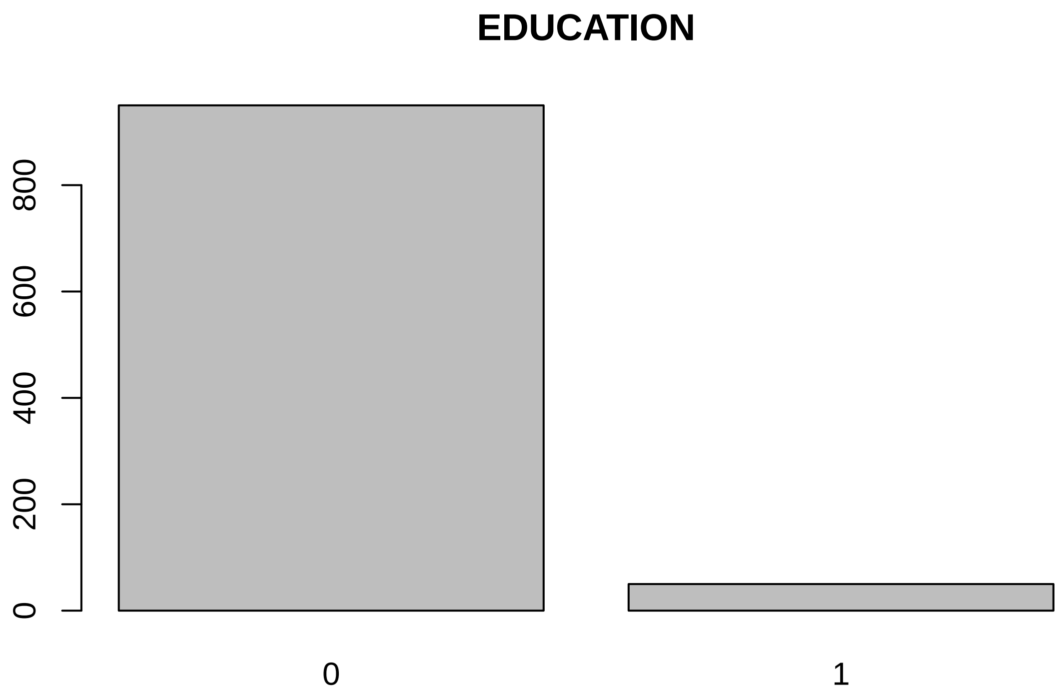

HISTORY
0
100
200
300
400
500
0
1
2
3
4

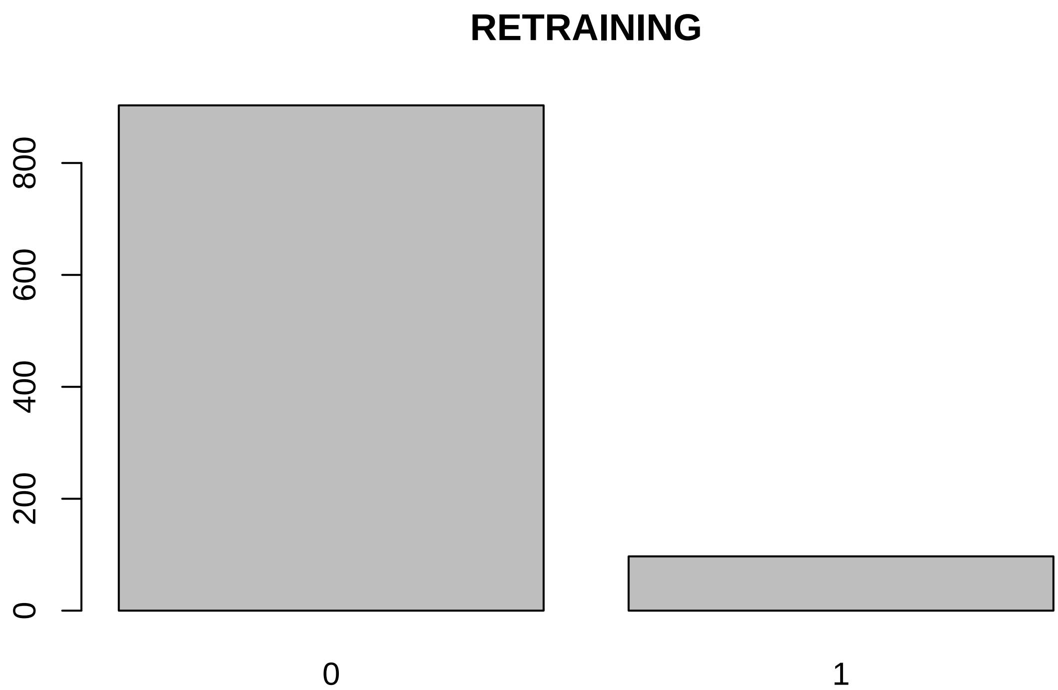

NEW_CAR
0
100
300
500
700
0
1

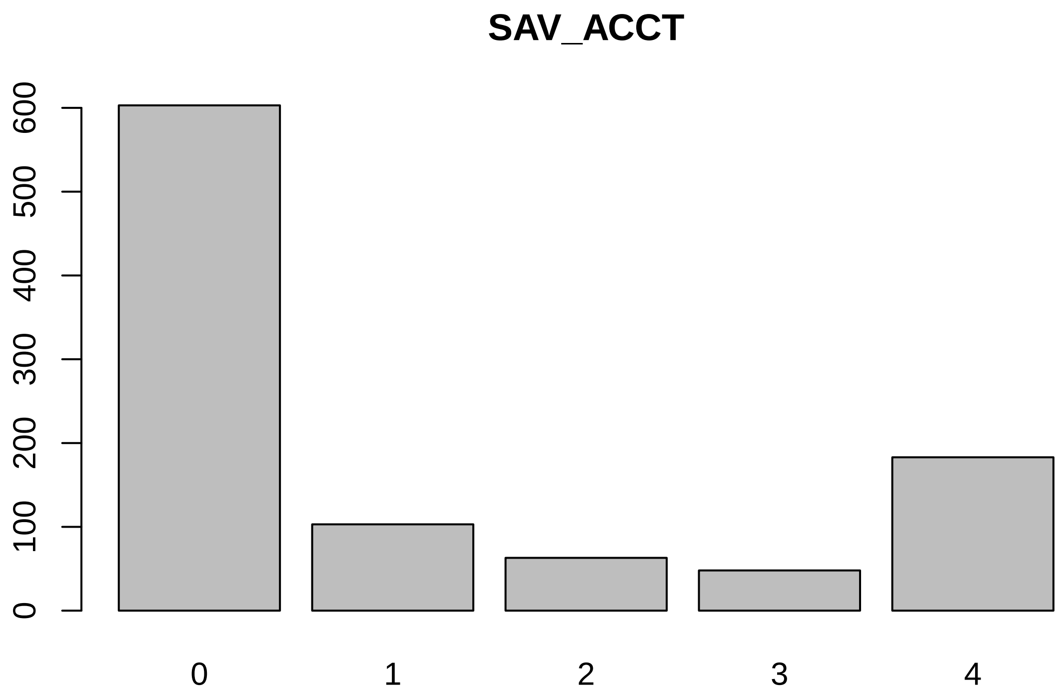

USED_CAR
0
200
400
600
800
0
1

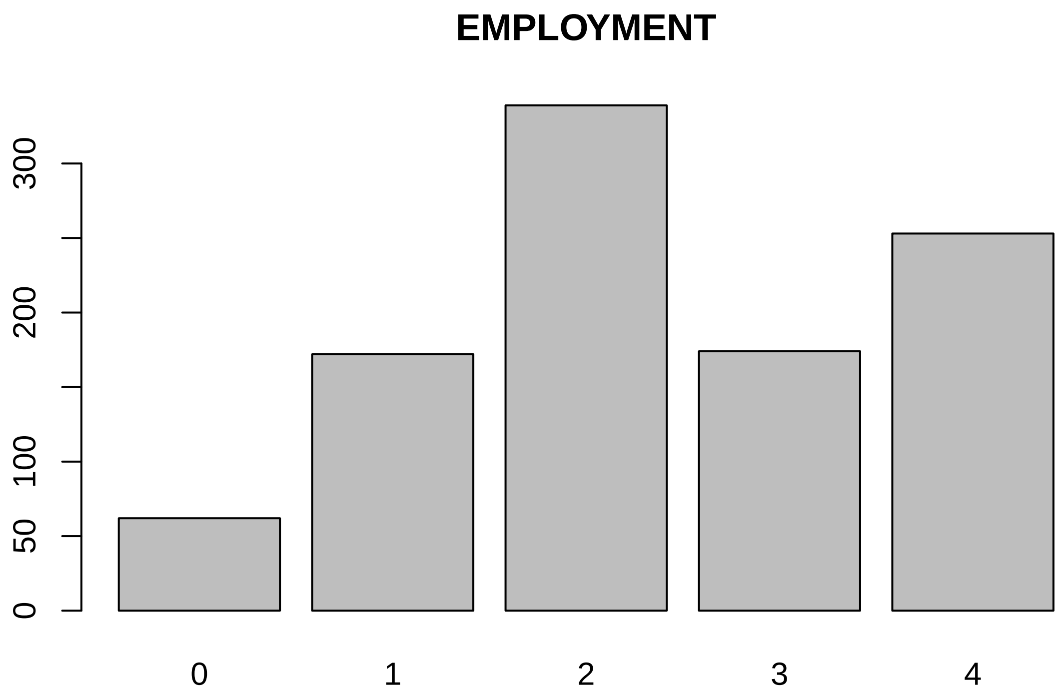

FURNITURE
800
600
400
200
0
0
1


RADIO.TV
0
100
300
500
700
0
1

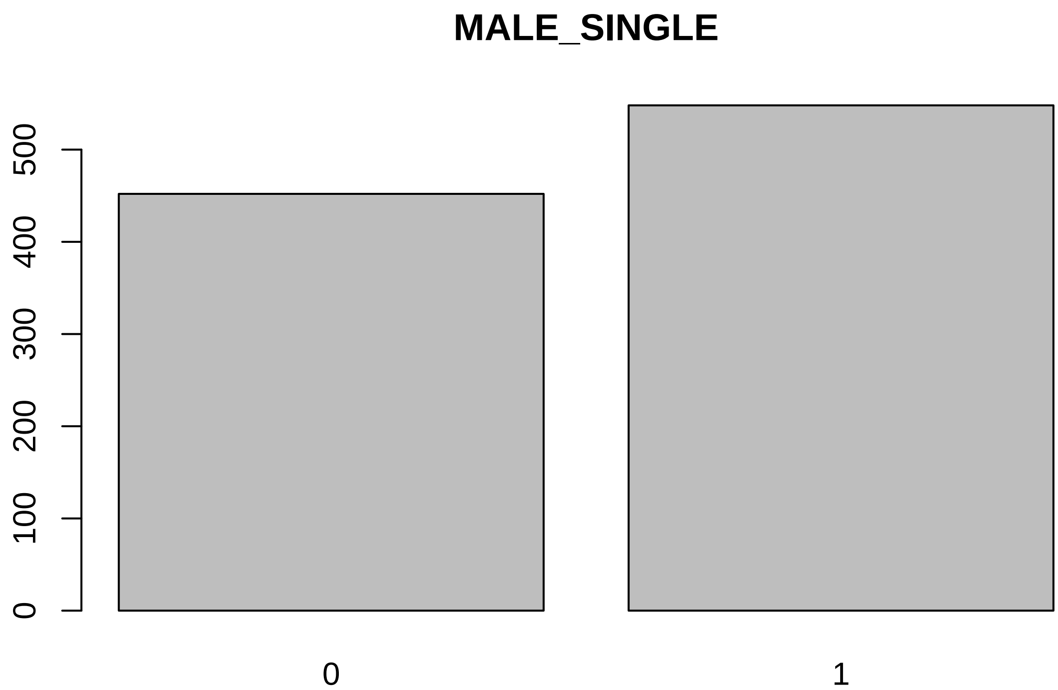

EDUCATION
800
600
400
200
0
0
1


RETRAINING
800
600
400
200
0
0
1


SAV_ACCT
0
100
200
300
400
500
600
0
1
2
3
4

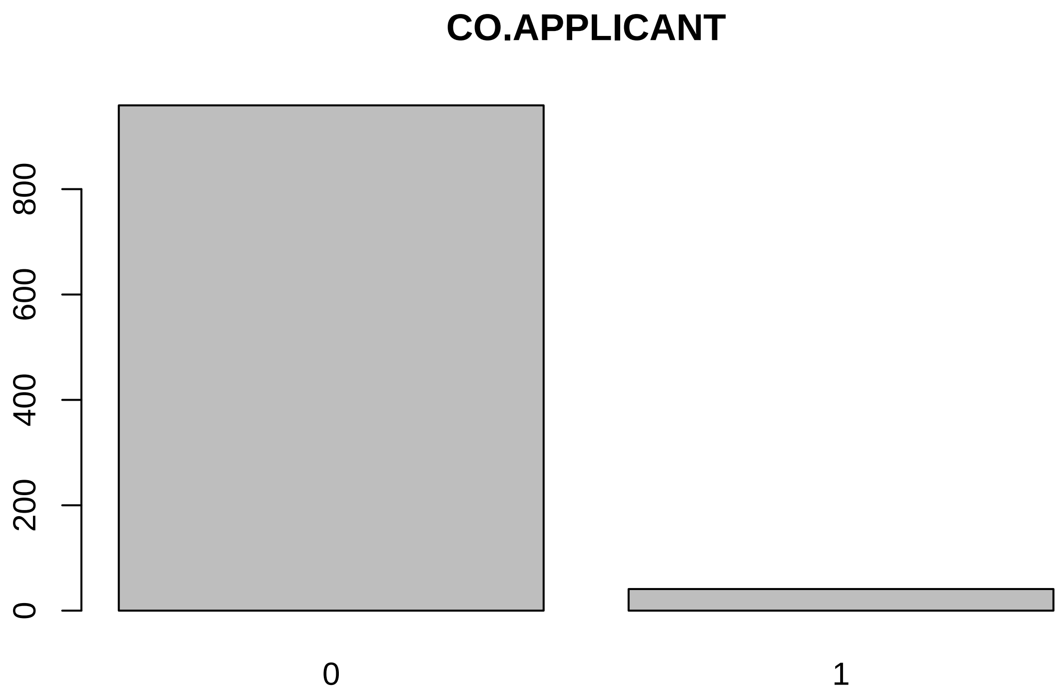

EMPLOYMENT
0
50
100
200
300
0
1
2
3
4

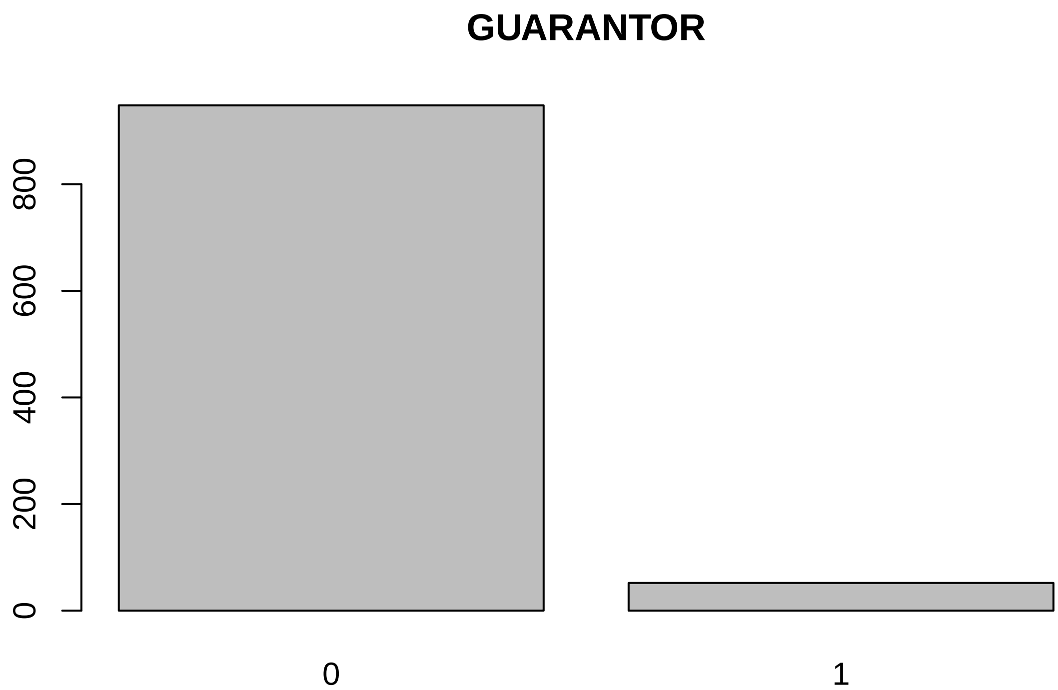

MALE_DIV
0
200
400
600
800
0
1

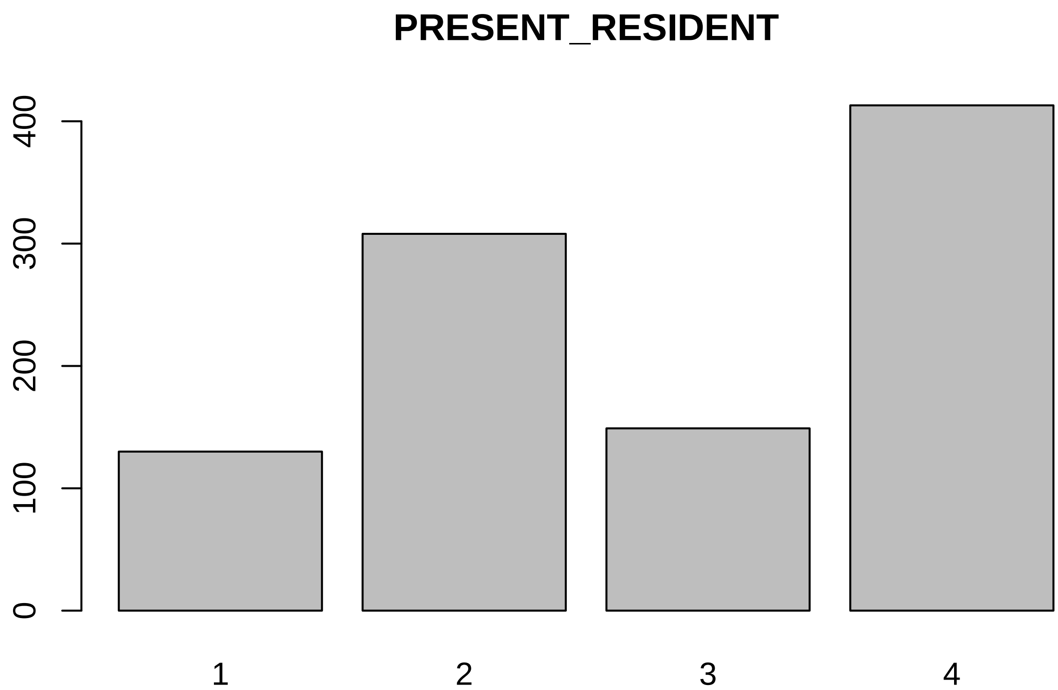


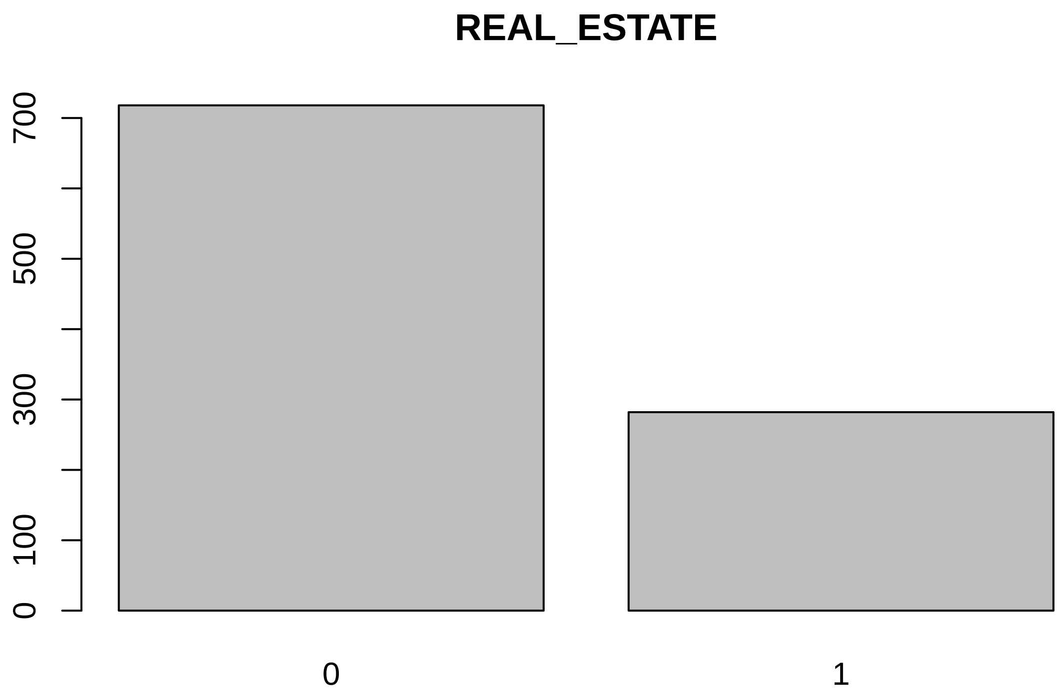
MALE_SINGLE
0
100
200
300
400
500
0
1

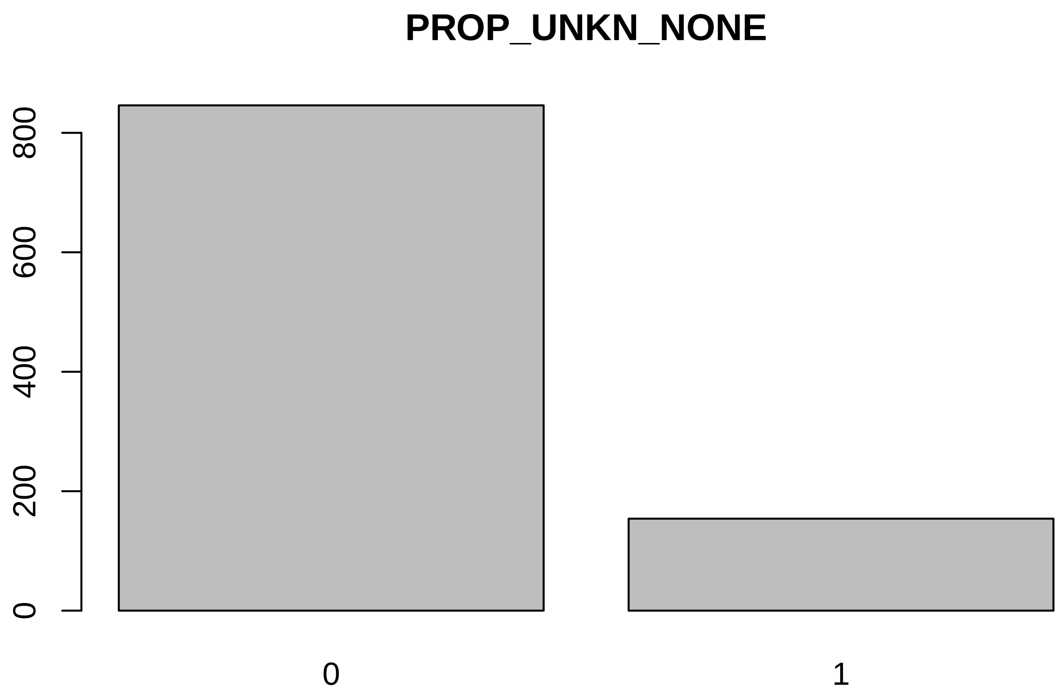

MALE_MAR_or_WID
0
200
400
600
800
0
1

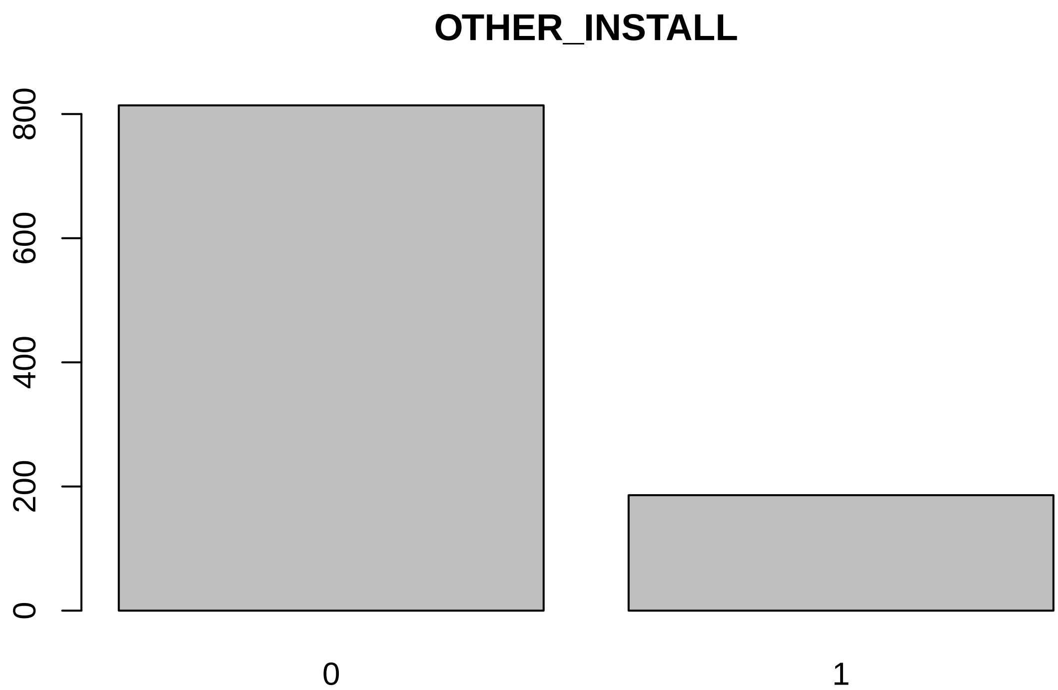

CO.APPLICANT
800
600
400
200
0
0
1

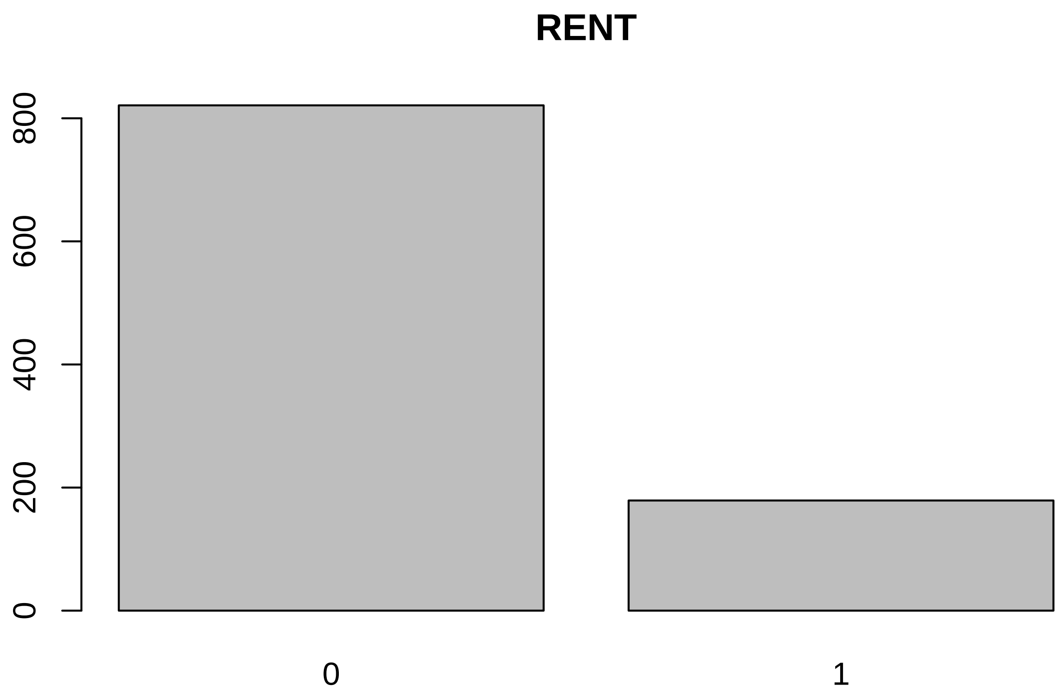

GUARANTOR
800
600
400
200
0
0
1

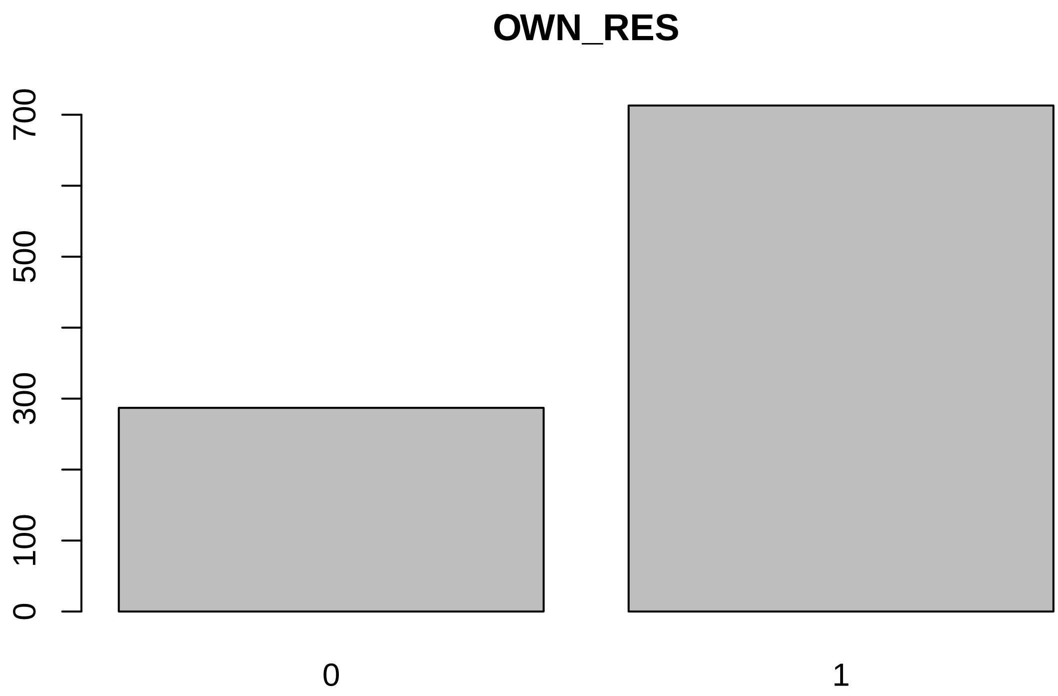

PRESENT_RESIDENT
0
100
200
300
400
1
2
3
4

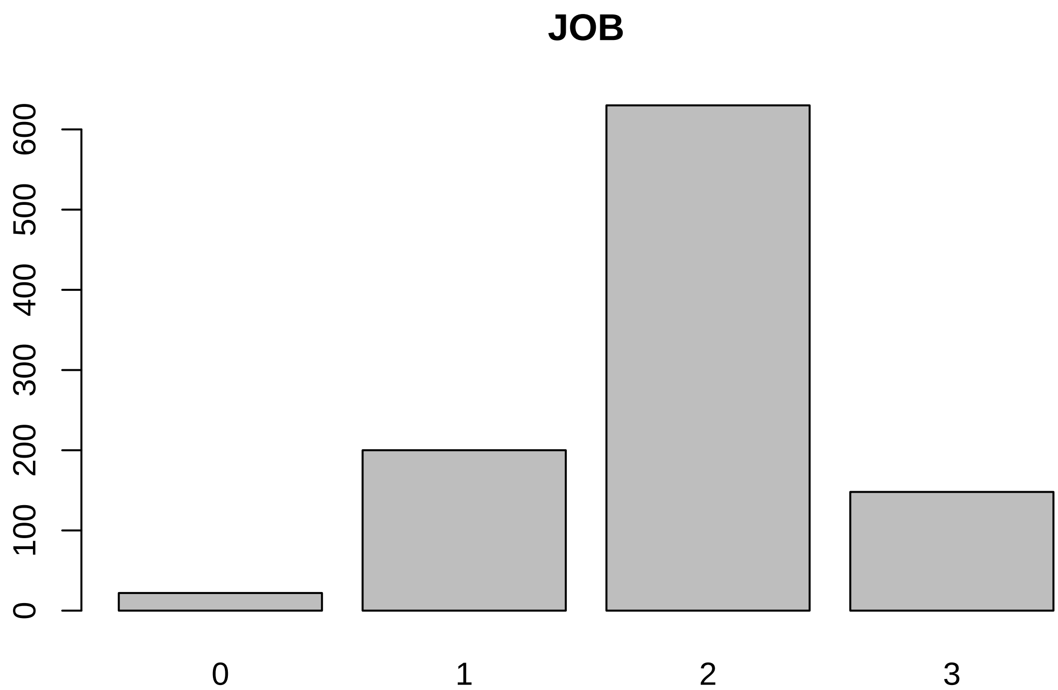

REAL_ESTATE
0
100
300
500
700
0
1

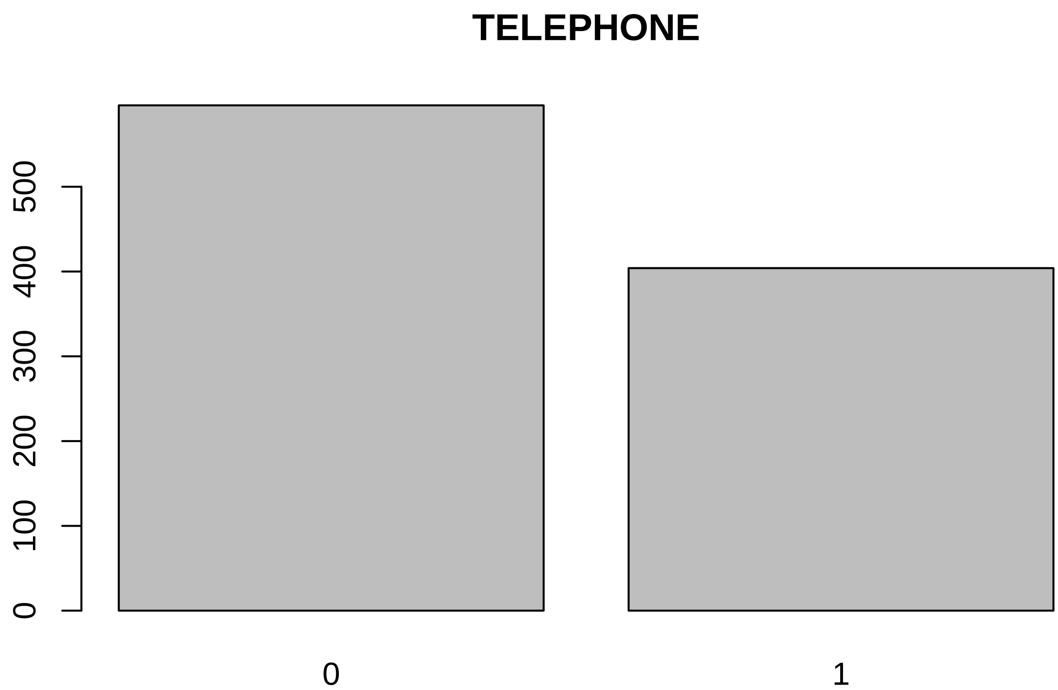

PROP_UNKN_NONE
0
200
400
600
800
0
1

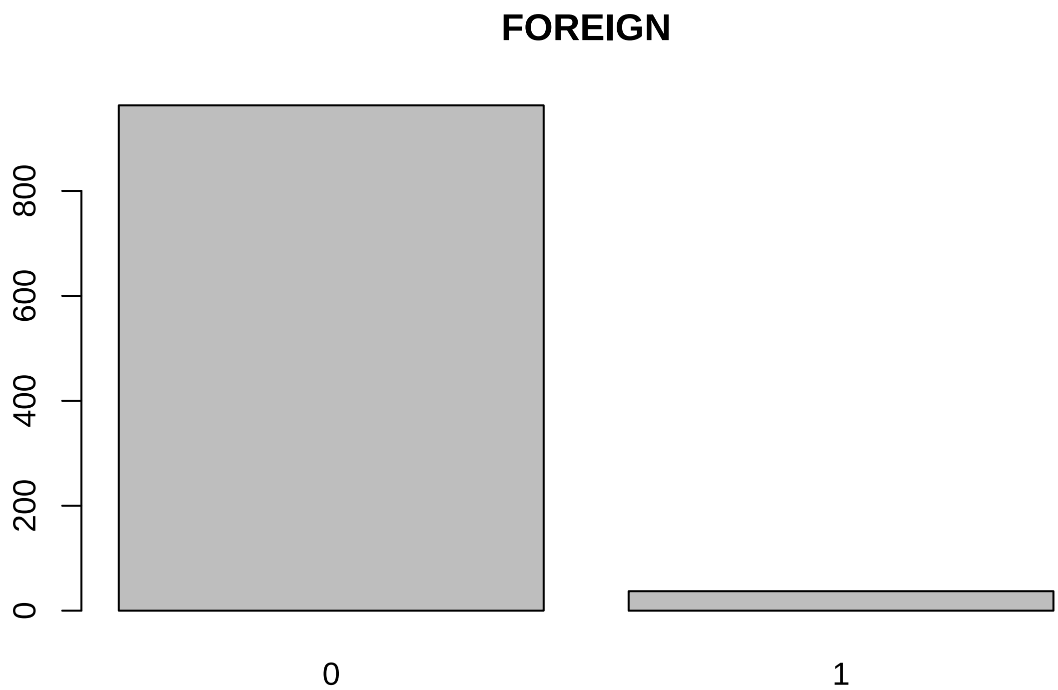

OTHER_INSTALL
800
600
400
200
0
0
1

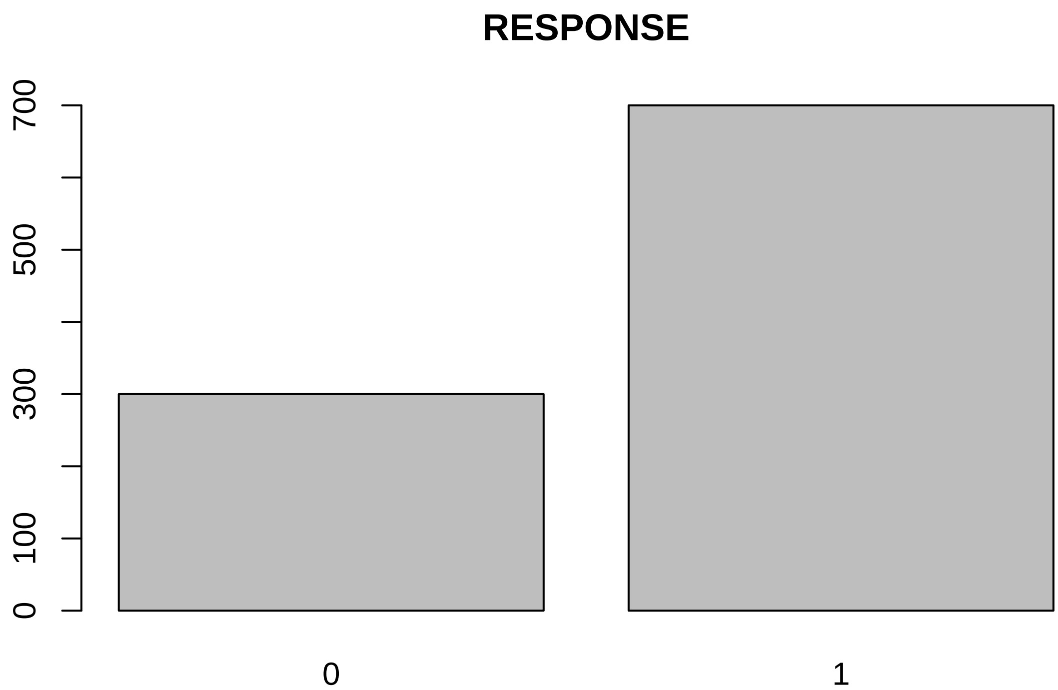

RENT
800
600
400
200
0
0
1


OWN_RES
0
100
300
500
700
0
1


JOB
0
100
200
300
400
500
600
0
1
2
3


TELEPHONE
500
400
300
200
100
0
0
1


FOREIGN
0
200
400
600
800
0
1



Categorical variable Analysis with respect to Response variable

```
cat("credit history vs responce")
```

```
## credit history vs responce
```

```
#aggregate.data.frame(Data$HISTORY,by=list(Data$RESPONSE),table)
by(Data$HISTORY,list(Data$RESPONSE),table)
```

```
## : 0
##
##   0   1   2   3   4
##  25  28 169  28  50
## ------------------------------------------------------------
## : 1
##
##   0   1   2   3   4
##  15  21 361  60 243
```

```
cat("Education vs responce")
```

```
## Education vs responce
```

```r
by(Data$EDUCATION,list(Data$RESPONSE),table)
```

```
## : 0
##
##   0   1
## 278  22
## ------------------------------------------------------------
## : 1
##
##   0   1
## 672  28
```

```r
cat("saving account vs responce")
```

```
## saving account vs responce
```

```r
by(Data$SAV_ACCT,list(Data$RESPONSE),table)
```

```
## : 0
##
##   0   1   2   3   4
## 217  34  11   6  32
## ------------------------------------------------------------
## : 1
##
##   0   1   2   3   4
## 386  69  52  42 151
```

```r
cat("emploment vs responce")
```

```
## emploment vs responce
```

```r
by(Data$EMPLOYMENT,list(Data$RESPONSE),table)
```

```
## : 0
##
##   0   1   2   3   4
##  23  70 104  39  64
## ------------------------------------------------------------
## : 1
##
##   0   1   2   3   4
##  39 102 235 135 189
```

```r
cat("owns real estate vs responce")
```

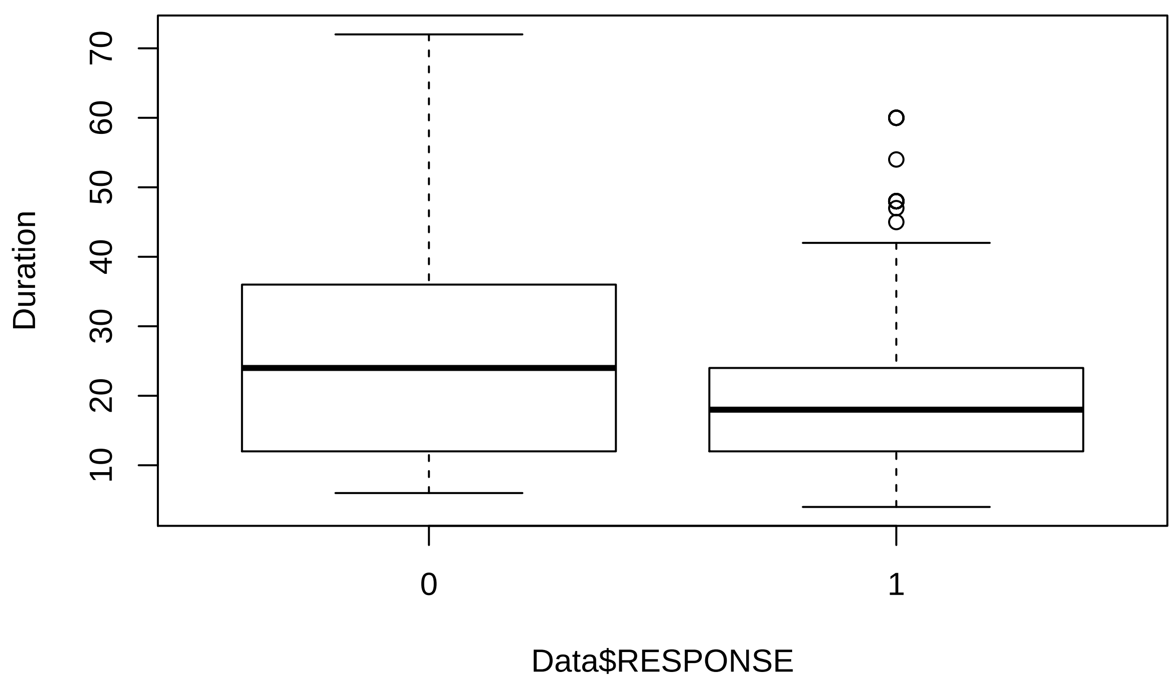
```
## owns real estate vs responce
```

```r
by(Data$REAL_ESTATE,list(Data$RESPONSE),table)
```

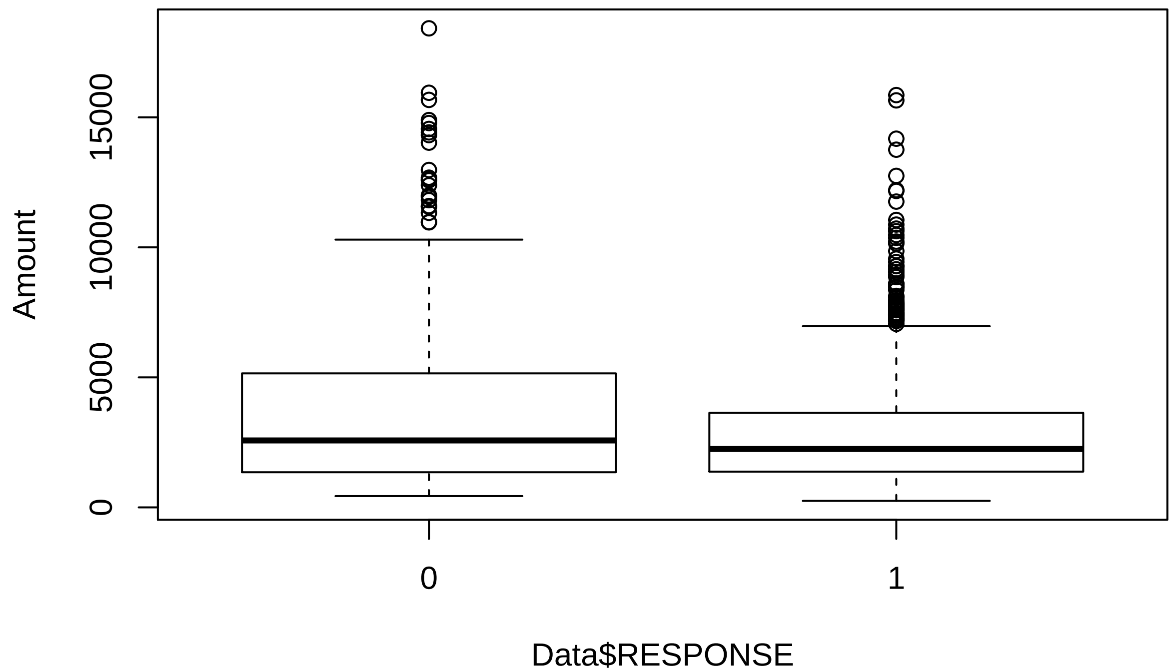
```
## : 0
##
##   0   1
## 240  60
## ------------------------------------------------------------
## : 1
##
##   0   1
## 478 222
```

Numerical variable anlysis with respect to response variable

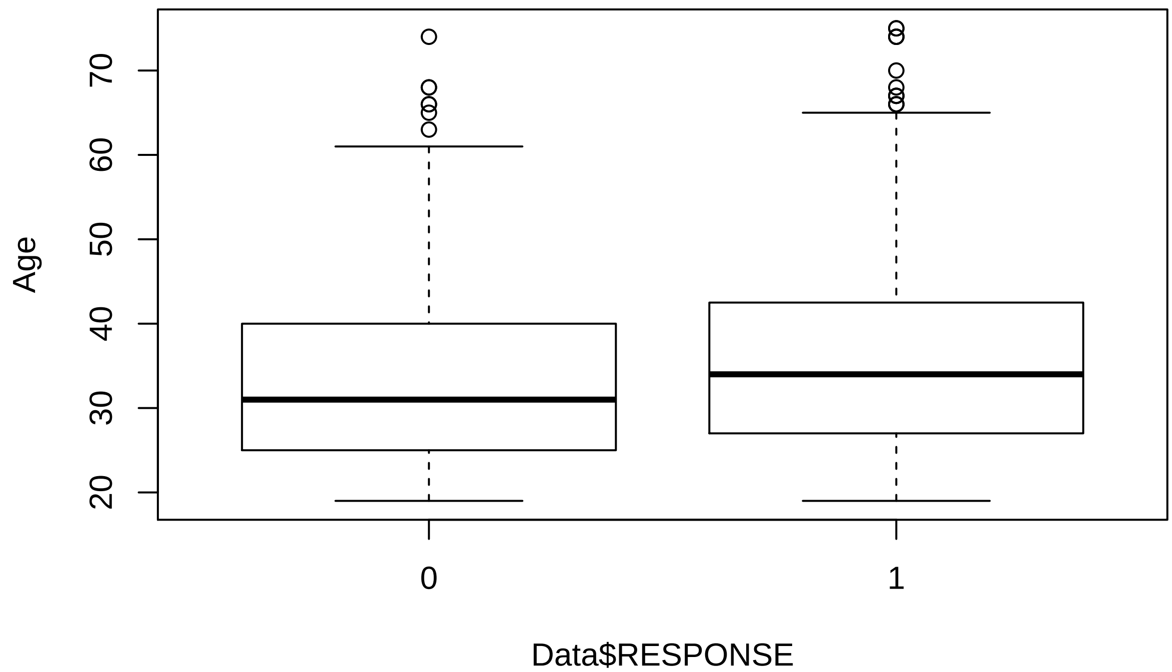
```r
boxplot(num_data$DURATION~Data$RESPONSE,ylab="Duration")
```




```r
boxplot(num_data$AMOUNT~Data$RESPONSE,ylab="Amount")
```




```
#boxplot(num_data$INSTALL_RATE~Data$RESPONSE,ylab="installment" )
boxplot(num_data$AGE~Data$RESPONSE,ylab="Age")
```




```r
#boxplot(num_data$NUM_CREDITS~Data$RESPONSE,ylab="NUM_CREDITS")
#boxplot(num_data$NUM_DEPENDENTS~Data$RESPONSE,ylab="NUM_DEPENDENTS")
```

Feature Selection using randomForest()

```r
library(randomForest)
```

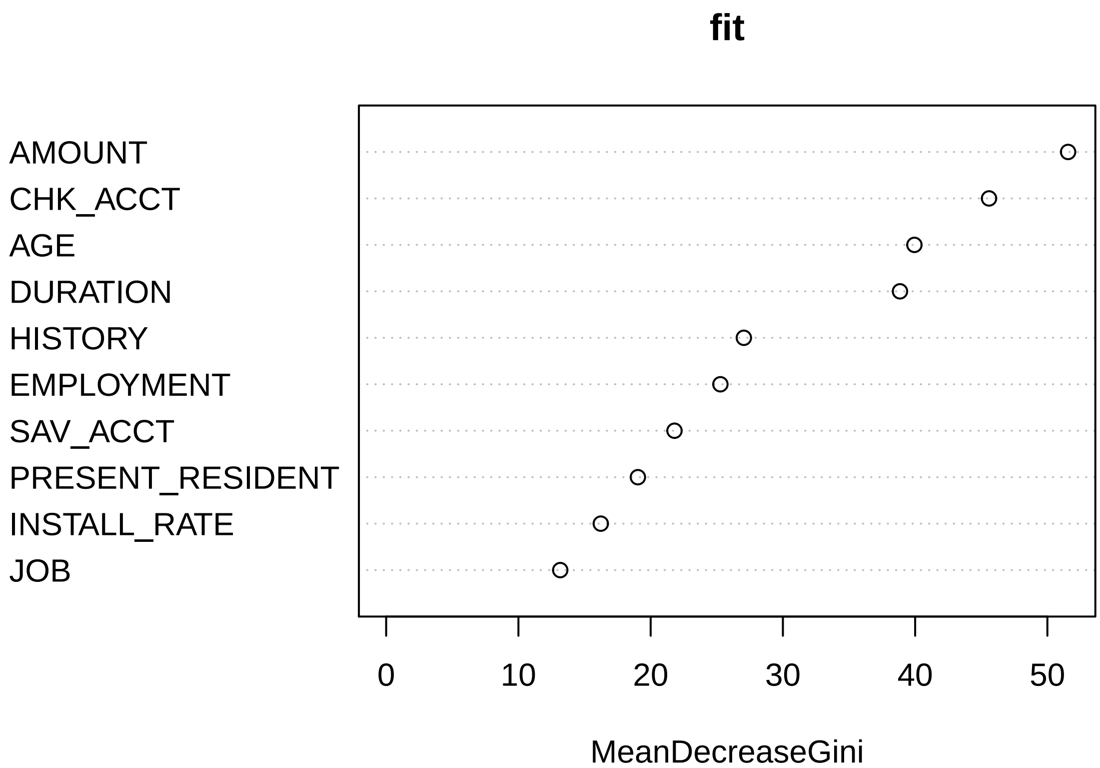
```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:psych':
##
##     outlier
```

```r
fit = randomForest(Data$RESPONSE ~., data = Data)
varImpPlot(fit,n.var=10)
```




```r
importance(fit)
```

```
##                   MeanDecreaseGini
## CHK_ACCT                 45.588171
## DURATION                 38.853502
## HISTORY                  27.045860
## NEW_CAR                   8.285138
## USED_CAR                  4.539689
## FURNITURE                 5.493281
## RADIO.TV                  5.873967
## EDUCATION                 3.995256
## RETRAINING                3.974351
## AMOUNT                   51.565864
## SAV_ACCT                 21.802129
## EMPLOYMENT               25.272832
## INSTALL_RATE             16.229811
## MALE_DIV                  3.494344
## MALE_SINGLE               7.134733
## MALE_MAR_or_WID           3.629730
## CO.APPLICANT              3.697698
## GUARANTOR                 3.873587
## PRESENT_RESIDENT         19.034461
## REAL_ESTATE               7.251261
## PROP_UNKN_NONE            5.516370
## AGE                      39.945083
```

```
## OTHER_INSTALL           8.596467
## RENT                    4.846237
## OWN_RES                 6.584980
## NUM_CREDITS             9.144857
## JOB                    13.163944
## NUM_DEPENDENTS          5.681426
## TELEPHONE               7.528464
## FOREIGN                 1.802749
```

Classification Model Building for prediction of good rating or bad rating

```r
#splitting data
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```r
set.seed(123)
trainDataIndex <- createDataPartition(Data$RESPONSE, p=0.7, list = F)  # 70% training data
trainData <- Data[trainDataIndex, ]
testData <- Data[-trainDataIndex, ]
prop.table(table(Data$RESPONSE))
```

```
##
##   0   1
## 0.3 0.7
```

```r
prop.table(table(trainData$RESPONSE))
```

```
##
##   0   1
## 0.3 0.7
```

```r
prop.table(table(testData$RESPONSE))
```

```
##
##   0   1
## 0.3 0.7
```

```r
#proportion of response variable is same in original and splitted data
```

Model 1

Logistic Regression

```r
attach(Data)
logit<-glm(RESPONSE~.,family = binomial,data = trainData)
summary(logit)#AIC: 688.62
```

```
##
## Call:
## glm(formula = RESPONSE ~ ., family = binomial, data = trainData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8040  -0.6404   0.3310   0.6683   2.8017
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         2.231e+00  1.555e+00   1.435 0.151396
## CHK_ACCT1           4.630e-01  2.624e-01   1.765 0.077594 .
## CHK_ACCT2           1.724e+00  5.366e-01   3.213 0.001314 **
## CHK_ACCT3           1.950e+00  2.864e-01   6.808 9.87e-12 ***
## DURATION           -3.286e-02  1.201e-02  -2.735 0.006238 **
## HISTORY1           -1.120e-01  7.049e-01  -0.159 0.873737
## HISTORY2            5.565e-01  5.453e-01   1.021 0.307417
## HISTORY3            9.956e-01  5.996e-01   1.660 0.096834 .
## HISTORY4            1.555e+00  5.487e-01   2.834 0.004600 **
## NEW_CAR1           -5.047e-01  5.065e-01  -0.997 0.319004
## USED_CAR1           1.308e+00  6.453e-01   2.027 0.042667 *
## FURNITURE1          1.077e-01  5.220e-01   0.206 0.836541
## RADIO.TV1           4.200e-01  5.095e-01   0.824 0.409731
## EDUCATION1         -6.484e-01  6.628e-01  -0.978 0.327944
## RETRAINING1        -1.925e-01  5.878e-01  -0.327 0.743321
## AMOUNT             -1.133e-04  5.823e-05  -1.946 0.051659 .
## SAV_ACCT1           3.121e-01  3.511e-01   0.889 0.373952
## SAV_ACCT2           6.882e-02  4.435e-01   0.155 0.876702
## SAV_ACCT3           1.318e+00  6.793e-01   1.940 0.052437 .
## SAV_ACCT4           1.206e+00  3.397e-01   3.549 0.000386 ***
## EMPLOYMENT1         5.023e-01  5.459e-01   0.920 0.357496
## EMPLOYMENT2         5.921e-01  5.191e-01   1.141 0.254057
## EMPLOYMENT3         1.089e+00  5.523e-01   1.971 0.048703 *
```

```
## EMPLOYMENT4         3.725e-01  5.270e-01   0.707 0.479630
## INSTALL_RATE       -3.137e-01  1.161e-01  -2.703 0.006864 **
## MALE_DIV1          -2.908e-01  4.901e-01  -0.593 0.552918
## MALE_SINGLE1        5.980e-01  2.730e-01   2.191 0.028459 *
## MALE_MAR_or_WID1    2.867e-02  3.745e-01   0.077 0.938985
## CO.APPLICANT1      -8.263e-01  4.807e-01  -1.719 0.085593 .
## GUARANTOR1          9.202e-01  4.970e-01   1.852 0.064076 .
## PRESENT_RESIDENT2 -9.796e-01  3.756e-01  -2.608 0.009094 **
## PRESENT_RESIDENT3 -5.136e-01  4.128e-01  -1.244 0.213485
## PRESENT_RESIDENT4 -4.870e-01  3.730e-01  -1.305 0.191724
## REAL_ESTATE1        3.799e-01  2.644e-01   1.437 0.150800
## PROP_UNKN_NONE1    -2.967e-01  4.774e-01  -0.621 0.534367
## AGE                 1.350e-02  1.168e-02   1.156 0.247493
## OTHER_INSTALL1     -4.226e-01  2.863e-01  -1.476 0.139903
## RENT1              -9.326e-01  5.982e-01  -1.559 0.118985
## OWN_RES1           -3.446e-01  5.725e-01  -0.602 0.547238
## NUM_CREDITS        -1.399e-01  2.582e-01  -0.542 0.587942
## JOB1               -1.811e+00  9.475e-01  -1.911 0.055966 .
## JOB2               -1.607e+00  9.190e-01  -1.749 0.080366 .
## JOB3               -1.578e+00  9.407e-01  -1.677 0.093476 .
## NUM_DEPENDENTS      2.691e-02  3.194e-01   0.084 0.932853
## TELEPHONE1          4.314e-01  2.564e-01   1.682 0.092551 .
## FOREIGN1            1.777e+00  8.392e-01   2.118 0.034197 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 855.21  on 699  degrees of freedom
## Residual deviance: 593.69  on 654  degrees of freedom
## AIC: 685.69
##
## Number of Fisher Scoring iterations: 5
```

```r
#Remove statistically insignifucant variable(as employment,rent) one by one with high p value
logit<-glm(RESPONSE~.,family = binomial,data = trainData[,-c(12,3)])
summary(logit)#AIC: 695.84
```

```
##
## Call:
## glm(formula = RESPONSE ~ ., family = binomial, data = trainData[,
##     -c(12, 3)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9998  -0.6744   0.3635   0.6920   2.6861
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.9645879  1.3507879   1.454 0.145835
## CHK_ACCT1          0.5150873  0.2538013   2.029 0.042408 *
## CHK_ACCT2          1.6777753  0.5175800   3.242 0.001189 **
## CHK_ACCT3          2.0428326  0.2784901   7.335 2.21e-13 ***
## DURATION          -0.0306214  0.0115507  -2.651 0.008024 **
```

```
## NEW_CAR1           -0.2516264  0.4933038  -0.510 0.609993
## USED_CAR1           1.5813267  0.6262349   2.525 0.011565 *
## FURNITURE1          0.3518391  0.5094378   0.691 0.489791
## RADIO.TV1           0.6453908  0.4973225   1.298 0.194380
## EDUCATION1         -0.2245867  0.6461611  -0.348 0.728163
## RETRAINING1        -0.0676044  0.5568915  -0.121 0.903377
## AMOUNT             -0.0001200  0.0000555  -2.161 0.030683 *
## SAV_ACCT1           0.2389652  0.3351808   0.713 0.475880
## SAV_ACCT2           0.0750556  0.4360928   0.172 0.863352
## SAV_ACCT3           1.2943110  0.6447882   2.007 0.044713 *
## SAV_ACCT4           1.1853004  0.3324852   3.565 0.000364 ***
## INSTALL_RATE       -0.3236033  0.1119087  -2.892 0.003832 **
## MALE_DIV1          -0.3517464  0.4735299  -0.743 0.457592
## MALE_SINGLE1        0.6887105  0.2598355   2.651 0.008036 **
## MALE_MAR_or_WID1    0.0747508  0.3688392   0.203 0.839397
## CO.APPLICANT1      -0.8118167  0.4642773  -1.749 0.080367 .
## GUARANTOR1          0.9400409  0.4869474   1.930 0.053548 .
## PRESENT_RESIDENT2 -0.8932574  0.3448036  -2.591 0.009580 **
## PRESENT_RESIDENT3 -0.3108481  0.3941144  -0.789 0.430272
## PRESENT_RESIDENT4 -0.4129890  0.3461145  -1.193 0.232785
## REAL_ESTATE1        0.4311199  0.2580128   1.671 0.094737 .
## PROP_UNKN_NONE1    -0.2894497  0.4705631  -0.615 0.538480
## AGE                 0.0123128  0.0108579   1.134 0.256798
## OTHER_INSTALL1     -0.6162762  0.2629045  -2.344 0.019073 *
## RENT1              -0.8082864  0.5827034  -1.387 0.165402
## OWN_RES1           -0.1658196  0.5594267  -0.296 0.766917
## NUM_CREDITS         0.2738873  0.2012964   1.361 0.173635
## JOB1               -1.0781354  0.8414902  -1.281 0.200116
## JOB2               -0.9214666  0.8170069  -1.128 0.259381
## JOB3               -0.9598913  0.8662485  -1.108 0.267818
## NUM_DEPENDENTS     -0.0961821  0.3119599  -0.308 0.757842
## TELEPHONE1          0.5154357  0.2466064   2.090 0.036608 *
## FOREIGN1            2.0049754  0.8429083   2.379 0.017377 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 855.21  on 699  degrees of freedom
## Residual deviance: 616.80  on 662  degrees of freedom
## AIC: 692.8
##
## Number of Fisher Scoring iterations: 5
```

```r
logit<-glm(RESPONSE~.,family = binomial,data = trainData[,-c(12,11,28,24,6,14,9,20,19,16,27,17,21,4,22,3
summary(logit)#AIC: 671.24
```

```
##
## Call:
## glm(formula = RESPONSE ~ ., family = binomial, data = trainData[,
##     -c(12, 11, 28, 24, 6, 14, 9, 20, 19, 16, 27, 17, 21, 4, 22,
##         30, 26, 18, 2)])
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -2.3488  -0.7890   0.4313   0.7381   2.3984
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.243e-01  6.122e-01  -0.203 0.839060
## CHK_ACCT1       5.120e-01  2.335e-01   2.193 0.028310 *
## CHK_ACCT2       1.525e+00  4.940e-01   3.087 0.002025 **
## CHK_ACCT3       1.857e+00  2.542e-01   7.305 2.78e-13 ***
## HISTORY1        1.863e-01  6.290e-01   0.296 0.767054
## HISTORY2        8.239e-01  4.736e-01   1.740 0.081877 .
## HISTORY3        8.998e-01  5.606e-01   1.605 0.108435
## HISTORY4        1.638e+00  5.022e-01   3.262 0.001105 **
## USED_CAR1       1.491e+00  4.300e-01   3.467 0.000526 ***
## RADIO.TV1       5.412e-01  2.327e-01   2.326 0.020040 *
## EDUCATION1     -2.194e-01  4.592e-01  -0.478 0.632816
## AMOUNT         -2.012e-04  4.135e-05  -4.866 1.14e-06 ***
## INSTALL_RATE   -3.585e-01  9.763e-02  -3.672 0.000241 ***
## MALE_SINGLE1    6.511e-01  2.080e-01   3.130 0.001748 **
## OTHER_INSTALL1 -3.546e-01  2.634e-01  -1.346 0.178283
## OWN_RES1        3.897e-01  2.131e-01   1.829 0.067366 .
## TELEPHONE1      3.951e-01  2.126e-01   1.858 0.063117 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 855.21  on 699  degrees of freedom
## Residual deviance: 659.68  on 683  degrees of freedom
## AIC: 693.68
##
## Number of Fisher Scoring iterations: 5
```

```
#credit history with -('1: all credits at this bank paid back duly )have no significance
#and with critical account have significance
logit<-glm(RESPONSE~.,family = binomial,data = trainData[,-c(12,11,28,24,6,14,9,20,19,16,27,17,21,4,22,3
summary(logit)
```

```
##
## Call:
## glm(formula = RESPONSE ~ ., family = binomial, data = trainData[,
##     -c(12, 11, 28, 24, 6, 14, 9, 20, 19, 16, 27, 17, 21, 4, 22,
##         30, 26, 18, 2, 3)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4459  -0.8373   0.4428   0.7613   2.4248
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     8.268e-01  4.007e-01   2.063 0.039094 *
## CHK_ACCT1       4.752e-01  2.261e-01   2.101 0.035606 *
## CHK_ACCT2       1.443e+00  4.844e-01   2.978 0.002900 **
## CHK_ACCT3       1.959e+00  2.505e-01   7.821 5.26e-15 ***
```

```
## USED_CAR1       1.579e+00  4.272e-01   3.697 0.000218 ***
## RADIO.TV1       5.081e-01  2.271e-01   2.237 0.025286 *
## EDUCATION1     -1.144e-01  4.439e-01  -0.258 0.796666
## AMOUNT         -2.165e-04  3.984e-05  -5.435 5.48e-08 ***
## INSTALL_RATE   -3.749e-01  9.600e-02  -3.905 9.43e-05 ***
## MALE_SINGLE1    7.087e-01  2.030e-01   3.492 0.000479 ***
## OTHER_INSTALL1 -5.229e-01  2.437e-01  -2.146 0.031894 *
## OWN_RES1        4.710e-01  2.080e-01   2.265 0.023523 *
## TELEPHONE1      4.871e-01  2.070e-01   2.353 0.018621 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 855.21  on 699  degrees of freedom
## Residual deviance: 679.86  on 687  degrees of freedom
## AIC: 705.86
##
## Number of Fisher Scoring iterations: 5
```

```
#AIC value should decrese after elimination of variable in this way we select our statistically signifi
```

```
# Odds Ratio
exp(coef(logit))#OR>1 positively correlated,OR<1 -ive correlation,lowest p value suggest highest associ
```

```
##    (Intercept)      CHK_ACCT1      CHK_ACCT2      CHK_ACCT3      USED_CAR1
##      2.2859553      1.6082713      4.2315403      7.0916743      4.8504948
##      RADIO.TV1     EDUCATION1         AMOUNT   INSTALL_RATE   MALE_SINGLE1
##      1.6620685      0.8919193      0.9997835      0.6873924      2.0313773
## OTHER_INSTALL1       OWN_RES1     TELEPHONE1
##      0.5928150      1.6016482      1.6275647
```

```
#chk_acct,history () +ive correlated)
#Amount,instalment rate,education1 (-ive correlated) with the response variable
```

```
# Confusion matrix table
prob <- predict(logit,type=c("response"),testData)
head(prob)
```

```
##         1         4         5         7         9        10
## 0.7771511 0.2847563 0.3444301 0.9026485 0.9131485 0.2973799
```

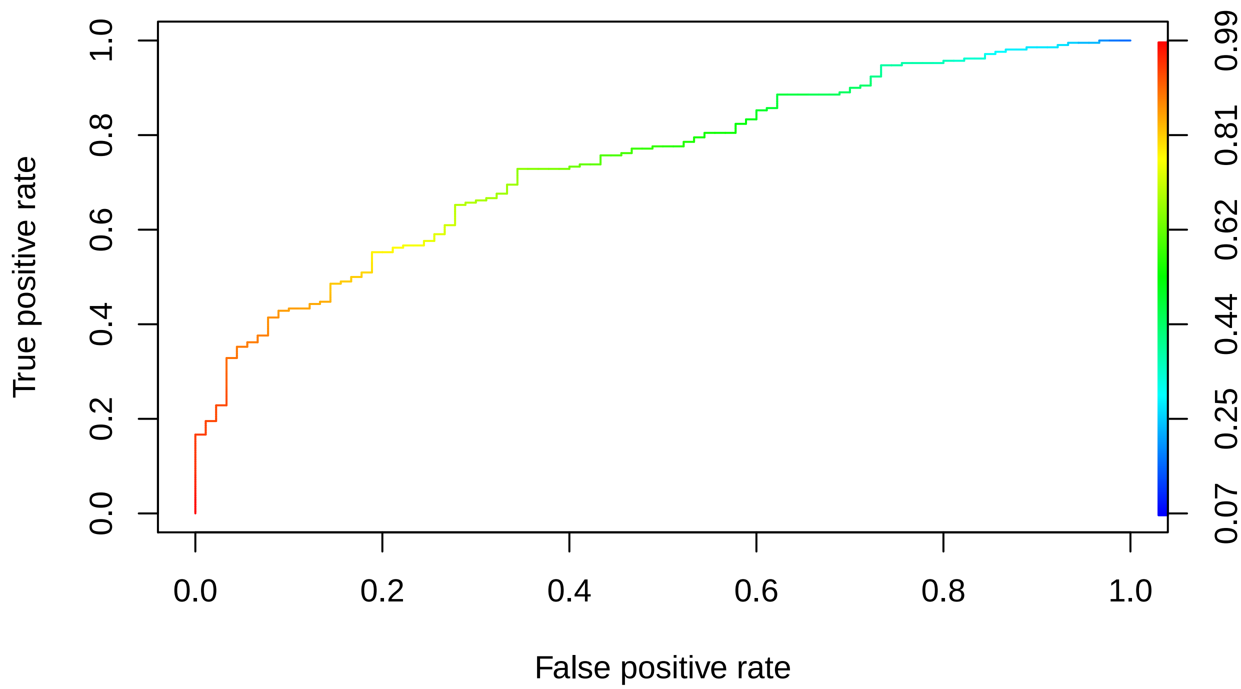
```
confusion<-table(prob>0.5,testData$RESPONSE)
confusion#,person with p>0.5 have good rating
```

```
##
##           0   1
##   FALSE  36  35
##   TRUE   54 175
```

```
# Model Accuracy
Accuracy<-sum(diag(confusion)/sum(confusion))
Accuracy# 0.6766667
```

```
## [1] 0.7033333
```

```
# ROC Curve
library(ROCR)
rocrpred<-prediction(prob,testData$RESPONSE)
rocrperf<-performance(rocrpred,'tpr','fpr')
plot(rocrperf,colorize=T,text.adj=c(-0.2,1.7))
```




```
# More area under the ROC Curve better is the logistic regression model obtained
#Area under TP(senstivity) should be more here TP means probability of correct prediction
#FP(type 1 error)
```

KNN model with cross validation and parameter tuning

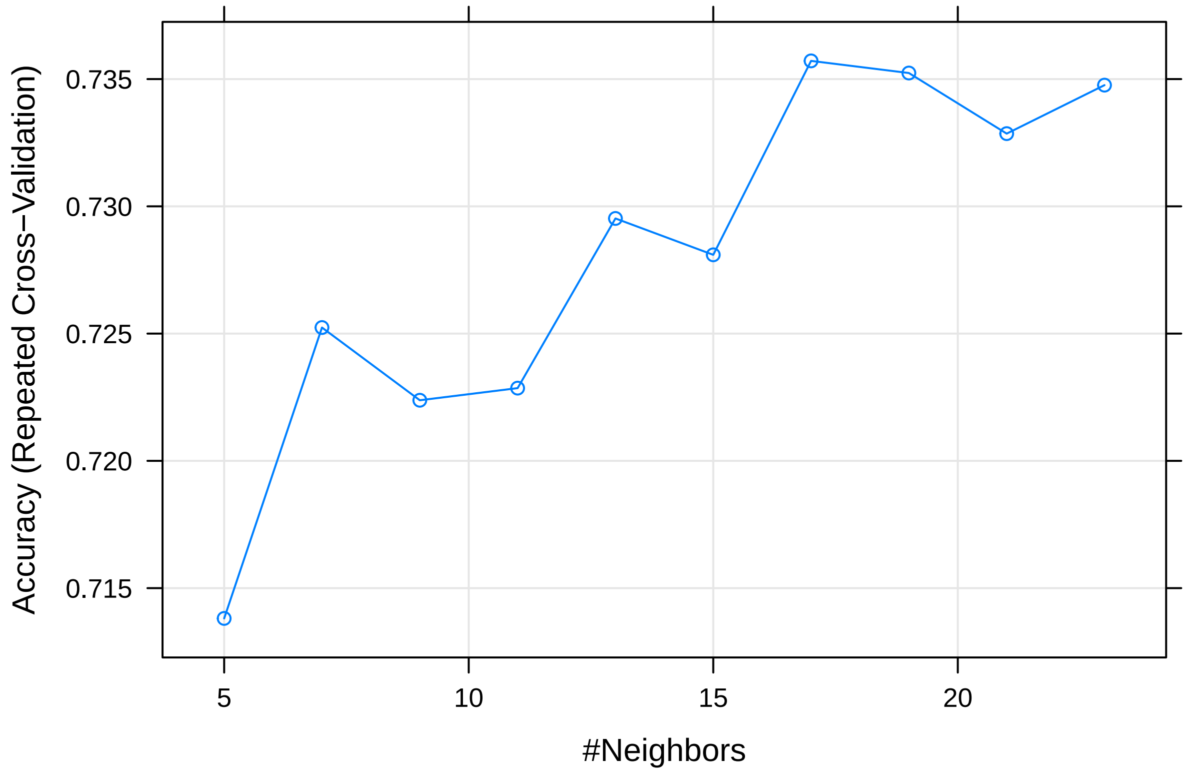
```
library(e1071)
#training and train control
set.seed(400)
```

```r
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
knn_fit <- train(RESPONSE ~., data = trainData, method = "knn", trControl=trctrl,preProcess = c("center

knn_fit #knn classifier
```

```
## k-Nearest Neighbors
##
## 700 samples
##  30 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (45), scaled (45)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 630, 630, 630, 630, 630, 630, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.7138095  0.2569957
##    7  0.7252381  0.2673210
##    9  0.7223810  0.2478685
##   11  0.7228571  0.2386742
##   13  0.7295238  0.2437923
##   15  0.7280952  0.2322293
##   17  0.7357143  0.2529915
##   19  0.7352381  0.2453702
##   21  0.7328571  0.2333846
##   23  0.7347619  0.2328125
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 17.
```

```r
#plot accuracy vs K Value graph
plot(knn_fit)
```




```r
#predict classes for test set using knn classifier
test_pred <- predict(knn_fit, newdata = testData[,-31])
test_pred
```

```
##   [1] 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0
##  [38] 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1
##  [75] 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 0 1 1 1 1 1 0 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0
## [186] 1 1 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [223] 0 1 0 1 0 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
## [260] 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1
## [297] 1 0 0 1
## Levels: 0 1
```

```r
#Test set Statistics
confusionMatrix(test_pred, testData$RESPONSE )  #Accuracy : 0.6733
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  23  20
##          1  67 190
##
```

```
##                Accuracy : 0.71
##                  95% CI : (0.6551, 0.7607)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.3793
##
##                   Kappa : 0.1884
##
##  Mcnemar's Test P-Value : 8.151e-07
##
##             Sensitivity : 0.25556
##             Specificity : 0.90476
##          Pos Pred Value : 0.53488
##          Neg Pred Value : 0.73930
##              Prevalence : 0.30000
##          Detection Rate : 0.07667
##    Detection Prevalence : 0.14333
##       Balanced Accuracy : 0.58016
##
##        'Positive' Class : 0
##
```

SVM model with cross validation and parameter tuning

```r
library(kernlab)
```

```
##
## Attaching package: 'kernlab'

## The following object is masked from 'package:ggplot2':
##
##     alpha

## The following object is masked from 'package:psych':
##
##     alpha
```

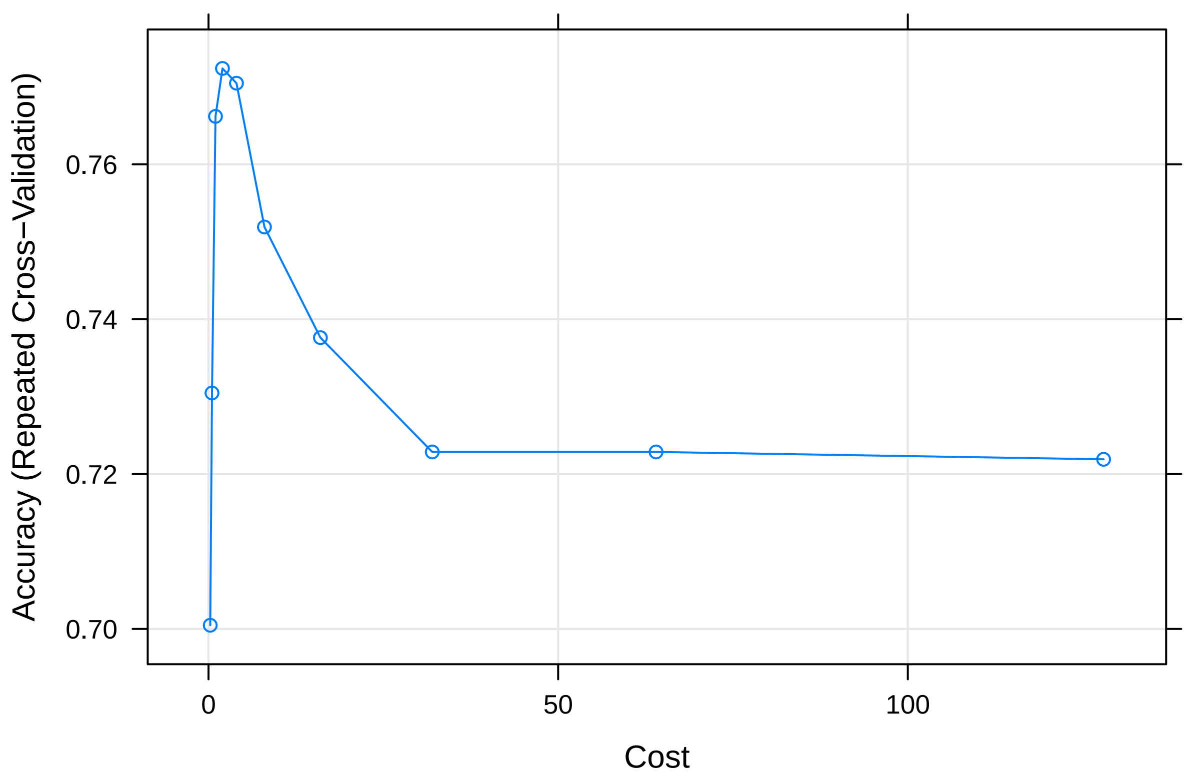
```r
set.seed(400)
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
SVM_fit <- train(RESPONSE ~., data = trainData, method = "svmRadial", trControl=trctrl,preProcess = c("

SVM_fit #SVM classifier
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 700 samples
##  30 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (45), scaled (45)
```

```
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 630, 630, 630, 630, 630, 630, ...
## Resampling results across tuning parameters:
##
##   C        Accuracy   Kappa
##     0.25  0.7004762  0.002180685
##     0.50  0.7304762  0.182102541
##     1.00  0.7661905  0.370841452
##     2.00  0.7723810  0.417288427
##     4.00  0.7704762  0.425658817
##     8.00  0.7519048  0.381004277
##    16.00  0.7376190  0.350843057
##    32.00  0.7228571  0.326083159
##    64.00  0.7228571  0.330620102
##   128.00  0.7219048  0.329533024
##
## Tuning parameter 'sigma' was held constant at a value of 0.01268028
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.01268028 and C = 2.
```

```r
#The final values used for the model were sigma = 0.01270901 and C = 2.
#plot accuracy vs K Value graph
plot(SVM_fit)
```



```r
#predict classes for test set using knn classifier
test_pred <- predict(SVM_fit, newdata = testData[,-31])
#test_pred
#Test set Statistics
confusionMatrix(test_pred, testData$RESPONSE )  #Accuracy : 0.7333
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  40  31
##          1  50 179
##
##                Accuracy : 0.73
##                  95% CI : (0.676, 0.7794)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.1418
##
##                   Kappa : 0.3159
##
##  Mcnemar's Test P-Value : 0.0455
##
##             Sensitivity : 0.4444
##             Specificity : 0.8524
##          Pos Pred Value : 0.5634
##          Neg Pred Value : 0.7817
##              Prevalence : 0.3000
##          Detection Rate : 0.1333
##    Detection Prevalence : 0.2367
##       Balanced Accuracy : 0.6484
##
##        'Positive' Class : 0
##
```

Random forest classifier

```r
library(randomForest)
attach(trainData)
```

```
## The following objects are masked from Data:
##
##     AGE, AMOUNT, CHK_ACCT, CO.APPLICANT, DURATION, EDUCATION,
##     EMPLOYMENT, FOREIGN, FURNITURE, GUARANTOR, HISTORY, INSTALL_RATE,
##     JOB, MALE_DIV, MALE_MAR_or_WID, MALE_SINGLE, NEW_CAR, NUM_CREDITS,
##     NUM_DEPENDENTS, OTHER_INSTALL, OWN_RES, PRESENT_RESIDENT,
##     PROP_UNKN_NONE, RADIO.TV, REAL_ESTATE, RENT, RESPONSE, RETRAINING,
##     SAV_ACCT, TELEPHONE, USED_CAR
```

```r
fit <- randomForest(RESPONSE~.,data=trainData,ntree=500)
print(fit) # view results
```

```
##
## Call:
##  randomForest(formula = RESPONSE ~ ., data = trainData, ntree = 500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 22.71%
## Confusion matrix:
##    0   1 class.error
## 0 92 118  0.56190476
## 1 41 449  0.08367347
```

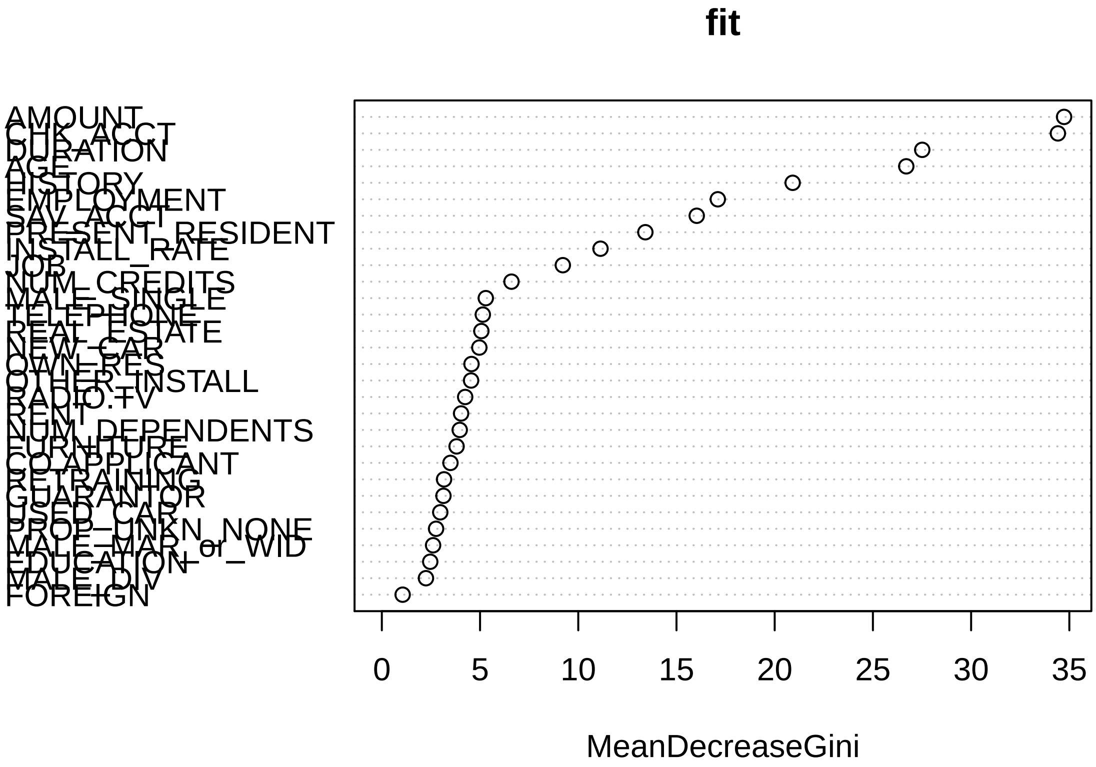
```r
fit$importance#gives gini index(priority of variables)
```

```
##                  MeanDecreaseGini
## CHK_ACCT                34.417796
## DURATION                27.511913
## HISTORY                 20.915794
## NEW_CAR                  4.963154
## USED_CAR                 2.974716
## FURNITURE                3.804749
## RADIO.TV                 4.241623
## EDUCATION                2.463373
## RETRAINING               3.166230
## AMOUNT                  34.732713
## SAV_ACCT                16.031486
## EMPLOYMENT              17.103832
## INSTALL_RATE            11.131559
## MALE_DIV                 2.245439
## MALE_SINGLE              5.291719
## MALE_MAR_or_WID          2.608791
## CO.APPLICANT             3.492115
## GUARANTOR                3.135902
## PRESENT_RESIDENT        13.415792
## REAL_ESTATE              5.066087
## PROP_UNKN_NONE           2.761273
## AGE                     26.699310
## OTHER_INSTALL            4.543314
## RENT                     4.035001
## OWN_RES                  4.563840
## NUM_CREDITS              6.599713
## JOB                      9.213446
## NUM_DEPENDENTS           3.964779
## TELEPHONE                5.141495
## FOREIGN                  1.058544
```

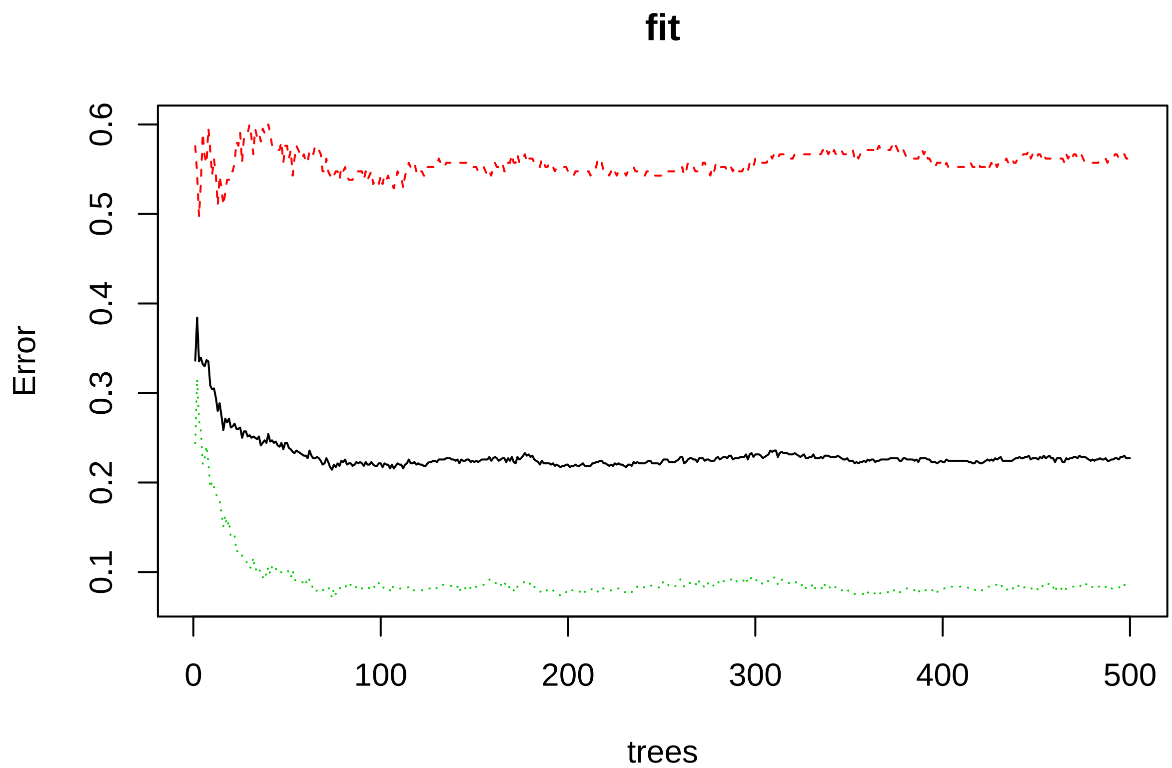
```r
importance(fit) # importance of each predictor max value more imp variables
```

```
##                    MeanDecreaseGini
## CHK_ACCT                 34.417796
## DURATION                 27.511913
## HISTORY                  20.915794
## NEW_CAR                   4.963154
## USED_CAR                  2.974716
## FURNITURE                 3.804749
## RADIO.TV                  4.241623
## EDUCATION                 2.463373
## RETRAINING                3.166230
## AMOUNT                   34.732713
## SAV_ACCT                 16.031486
## EMPLOYMENT               17.103832
## INSTALL_RATE             11.131559
## MALE_DIV                  2.245439
## MALE_SINGLE               5.291719
## MALE_MAR_or_WID           2.608791
## CO.APPLICANT              3.492115
## GUARANTOR                 3.135902
## PRESENT_RESIDENT         13.415792
## REAL_ESTATE               5.066087
## PROP_UNKN_NONE            2.761273
## AGE                      26.699310
## OTHER_INSTALL             4.543314
## RENT                      4.035001
## OWN_RES                   4.563840
## NUM_CREDITS               6.599713
## JOB                       9.213446
## NUM_DEPENDENTS            3.964779
## TELEPHONE                 5.141495
## FOREIGN                   1.058544
```

```
varImpPlot(fit)
```


fit
AMOUNT
CHK_ACCT
DURATION
AGE
HISTORY
EMPLOYMENT
SAV_ACCT
PRESENT_RESIDENT
INSTALL_RATE
JOB
NUM_CREDITS
MALE_SINGLE
TELEPHONE
REAL_ESTATE
NEW_CAR
OWN_RES
OTHER_INSTALL
RADIO.TV
RENT
NUM_DEPENDENTS
FURNITURE
CO.APPLICANT
RETRAINING
GUARANTOR
USED_CAR
PROP_UNKN_NONE
MALE_MAR_or_WID
EDUCATION
MALE_DIV
FOREIGN
0
5
10
15
20
25
30
35
MeanDecreaseGini

```
plot(fit)
```




```r
votes<-as.data.frame(fit$votes)

# Predicting test data
pred_test <-predict(fit,testData)
confusionMatrix(table(pred_test,testData$RESPONSE))#Accuracy : 0.76
```

```
## Confusion Matrix and Statistics
##
##
## pred_test   0   1
##         0  34  18
##         1  56 192
##
##                Accuracy : 0.7533
##                  95% CI : (0.7005, 0.8011)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.02388
##
##                   Kappa : 0.3321
##
##  Mcnemar's Test P-Value : 1.699e-05
##
##             Sensitivity : 0.3778
##             Specificity : 0.9143
##          Pos Pred Value : 0.6538
```

```
##          Neg Pred Value : 0.7742
##              Prevalence : 0.3000
##          Detection Rate : 0.1133
##    Detection Prevalence : 0.1733
##       Balanced Accuracy : 0.6460
##
##        'Positive' Class : 0
##
```

```
pred_train <-as.data.frame( predict(fit,trainData))
confusionMatrix(table(pred_train$`predict(fit, trainData)`,trainData$RESPONSE))
```

```
## Confusion Matrix and Statistics
##
##
##       0   1
##   0 210   0
##   1   0 490
##
##                Accuracy : 1
##                  95% CI : (0.9947, 1)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.3
##          Detection Rate : 0.3
##    Detection Prevalence : 0.3
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : 0
##
```

Model Selection by F1 score of SVM and Random Forest model

f1 is defined as 2 * precision * recall / (precision + recall). precision is the proportion of retrieved documents that are relevant to a query and recall is the proportion of relevant documents that are successfully retrieved by a query. If there are zero relevant documents that are retrieved, zero relevant documents, or zero predicted documents, f1 is defined as 0.

```
#for SVM
library(Metrics)
```

```
##
## Attaching package: 'Metrics'
```

```
## The following objects are masked from 'package:caret':
##
##     precision, recall
```

```r
f1(testData$RESPONSE,test_pred)#1
```

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

```r
#for random Forest
f1(testData$RESPONSE,pred_test)#1
```

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

```r
#
```

both model is good model because F1 score is 1 (perfect precision and recall) but on the basis of accuracy random forest model is good. svm model is good on the basis of senstivity(TP)value