Exploratory Data Analysis and Missing value Imputation - BINGO BONUS ASSIGNMENT

Load the data set

```
INFILE <- "C:/Users/aroon/Desktop/MSDS/422/MSDS422/Week 1/HMEQ_Loss.csv"
df <- read.csv(INFILE)
copy.df <- df
```

Preliminary exploration of dataset

```
# Exploring Raw data
head(df) # first 6
```

```
##   TARGET_BAD_FLAG TARGET_LOSS_AMT LOAN MORTDUE  VALUE  REASON    JOB  YOJ DEROG
## 1               1             641 1100   25860  39025 HomeImp  Other 10.5     0
## 2               1            1109 1300   70053  68400 HomeImp  Other  7.0     0
## 3               1             767 1500   13500  16700 HomeImp  Other  4.0     0
## 4               1            1425 1500      NA     NA                 NA    NA
## 5               0              NA 1700   97800 112000 HomeImp Office  3.0     0
## 6               1             335 1700   30548  40320 HomeImp  Other  9.0     0
##   DELINQ     CLAGE NINQ CLNO  DEBTINC
## 1      0  94.36667    1    9       NA
## 2      2 121.83333    0   14       NA
## 3      0 149.46667    1   10       NA
## 4     NA        NA   NA   NA       NA
## 5      0  93.33333    0   14       NA
## 6      0 101.46600    1    8 37.11361
```

```
tail(df) # last 6
```

```
##      TARGET_BAD_FLAG TARGET_LOSS_AMT  LOAN MORTDUE VALUE  REASON   JOB YOJ
## 5955               0              NA 88900   48919 93371 DebtCon Other  15
## 5956               0              NA 88900   57264 90185 DebtCon Other  16
## 5957               0              NA 89000   54576 92937 DebtCon Other  16
## 5958               0              NA 89200   54045 92924 DebtCon Other  15
## 5959               0              NA 89800   50370 91861 DebtCon Other  14
## 5960               0              NA 89900   48811 88934 DebtCon Other  15
##      DEROG DELINQ    CLAGE NINQ CLNO  DEBTINC
## 5955     0      1 205.6502    0   15 34.81826
## 5956     0      0 221.8087    0   16 36.11235
## 5957     0      0 208.6921    0   15 35.85997
## 5958     0      0 212.2797    0   15 35.55659
## 5959     0      0 213.8927    0   16 34.34088
## 5960     0      0 219.6010    0   16 34.57152
```

```
# view the class of the data input
class(df)
```

```
## [1] "data.frame"
```

```
print(str(df)) # structure of data frame
```

```
## 'data.frame':    5960 obs. of  14 variables:
##  $ TARGET_BAD_FLAG: int  1 1 1 1 0 1 1 1 1 1 ...
##  $ TARGET_LOSS_AMT: int  641 1109 767 1425 NA 335 1841 373 1217 1523 ...
##  $ LOAN           : int  1100 1300 1500 1500 1700 1700 1800 1800 2000 2000 ...
##  $ MORTDUE        : num  25860 70053 13500 NA 97800 ...
##  $ VALUE          : num  39025 68400 16700 NA 112000 ...
##  $ REASON         : Factor w/ 3 levels "","DebtCon","HomeImp": 3 3 3 1 3 3 3 3 3 3 ...
##  $ JOB            : Factor w/ 7 levels "","Mgr","Office",..: 4 4 4 1 3 4 4 4 4 6 ...
##  $ YOJ            : num  10.5 7 4 NA 3 9 5 11 3 16 ...
##  $ DEROG          : int  0 0 0 NA 0 0 3 0 0 0 ...
##  $ DELINQ         : int  0 2 0 NA 0 0 2 0 2 0 ...
##  $ CLAGE          : num  94.4 121.8 149.5 NA 93.3 ...
##  $ NINQ           : int  1 0 1 NA 0 1 1 0 1 0 ...
##  $ CLNO           : int  9 14 10 NA 14 8 17 8 12 13 ...
##  $ DEBTINC        : num  NA NA NA NA NA ...
## NULL
```

```
# view diamension #rows and columns
dim(df)
```

```
## [1] 5960   14
```

```
nrow(df)
```

```
## [1] 5960
```

```
names(df) # column names
```

```
##  [1] "TARGET_BAD_FLAG" "TARGET_LOSS_AMT" "LOAN"            "MORTDUE"
##  [5] "VALUE"           "REASON"          "JOB"             "YOJ"
##  [9] "DEROG"           "DELINQ"          "CLAGE"           "NINQ"
## [13] "CLNO"            "DEBTINC"
```

```
summary(df) # summary of Data Frame
```

```
##  TARGET_BAD_FLAG  TARGET_LOSS_AMT      LOAN          MORTDUE
##  Min.   :0.0000   Min.   :  224   Min.   : 1100   Min.   :  2063
##  1st Qu.:0.0000   1st Qu.: 5639   1st Qu.:11100   1st Qu.: 46276
##  Median :0.0000   Median :11003   Median :16300   Median : 65019
##  Mean   :0.1995   Mean   :13415   Mean   :18608   Mean   : 73761
##  3rd Qu.:0.0000   3rd Qu.:17634   3rd Qu.:23300   3rd Qu.: 91488
##  Max.   :1.0000   Max.   :78987   Max.   :89900   Max.   :399550
```

```
##                   NA's   :4771                     NA's   :518
##      VALUE            REASON          JOB            YOJ
##  Min.   :  8000          : 252          : 279   Min.   : 0.000
##  1st Qu.: 66076   DebtCon:3928   Mgr    : 767   1st Qu.: 3.000
##  Median : 89236   HomeImp:1780   Office : 948   Median : 7.000
##  Mean   :101776                  Other  :2388   Mean   : 8.922
##  3rd Qu.:119824                  ProfExe:1276   3rd Qu.:13.000
##  Max.   :855909                  Sales  : 109   Max.   :41.000
##  NA's   :112                     Self   : 193   NA's   :515
##      DEROG             DELINQ            CLAGE             NINQ
##  Min.   : 0.0000   Min.   : 0.0000   Min.   :   0.0   Min.   : 0.000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 115.1   1st Qu.: 0.000
##  Median : 0.0000   Median : 0.0000   Median : 173.5   Median : 1.000
##  Mean   : 0.2546   Mean   : 0.4494   Mean   : 179.8   Mean   : 1.186
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 231.6   3rd Qu.: 2.000
##  Max.   :10.0000   Max.   :15.0000   Max.   :1168.2   Max.   :17.000
##  NA's   :708       NA's   :580       NA's   :308      NA's   :510
##       CLNO          DEBTINC
##  Min.   : 0.0   Min.   :  0.5245
##  1st Qu.:15.0   1st Qu.: 29.1400
##  Median :20.0   Median : 34.8183
##  Mean   :21.3   Mean   : 33.7799
##  3rd Qu.:26.0   3rd Qu.: 39.0031
##  Max.   :71.0   Max.   :203.3121
##  NA's   :222    NA's   :1267
```

```
sum(complete.cases(df)) # Check for number of complete rows
```

```
## [1] 309
```

```
summary(df$LOAN)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1100   11100   16300   18608   23300   89900
```

```
summary(df$MORTDUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    2063   46276   65019   73761   91488  399550     518
```

```
# load dplyr
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
glimpse(df)
```

```
## Rows: 5,960
## Columns: 14
## $ TARGET_BAD_FLAG <int> 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, ...
## $ TARGET_LOSS_AMT <int> 641, 1109, 767, 1425, NA, 335, 1841, 373, 1217, 152...
## $ LOAN            <int> 1100, 1300, 1500, 1500, 1700, 1700, 1800, 1800, 200...
## $ MORTDUE         <dbl> 25860, 70053, 13500, NA, 97800, 30548, 48649, 28502...
## $ VALUE           <dbl> 39025, 68400, 16700, NA, 112000, 40320, 57037, 4303...
## $ REASON          <fct> HomeImp, HomeImp, HomeImp, , HomeImp, HomeImp, Home...
## $ JOB             <fct> Other, Other, Other, , Office, Other, Other, Other,...
## $ YOJ             <dbl> 10.5, 7.0, 4.0, NA, 3.0, 9.0, 5.0, 11.0, 3.0, 16.0,...
## $ DEROG           <int> 0, 0, 0, NA, 0, 0, 3, 0, 0, 0, NA, 0, 0, 0, 0, 0, 2...
## $ DELINQ          <int> 0, 2, 0, NA, 0, 0, 2, 0, 2, 0, NA, 1, 0, 0, 1, 1, 6...
## $ CLAGE           <dbl> 94.36667, 121.83333, 149.46667, NA, 93.33333, 101.4...
## $ NINQ            <int> 1, 0, 1, NA, 0, 1, 1, 0, 1, 0, NA, 1, 2, 0, 0, 0, 1...
## $ CLNO            <int> 9, 14, 10, NA, 14, 8, 17, 8, 12, 13, NA, 9, 25, 24,...
## $ DEBTINC         <dbl> NA, NA, NA, NA, NA, 37.113614, NA, 36.884894, NA, N...
```

```
# find the mean of debt to income ratio by variable BAD
print( with( df, tapply( DEBTINC, TARGET_BAD_FLAG, mean, na.rm=TRUE ) ) )
```

```
##        0        1
## 33.25313 39.38764
```

```
# find the mean of mortgage due by variable BAD
print( with( df, tapply( MORTDUE, TARGET_BAD_FLAG, mean, na.rm=TRUE ) ) )
```

```
##        0        1
## 74829.25 69460.45
```

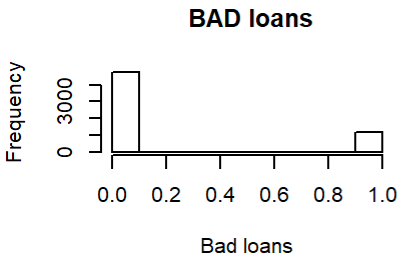
Graphs

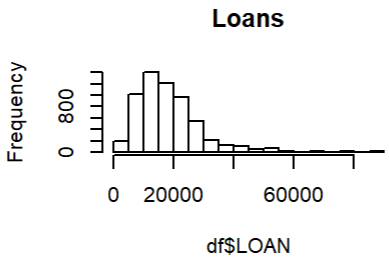
```
par(mfrow = c(3,3))

hist(df$TARGET_BAD_FLAG, xlab="Bad loans", main=" BAD loans") # histogram of bad loans
#Amount of the loan request
hist(df$LOAN,main="Loans")
plot(df$MORTDUE,df$LOAN)
hist(df$VALUE/df$MORTDUE ,main= "Value to Mortgage Ratio")
```

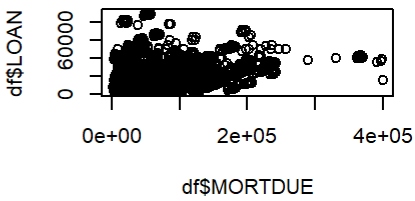
```
# find the mean of debt to income ratio by variable BAD
barplot( with( df, tapply( DEBTINC, TARGET_BAD_FLAG, mean, na.rm=TRUE ,main="Debt to income cl
assified by Bad debt") ) )
```

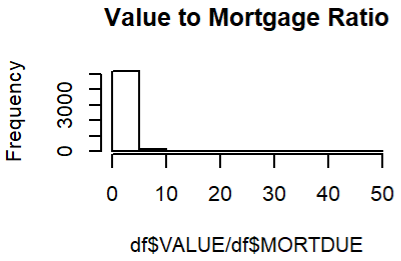
```
df$M_DEBTINC = is.na( df$DEBTINC ) + 0
df$M_VALUE = is.na( df$VALUE ) + 0
df$M_MORTDUE = is.na( df$MORTDUE ) + 0
df$M_YOJ = is.na( df$YOJ ) + 0
df$M_DEROG = is.na( df$DEROG ) + 0
df$M_DELINQ = is.na( df$DELINQ ) + 0
df$M_CLAGE = is.na( df$CLAGE ) + 0
df$M_NINQ = is.na( df$NINQ ) + 0
df$M_CLNO = is.na( df$CLNO ) + 0

head(df)
```

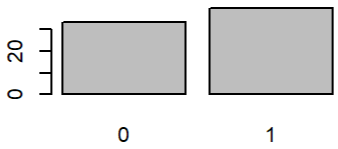
```
##   TARGET_BAD_FLAG TARGET_LOSS_AMT LOAN MORTDUE  VALUE  REASON    JOB  YOJ DEROG
## 1               1             641 1100   25860  39025 HomeImp  Other 10.5     0
## 2               1            1109 1300   70053  68400 HomeImp  Other  7.0     0
## 3               1             767 1500   13500  16700 HomeImp  Other  4.0     0
## 4               1            1425 1500      NA     NA                  NA    NA
## 5               0              NA 1700   97800 112000 HomeImp Office  3.0     0
## 6               1             335 1700   30548  40320 HomeImp  Other  9.0     0
##   DELINQ     CLAGE NINQ CLNO  DEBTINC M_DEBTINC M_VALUE M_MORTDUE M_YOJ M_DEROG
## 1      0  94.36667    1    9       NA         1       0         0     0       0
## 2      2 121.83333    0   14       NA         1       0         0     0       0
## 3      0 149.46667    1   10       NA         1       0         0     0       0
## 4     NA        NA   NA   NA       NA         1       1         1     1       1
## 5      0  93.33333    0   14       NA         1       0         0     0       0
## 6      0 101.46600    1    8 37.11361         0       0         0     0       0
##   M_DELINQ M_CLAGE M_NINQ M_CLNO
## 1        0       0      0      0
## 2        0       0      0      0
## 3        0       0      0      0
## 4        1       1      1      1
## 5        0       0      0      0
## 6        0       0      0      0
```











Find

the mean of the columns. Copy the column value into new column and call it IMP that will be corrected values.

```
#mean( df$DEBTINC, na.rm=TRUE ) # used to find mean.
df$IMP_DEBTINC = df$DEBTINC
df$IMP_VALUE = df$VALUE
df$IMP_MORTDUE = df$MORTDUE
df$IMP_YOJ = df$YOJ
df$IMP_DEROG = df$DEROG
df$IMP_DELINQ = df$DELINQ
df$IMP_CLAGE =  df$CLAGE
df$IMP_NINQ =  df$NINQ
df$IMP_CLNO = df$CLNO
```

Create new column for every column that will have imputed values and have column name with IMP prefixed. Fill the missing values with the average value.

```
####impute using mean
df$IMP_DEBTINC = ifelse(is.na( df$IMP_DEBTINC ), mean( df$DEBTINC, na.rm=TRUE ), df$IMP_DEBTIN
C )
df$IMP_CLAGE = ifelse(is.na(df$CLAGE), mean( df$CLAGE, na.rm= TRUE ), df$IMP_CLAGE )
df$IMP_YOJ = ifelse(is.na(df$YOJ), mean( df$YOJ, na.rm= TRUE ), df$IMP_YOJ )
# though there is only 1 maximum value of 10 and mean = 0.2546 median is 0. It's safe to assum
e the mean value for the missing value.
df$IMP_DEROG = ifelse(is.na(df$DEROG), mean( df$DEROG, na.rm= TRUE ), df$IMP_DEROG )
# safer to assume mean value when the data is missing because median, 1st,3rd quartile value a
re 0
df$IMP_DELINQ = ifelse(is.na(df$DELINQ), mean( df$DELINQ, na.rm= TRUE ), df$IMP_DELINQ )
```

```
####impute using median

df$IMP_VALUE = ifelse(is.na( df$IMP_VALUE ), median( df$VALUE, na.rm=TRUE ), df$IMP_VALUE )
df$IMP_MORTDUE = ifelse(is.na( df$MORTDUE), median( df$MORTDUE, na.rm=TRUE),df$IMP_MORTDUE )
# Number of recent credit inquiries are usually whole numbers. mean is only.1 more than median
.
df$IMP_NINQ = ifelse(is.na(df$NINQ), median( df$NINQ, na.rm= TRUE ), df$IMP_NINQ )
# number of credit line is a whole number and since mean is only .1 greater than median.I used
 median value
df$IMP_CLNO = ifelse(is.na(df$CLNO), mean( df$CLNO, na.rm= TRUE ), df$IMP_CLNO )
```

Getting rid of the columns with missing values

```
print( head(df) )
```

```
##   TARGET_BAD_FLAG TARGET_LOSS_AMT LOAN MORTDUE  VALUE  REASON    JOB  YOJ DEROG
## 1               1             641 1100   25860  39025 HomeImp  Other 10.5     0
## 2               1            1109 1300   70053  68400 HomeImp  Other  7.0     0
## 3               1             767 1500   13500  16700 HomeImp  Other  4.0     0
## 4               1            1425 1500      NA     NA                 NA    NA
## 5               0              NA 1700   97800 112000 HomeImp Office  3.0     0
## 6               1             335 1700   30548  40320 HomeImp  Other  9.0     0
##   DELINQ     CLAGE NINQ CLNO  DEBTINC M_DEBTINC M_VALUE M_MORTDUE M_YOJ M_DEROG
## 1      0  94.36667    1    9       NA         1       0         0     0       0
## 2      2 121.83333    0   14       NA         1       0         0     0       0
## 3      0 149.46667    1   10       NA         1       0         0     0       0
## 4     NA        NA   NA   NA       NA         1       1         1     1       1
## 5      0  93.33333    0   14       NA         1       0         0     0       0
## 6      0 101.46600    1    8 37.11361         0       0         0     0       0
##   M_DELINQ M_CLAGE M_NINQ M_CLNO IMP_DEBTINC IMP_VALUE IMP_MORTDUE   IMP_YOJ
## 1        0       0      0      0    33.77992   39025.0       25860 10.500000
## 2        0       0      0      0    33.77992   68400.0       70053  7.000000
## 3        0       0      0      0    33.77992   16700.0       13500  4.000000
## 4        1       1      1      1    33.77992   89235.5       65019  8.922268
## 5        0       0      0      0    33.77992  112000.0       97800  3.000000
## 6        0       0      0      0    37.11361   40320.0       30548  9.000000
##   IMP_DEROG IMP_DELINQ IMP_CLAGE IMP_NINQ IMP_CLNO
## 1 0.0000000  0.0000000  94.36667        1   9.0000
## 2 0.0000000  2.0000000 121.83333        0  14.0000
## 3 0.0000000  0.0000000 149.46667        1  10.0000
## 4 0.2545697  0.4494424 179.76628        1  21.2961
## 5 0.0000000  0.0000000  93.33333        0  14.0000
## 6 0.0000000  0.0000000 101.46600        1   8.0000
```

```
df = subset(df, select = -c( DEBTINC,VALUE,MORTDUE,YOJ,CLAGE,DEROG,DELINQ,NINQ,CLNO ) )
print(head(df))
```

```
##   TARGET_BAD_FLAG TARGET_LOSS_AMT LOAN  REASON    JOB M_DEBTINC M_VALUE
## 1               1             641 1100 HomeImp  Other         1       0
## 2               1            1109 1300 HomeImp  Other         1       0
## 3               1             767 1500 HomeImp  Other         1       0
```

```
## 4               1            1425 1500                          1       1
## 5               0              NA 1700 HomeImp Office           1       0
## 6               1             335 1700 HomeImp  Other           0       0
##   M_MORTDUE M_YOJ M_DEROG M_DELINQ M_CLAGE M_NINQ M_CLNO IMP_DEBTINC IMP_VALUE
## 1         0     0       0        0       0      0      0    33.77992   39025.0
## 2         0     0       0        0       0      0      0    33.77992   68400.0
## 3         0     0       0        0       0      0      0    33.77992   16700.0
## 4         1     1       1        1       1      1      1    33.77992   89235.5
## 5         0     0       0        0       0      0      0    33.77992  112000.0
## 6         0     0       0        0       0      0      0    37.11361   40320.0
##   IMP_MORTDUE   IMP_YOJ IMP_DEROG IMP_DELINQ IMP_CLAGE IMP_NINQ IMP_CLNO
## 1       25860 10.500000 0.0000000  0.0000000  94.36667        1   9.0000
## 2       70053  7.000000 0.0000000  2.0000000 121.83333        0  14.0000
## 3       13500  4.000000 0.0000000  0.0000000 149.46667        1  10.0000
## 4       65019  8.922268 0.2545697  0.4494424 179.76628        1  21.2961
## 5       97800  3.000000 0.0000000  0.0000000  93.33333        0  14.0000
## 6       30548  9.000000 0.0000000  0.0000000 101.46600        1   8.0000
```

Handle Categorical variable

```
print( with( df, tapply( TARGET_BAD_FLAG, JOB, mean, na.rm=TRUE ) ) ) # get the bad loan based
 on job title.
```

```
##                   Mgr     Office      Other    ProfExe      Sales       Self
## 0.08243728 0.23337679 0.13185654 0.23199330 0.16614420 0.34862385 0.30051813
```

```
df$IMP_JOB = df$JOB # copy JOB into IMP_JOB
df$IMP_REASON = df$REASON # copy REASON into IMP_REASON
# id the job is blank then mark it unknown

df$IMP_JOB = ifelse(df$JOB == "", "UNKNOWN", as.character(df$IMP_JOB) )
df$IMP_REASON = ifelse(df$REASON == "", "UNKNOWN", as.character(df$IMP_REASON) )

print( head(df) )
```

```
##   TARGET_BAD_FLAG TARGET_LOSS_AMT LOAN  REASON    JOB M_DEBTINC M_VALUE
## 1               1             641 1100 HomeImp  Other         1       0
## 2               1            1109 1300 HomeImp  Other         1       0
## 3               1             767 1500 HomeImp  Other         1       0
## 4               1            1425 1500                        1       1
## 5               0              NA 1700 HomeImp Office         1       0
## 6               1             335 1700 HomeImp  Other         0       0
##   M_MORTDUE M_YOJ M_DEROG M_DELINQ M_CLAGE M_NINQ M_CLNO IMP_DEBTINC IMP_VALUE
## 1         0     0       0        0       0      0      0    33.77992   39025.0
## 2         0     0       0        0       0      0      0    33.77992   68400.0
## 3         0     0       0        0       0      0      0    33.77992   16700.0
## 4         1     1       1        1       1      1      1    33.77992   89235.5
## 5         0     0       0        0       0      0      0    33.77992  112000.0
## 6         0     0       0        0       0      0      0    37.11361   40320.0
##   IMP_MORTDUE   IMP_YOJ IMP_DEROG IMP_DELINQ IMP_CLAGE IMP_NINQ IMP_CLNO
## 1       25860 10.500000 0.0000000  0.0000000  94.36667        1   9.0000
```

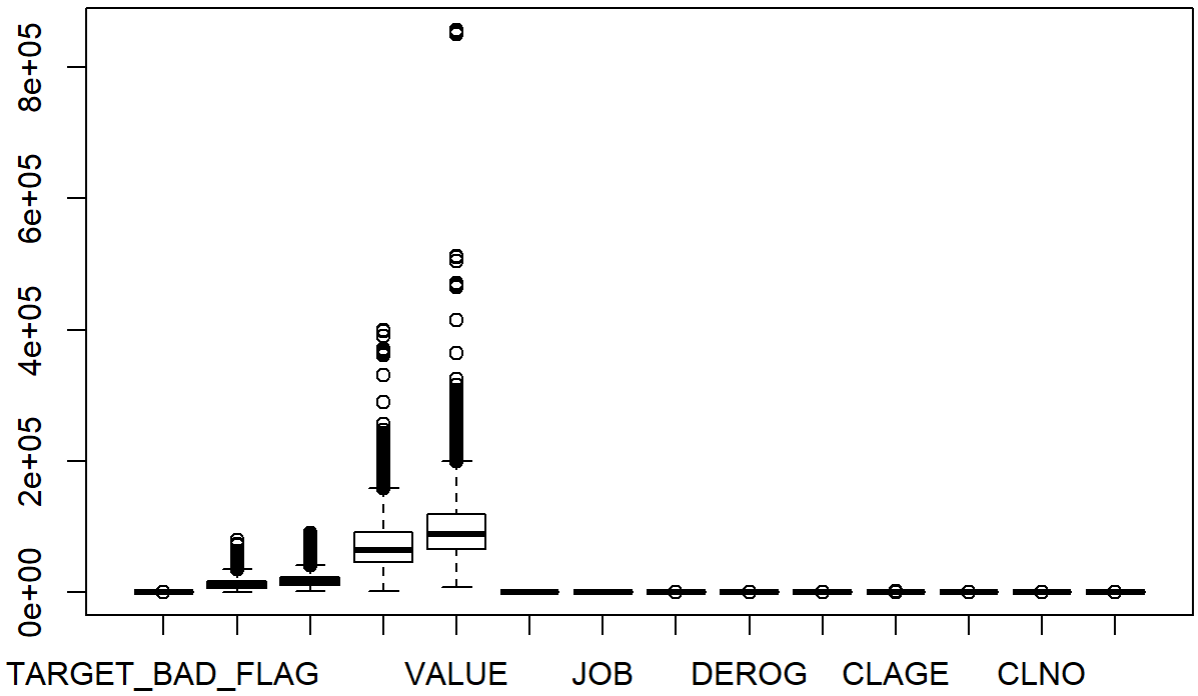
```
## 2       70053  7.000000 0.0000000  2.0000000 121.83333        0  14.0000
## 3       13500  4.000000 0.0000000  0.0000000 149.46667        1  10.0000
## 4       65019  8.922268 0.2545697  0.4494424 179.76628        1  21.2961
## 5       97800  3.000000 0.0000000  0.0000000  93.33333        0  14.0000
## 6       30548  9.000000 0.0000000  0.0000000 101.46600        1   8.0000
##   IMP_JOB IMP_REASON
## 1   Other    HomeImp
## 2   Other    HomeImp
## 3   Other    HomeImp
## 4 UNKNOWN    UNKNOWN
## 5  Office    HomeImp
## 6   Other    HomeImp
```

```
df = subset(df, select = -c( JOB, REASON ) )
print( head(df) )
```

```
##   TARGET_BAD_FLAG TARGET_LOSS_AMT LOAN M_DEBTINC M_VALUE M_MORTDUE M_YOJ
## 1               1              641 1100         1       0         0     0
## 2               1             1109 1300         1       0         0     0
## 3               1              767 1500         1       0         0     0
## 4               1             1425 1500         1       1         1     1
## 5               0               NA 1700         1       0         0     0
## 6               1              335 1700         0       0         0     0
##   M_DEROG M_DELINQ M_CLAGE M_NINQ M_CLNO IMP_DEBTINC IMP_VALUE IMP_MORTDUE
## 1       0        0       0      0      0    33.77992   39025.0       25860
## 2       0        0       0      0      0    33.77992   68400.0       70053
## 3       0        0       0      0      0    33.77992   16700.0       13500
## 4       1        1       1      1      1    33.77992   89235.5       65019
## 5       0        0       0      0      0    33.77992  112000.0       97800
## 6       0        0       0      0      0    37.11361   40320.0       30548
##     IMP_YOJ IMP_DEROG IMP_DELINQ IMP_CLAGE IMP_NINQ IMP_CLNO IMP_JOB IMP_REASON
## 1 10.500000 0.0000000  0.0000000  94.36667        1   9.0000   Other    HomeImp
## 2  7.000000 0.0000000  2.0000000 121.83333        0  14.0000   Other    HomeImp
## 3  4.000000 0.0000000  0.0000000 149.46667        1  10.0000   Other    HomeImp
## 4  8.922268 0.2545697  0.4494424 179.76628        1  21.2961 UNKNOWN    UNKNOWN
## 5  3.000000 0.0000000  0.0000000  93.33333        0  14.0000  Office    HomeImp
## 6  9.000000 0.0000000  0.0000000 101.46600        1   8.0000   Other    HomeImp
```

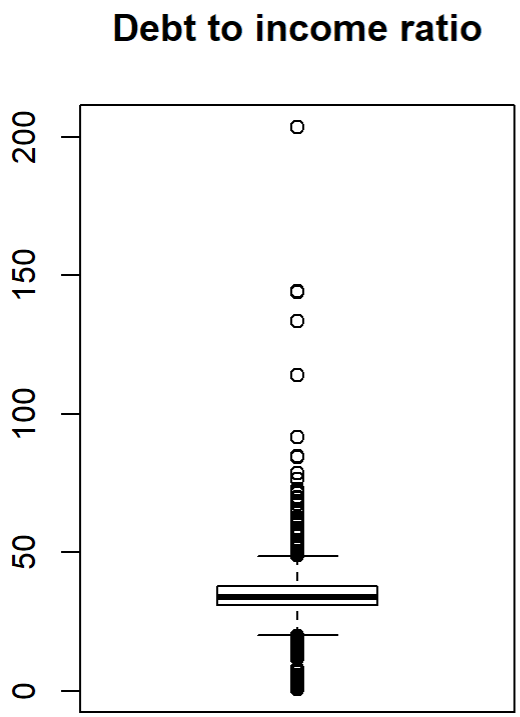
* Outlier Stats

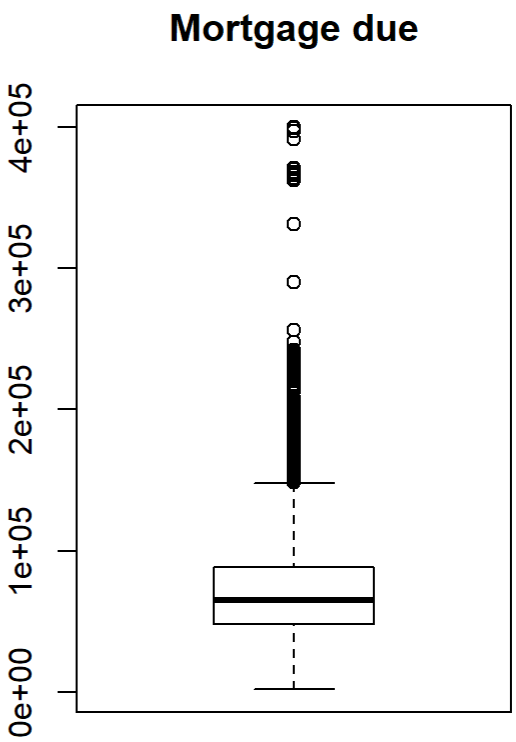
```
boxplot(copy.df)
```




```
par(mfrow = c(1,2))
# the 2 variables with outliers that needs to be fixed
boxplot(df$IMP_DEBTINC,main="Debt to income ratio")
boxplot(df$IMP_MORTDUE, main="Mortgage due")
```

Get





the MAX, MIN, MEAN, SD for 2 variables - DEBTINC and MARTDUE

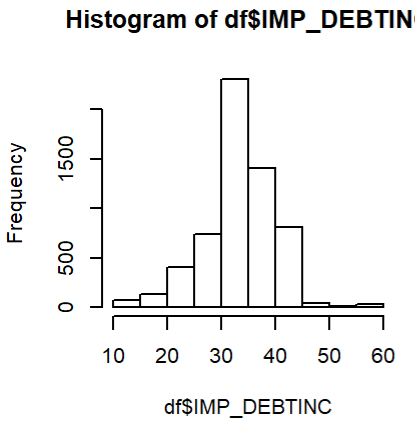
```
a1 = max( df$IMP_DEBTINC, na.rm=TRUE )
z1 = min( df$IMP_DEBTINC, na.rm=TRUE )
m1 = mean( df$IMP_DEBTINC, na.rm=TRUE )
s1 = sd( df$IMP_DEBTINC, na.rm=TRUE )
a2 = max( df$IMP_MORTDUE, na.rm=TRUE )
z2 = min( df$IMP_MORTDUE, na.rm=TRUE )
m2 = mean( df$IMP_MORTDUE, na.rm=TRUE )
s2 = sd( df$IMP_MORTDUE, na.rm=TRUE )
```

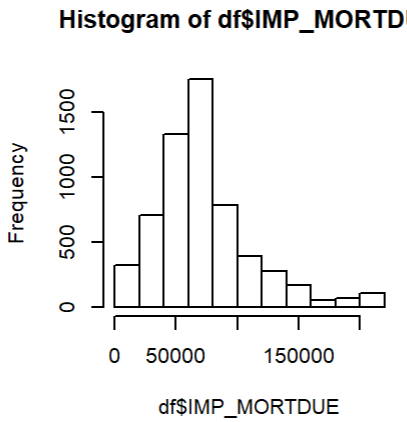
If the value beyond mean +3 standard deviation then set it to mean+3*SD and if value is lesser than mean-3SD ,then replace it with mean-3SD.

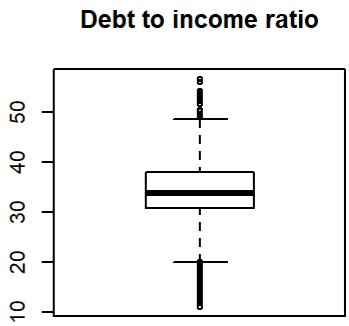
```
df$IMP_DEBTINC = ifelse( df$IMP_DEBTINC > m1+3*s1, m1+3*s1, df$IMP_DEBTINC )
df$IMP_DEBTINC = ifelse( df$IMP_DEBTINC < m1-3*s1, m1-3*s1, df$IMP_DEBTINC )
df$IMP_MORTDUE = ifelse( df$IMP_MORTDUE > m2+3*s2, m2+3*s2, df$IMP_MORTDUE )
df$IMP_MORTDUE = ifelse( df$IMP_MORTDUE < m2-3*s2, m2-3*s2, df$IMP_MORTDUE )
```

Histogram and box plot after fixing outliers

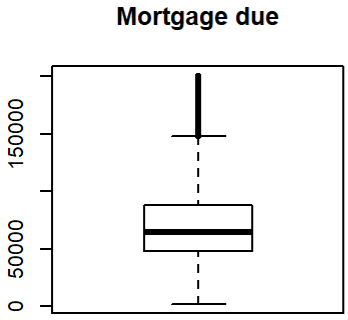
```
par(mfrow=c(2,3))
hist( df$IMP_DEBTINC  )
hist( df$IMP_MORTDUE  )
boxplot( df$IMP_DEBTINC ,main= "Debt to income ratio" )
boxplot( df$IMP_MORTDUE , main="Mortgage due"  )
```







#####

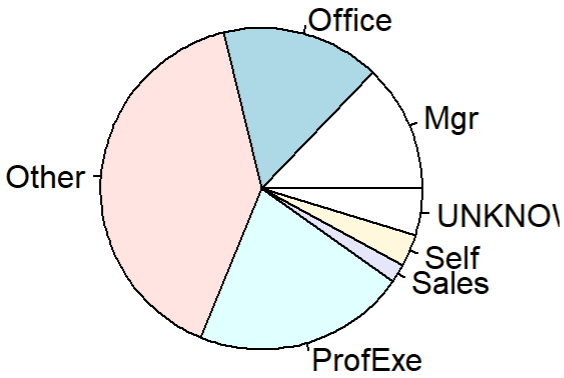


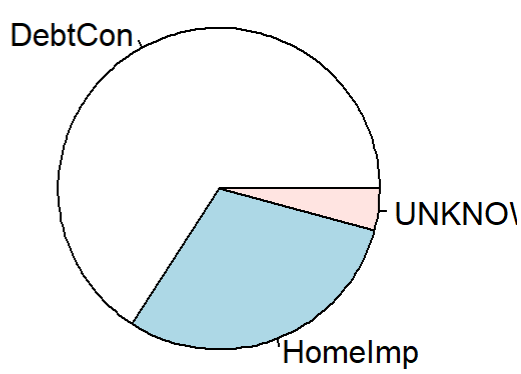
Recalculate the MAX, MIN, MEAN, SD for 2 variables - DEBTINC and MARTDUE

```
a1 = max( df$IMP_DEBTINC, na.rm=TRUE )
z1 = min( df$IMP_DEBTINC, na.rm=TRUE )
m1 = mean( df$IMP_DEBTINC, na.rm=TRUE )
s1 = sd( df$IMP_DEBTINC, na.rm=TRUE )
a2 = max( df$IMP_MORTDUE, na.rm=TRUE )
z2 = min( df$IMP_MORTDUE, na.rm=TRUE )
m2 = mean( df$IMP_MORTDUE, na.rm=TRUE )
s2 = sd( df$IMP_MORTDUE, na.rm=TRUE )
```

Create table and pie chart of the categorical table

```
tbl1 <- table( df$IMP_JOB )
tbl2 <- table( df$IMP_REASON )
par(mfrow=c(1,2))
pie( tbl1 , main= "JOB" )
pie( tbl2, main= "REASON" )
```





Create new variable to identift risky jobs

```
print( with( df, tapply( TARGET_BAD_FLAG, IMP_JOB, mean ) ) )
```

```
##        Mgr     Office      Other    ProfExe      Sales       Self    UNKNOWN
## 0.23337679 0.13185654 0.23199330 0.16614420 0.34862385 0.30051813 0.08243728
```

```
df$FLAG_JOB_RISKY = ifelse(df$IMP_JOB %in% c("Self","Sales"), 1, 0 )
head(df,n=25)
```

```
##    TARGET_BAD_FLAG TARGET_LOSS_AMT LOAN M_DEBTINC M_VALUE M_MORTDUE M_YOJ
## 1                1              641 1100         1       0         0     0
## 2                1             1109 1300         1       0         0     0
## 3                1              767 1500         1       0         0     0
## 4                1             1425 1500         1       1         1     1
## 5                0               NA 1700         1       0         0     0
## 6                1              335 1700         0       0         0     0
## 7                1             1841 1800         1       0         0     0
## 8                1              373 1800         0       0         0     0
## 9                1             1217 2000         1       0         0     0
## 10               1             1523 2000         1       0         1     0
## 11               1             1822 2000         1       1         0     0
## 12               1             1224 2000         1       0         0     0
## 13               1             1928 2000         1       0         0     0
```

```
## 14              0              NA 2000         1       0         0     0
## 15              1            1680 2100         1       0         0     0
## 16              1             407 2200         1       0         0     1
## 17              1            2375 2200         1       0         0     0
## 18              1             224 2200         0       1         0     0
## 19              1            2017 2300         1       0         0     0
## 20              0              NA 2300         0       0         0     0
## 21              1            1825 2300         1       0         0     0
## 22              1             589 2400         1       0         0     0
## 23              1            2192 2400         1       0         0     0
## 24              1            1694 2400         1       1         0     0
## 25              1            1638 2400         1       0         1     1
##    M_DEROG M_DELINQ M_CLAGE M_NINQ M_CLNO IMP_DEBTINC IMP_VALUE IMP_MORTDUE
## 1        0        0       0      0      0    33.77992   39025.0       25860
## 2        0        0       0      0      0    33.77992   68400.0       70053
## 3        0        0       0      0      0    33.77992   16700.0       13500
## 4        1        1       1      1      1    33.77992   89235.5       65019
## 5        0        0       0      0      0    33.77992  112000.0       97800
## 6        0        0       0      0      0    37.11361   40320.0       30548
## 7        0        0       0      0      0    33.77992   57037.0       48649
## 8        0        0       0      0      0    36.88489   43034.0       28502
## 9        0        0       0      0      0    33.77992   46740.0       32700
## 10       0        0       0      0      0    33.77992   62250.0       65019
## 11       1        1       1      1      1    33.77992   89235.5       22608
## 12       0        0       0      0      0    33.77992   29800.0       20627
## 13       0        0       0      0      0    33.77992   55000.0       45000
## 14       0        0       0      0      0    33.77992   87400.0       64536
## 15       0        0       0      0      0    33.77992   83850.0       71000
## 16       0        0       0      0      0    33.77992   34687.0       24280
## 17       0        0       0      0      0    33.77992  102600.0       90957
## 18       1        1       1      1      1    10.88178   89235.5       23030
## 19       0        0       0      0      0    33.77992   40150.0       28192
## 20       0        0       0      0      0    31.58850  120953.0      102370
## 21       0        0       0      0      0    33.77992   46200.0       37626
## 22       0        0       1      0      0    33.77992   73395.0       50000
## 23       0        0       0      0      0    33.77992   40800.0       28000
## 24       1        0       0      0      0    33.77992   89235.5       18000
## 25       0        0       0      0      0    33.77992   17180.0       65019
##      IMP_YOJ IMP_DEROG IMP_DELINQ IMP_CLAGE IMP_NINQ IMP_CLNO IMP_JOB
## 1  10.500000 0.0000000  0.0000000  94.36667        1   9.0000   Other
## 2   7.000000 0.0000000  2.0000000 121.83333        0  14.0000   Other
## 3   4.000000 0.0000000  0.0000000 149.46667        1  10.0000   Other
## 4   8.922268 0.2545697  0.4494424 179.76628        1  21.2961 UNKNOWN
## 5   3.000000 0.0000000  0.0000000  93.33333        0  14.0000  Office
## 6   9.000000 0.0000000  0.0000000 101.46600        1   8.0000   Other
## 7   5.000000 3.0000000  2.0000000  77.10000        1  17.0000   Other
## 8  11.000000 0.0000000  0.0000000  88.76603        0   8.0000   Other
## 9   3.000000 0.0000000  2.0000000 216.93333        1  12.0000   Other
## 10 16.000000 0.0000000  0.0000000 115.80000        0  13.0000   Sales
## 11 18.000000 0.2545697  0.4494424 179.76628        1  21.2961 UNKNOWN
## 12 11.000000 0.0000000  1.0000000 122.53333        1   9.0000  Office
## 13  3.000000 0.0000000  0.0000000  86.06667        2  25.0000   Other
## 14  2.500000 0.0000000  0.0000000 147.13333        0  24.0000     Mgr
## 15  8.000000 0.0000000  1.0000000 123.00000        0  16.0000   Other
```

```
## 16  8.922268 0.0000000  1.0000000 300.86667        0   8.0000   Other
## 17  7.000000 2.0000000  6.0000000 122.90000        1  22.0000     Mgr
## 18 19.000000 0.2545697  0.4494424 179.76628        1  21.2961 UNKNOWN
## 19  4.500000 0.0000000  0.0000000  54.60000        1  16.0000   Other
## 20  2.000000 0.0000000  0.0000000  90.99253        0  13.0000  Office
## 21  3.000000 0.0000000  1.0000000 122.26667        1  14.0000   Other
## 22  5.000000 1.0000000  0.0000000 179.76628        1   0.0000 ProfExe
## 23 12.000000 0.0000000  0.0000000  67.20000        2  22.0000     Mgr
## 24 22.000000 0.2545697  2.0000000 121.73333        0  10.0000     Mgr
## 25  8.922268 0.0000000  0.0000000  14.56667        3   4.0000   Other
##    IMP_REASON FLAG_JOB_RISKY
## 1     HomeImp              0
## 2     HomeImp              0
## 3     HomeImp              0
## 4     UNKNOWN              0
## 5     HomeImp              0
## 6     HomeImp              0
## 7     HomeImp              0
## 8     HomeImp              0
## 9     HomeImp              0
## 10    HomeImp              1
## 11    UNKNOWN              0
## 12    HomeImp              0
## 13    HomeImp              0
## 14    UNKNOWN              0
## 15    HomeImp              0
## 16    HomeImp              0
## 17    HomeImp              0
## 18    UNKNOWN              0
## 19    HomeImp              0
## 20    HomeImp              0
## 21    HomeImp              0
## 22    HomeImp              0
## 23    HomeImp              0
## 24    HomeImp              0
## 25    HomeImp              0
```

```
summary(df)
```

```
##  TARGET_BAD_FLAG  TARGET_LOSS_AMT      LOAN         M_DEBTINC
##  Min.   :0.0000   Min.   :  224   Min.   : 1100   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.: 5639   1st Qu.:11100   1st Qu.:0.0000
##  Median :0.0000   Median :11003   Median :16300   Median :0.0000
##  Mean   :0.1995   Mean   :13415   Mean   :18608   Mean   :0.2126
##  3rd Qu.:0.0000   3rd Qu.:17634   3rd Qu.:23300   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :78987   Max.   :89900   Max.   :1.0000
##                   NA's   :4771
##     M_VALUE          M_MORTDUE           M_YOJ            M_DEROG
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.0000
##  Mean   :0.01879   Mean   :0.08691   Mean   :0.08641   Mean   :0.1188
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
```

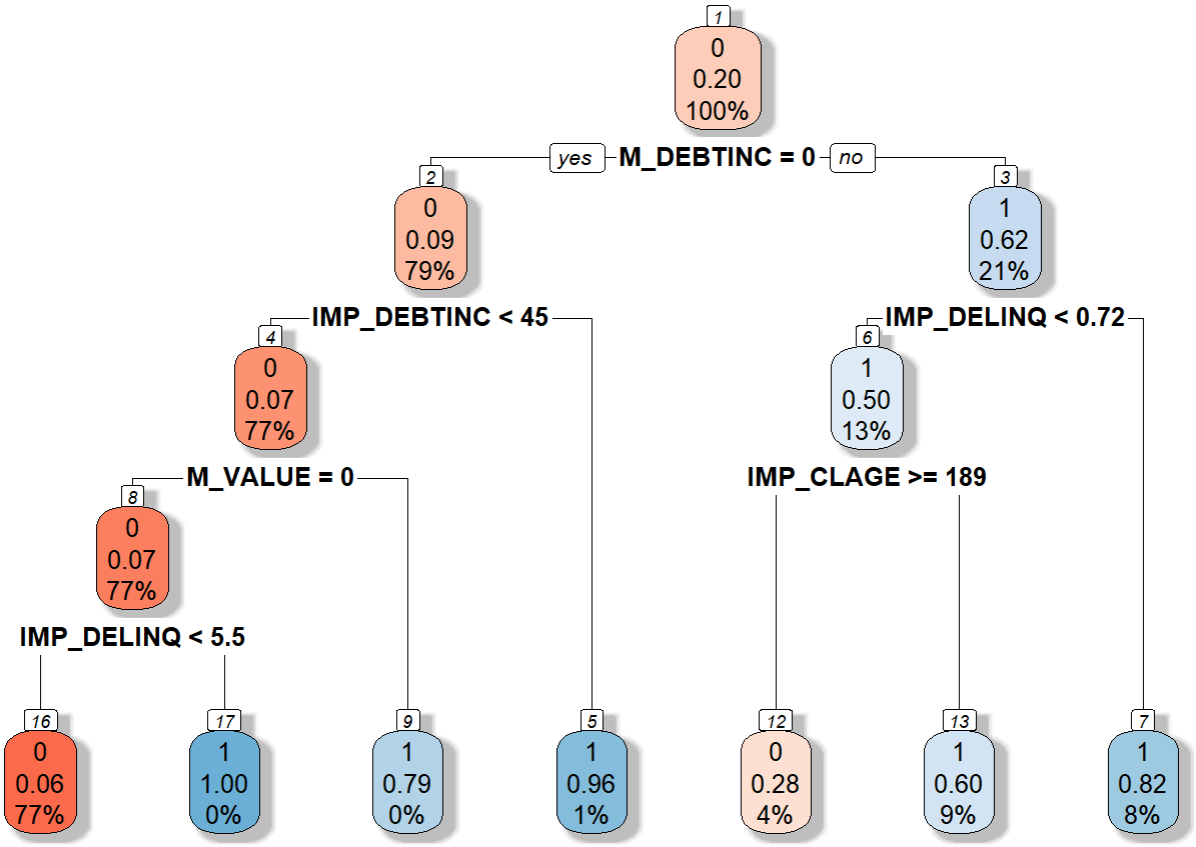
```
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
##
##     M_DELINQ          M_CLAGE            M_NINQ            M_CLNO
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.09732   Mean   :0.05168   Mean   :0.08557   Mean   :0.03725
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##   IMP_DEBTINC      IMP_VALUE       IMP_MORTDUE        IMP_YOJ
##  Min.   :10.88   Min.   :  8000   Min.   :  2063   Min.   : 0.000
##  1st Qu.:30.76   1st Qu.: 66490   1st Qu.: 48139   1st Qu.: 3.000
##  Median :33.78   Median : 89236   Median : 65019   Median : 8.000
##  Mean   :33.69   Mean   :101540   Mean   : 72202   Mean   : 8.922
##  3rd Qu.:37.95   3rd Qu.:119005   3rd Qu.: 88200   3rd Qu.:12.000
##  Max.   :56.68   Max.   :855909   Max.   :200659   Max.   :41.000
##
##    IMP_DEROG         IMP_DELINQ        IMP_CLAGE         IMP_NINQ
##  Min.   : 0.0000   Min.   : 0.0000   Min.   :   0.0   Min.   : 0.00
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 117.4   1st Qu.: 0.00
##  Median : 0.0000   Median : 0.0000   Median : 178.1   Median : 1.00
##  Mean   : 0.2546   Mean   : 0.4494   Mean   : 179.8   Mean   : 1.17
##  3rd Qu.: 0.0000   3rd Qu.: 0.4494   3rd Qu.: 227.1   3rd Qu.: 2.00
##  Max.   :10.0000   Max.   :15.0000   Max.   :1168.2   Max.   :17.00
##
##     IMP_CLNO       IMP_JOB           IMP_REASON        FLAG_JOB_RISKY
##  Min.   : 0.0   Length:5960        Length:5960        Min.   :0.00000
##  1st Qu.:15.0   Class :character   Class :character   1st Qu.:0.00000
##  Median :21.0   Mode  :character   Mode  :character   Median :0.00000
##  Mean   :21.3                                         Mean   :0.05067
##  3rd Qu.:26.0                                         3rd Qu.:0.00000
##  Max.   :71.0                                         Max.   :1.00000
##
```

Prediction using decision tree

```
library( rpart ) # model
library( rpart.plot )
```

```
## Warning: package 'rpart.plot' was built under R version 3.6.3
```

```
df2 <- df
t = rpart( TARGET_BAD_FLAG ~. , method="class", data=df)
rpart.plot( t, box.palette="RdBu", shadow.col="gray", nn=TRUE)
```




```
p_bad = predict( t, data=df2, type=c("class") )
p_Prob = predict( t, data=df2, type=c("prob") )
df2 = cbind( df2, p_bad )
df2 = cbind( df2, p_Prob )

accuracy = sum(df2$BAD == df2$p_bad)/length(df2$BAD)
cat( "The accuracy of prediction is",accuracy )
```

```
Comparing R to Python.
```

I find both R and Python are good for Exploratory Data Analysis. I personally felt R was more clear and straightforward. I could do everything with just one library Dplyr. I have worked with R in MSDS 401 course that gave me bit more familiarity with R than Python. Also R has GGPLOT library that presents the graphs in a more visually appealing manner.