Data Mining and Predictive Modelling Assignment 1

Name: Rushikesh Jyoti

Division: A

Roll no: 27

SRN: 201901139

Code

```
library(moments)

library(corrplot)

dataset <- read.csv('C:\\VS_Workshop\\Sem 6\\Data Mining and Predictive Modelling\\Assignments\\Ass1\\pva97nk.csv')


# 2. Identify the variables in the file "pva97nk.csv" and

# determine whether any variable has any missing values.

colnames(dataset)

sprintf("There are %d NA values in dataset", sum(is.na(dataset)))

# OR

# table(is.na(dataset))



# 3. Impute some of the variables that have missing values using their corresponding mean values.

# Verify whether your task has been correctly done.

for(i in 1:ncol(dataset)){

  if (is.numeric(dataset[,i])){
        dataset[is.na(dataset[,i]), i] <- mean(dataset[,i], na.rm = TRUE)
    }
}
# Verification
sprintf("There are %d NA values in dataset", sum(is.na(dataset)))

# 4. Compute Skewness and Kurtosis
skurtosis <- data.frame("Category", "Skewness", "Kurtosis")
for(i in 1:ncol(dataset)) {
    if(is.numeric(dataset[,i])){
         skurtosis[nrow(skurtosis) + 1,] = c(
            colnames(dataset)[i],
            round(skewness(dataset[,i]), 5),
            round(kurtosis(dataset[,i]), 5)
        )
    }
}
skurtosis
# Histogram of GiftCntAll
hist(dataset$GiftCntAll)

# 5. Determine the "summary" information for the numerical variables.
summary(dataset)

# 6. Identify the "distributions" of the numerical variables
# and plot the distributions.
for(i in 1:ncol(dataset)) {
    if (is.numeric(dataset[,i])) {
        hist(dataset[,i], main=colnames(dataset)[i])
    }
}

# 7. Transform the numeric variables into their natural log values
# and scale [0 - 1] values.
numericset = Filter(is.numeric, dataset)
for (i in 1:ncol(numericset)) {
    print(colnames(numericset)[i])
    print(head(log(numericset[,i])))

}

# 8. Check whether the numeric variables follow normality conditions.
qqnorm(numericset$GiftCntAll)
qqline(numericset$GiftCntAll)

qqnorm(numericset$PromCntAll)
qqline(numericset$PromCntAll)

qqnorm(numericset$DemAge)
qqline(numericset$DemAge)



# 9. Find the correlation matrix for all the variables in the dataset
# and plot the graph of the correlation matrix.
corrplot(cor(numericset, method = c("spearman")), diag=FALSE)

# 10. From the given dataset partition the data into 70-15-15 divisions
# so to construct the training, validation and test datasets.
```

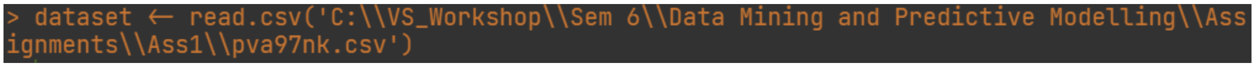
```
spec = c(train = .70, test = .15, validate = .15)


g = sample(cut(
   seq(nrow(numericset)),
   nrow(numericset) * cumsum(c(0, spec)),
   labels = names(spec)
))


result = split(numericset, g)
sapply(result, nrow) / nrow(numericset)
# To see the dataset
# head(result$train)
# head(result$test)
# head(result$validate)
```

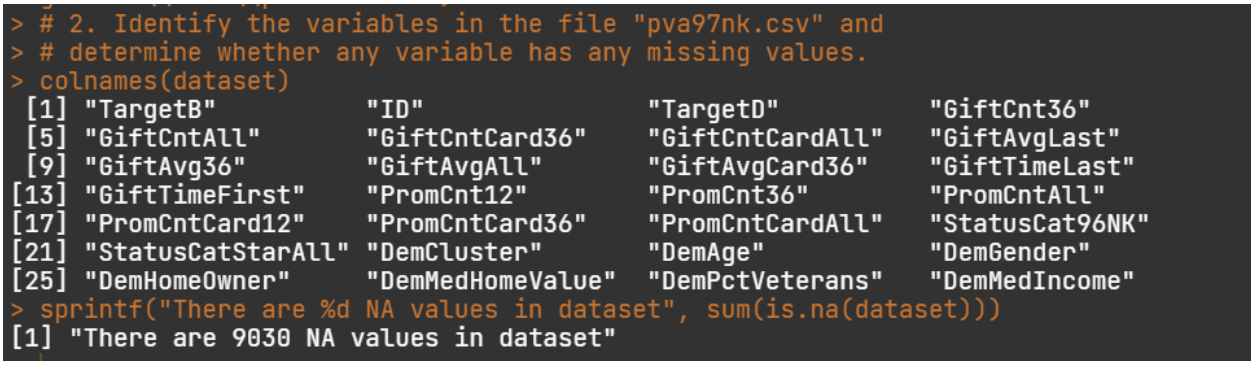
Output

1. Read the file



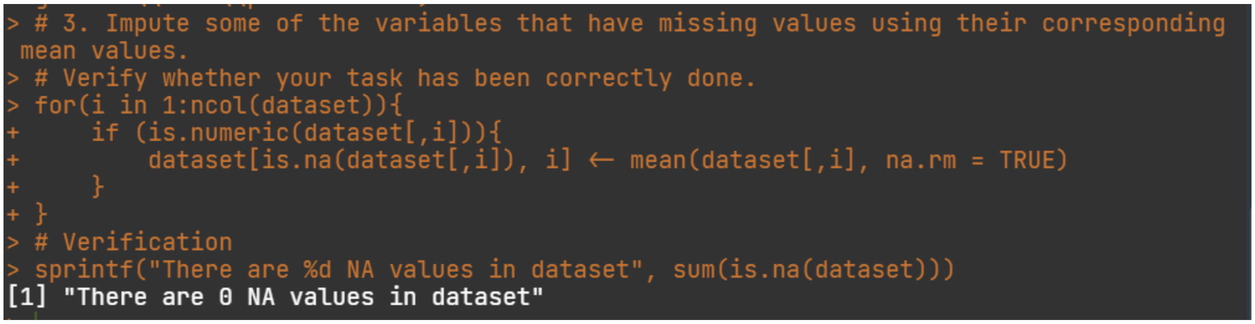
```
> dataset <- read.csv('C:\\VS_Workshop\\Sem 6\\Data Mining and Predictive Modelling\\Ass
ignments\\Ass1\\pva97nk.csv')
```

2. The variables and NA values



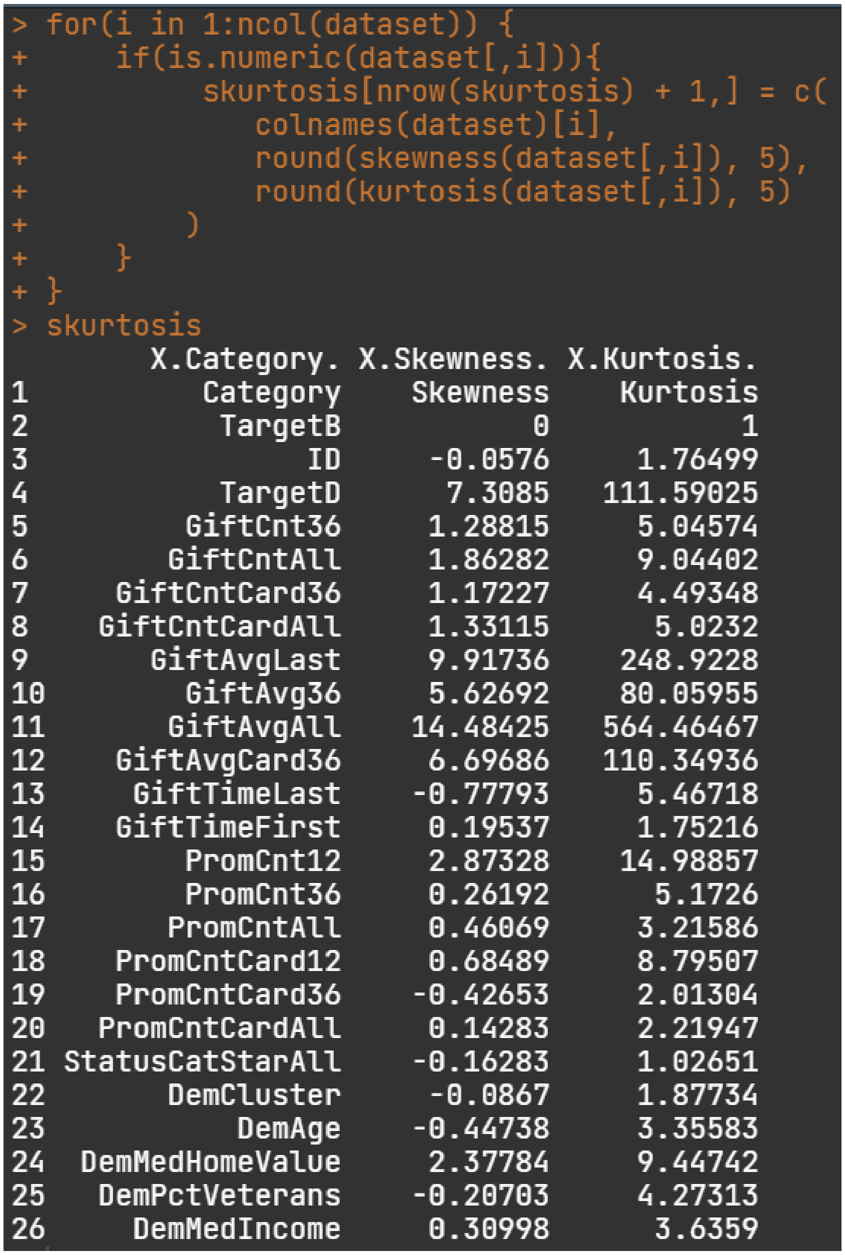
```
> # 2. Identify the variables in the file "pva97nk.csv" and
> # determine whether any variable has any missing values.
> colnames(dataset)
 [1] "TargetB"          "ID"               "TargetD"          "GiftCnt36"
 [5] "GiftCntAll"       "GiftCntCard36"    "GiftCntCardAll"   "GiftAvgLast"
 [9] "GiftAvg36"        "GiftAvgAll"       "GiftAvgCard36"    "GiftTimeLast"
[13] "GiftTimeFirst"    "PromCnt12"        "PromCnt36"        "PromCntAll"
[17] "PromCntCard12"    "PromCntCard36"    "PromCntCardAll"   "StatusCat96NK"
[21] "StatusCatStarAll" "DemCluster"       "DemAge"           "DemGender"
[25] "DemHomeOwner"     "DemMedHomeValue"  "DemPctVeterans"   "DemMedIncome"
> sprintf("There are %d NA values in dataset", sum(is.na(dataset)))
[1] "There are 9030 NA values in dataset"
```

3. Filling NA values with mean and verification of values



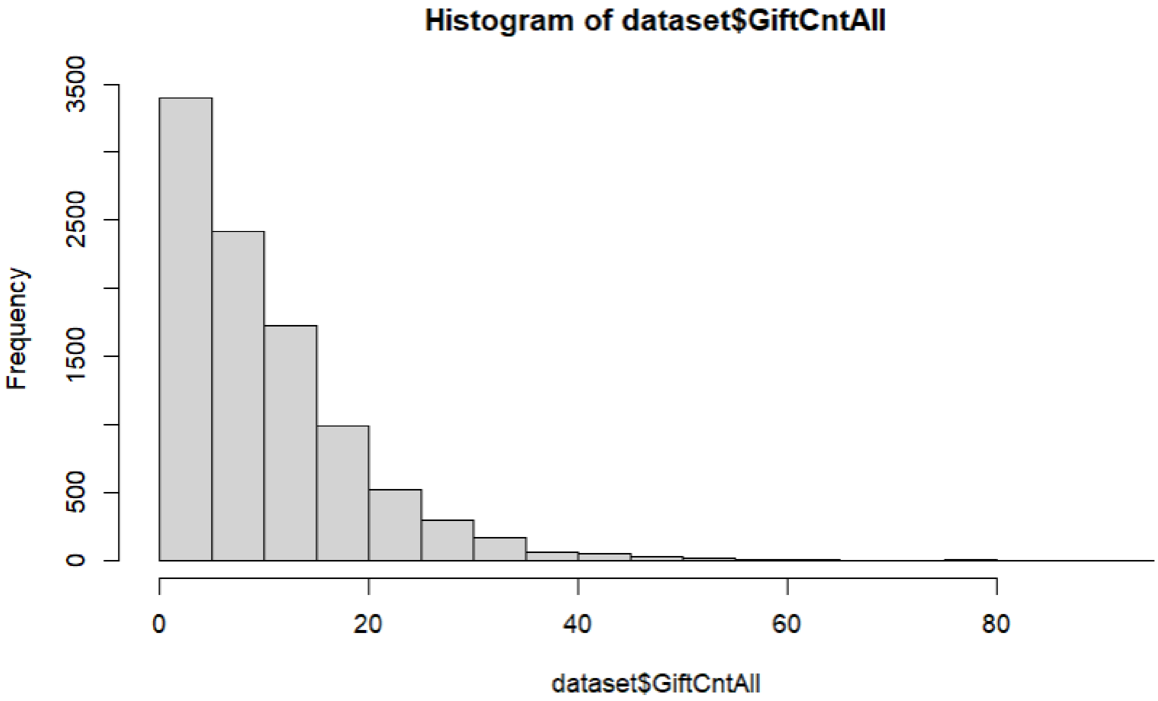
```
> # 3. Impute some of the variables that have missing values using their corresponding
 mean values.
> # Verify whether your task has been correctly done.
> for(i in 1:ncol(dataset)){
+     if (is.numeric(dataset[,i])){
+         dataset[is.na(dataset[,i]), i] ← mean(dataset[,i], na.rm = TRUE)
+     }
+ }
> # Verification
> sprintf("There are %d NA values in dataset", sum(is.na(dataset)))
[1] "There are 0 NA values in dataset"
```

4. Computing the skewness and kurtosis

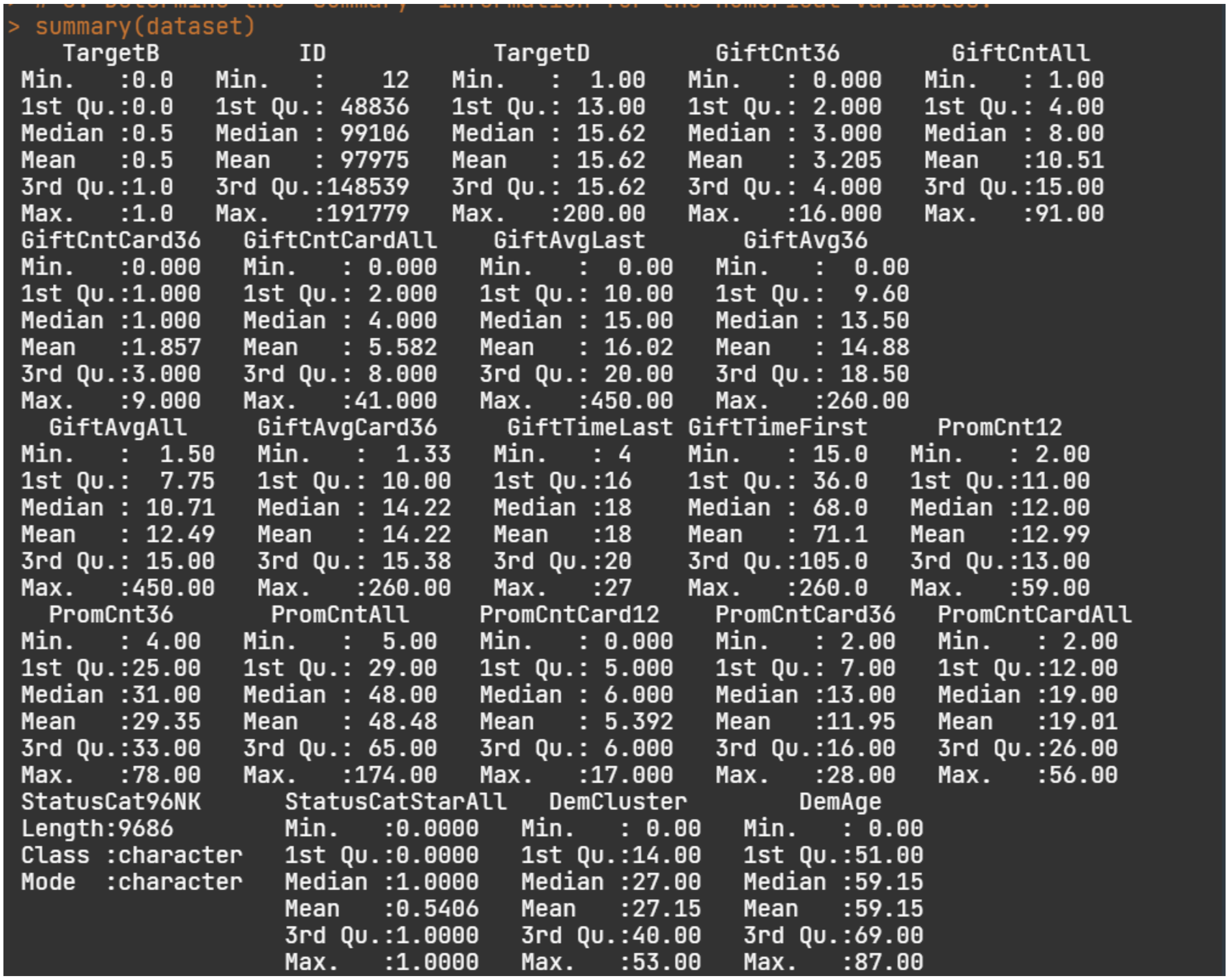


```
> for(i in 1:ncol(dataset)) {
+     if(is.numeric(dataset[,i])){
+         skurtosis[nrow(skurtosis) + 1,] = c(
+            colnames(dataset)[i],
+            round(skewness(dataset[,i]), 5),
+            round(kurtosis(dataset[,i]), 5)
+        )
+     }
+ }
> skurtosis
        X.Category. X.Skewness. X.Kurtosis.
1          Category    Skewness    Kurtosis
2           TargetB           0           1
3                ID     -0.0576     1.76499
4           TargetD      7.3085   111.59025
5         GiftCnt36     1.28815     5.04574
6        GiftCntAll     1.86282     9.04402
7     GiftCntCard36     1.17227     4.49348
8    GiftCntCardAll     1.33115      5.0232
9       GiftAvgLast     9.91736    248.9228
10        GiftAvg36     5.62692    80.05955
11       GiftAvgAll    14.48425   564.46467
12    GiftAvgCard36     6.69686   110.34936
13     GiftTimeLast    -0.77793     5.46718
14    GiftTimeFirst     0.19537     1.75216
15        PromCnt12     2.87328    14.98857
16        PromCnt36     0.26192      5.1726
17       PromCntAll     0.46069     3.21586
18    PromCntCard12     0.68489     8.79507
19    PromCntCard36    -0.42653     2.01304
20   PromCntCardAll     0.14283     2.21947
21 StatusCatStarAll    -0.16283     1.02651
22       DemCluster     -0.0867     1.87734
23           DemAge    -0.44738     3.35583
24  DemMedHomeValue     2.37784     9.44742
25   DemPctVeterans    -0.20703     4.27313
26     DemMedIncome     0.30998      3.6359
```

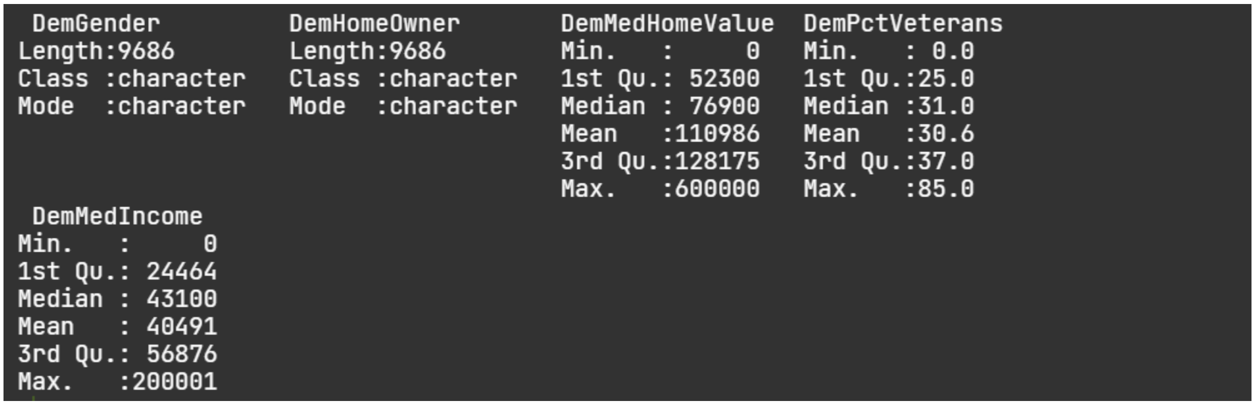
Skewness and Kurtosis of GiftCntAll



5. Summary of dataset

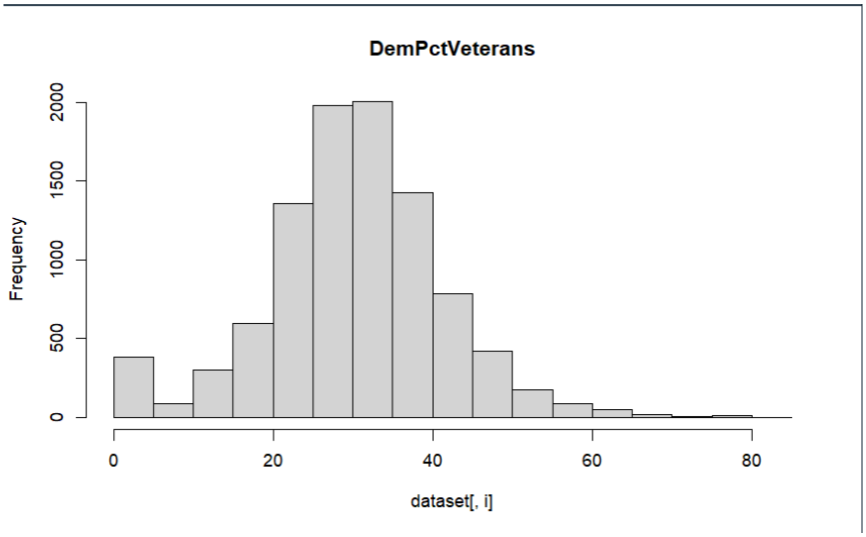


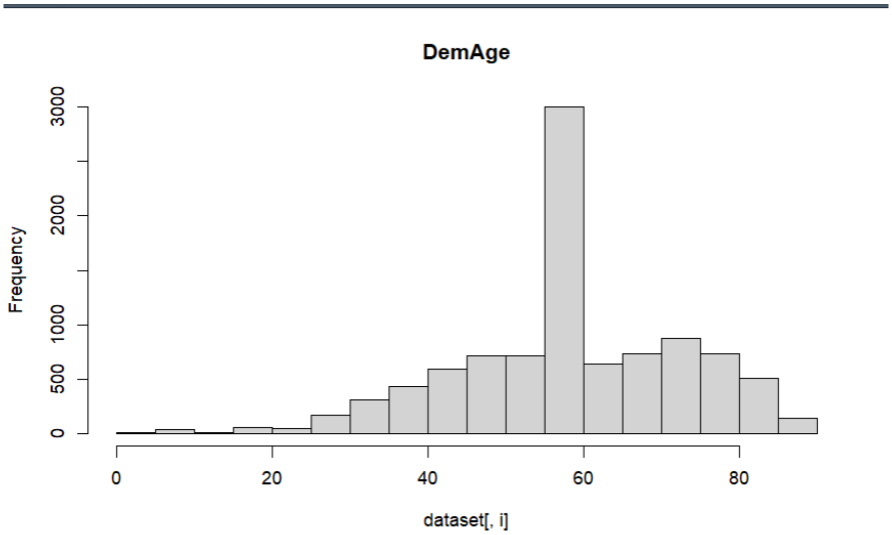
```
> summary(dataset)
    TargetB          ID            TargetD         GiftCnt36        GiftCntAll
 Min.   :0.0   Min.   :    12   Min.   :  1.00   Min.   : 0.000   Min.   : 1.00
 1st Qu.:0.0   1st Qu.: 48836   1st Qu.: 13.00   1st Qu.: 2.000   1st Qu.: 4.00
 Median :0.5   Median : 99106   Median : 15.62   Median : 3.000   Median : 8.00
 Mean   :0.5   Mean   : 97975   Mean   : 15.62   Mean   : 3.205   Mean   :10.51
 3rd Qu.:1.0   3rd Qu.:148539   3rd Qu.: 15.62   3rd Qu.: 4.000   3rd Qu.:15.00
 Max.   :1.0   Max.   :191779   Max.   :200.00   Max.   :16.000   Max.   :91.00
 GiftCntCard36   GiftCntCardAll    GiftAvgLast        GiftAvg36
 Min.   :0.000   Min.   : 0.000   Min.   :  0.00   Min.   :  0.00
 1st Qu.:1.000   1st Qu.: 2.000   1st Qu.: 10.00   1st Qu.:  9.60
 Median :1.000   Median : 4.000   Median : 15.00   Median : 13.50
 Mean   :1.857   Mean   : 5.582   Mean   : 16.02   Mean   : 14.88
 3rd Qu.:3.000   3rd Qu.: 8.000   3rd Qu.: 20.00   3rd Qu.: 18.50
 Max.   :9.000   Max.   :41.000   Max.   :450.00   Max.   :260.00
   GiftAvgAll      GiftAvgCard36      GiftTimeLast GiftTimeFirst      PromCnt12
 Min.   :  1.50   Min.   :  1.33   Min.   : 4   Min.   : 15.0   Min.   : 2.00
 1st Qu.:  7.75   1st Qu.: 10.00   1st Qu.:16   1st Qu.: 36.0   1st Qu.:11.00
 Median : 10.71   Median : 14.22   Median :18   Median : 68.0   Median :12.00
 Mean   : 12.49   Mean   : 14.22   Mean   :18   Mean   : 71.1   Mean   :12.99
 3rd Qu.: 15.00   3rd Qu.: 15.38   3rd Qu.:20   3rd Qu.:105.0   3rd Qu.:13.00
 Max.   :450.00   Max.   :260.00   Max.   :27   Max.   :260.0   Max.   :59.00
   PromCnt36       PromCntAll      PromCntCard12    PromCntCard36    PromCntCardAll
 Min.   : 4.00   Min.   :  5.00   Min.   : 0.000   Min.   : 2.00   Min.   : 2.00
 1st Qu.:25.00   1st Qu.: 29.00   1st Qu.: 5.000   1st Qu.: 7.00   1st Qu.:12.00
 Median :31.00   Median : 48.00   Median : 6.000   Median :13.00   Median :19.00
 Mean   :29.35   Mean   : 48.48   Mean   : 5.392   Mean   :11.95   Mean   :19.01
 3rd Qu.:33.00   3rd Qu.: 65.00   3rd Qu.: 6.000   3rd Qu.:16.00   3rd Qu.:26.00
 Max.   :78.00   Max.   :174.00   Max.   :17.000   Max.   :28.00   Max.   :56.00
 StatusCat96NK      StatusCatStarAll   DemCluster         DemAge
 Length:9686        Min.   :0.0000   Min.   : 0.00   Min.   : 0.00
 Class :character   1st Qu.:0.0000   1st Qu.:14.00   1st Qu.:51.00
 Mode  :character   Median :1.0000   Median :27.00   Median :59.15
                    Mean   :0.5406   Mean   :27.15   Mean   :59.15
                    3rd Qu.:1.0000   3rd Qu.:40.00   3rd Qu.:69.00
                    Max.   :1.0000   Max.   :53.00   Max.   :87.00
```

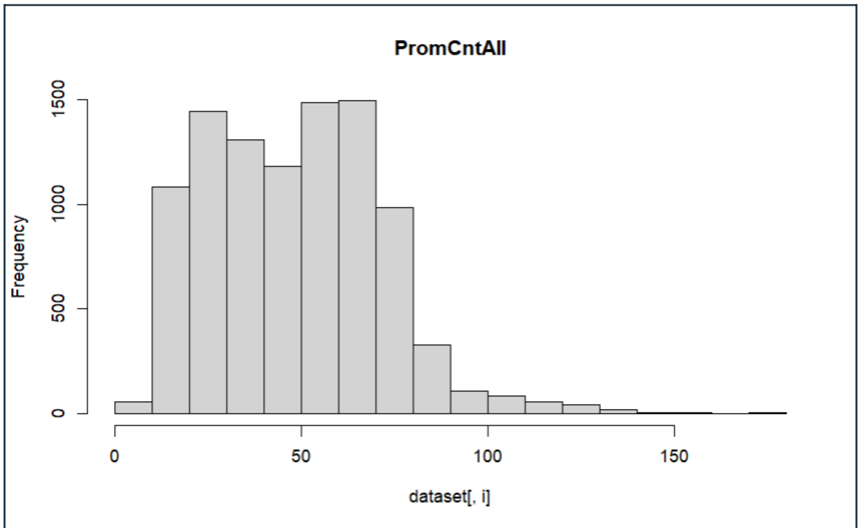


```
  DemGender         DemHomeOwner       DemMedHomeValue  DemPctVeterans
 Length:9686        Length:9686        Min.   :     0   Min.   : 0.0
 Class :character   Class :character   1st Qu.: 52300   1st Qu.:25.0
 Mode  :character   Mode  :character   Median : 76900   Median :31.0
                                       Mean   :110986   Mean   :30.6
                                       3rd Qu.:128175   3rd Qu.:37.0
                                       Max.   :600000   Max.   :85.0
  DemMedIncome
 Min.   :     0
 1st Qu.: 24464
 Median : 43100
 Mean   : 40491
 3rd Qu.: 56876
 Max.   :200001
```

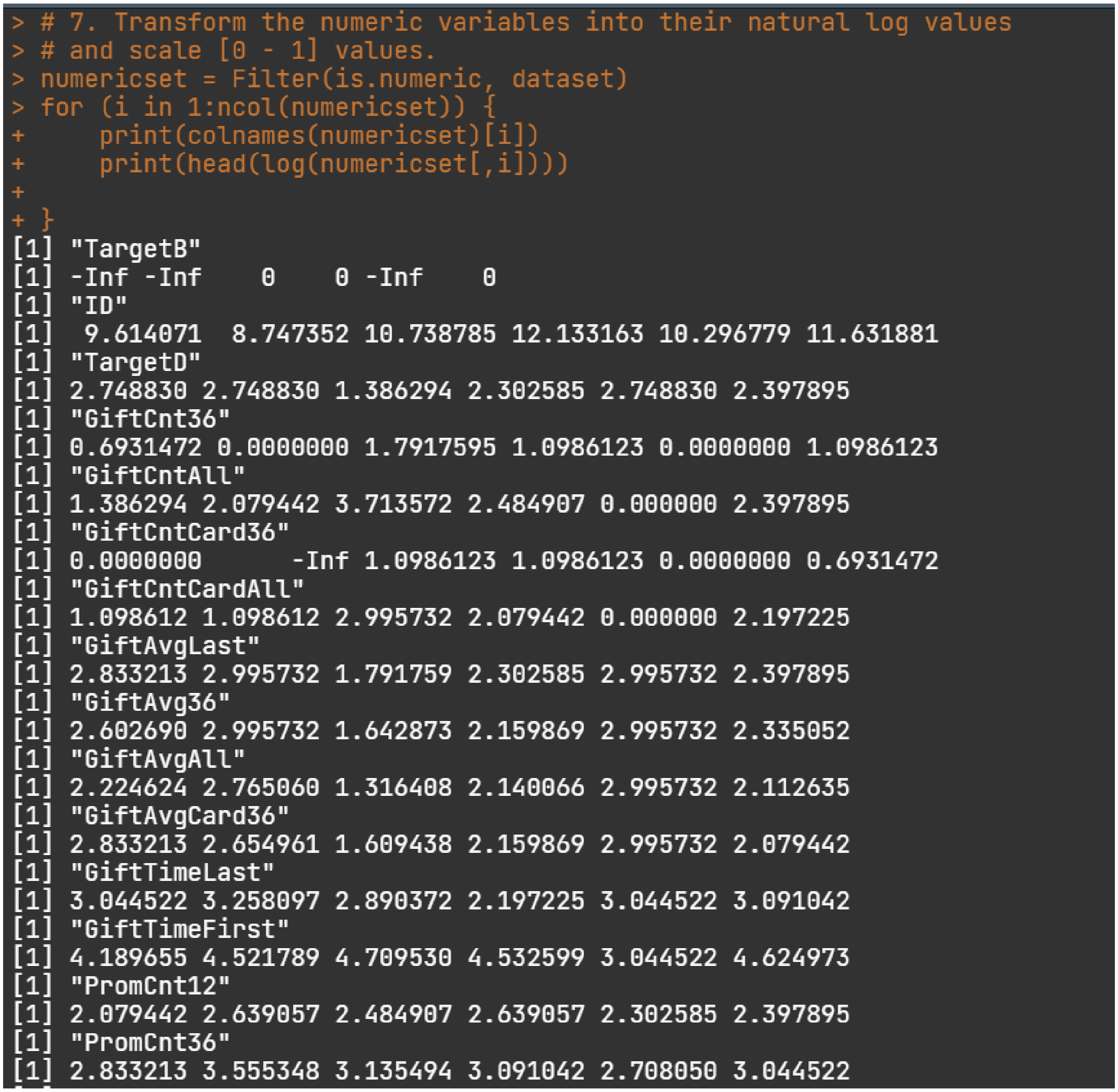
6. Distributions of numeric variables and plotting the distributions



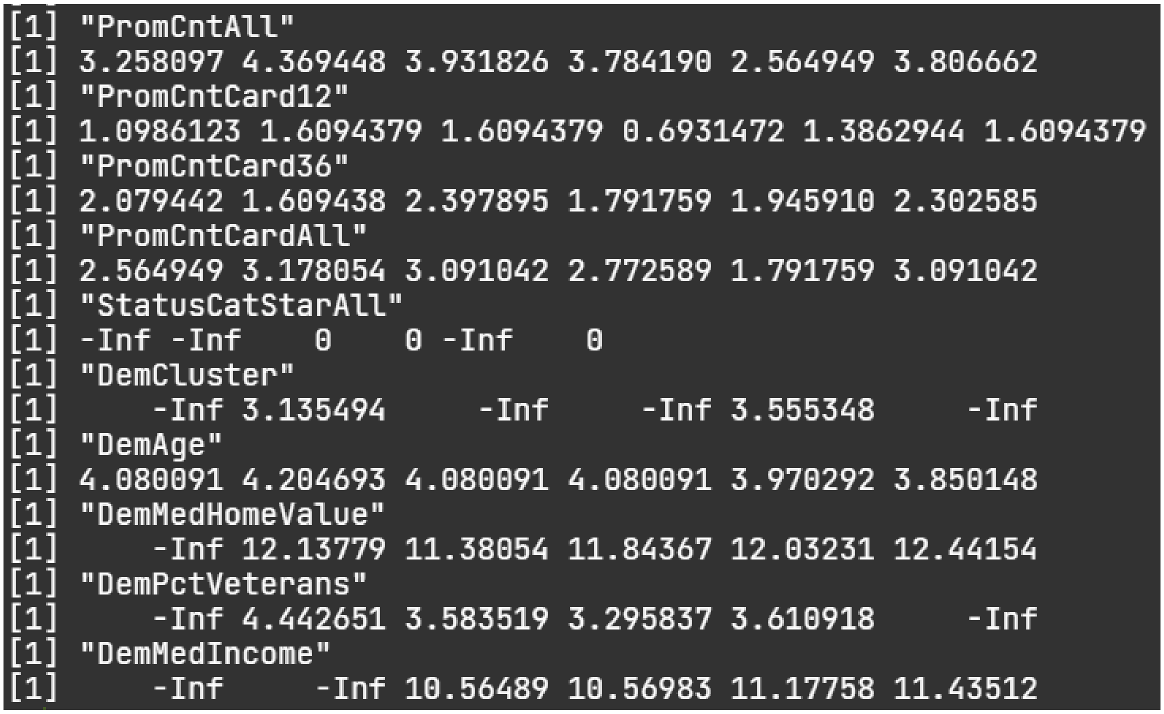




7. Transform numeric variables to their natural log



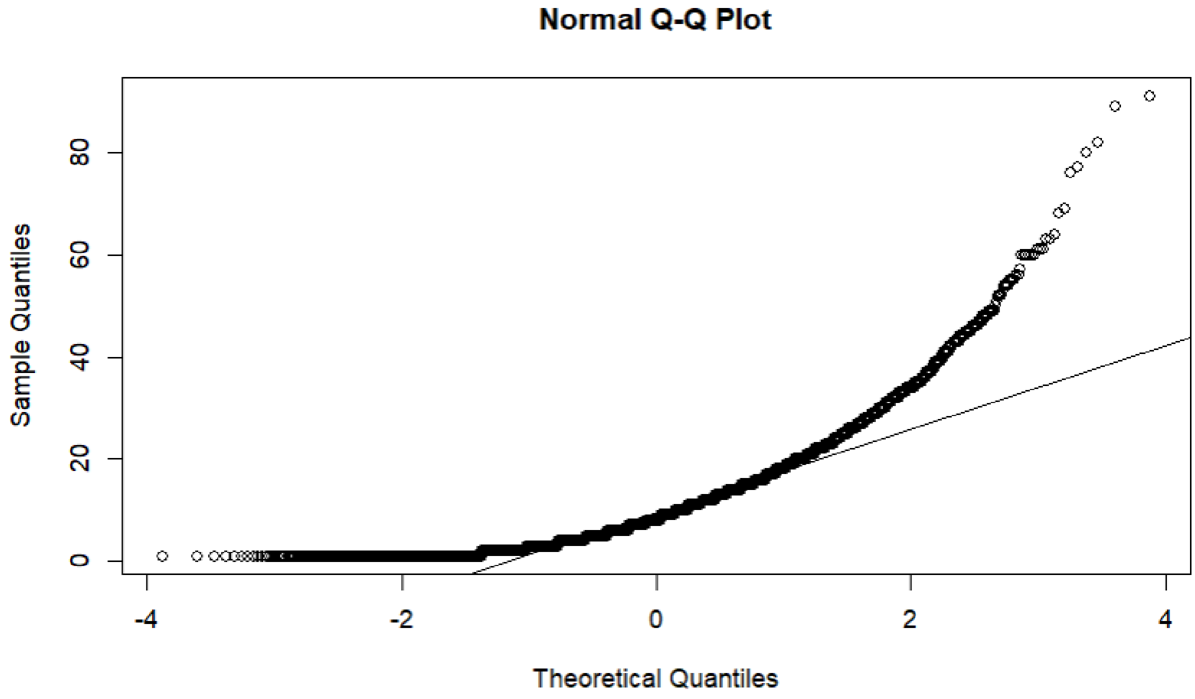
```
> # 7. Transform the numeric variables into their natural log values
> # and scale [0 - 1] values.
> numericset = Filter(is.numeric, dataset)
> for (i in 1:ncol(numericset)) {
+     print(colnames(numericset)[i])
+     print(head(log(numericset[,i])))
+
+ }
[1] "TargetB"
[1] -Inf -Inf    0    0 -Inf    0
[1] "ID"
[1]  9.614071  8.747352 10.738785 12.133163 10.296779 11.631881
[1] "TargetD"
[1] 2.748830 2.748830 1.386294 2.302585 2.748830 2.397895
[1] "GiftCnt36"
[1] 0.6931472 0.0000000 1.7917595 1.0986123 0.0000000 1.0986123
[1] "GiftCntAll"
[1] 1.386294 2.079442 3.713572 2.484907 0.000000 2.397895
[1] "GiftCntCard36"
[1] 0.0000000      -Inf 1.0986123 1.0986123 0.0000000 0.6931472
[1] "GiftCntCardAll"
[1] 1.098612 1.098612 2.995732 2.079442 0.000000 2.197225
[1] "GiftAvgLast"
[1] 2.833213 2.995732 1.791759 2.302585 2.995732 2.397895
[1] "GiftAvg36"
[1] 2.602690 2.995732 1.642873 2.159869 2.995732 2.335052
[1] "GiftAvgAll"
[1] 2.224624 2.765060 1.316408 2.140066 2.995732 2.112635
[1] "GiftAvgCard36"
[1] 2.833213 2.654961 1.609438 2.159869 2.995732 2.079442
[1] "GiftTimeLast"
[1] 3.044522 3.258097 2.890372 2.197225 3.044522 3.091042
[1] "GiftTimeFirst"
[1] 4.189655 4.521789 4.709530 4.532599 3.044522 4.624973
[1] "PromCnt12"
[1] 2.079442 2.639057 2.484907 2.639057 2.302585 2.397895
[1] "PromCnt36"
[1] 2.833213 3.555348 3.135494 3.091042 2.708050 3.044522
```



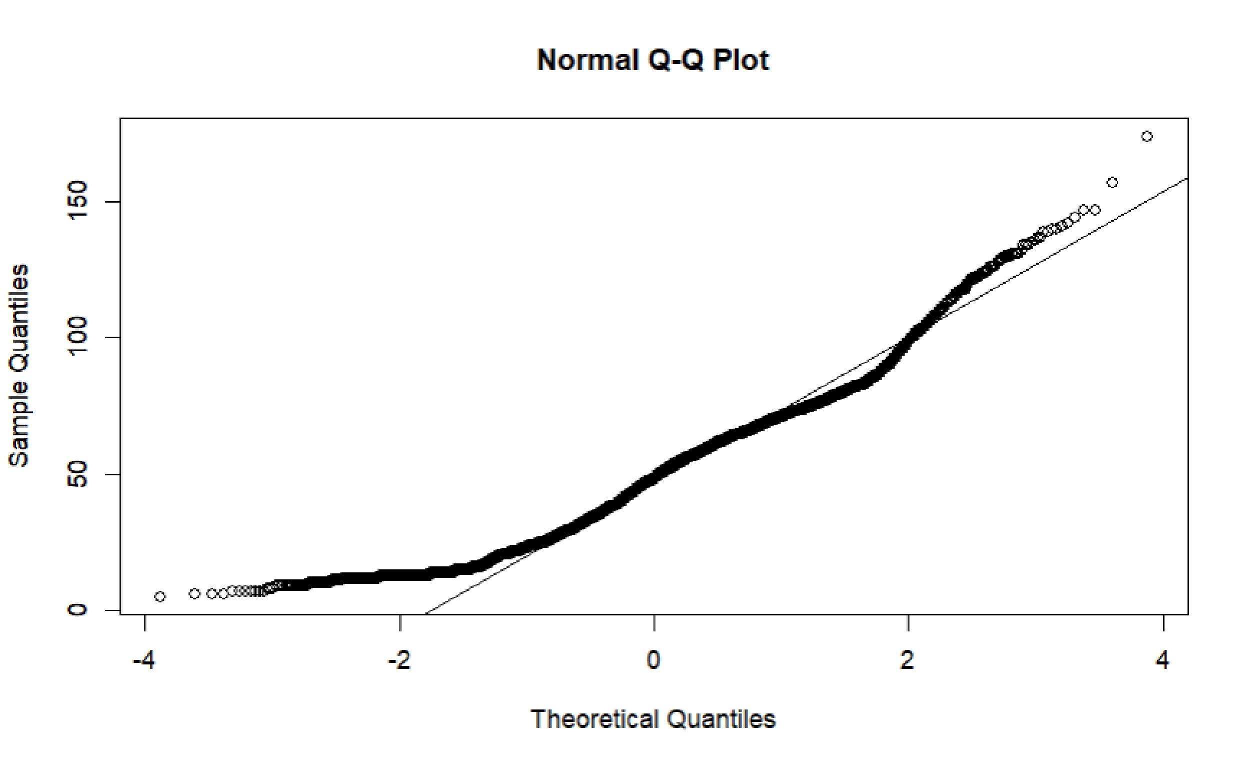
```
[1] "PromCntAll"
[1] 3.258097 4.369448 3.931826 3.784190 2.564949 3.806662
[1] "PromCntCard12"
[1] 1.0986123 1.6094379 1.6094379 0.6931472 1.3862944 1.6094379
[1] "PromCntCard36"
[1] 2.079442 1.609438 2.397895 1.791759 1.945910 2.302585
[1] "PromCntCardAll"
[1] 2.564949 3.178054 3.091042 2.772589 1.791759 3.091042
[1] "StatusCatStarAll"
[1] -Inf -Inf    0    0 -Inf    0
[1] "DemCluster"
[1]     -Inf 3.135494     -Inf     -Inf 3.555348     -Inf
[1] "DemAge"
[1] 4.080091 4.204693 4.080091 4.080091 3.970292 3.850148
[1] "DemMedHomeValue"
[1]     -Inf 12.13779 11.38054 11.84367 12.03231 12.44154
[1] "DemPctVeterans"
[1]     -Inf 4.442651 3.583519 3.295837 3.610918     -Inf
[1] "DemMedIncome"
[1]     -Inf     -Inf 10.56489 10.56983 11.17758 11.43512
```

8. Check whether the numeric variables follow normality conditions.

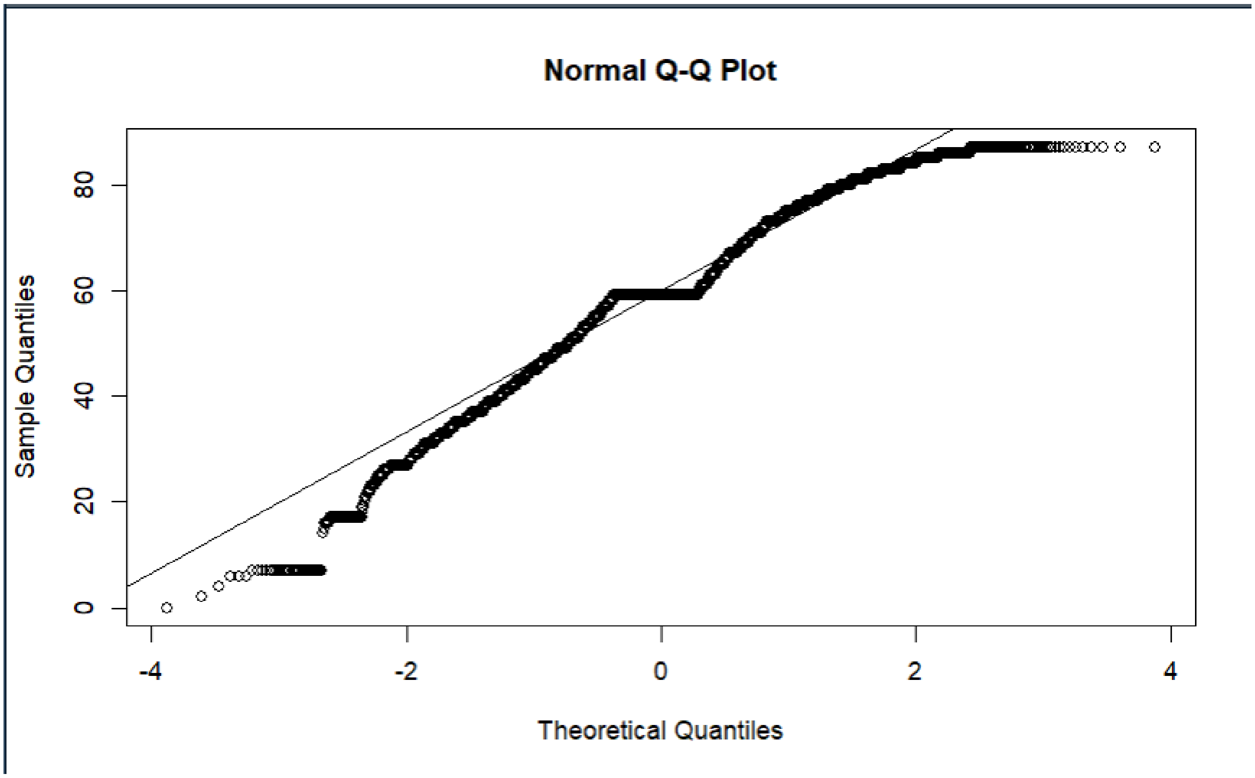
Normality of GiftCntAll



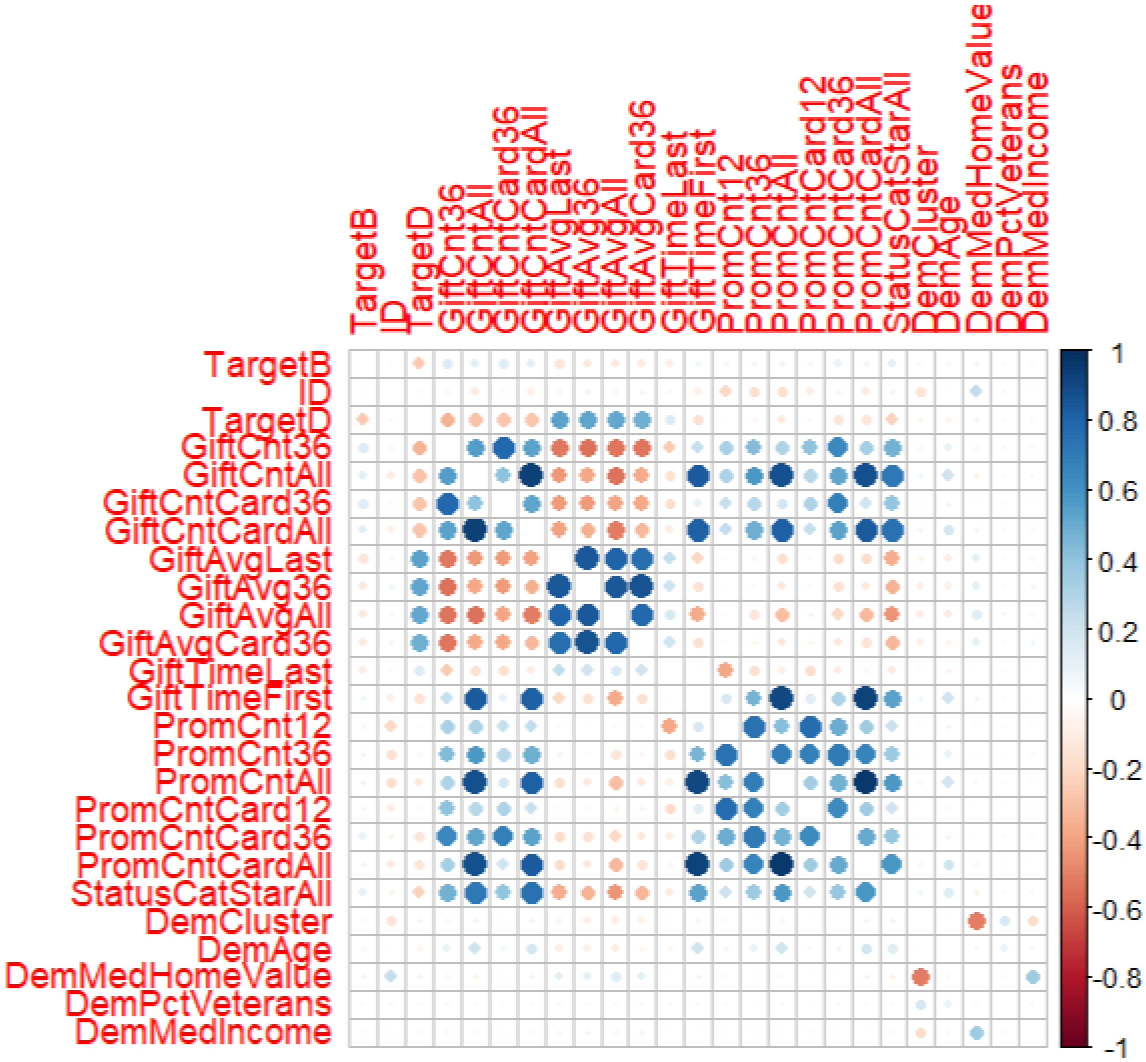
Normality of PromCntAll



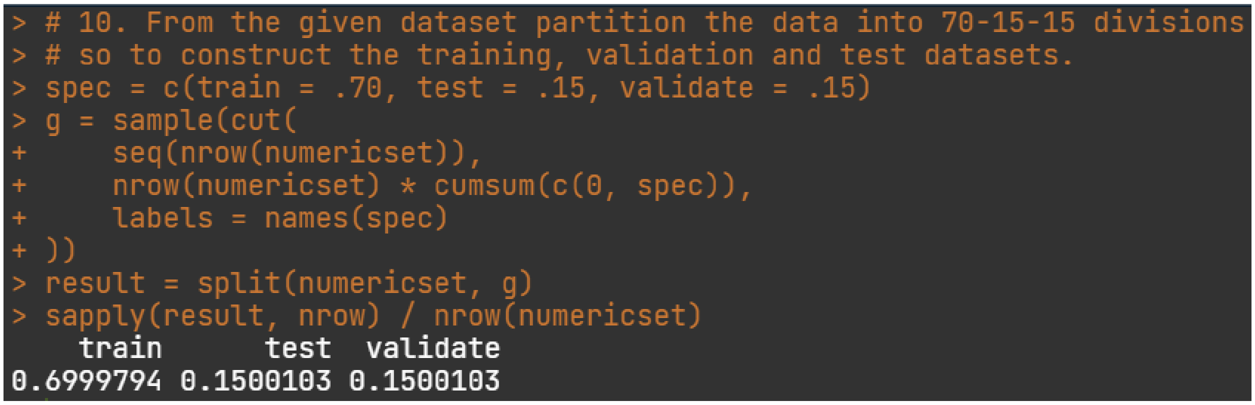
Normality of DemAge



9. Find the correlation matrix



10. From the given dataset partition the data into 70-15-15 divisions so to construct the training, validation and test datasets.



```
> # 10. From the given dataset partition the data into 70-15-15 divisions
> # so to construct the training, validation and test datasets.
> spec = c(train = .70, test = .15, validate = .15)
> g = sample(cut(
+     seq(nrow(numericset)),
+     nrow(numericset) * cumsum(c(0, spec)),
+     labels = names(spec)
+ ))
> result = split(numericset, g)
> sapply(result, nrow) / nrow(numericset)
    train      test  validate 
0.6999794 0.1500103 0.1500103 
```