Project 3

2022-10-18

Part I

Part I

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{i' \in C_k} \| x_i - x_{i'} \|^2$$

1. Add and subtract mean vector $\bar{x}_k$

$$= \frac{1}{2} \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{i' \in C_k} \| (x_i - \bar{x}_k) + (\bar{x}_k - x_{i'}) \|^2$$

2. Obtain an inner product

$$= \frac{1}{2} \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{i' \in C_k} \left(\| x_i - \bar{x}_k \|^2 + \| x_{i'} - \bar{x}_k \|^2 - 2 \langle x_i - \bar{x}_k, x_{i'} - \bar{x}_k \rangle \right)$$

3. Recover n_k by properties of Summations

$$= \frac{1}{2} \sum_{k=1}^{K} \left[n_k \sum_{i \in C_k} \| x_i - \bar{x}_k \|^2 + n_k \sum_{i' \in C_k} \| x_{i'} - \bar{x}_k \|^2 - 2 n_k \left\langle \sum_{i \in C_k} (x_i - \bar{x}_k), \sum_{i' \in C_k} x_{i'} - \bar{x}_k \right\rangle \right]$$

4. simplify

$$= \frac{1}{2} \sum_{k=1}^{K} 2 n_k \sum_{i \in C_k} \| x_i - \bar{x}_k \|^2$$

5. Final answer

$$\sum_{k=1}^{K} n_k \sum_{i \in C_k} \| x_i - \bar{x}_k \|^2$$

Part II

```
library(mice)
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(cluster)
library(Rtsne)
library(clv)
```

```
## Loading required package: class
```

```
set.seed(0)
```

(i) Reading Data

```
hmeq=read.csv("hmeq.csv")
m=dim(hmeq)[1]
n=dim(hmeq)[2]
col_names=colnames(hmeq)
missing_matrix=is.na(hmeq) | hmeq==""
for (i in 1:n) {
  out=c(0,0)
  missing=sum(missing_matrix[,i])
  per_missing=(missing/m)*100
  out[1]=col_names[i]
  out[2]=per_missing
  print(out)
}
```

```
## [1] "BAD" "0"
## [1] "LOAN" "0"
## [1] "MORTDUE"          "8.69127516778524"
## [1] "VALUE"            "1.87919463087248"
## [1] "REASON"           "4.22818791946309"
## [1] "JOB"              "4.68120805369128"
## [1] "YOJ"              "8.64093959731544"
## [1] "DEROG"            "11.8791946308725"
## [1] "DELINQ"           "9.73154362416107"
## [1] "CLAGE"            "5.16778523489933"
## [1] "NINQ"             "8.55704697986577"
## [1] "CLNO"            "3.7248322147651"
## [1] "DEBTINC"         "21.258389261745"
```

(ii) Data Cleaning

(a) Missing Values

```
hmeq$JOB[hmeq$JOB == "" | is.na(hmeq$JOB)] <- "Unknown"
hmeq$REASON[hmeq$REASON == "" | is.na(hmeq$REASON)] <- "Unknown"
table(hmeq[,c("JOB", "REASON")])
```

```
##          REASON
## JOB       DebtCon HomeImp Unknown
##   Mgr         572     174      21
##   Office      620     301      27
##   Other      1604     716      68
##   ProfExe     847     405      24
##   Sales        97      12       0
##   Self         73     115       5
##   Unknown     115      57     107
```

(b) Logarithm Transformation

```
has_zero=function(x) {
  return(any(x==0, na.rm = TRUE))
}

log_columns=c("LOAN", "VALUE", "MORTDUE", "YOJ", "CLAGE")

for (j in log_columns) {
  if (has_zero(hmeq[,j])) {
    hmeq[,j]=log(hmeq[,j]+1)
  }
  else {
    hmeq[,j]=log(hmeq[,j])
  }
}
```

(c) Imputation

```
temp_hmeq=mice(hmeq, m = 1)
```

```
##
##  iter imp variable
##   1   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   2   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   3   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   4   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   5   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
```

```
## Warning: Number of logged events: 2
```

```
imputed_hmeq=complete(temp_hmeq, 1)
```

(iii) Distance Matrix

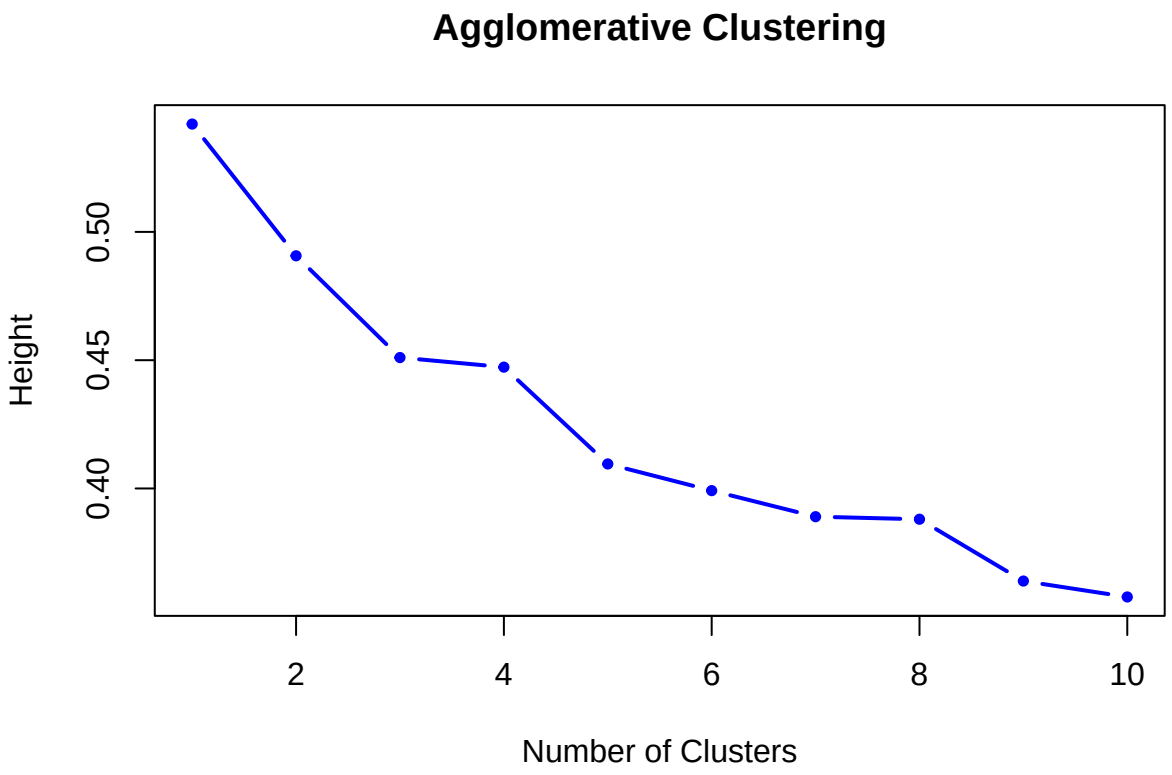
```
imputed_hmeq$REASON=as.factor(imputed_hmeq$REASON)
imputed_hmeq$JOB=as.factor(imputed_hmeq$JOB)
```

```
d=daisy(imputed_hmeq[,-1])
```

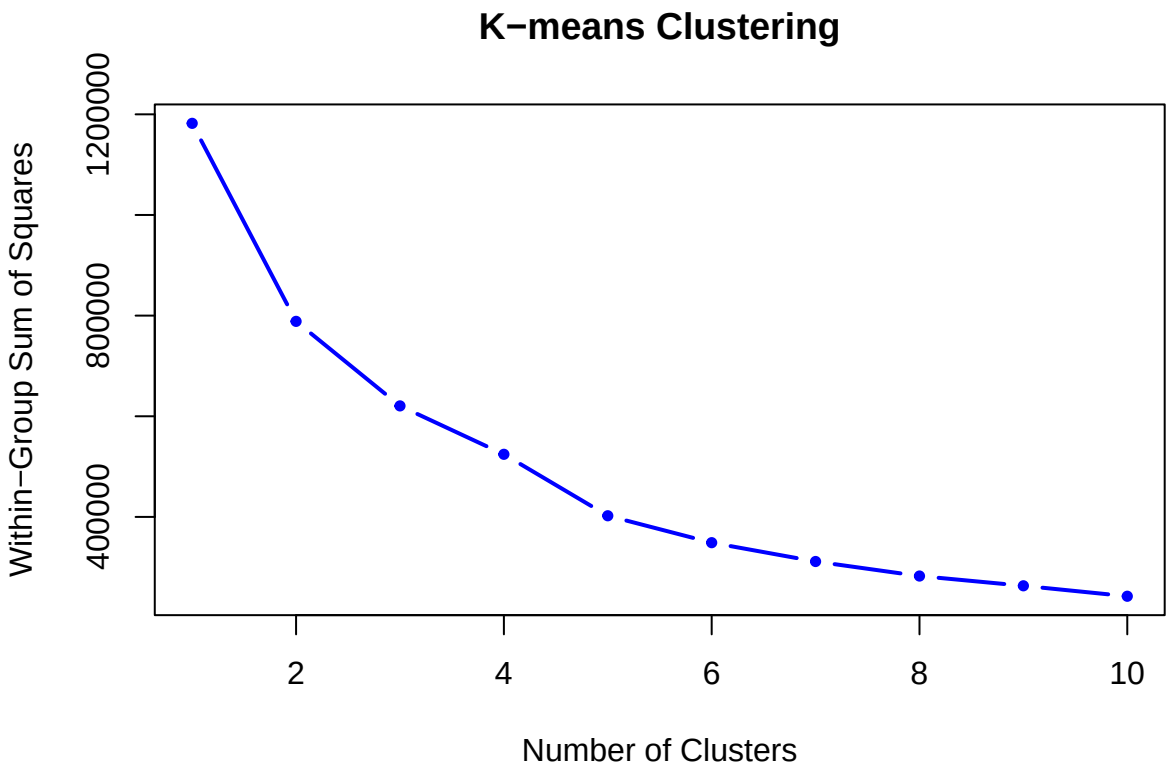
(iv) Clustering

(a) Implementation

```
fit=hclust(d, method="complete")
K.max <- 10
height <- tail(fit$height, n=K.max)
n.cluster <- tail((nrow(imputed_hmeq)-1):1, n=K.max)
plot(n.cluster, height,  type="b", pch=19, cex=.5, xlab="Number of Clusters",
    ylab="Height", col="blue", lwd=2, main = "Agglomerative Clustering")
```




```
K.max=10
wss=1:10
for (K in 1:K.max) wss[K]=sum(kmeans(imputed_hmeq[,-c(1,5,6)], centers = K)$withinss)
plot(1:K.max, wss, type="b", xlab = "Number of Clusters", ylab="Within-Group Sum of Squares",
      pch=19, cex=.5, col="blue", lwd=2, main = "K-means Clustering")
```




```
fit2=kmeans(imputed_hmeq[,-c(1,5,6)], centers = 2)
```

(b) Optimal Number of Clusters

For both hierarchical clustering and K-means clustering, the optimal number of clusters I determined was 2. I arrived at this number by plotting the number of clusters against a method-specific value. These values are height (calculated by the complete method) and within-group sum of squares for hierarchical clustering and K-means clustering, respectively.

(c) Important Options

Hierarchical Clustering The default parameters were chosen for running the hclust function. In my case, the only parameter worth describing is the method parameter. The allowable options for this parameter are: ward.D, ward.D2, single, complete, average, mcquitty, median, and centroid. The chosen parameter control how agglomeration occurs.

I chose the "complete" method through a process of elimination. The ward.D, ward.D2, and centroid methods all require access to the original data, but I wanted to see if a similarity matrix alone would suffice to cluster the data. The mcquitty and median methods were not discussed extensively in class so I did not use these. The average method is highly sensitive to the distance measure used so I did not use this method. Lastly, the single method is very sensitive to outliers so I did not use this method.

K-Means Clustering For K-means clustering, I did not choose any non-default parameters consciously as they were not discussed extensively in class. I do bear in mind that solving the K-means objective function requires using an algorithm which may not converge to a global minimum so I know that obtaining a solution can be dependent on the choice of algorithm. In my case, I went with Hartigan-Wong.

(d) Cluster Membership

```
hcl=cutree(fit, k=2)
km=fit2$cluster
```

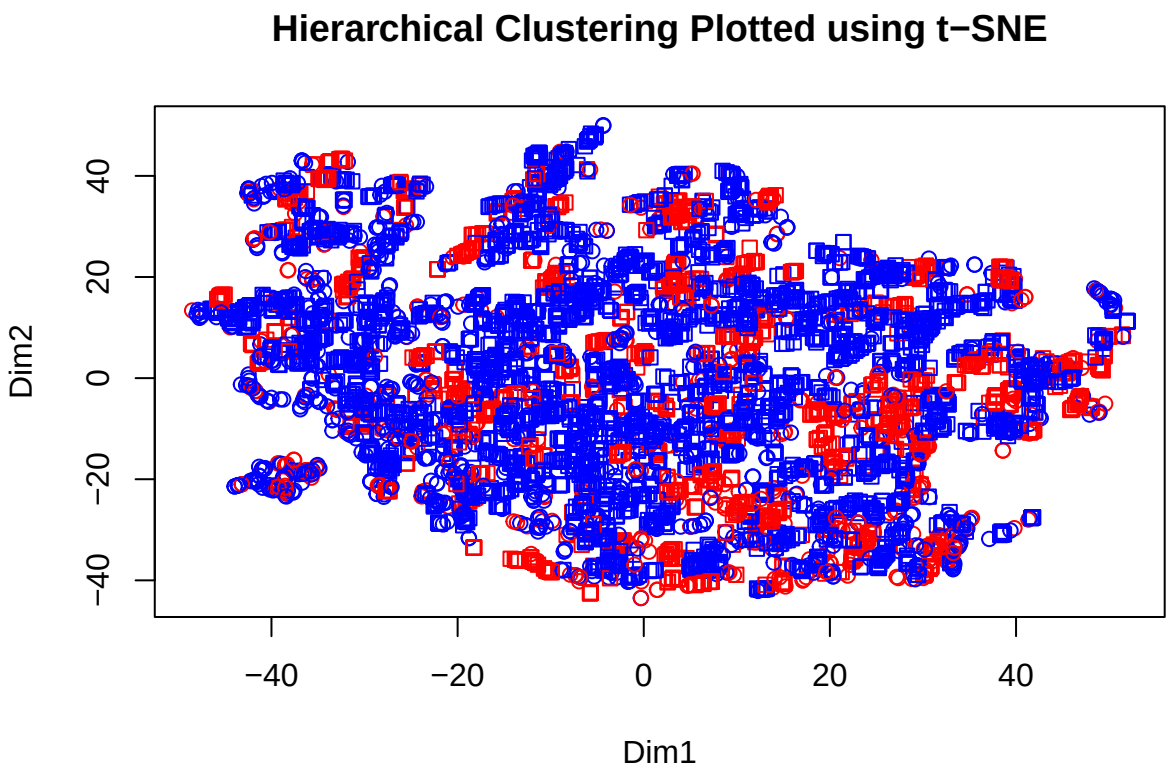
(e) MDS via t-SNE

Remark: Cluster 1 is represented by blue and cluster 2 by red. Also, a BAD value of 0 is represented by a square and 1 by a circle.

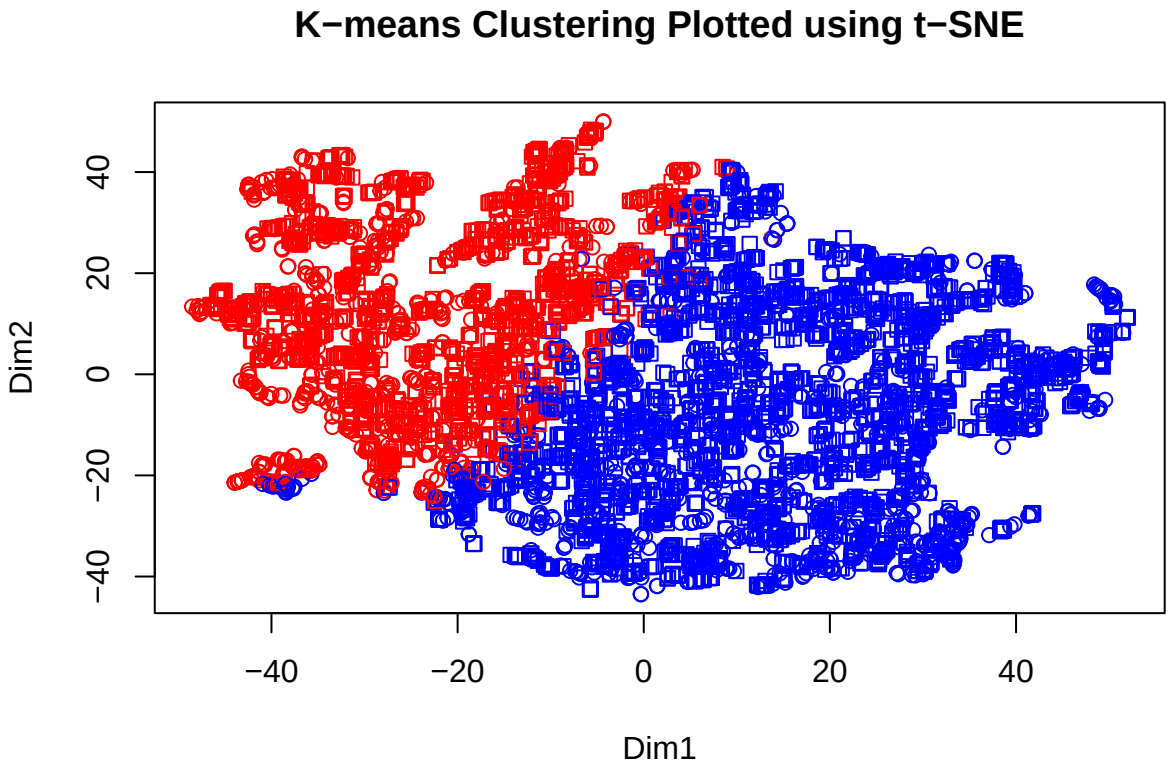
```
t=Rtsne(imputed_hmeq[,-1], num_threads=0)

assign_colors=function(num) {
  switch(num, "red", "blue")
}
h_cols=sapply(hcl, assign_colors)
k_cols=sapply(km, assign_colors)

plot(t$Y, xlab = "Dim1", ylab = "Dim2", main = "Hierarchical Clustering Plotted using t-SNE",
     col=h_cols, pch=imputed_hmeq$BAD)
```




```
plot(t$Y, xlab = "Dim1", ylab = "Dim2", main = "K-means Clustering Plotted using t-SNE",
     col=k_cols, pch=imputed_hmeq$BAD)
```



(f) Two-way Contingency Table

```
table(h_cols, k_cols)
```

```
##       k_cols
## h_cols blue  red
##   blue 2441 1739
##   red  1242  538
```

```
out=std.ext(hcl, km)
"Jaccard"
```

```
## [1] "Jaccard"
```

```
clv.Jaccard(out)
```

```
## [1] 0.378329
```

```
"Rand"
```

```
## [1] "Rand"
```

```
clv.Rand(out)
```

```
## [1] 0.4999161
```

(v) Analysis

```
imputed_hmeq$clus=km
clus1=imputed_hmeq[imputed_hmeq$clus==1,]
clus2=imputed_hmeq[imputed_hmeq$clus==2,]
"Contingency Table"
```

```
## [1] "Contingency Table"
```

```
table(imputed_hmeq$BAD, imputed_hmeq$clus)
```

```
##
##        1    2
##   0 1750 3021
##   1  527  662
```

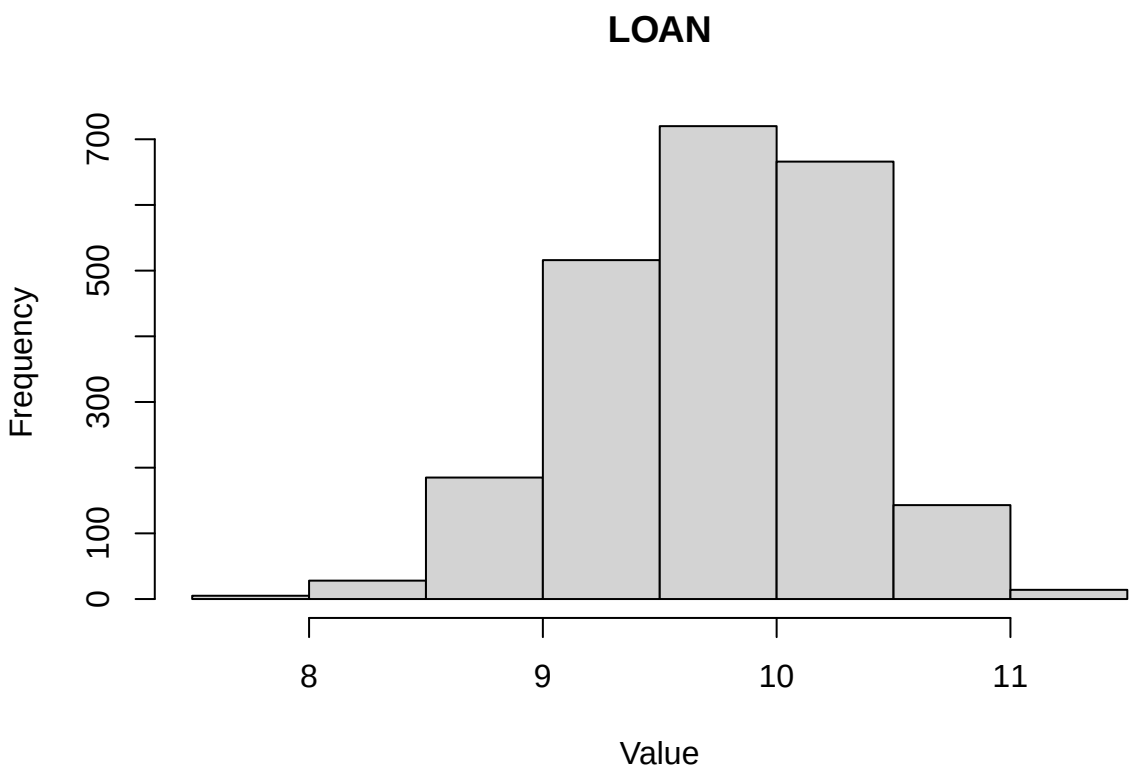
```
mat.data=c(1750/(1750+527),527/(1750+527),3021/(3021+662),662/(3021+662))
"Columnwise Percentage"
```

```
## [1] "Columnwise Percentage"
```

```
mat1=matrix(mat.data,nrow=2,ncol=2,byrow=FALSE, dimnames = list(c(0,1), c(1,2)))
mat1
```

```
##           1         2
## 0 0.7685551 0.8202552
## 1 0.2314449 0.1797448
```

```
for (i in c(2,3,4,8,9,10,11,12,13)) {
  name=colnames(imputed_hmeq)[i]
  print(hist(clus1[,name], main = name, xlab = "Value"))
}
```

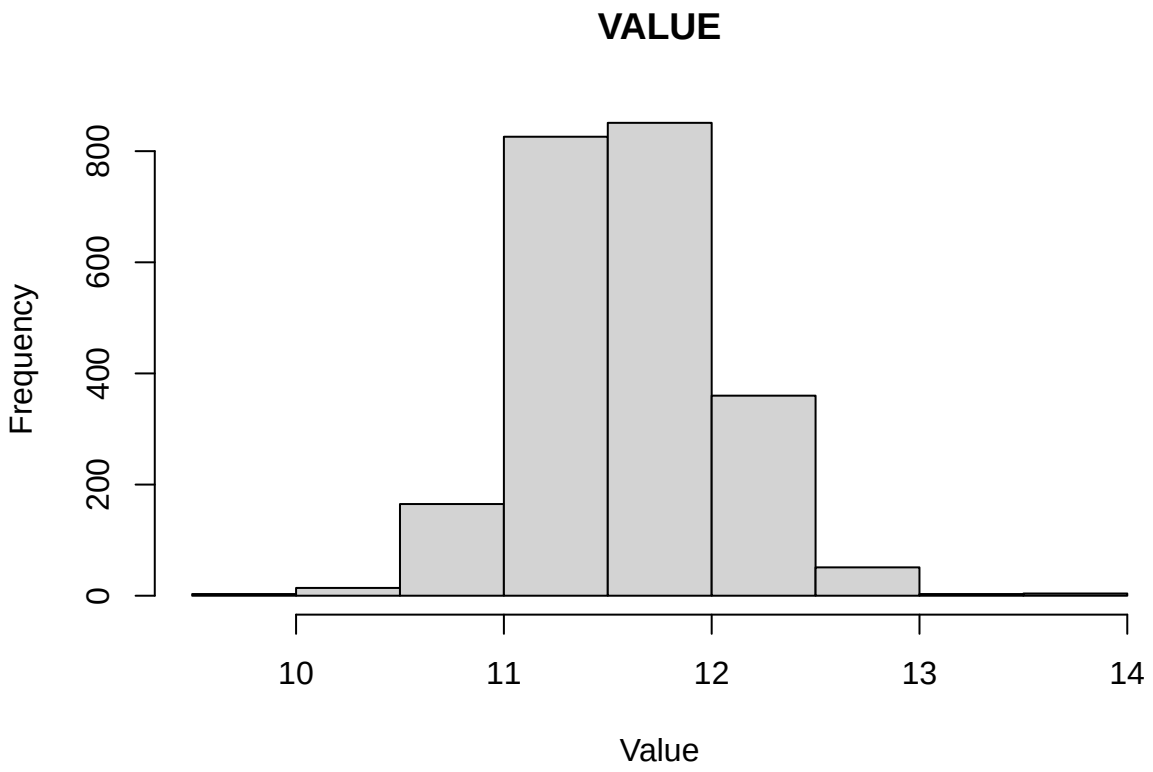



```
## $breaks
## [1]  7.5  8.0  8.5  9.0  9.5 10.0 10.5 11.0 11.5
##
## $counts
## [1]   5  28 185 516 720 666 143  14
##
## $density
```

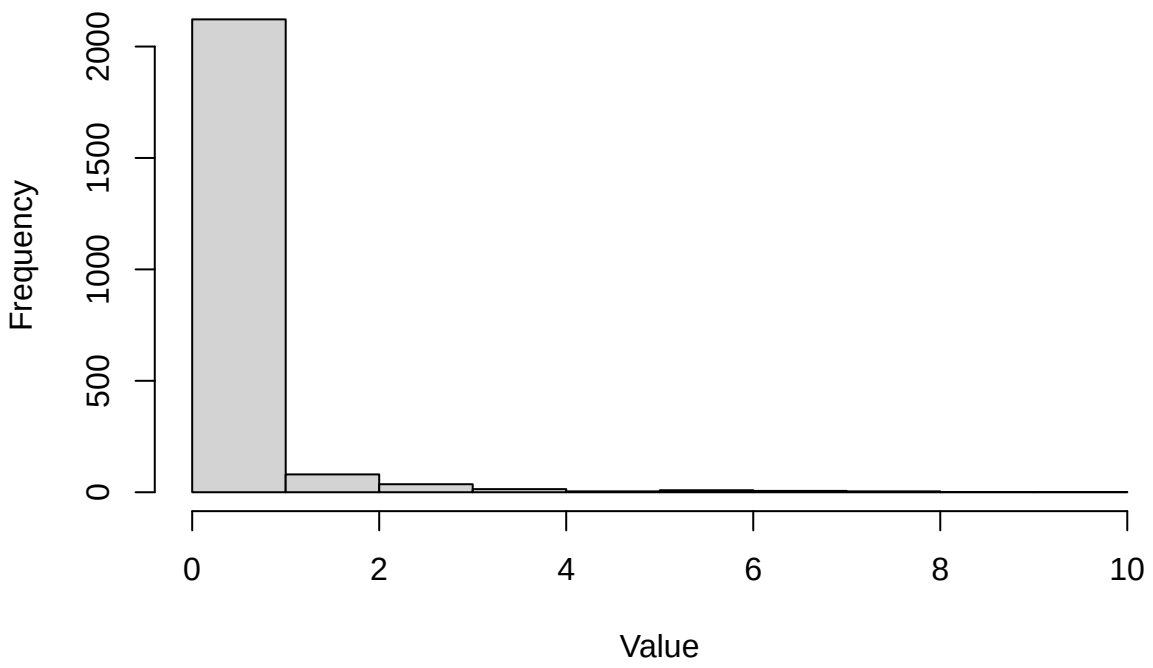
```
## [1] 0.004391744 0.024593764 0.162494510 0.453227931 0.632411067 0.584980237
## [7] 0.125603865 0.012296882
##
## $mids
## [1]  7.75  8.25  8.75  9.25  9.75 10.25 10.75 11.25
##
## $xname
## [1] "clus1[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```


```
## $breaks
##  [1]  8.5  9.0  9.5 10.0 10.5 11.0 11.5 12.0 12.5 13.0
##
## $counts
## [1]   3   9  37 127 479 968 482 164   8
##
## $density
## [1] 0.002635046 0.007905138 0.032498902 0.111550285 0.420729029 0.850241546
## [7] 0.423364076 0.144049188 0.007026790
##
## $mids
## [1]  8.75  9.25  9.75 10.25 10.75 11.25 11.75 12.25 12.75
##
## $xname
## [1] "clus1[, name]"
```

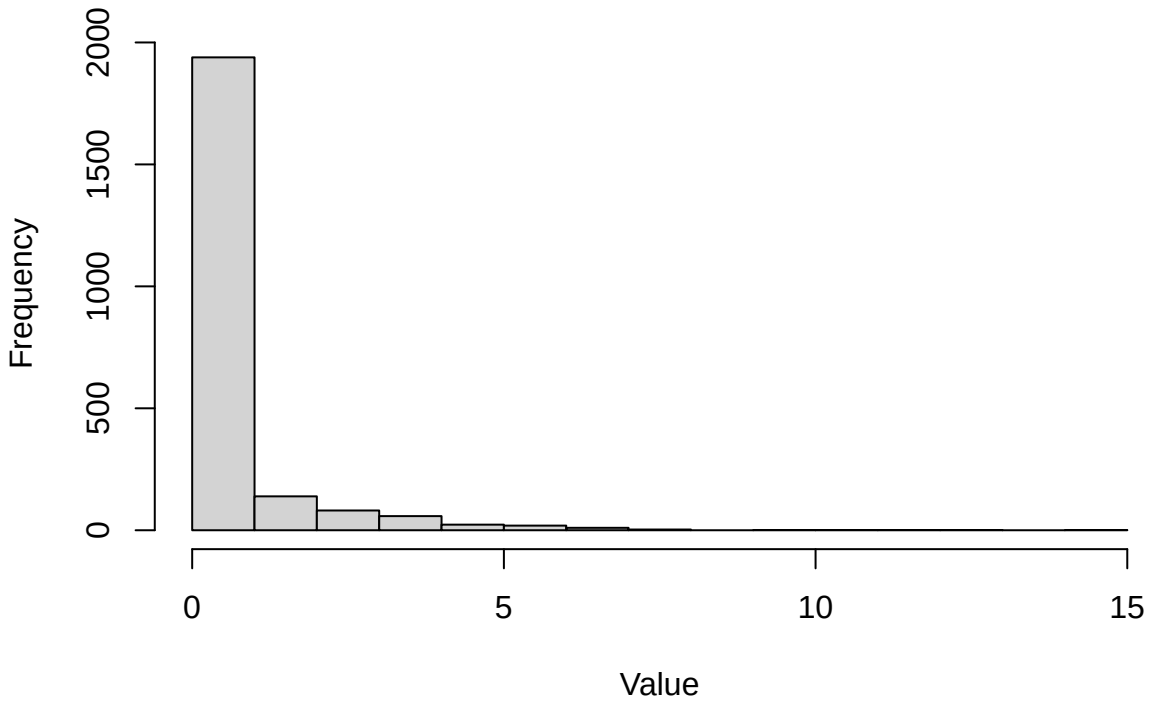
```
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```



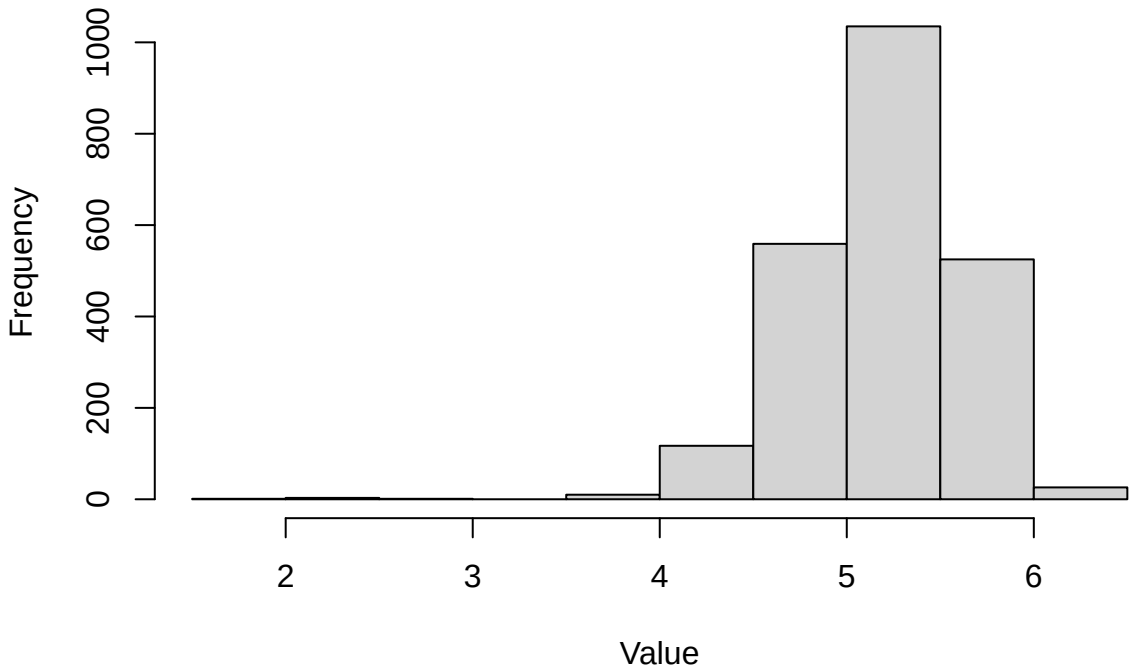

```
## $breaks
##  [1]  9.5 10.0 10.5 11.0 11.5 12.0 12.5 13.0 13.5 14.0
##
## $counts
## [1]   3  14 165 826 851 360  51   3   4
##
## $density
## [1] 0.002635046 0.012296882 0.144927536 0.725516030 0.747474747 0.316205534
## [7] 0.044795784 0.002635046 0.003513395
##
## $mids
## [1]  9.75 10.25 10.75 11.25 11.75 12.25 12.75 13.25 13.75
##
## $xname
## [1] "clus1[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```



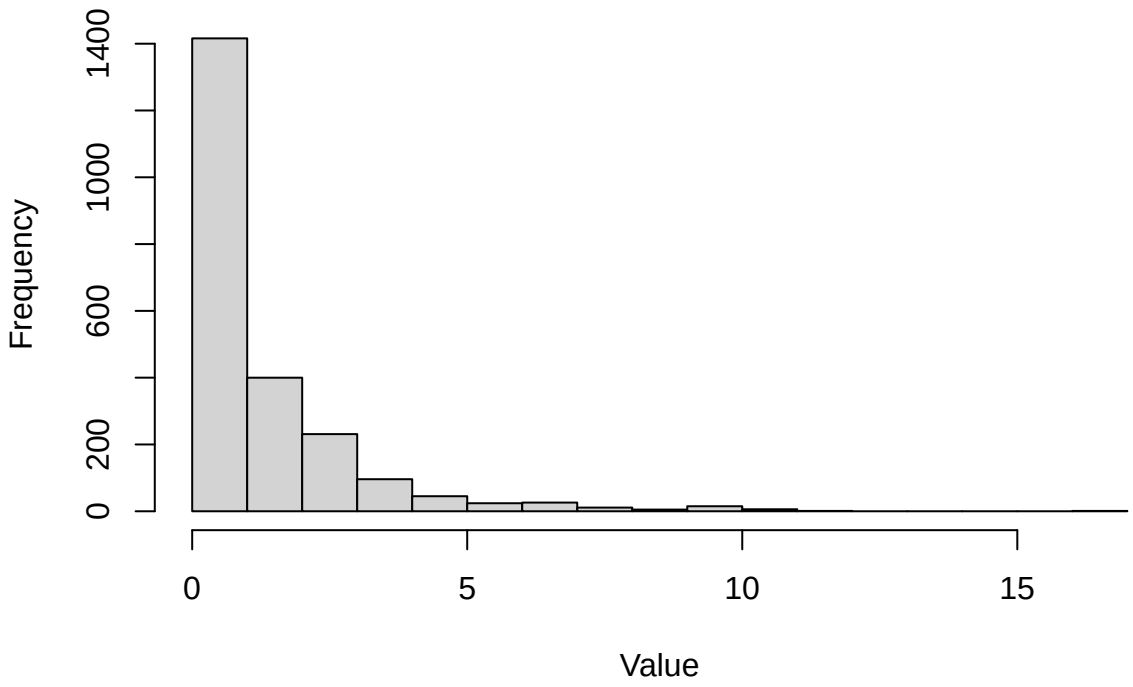

```
## $breaks
##  [1]  0  1  2  3  4  5  6  7  8  9 10
##
## $counts
##  [1] 2122   80   36   14    4    9    6    4    1    1
##
## $density
##  [1] 0.9319279754 0.0351339482 0.0158102767 0.0061484409 0.0017566974
##  [6] 0.0039525692 0.0026350461 0.0017566974 0.0004391744 0.0004391744
##
## $mids
##  [1] 0.5 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5
##
## $xname
## [1] "clus1[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```



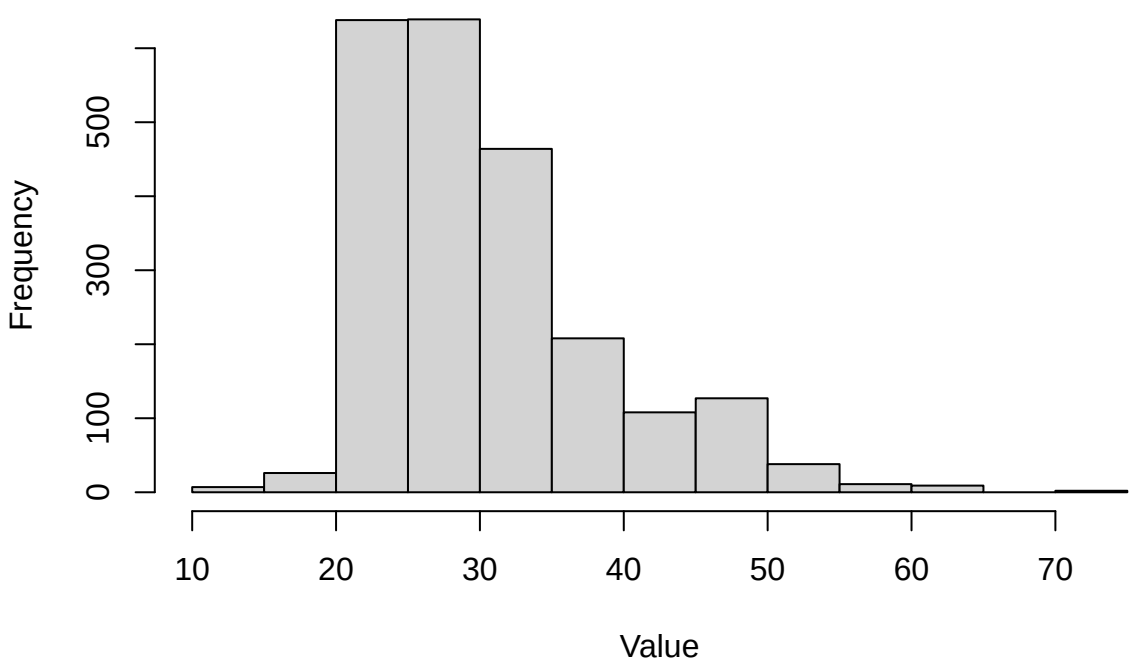

```
## $breaks
##  [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
##
## $counts
##  [1] 1939  139   81   58   23   19   10    3    0    1    1    1    1    0    1
##
## $density
##  [1] 0.8515590690 0.0610452350 0.0355731225 0.0254721124 0.0101010101
##  [6] 0.0083443127 0.0043917435 0.0013175231 0.0000000000 0.0004391744
## [11] 0.0004391744 0.0004391744 0.0004391744 0.0000000000 0.0004391744
##
## $mids
##  [1]  0.5  1.5  2.5  3.5  4.5  5.5  6.5  7.5  8.5  9.5 10.5 11.5 12.5 13.5 14.5
##
## $xname
## [1] "clus1[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```



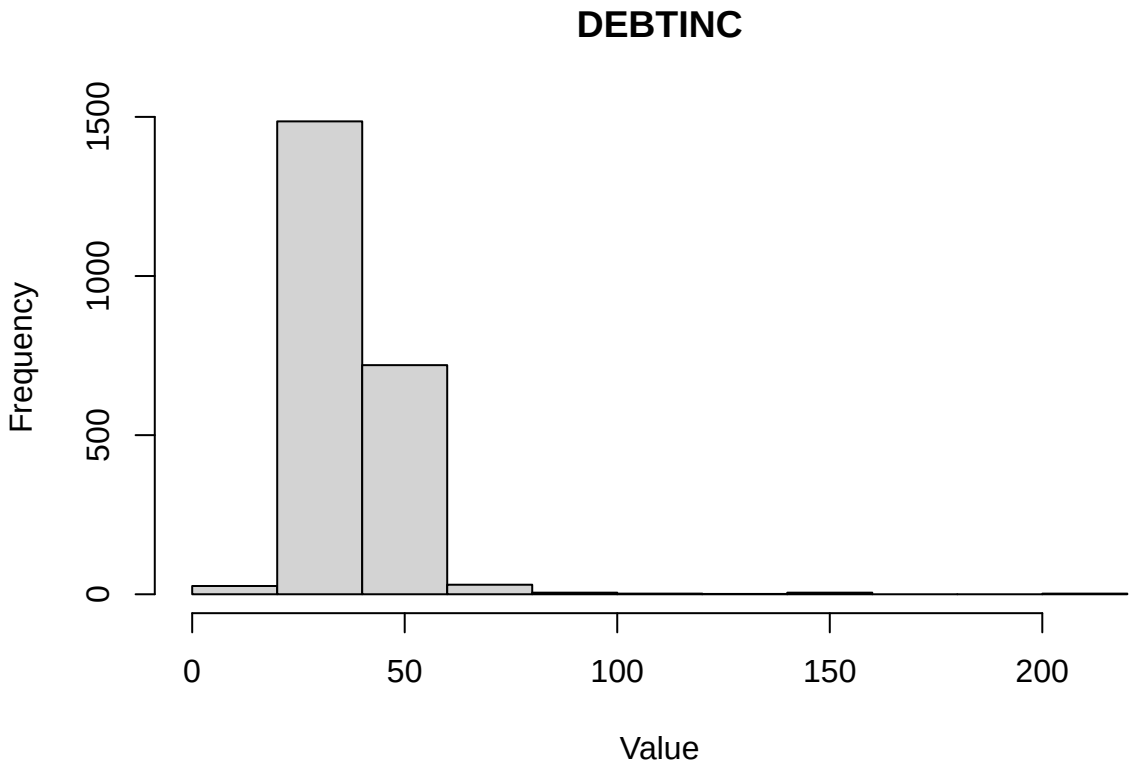

```
## $breaks
##  [1] 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5
##
## $counts
##  [1]    1    3    1    0   10  117  559 1035  525   26
##
## $density
##  [1] 0.0008783487 0.0026350461 0.0008783487 0.0000000000 0.0087834870
##  [6] 0.1027667984 0.4909969258 0.9090909091 0.4611330698 0.0228370663
##
## $mids
##  [1] 1.75 2.25 2.75 3.25 3.75 4.25 4.75 5.25 5.75 6.25
##
## $xname
## [1] "clus1[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```




```
## $breaks
##  [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
##
## $counts
##  [1] 1416  400  231   96   45   24   26   11    5   15    6    1    0    0    0
## [16]    0    1
##
## $density
##  [1] 0.6218708827 0.1756697409 0.1014492754 0.0421607378 0.0197628458
##  [6] 0.0105401845 0.0114185332 0.0048309179 0.0021958718 0.0065876153
## [11] 0.0026350461 0.0004391744 0.0000000000 0.0000000000 0.0000000000
## [16] 0.0000000000 0.0004391744
##
## $mids
##  [1]  0.5  1.5  2.5  3.5  4.5  5.5  6.5  7.5  8.5  9.5 10.5 11.5 12.5 13.5 14.5
## [16] 15.5 16.5
##
## $xname
## [1] "clus1[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

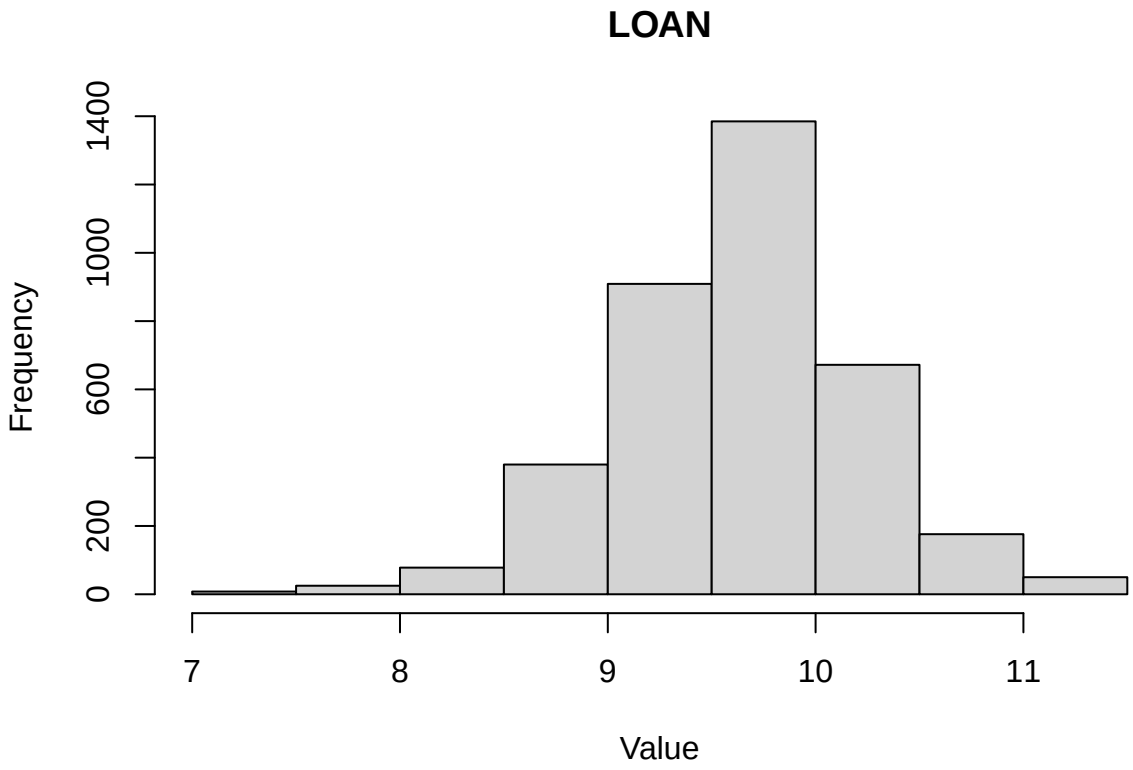



```
## $breaks
##  [1] 10 15 20 25 30 35 40 45 50 55 60 65 70 75
##
## $counts
##  [1]   7  26 638 639 464 208 108 127  38  11   9   0   2
##
## $density
##  [1] 0.0006148441 0.0022837066 0.0560386473 0.0561264822 0.0407553799
##  [6] 0.0182696531 0.0094861660 0.0111550285 0.0033377251 0.0009661836
## [11] 0.0007905138 0.0000000000 0.0001756697
##
## $mids
##  [1] 12.5 17.5 22.5 27.5 32.5 37.5 42.5 47.5 52.5 57.5 62.5 67.5 72.5
##
## $xname
## [1] "clus1[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

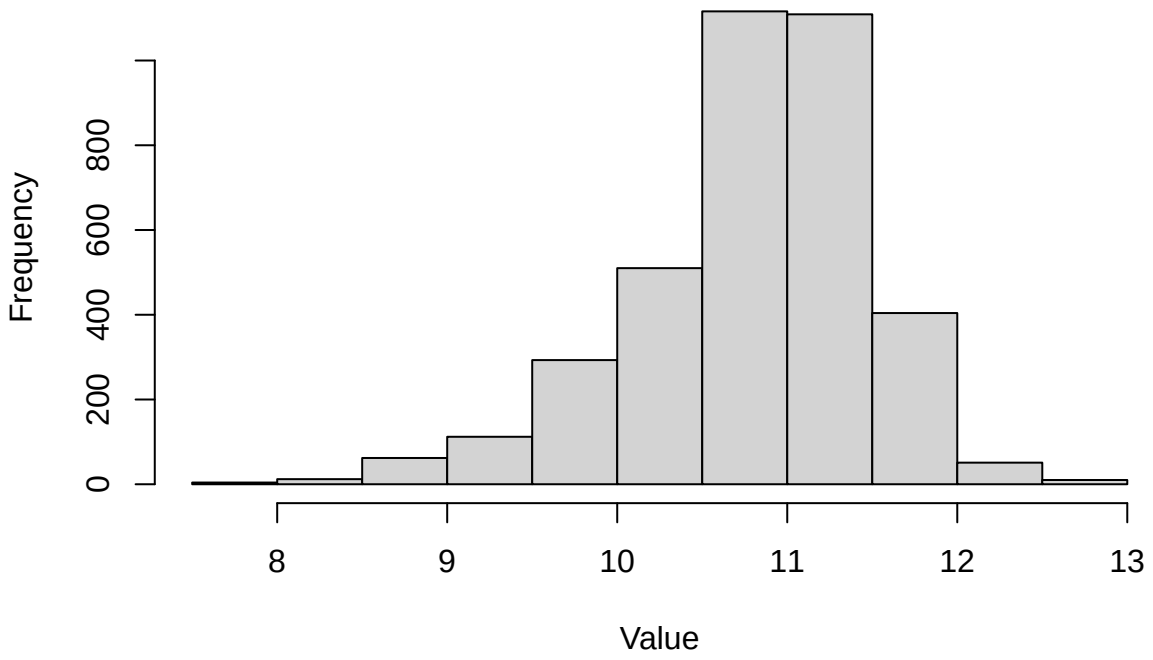



```
## $breaks
##  [1]   0  20  40  60  80 100 120 140 160 180 200 220
##
## $counts
##  [1]   26 1486  720   30    5    2    1    5    0    0    2
##
## $density
##  [1] 5.709267e-04 3.263065e-02 1.581028e-02 6.587615e-04 1.097936e-04
##  [6] 4.391744e-05 2.195872e-05 1.097936e-04 0.000000e+00 0.000000e+00
## [11] 4.391744e-05
##
## $mids
##  [1]  10  30  50  70  90 110 130 150 170 190 210
##
## $xname
## [1] "clus1[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

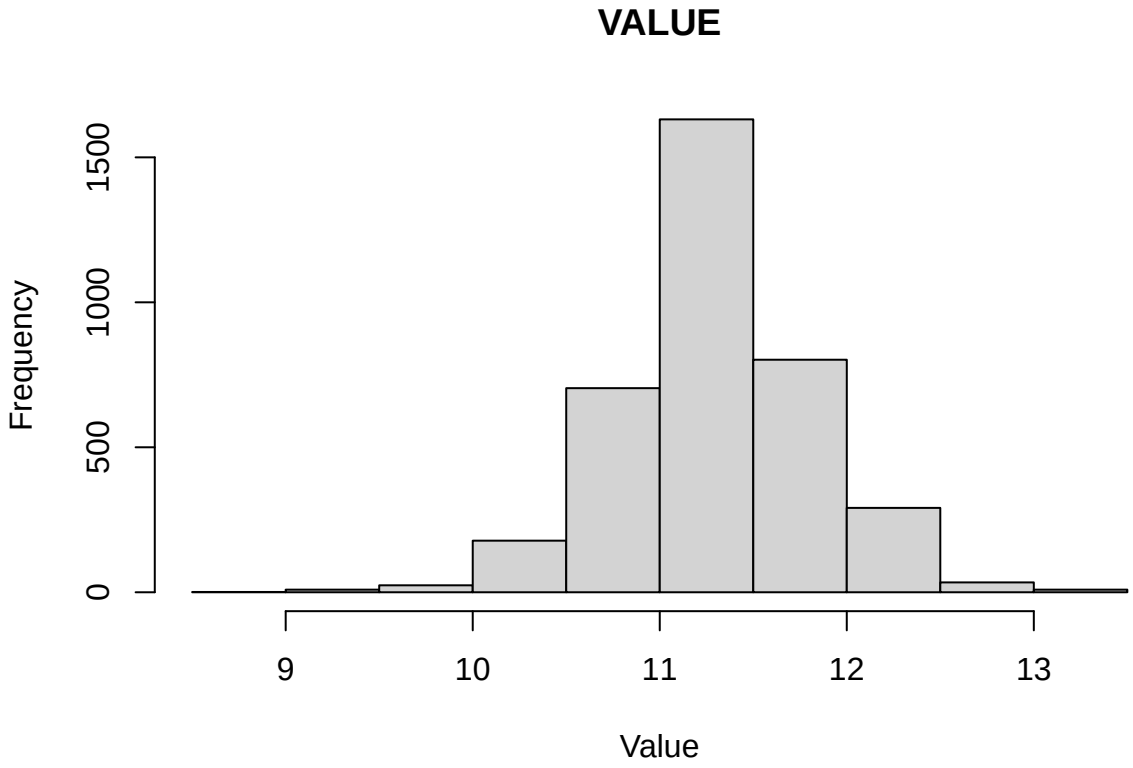
```
for (i in c(2,3,4,8,9,10,11,12,13)) {
  name=colnames(imputed_hmeq)[i]
  print(hist(clus2[,name], main = name, xlab = "Value"))
}
```



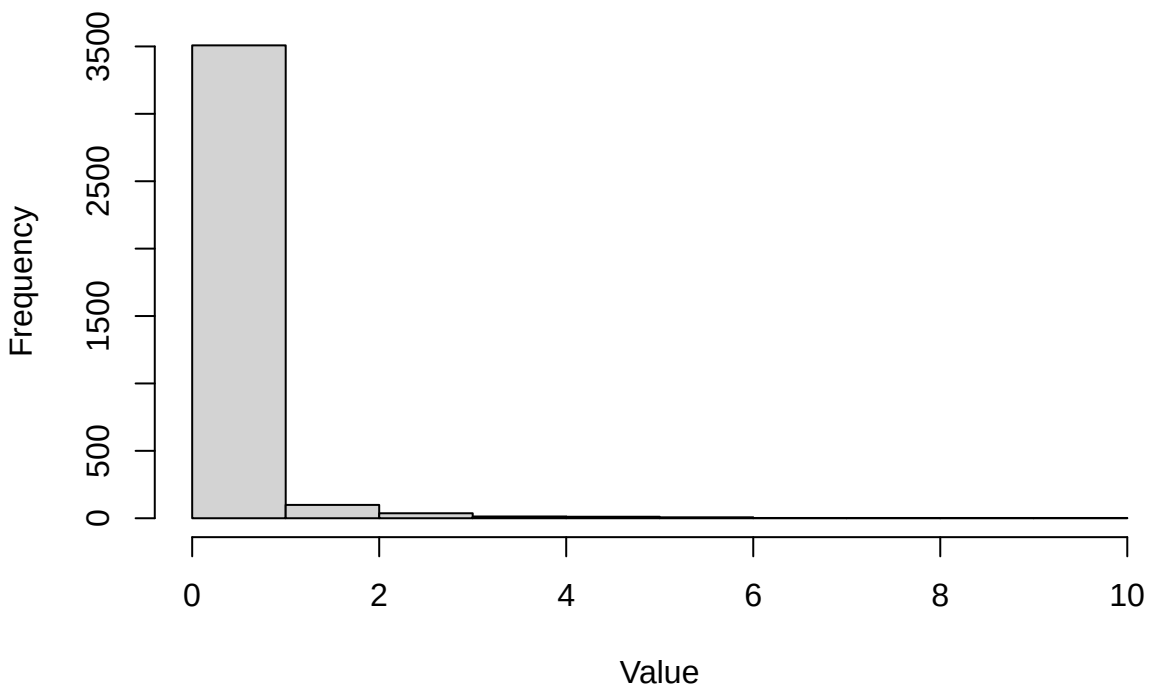

```
## $breaks
##  [1]  7.0  7.5  8.0  8.5  9.0  9.5 10.0 10.5 11.0 11.5
##
## $counts
## [1]    8   25   78  380  909 1385  672  176   50
##
## $density
## [1] 0.004344285 0.013575889 0.042356774 0.206353516 0.493619332 0.752104263
## [7] 0.364919902 0.095574260 0.027151778
##
## $mids
## [1]  7.25  7.75  8.25  8.75  9.25  9.75 10.25 10.75 11.25
##
## $xname
## [1] "clus2[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```



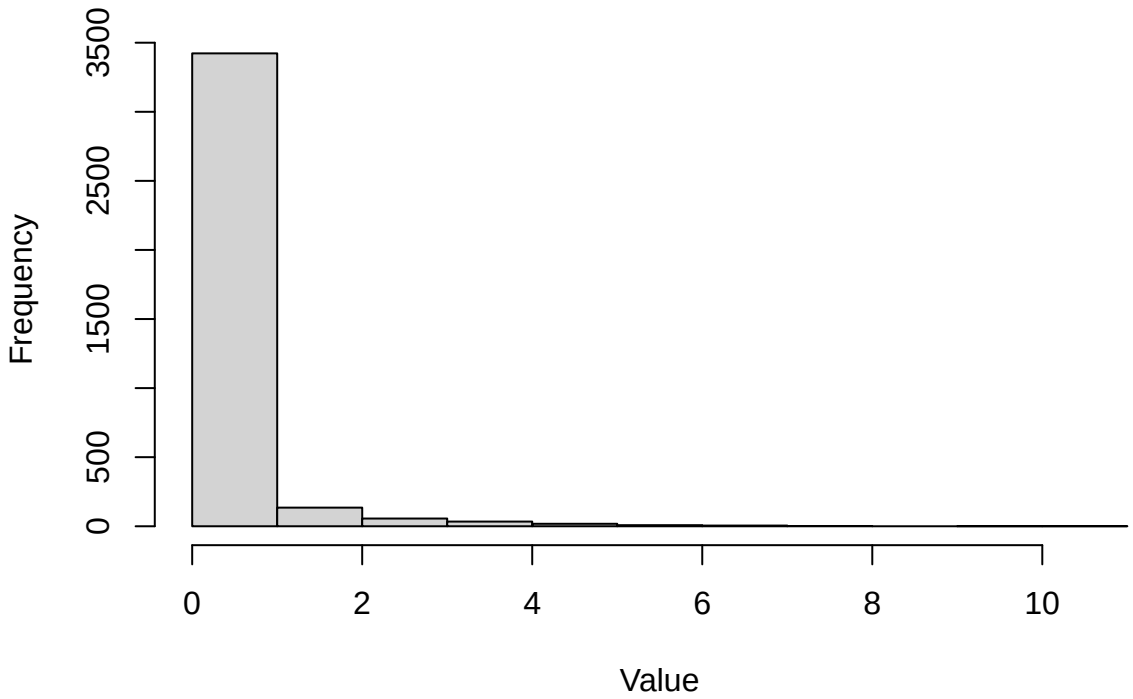

```
## $breaks
##  [1]  7.5  8.0  8.5  9.0  9.5 10.0 10.5 11.0 11.5 12.0 12.5 13.0
##
## $counts
##  [1]    4   12   62  112  293  510 1116 1109  404   51   10
##
## $density
##  [1] 0.002172142 0.006516427 0.033668205 0.060819984 0.159109422 0.276948140
##  [7] 0.606027695 0.602226446 0.219386370 0.027694814 0.005430356
##
## $mids
##  [1]  7.75  8.25  8.75  9.25  9.75 10.25 10.75 11.25 11.75 12.25 12.75
##
## $xname
## [1] "clus2[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```



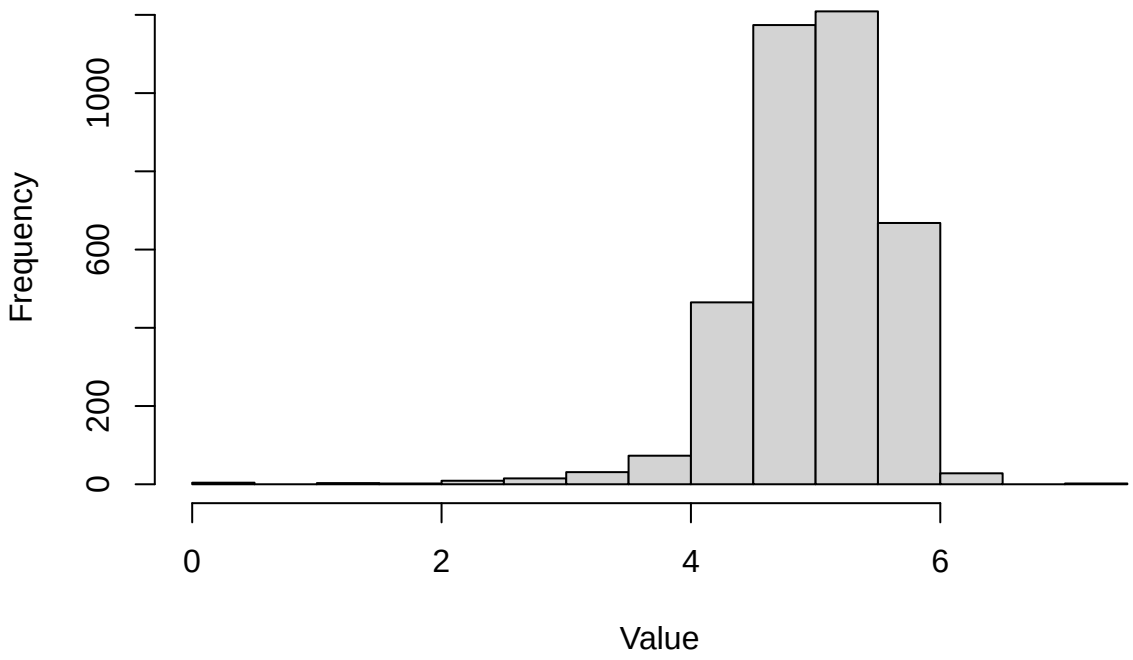

```
## $breaks
##  [1]  8.5  9.0  9.5 10.0 10.5 11.0 11.5 12.0 12.5 13.0 13.5
##
## $counts
##  [1]    1    9   24  178  704 1631  802  291   34    9
##
## $density
##  [1] 0.0005430356 0.0048873201 0.0130328537 0.0966603313 0.3822970405
##  [6] 0.8856910128 0.4355145262 0.1580233505 0.0184632093 0.0048873201
##
## $mids
##  [1]  8.75  9.25  9.75 10.25 10.75 11.25 11.75 12.25 12.75 13.25
##
## $xname
## [1] "clus2[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```



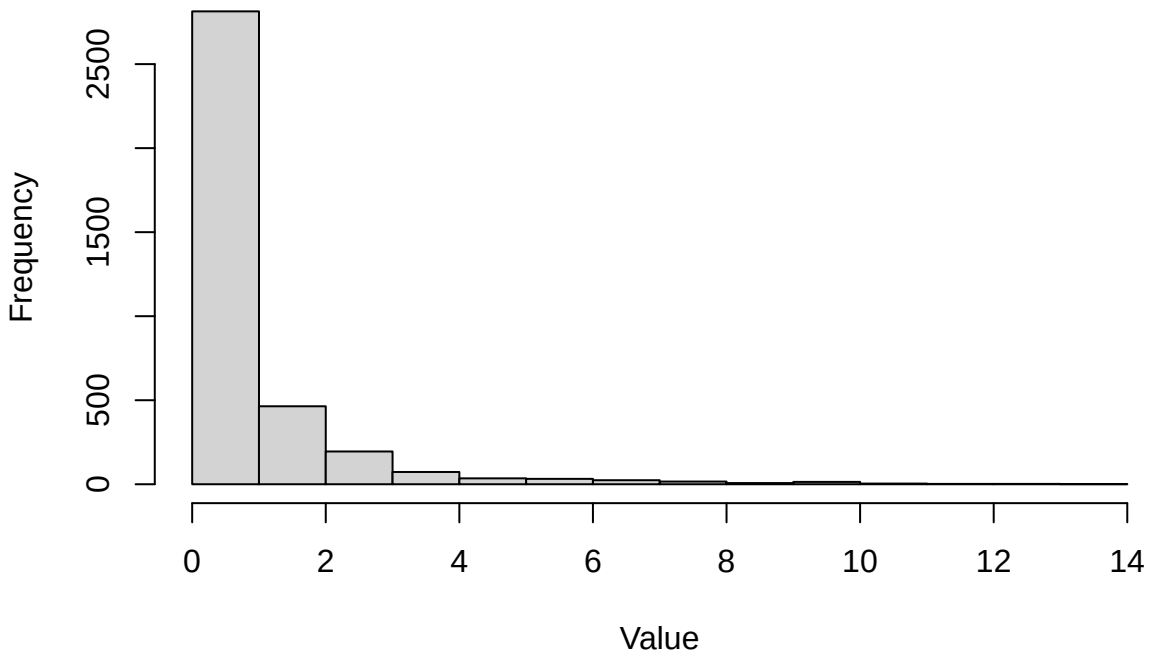

```
## $breaks
##  [1]  0  1  2  3  4  5  6  7  8  9 10
##
## $counts
##  [1] 3508   99   37   13   11    7    2    2    2    2
##
## $density
##  [1] 0.9524843877 0.0268802607 0.0100461580 0.0035297312 0.0029866956
##  [6] 0.0019006245 0.0005430356 0.0005430356 0.0005430356 0.0005430356
##
## $mids
##  [1] 0.5 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5
##
## $xname
## [1] "clus2[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```



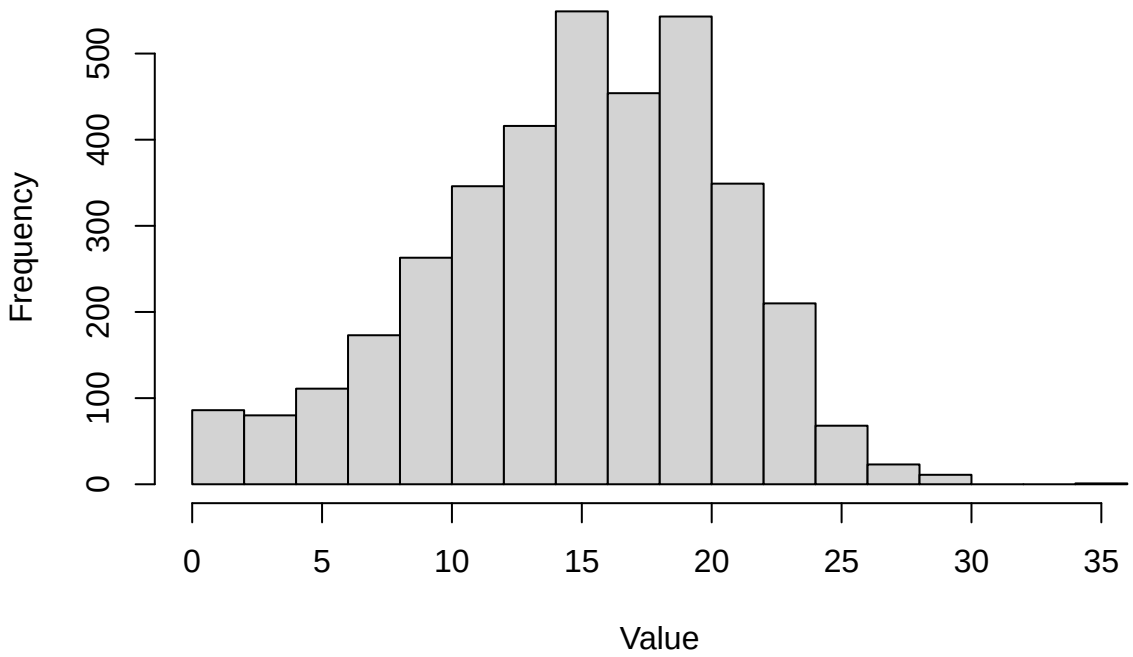

```
## $breaks
##  [1]  0  1  2  3  4  5  6  7  8  9 10 11
##
## $counts
##  [1] 3423  135   56   34   18    8    5    2    0    1    1
##
## $density
##  [1] 0.9294053761 0.0366549009 0.0152049959 0.0092316047 0.0048873201
##  [6] 0.0021721423 0.0013575889 0.0005430356 0.0000000000 0.0002715178
## [11] 0.0002715178
##
## $mids
##  [1]  0.5  1.5  2.5  3.5  4.5  5.5  6.5  7.5  8.5  9.5 10.5
##
## $xname
## [1] "clus2[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```



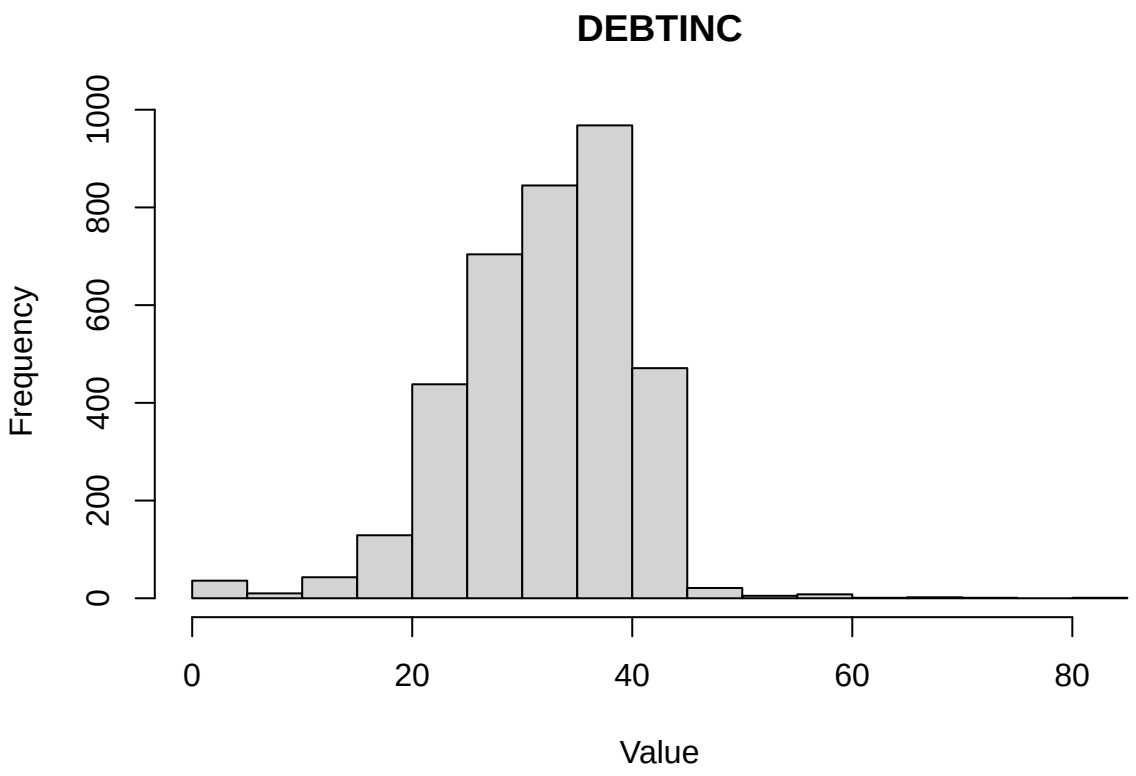

```
## $breaks
##  [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5
##
## $counts
##  [1]    4    0    3    2    9   15   31   73  465 1174 1209  668   28    0    2
##
## $density
##  [1] 0.002172142 0.000000000 0.001629107 0.001086071 0.004887320 0.008145534
##  [7] 0.016834103 0.039641597 0.252511540 0.637523758 0.656530003 0.362747760
## [13] 0.015204996 0.000000000 0.001086071
##
## $mids
##  [1] 0.25 0.75 1.25 1.75 2.25 2.75 3.25 3.75 4.25 4.75 5.25 5.75 6.25 6.75 7.25
##
## $xname
## [1] "clus2[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```




```
## $breaks
##  [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
##
## $counts
##  [1] 2814  464  195   73   35   32   24   16    7   14    4    2    2    1
##
## $density
##  [1] 0.7640510453 0.1259842520 0.0529459680 0.0198207983 0.0095031225
##  [6] 0.0086885691 0.0065164268 0.0043442846 0.0019006245 0.0038012490
## [11] 0.0010860711 0.0005430356 0.0005430356 0.0002715178
##
## $mids
##  [1]  0.5  1.5  2.5  3.5  4.5  5.5  6.5  7.5  8.5  9.5 10.5 11.5 12.5 13.5
##
## $xname
## [1] "clus2[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```




```
## $breaks
##  [1]  0  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36
##
## $counts
##  [1]  86  80 111 173 263 346 416 549 454 543 349 210  68  23  11   0   0   1
##
## $density
##  [1] 0.0116752647 0.0108607114 0.0150692370 0.0234862884 0.0357045887
##  [6] 0.0469725767 0.0564756992 0.0745316318 0.0616345371 0.0737170785
## [11] 0.0473798534 0.0285093674 0.0092316047 0.0031224545 0.0014933478
## [16] 0.0000000000 0.0000000000 0.0001357589
##
## $mids
##  [1]  1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35
##
## $xname
## [1] "clus2[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```



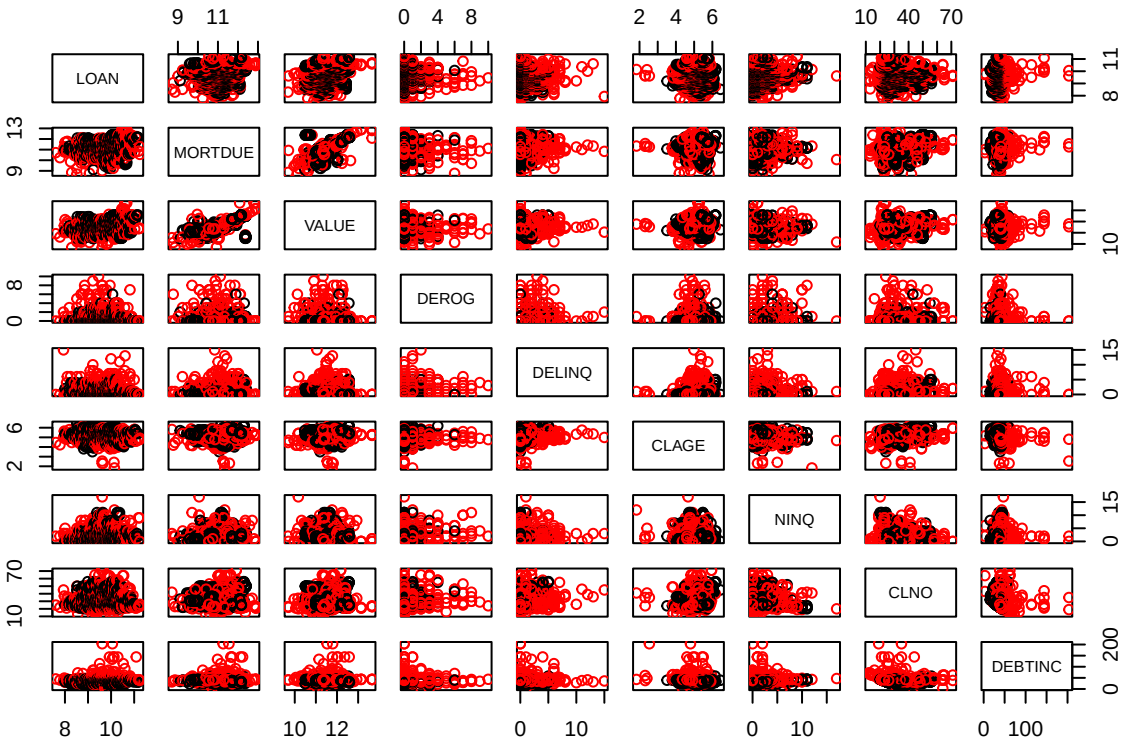

```
## $breaks
##  [1]  0  5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85
##
## $counts
##  [1]  36  10  43 129 438 704 845 968 471  21   5   8   1   2   1   0   1
##
## $density
##  [1] 1.954928e-03 5.430356e-04 2.335053e-03 7.005159e-03 2.378496e-02
##  [6] 3.822970e-02 4.588651e-02 5.256584e-02 2.557698e-02 1.140375e-03
## [11] 2.715178e-04 4.344285e-04 5.430356e-05 1.086071e-04 5.430356e-05
## [16] 0.000000e+00 5.430356e-05
##
## $mids
##  [1]  2.5  7.5 12.5 17.5 22.5 27.5 32.5 37.5 42.5 47.5 52.5 57.5 62.5 67.5 72.5
## [16] 77.5 82.5
##
## $xname
## [1] "clus2[, name]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
for (i in c(2,3,4,8,9,10,11,12,13)) {
  name=colnames(imputed_hmeq)[i]
  w=wilcox.test(clus1[,name], clus2[,name])
  print(name)
  print(w$p.value)
```

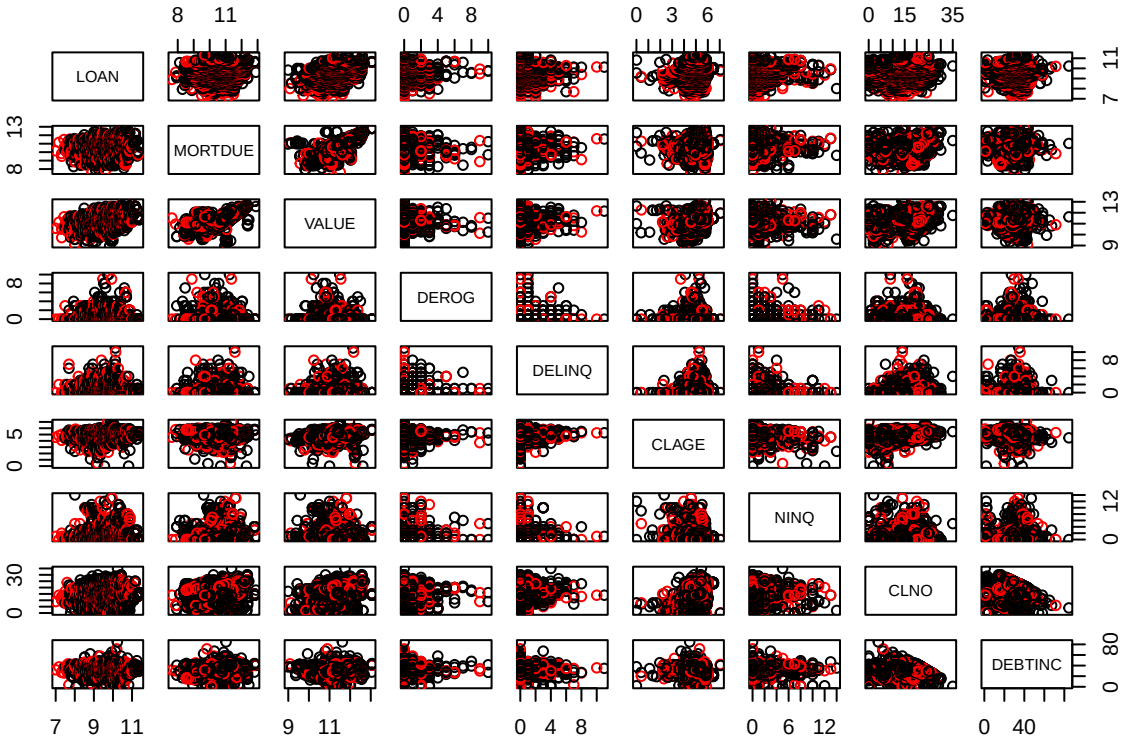
```
}
```

```
## [1] "LOAN"
## [1] 2.932464e-16
## [1] "MORTDUE"
## [1] 1.28775e-137
## [1] "VALUE"
## [1] 1.404887e-104
## [1] "DEROG"
## [1] 3.10203e-07
## [1] "DELINQ"
## [1] 2.785019e-32
## [1] "CLAGE"
## [1] 4.221841e-35
## [1] "NINQ"
## [1] 3.749283e-38
## [1] "CLNO"
## [1] 0
## [1] "DEBTINC"
## [1] 7.284695e-152
```

```
pairs(clus1[,c(2,3,4,8,9,10,11,12,13)], col=ifelse(clus1$BAD==0, "black", "red"))
```




```
pairs(clus2[,c(2,3,4,8,9,10,11,12,13)], col=ifelse(clus1$BAD==0, "black", "red"))
```



I chose to consider the clustering result involving K-means because the two clusters identified were separated well by t-SNE (note the horizontal line that can separate the blue and red clouds). This leads me to believe that both t-SNE and K-means found similar patterns in the data and therefore K-means may have identified important patterns.

In choosing this method I want to try to understand each cluster and explore whether there is a link between these and the "BAD" variable. Here, I note three interesting findings that address these two aspects of the project.

1. The first finding that I would like to convey is that there does not seem to be a major difference between the number and/or percentage of BAD observations between the two clusters. The contingency tables show 23% of cluster 1 variables were BAD compared to 17% of cluster 2 variables.

2. My second observation is that the variables coming from cluster 1 and cluster 2 are distributed differently. We can confirm this by eyeballing the histograms and also noting the Wilcoxon test p-values. This finding is especially confusing because from the previous item, the variables cannot discriminate well between BAD and non-BAD observations, in spite of being distributed differently.

3. Lastly, I chose to include scatter plot matrices for each cluster to investigate whether pairs of variables could separate the data. It is reassuring that no pairs of variables could separate the data–otherwise, a clustering analysis would have seemed superfluous.

Taken together these findings suggest that maybe there is some inherent structure in the data that is not well-described by two labels. Indeed, maybe the data can tell us more than just whether the borrowers defaulted on their loans or not.