PROJECT 3 STAT 5474

Prince Appiah

10/15/2021

PART II

(i) Read the data into R. List the missing rate (in percentage) for each variable.

```r
dat <- read.csv("HMEQ.csv")
dim(dat)
```

```
## [1] 5960   13
```

Comment

The dimesion is 5960 rows and 13 variables

```r
miss.perc <- function(dat, filename=NULL){
  vnames <- colnames(dat); vnames
  n <- nrow(dat)
  out <- NULL
  for (j in 1: ncol(dat)){
    vname <- colnames(dat)[j]
    x <- as.vector(dat[,j])
    n1 <- sum(is.na(x), na.rm=T)
    n2 <- sum(x=="NA", na.rm=T)
    n3 <- sum(x=="", na.rm=T)
    nmiss <- n1 + n2 + n3
    ncomplete <- n-nmiss
    out <- rbind(out, c(col.number=j, vname=vname,
                        mode=mode(x), n.levels=length(unique(x)),
                        ncomplete=ncomplete, miss.perc=nmiss/n))
  }
  out <- as.data.frame(out)
  row.names(out) <- NULL
  if (!is.null(filename)) write.csv(out, file = filename, row.names=F)
  return(out)
}
miss.perc(dat)
```

```
##    col.number   vname      mode n.levels ncomplete          miss.perc
## 1           1     BAD   numeric        2      5960                  0
## 2           2    LOAN   numeric      540      5960                  0
## 3           3 MORTDUE   numeric     5054      5442 0.0869127516778524
## 4           4   VALUE   numeric     5382      5848 0.0187919463087248
## 5           5  REASON character        3      5708 0.0422818791946309
## 6           6     JOB character        7      5681 0.0468120805369127
## 7           7     YOJ   numeric      100      5445 0.0864093959731544
## 8           8   DEROG   numeric       12      5252  0.118791946308725
## 9           9  DELINQ   numeric       15      5380 0.0973154362416107
## 10         10   CLAGE   numeric     5315      5652 0.0516778523489933
## 11         11    NINQ   numeric       17      5450 0.0855704697986577
## 12         12    CLNO   numeric       63      5738  0.037248322147651
## 13         13 DEBTINC   numeric     4694      4693   0.21258389261745
```

Comment

From the output we have DEBTINC has the highest percentage of missing values that is 21.258%. The variables "BAD" and "lOAN" have no missing values.

(ii)

(a) Replace missing values for both JOB and REASON with default constant "Unknown". Output the frequency table after the replacement

```r
dat$JOB[dat$JOB==""] <- "Unknown"
dat$REASON[dat$REASON==""] <- "Unknown"

table(dat$JOB)
```

```
##
##     Mgr  Office   Other ProfExe   Sales    Self Unknown
##     767     948    2388    1276     109     193     279
```

```r
table(dat$REASON)
```

```
##
## DebtCon HomeImp Unknown
##    3928    1780     252
```

Comment

Mgr Office Other ProfExe Sales Self Unknown

767 948 2388 1276 109 193 279

DebtCon HomeImp Unknown

3928 1780 252

From the above tables, we see that the missing values for the variables

"JOB" and "REASON" have been replaced with unknown. We replaced 279 missing ## values with unknow for the variable "JOB" and 252 for the variable

"REASON".

(b)

Perform the (natural) logarithm transformation on the following

variables: LOAN, VALUE, MORTDUE, YOJ, and CLAGE. If a variable has

value 0, then try log(x+1) for the transformation.

```
log.transf <- function(x)
  {
    a <- x
      if(sum(a==0, na.rm=TRUE) > 1)
        {
          x <- log(a+1)
        }
      else
        {
          x <- log(a)
        }
    return(x)
}
```

```
dat$LOAN <- log.transf(dat$LOAN)
head(dat$LOAN, 10)
```

```
##  [1] 7.003065 7.170120 7.313220 7.313220 7.438384 7.438384 7.495542 7.495542
##  [9] 7.600902 7.600902
```

```
dat$VALUE <- log.transf(dat$VALUE)
head(dat$VALUE, 10)
```

```
##  [1] 10.571958 11.133128  9.723164        NA 11.626254 10.604603 10.951455
##  [8] 10.669746 10.752356 11.038914
```

```r
dat$MORTDUE <- log.transf(dat$MORTDUE)
head(dat$MORTDUE, 10)
```

```
##  [1] 10.160453 11.157007  9.510445        NA 11.490680 10.327054 10.792387
##  [8] 10.257730 10.395130        NA
```

```r
dat$YOJ <- log.transf(dat$YOJ)
head(dat$YOJ, 10)
```

```
##  [1] 2.442347 2.079442 1.609438       NA 1.386294 2.302585 1.791759 2.484907
##  [9] 1.386294 2.833213
```

```r
dat$CLAGE <- log.transf(dat$CLAGE)
head(dat$CLAGE, 10)
```

```
##  [1] 4.557729 4.810828 5.013742       NA 4.546835 4.629531 4.357990 4.497207
##  [9] 5.384189 4.760463
```

(c)

Impute all the remaining values with an appropriate imputation procedure of your choice

```r
library(mice)
```

```
## Registered S3 methods overwritten by 'tibble':
##   method     from
##   format.tbl pillar
##   print.tbl  pillar

##
## Attaching package: 'mice'

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

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

```r
imputed.dat <- mice(dat,m=1, maxit = 50, method = 'pmm', seed = 500)
```

```
##
##  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
##   6   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   7   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   8   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   9   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   10   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   11   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   12   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   13   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   14   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   15   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   16   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   17   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   18   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   19   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   20   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   21   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   22   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   23   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   24   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   25   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   26   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   27   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   28   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   29   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   30   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   31   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   32   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   33   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   34   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   35   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   36   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   37   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   38   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   39   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   40   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   41   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   42   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   43   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   44   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   45   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   46   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   47   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   48   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   49   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
##   50   1  MORTDUE  VALUE  YOJ  DEROG  DELINQ  CLAGE  NINQ  CLNO  DEBTINC
```

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

```r
summary(imputed.dat)
```

```
## Class: mids
## Number of multiple imputations:  1
## Imputation methods:
##     BAD    LOAN MORTDUE   VALUE  REASON     JOB     YOJ   DEROG  DELINQ   CLAGE
##      ""      ""   "pmm"   "pmm"      ""      ""   "pmm"   "pmm"   "pmm"   "pmm"
##    NINQ    CLNO DEBTINC
##   "pmm"   "pmm"   "pmm"
## PredictorMatrix:
##         BAD LOAN MORTDUE VALUE REASON JOB YOJ DEROG DELINQ CLAGE NINQ CLNO
## BAD       0    1       1     1      0   0   1     1      1     1    1    1
## LOAN      1    0       1     1      0   0   1     1      1     1    1    1
## MORTDUE   1    1       0     1      0   0   1     1      1     1    1    1
## VALUE     1    1       1     0      0   0   1     1      1     1    1    1
## REASON    1    1       1     1      0   0   1     1      1     1    1    1
## JOB       1    1       1     1      0   0   1     1      1     1    1    1
##         DEBTINC
## BAD           1
## LOAN          1
## MORTDUE       1
## VALUE         1
## REASON        1
## JOB           1
## Number of logged events:  2
##   it im dep     meth    out
## 1  0  0     constant REASON
## 2  0  0     constant    JOB
```

```r
dat.complete <- complete(imputed.dat, 1)
dat <- as.data.frame(dat.complete)
```

(iii) Distance Matrix

```r
dat0 <- model.matrix(~.-1, data = dat.complete)
dim(dat0)
```

```
## [1] 5960   20
```

```r
dat.1 <- as.data.frame(dat0)
```

Removing BAD because it's our outcome

```r
dat.1 <- dat.1[, -1]
names(dat.1)
```

```
##  [1] "LOAN"          "MORTDUE"       "VALUE"         "REASONDebtCon"
```

```
##  [5] "REASONHomeImp" "REASONUnknown" "JOBOffice"     "JOBOther"
##  [9] "JOBProfExe"    "JOBSales"      "JOBSelf"       "JOBUnknown"
## [13] "YOJ"           "DEROG"         "DELINQ"        "CLAGE"
## [17] "NINQ"          "CLNO"          "DEBTINC"
```

```
library(cluster)
library(gower)
Dist. <- daisy(dat.1, metric = "gower")
```

```
## Warning in daisy(dat.1, metric = "gower"): binary variable(s) 4, 5, 6, 7, 8, 9,
## 10, 11, 12 treated as interval scaled
```

```
summary(Dist.)
```

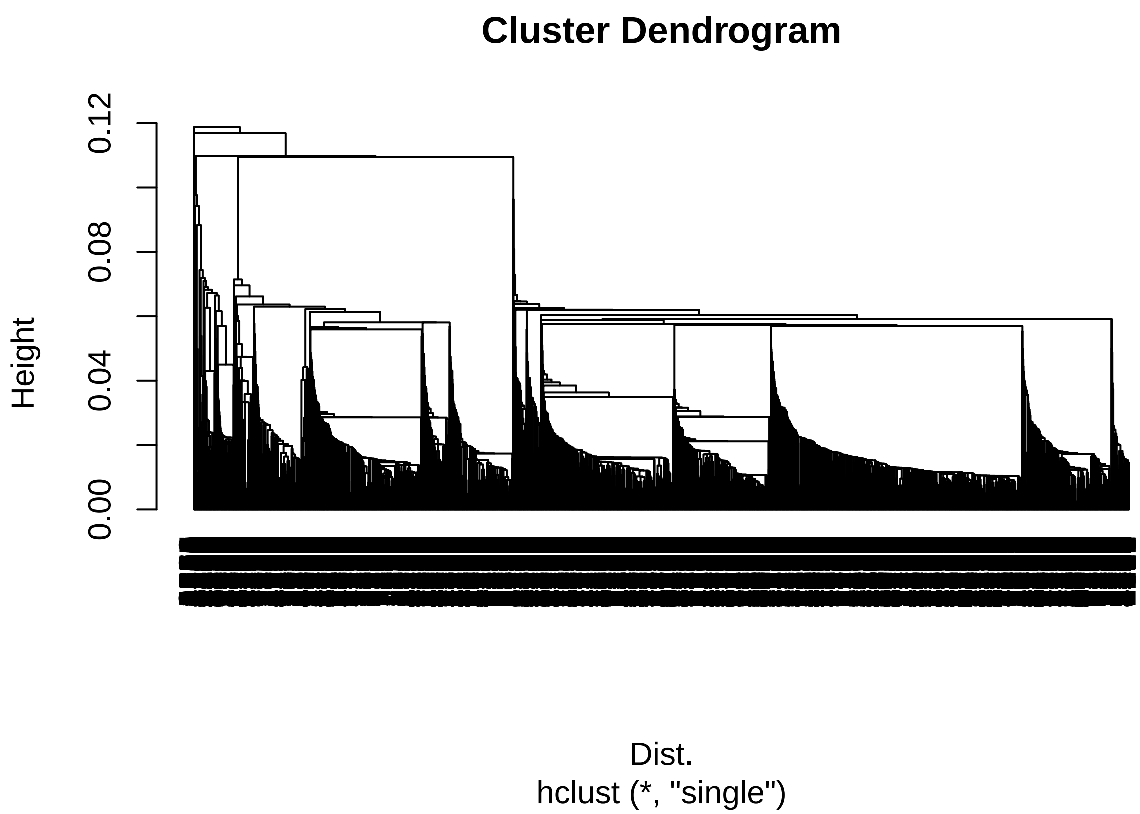
```
## 17757820 dissimilarities, summarized :
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0002954 0.1315000 0.1708100 0.1778700 0.2454400 0.4550600
## Metric :  mixed ;  Types = I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I
## Number of objects : 5960
```

(iv) CLUSTER ANALYSIS

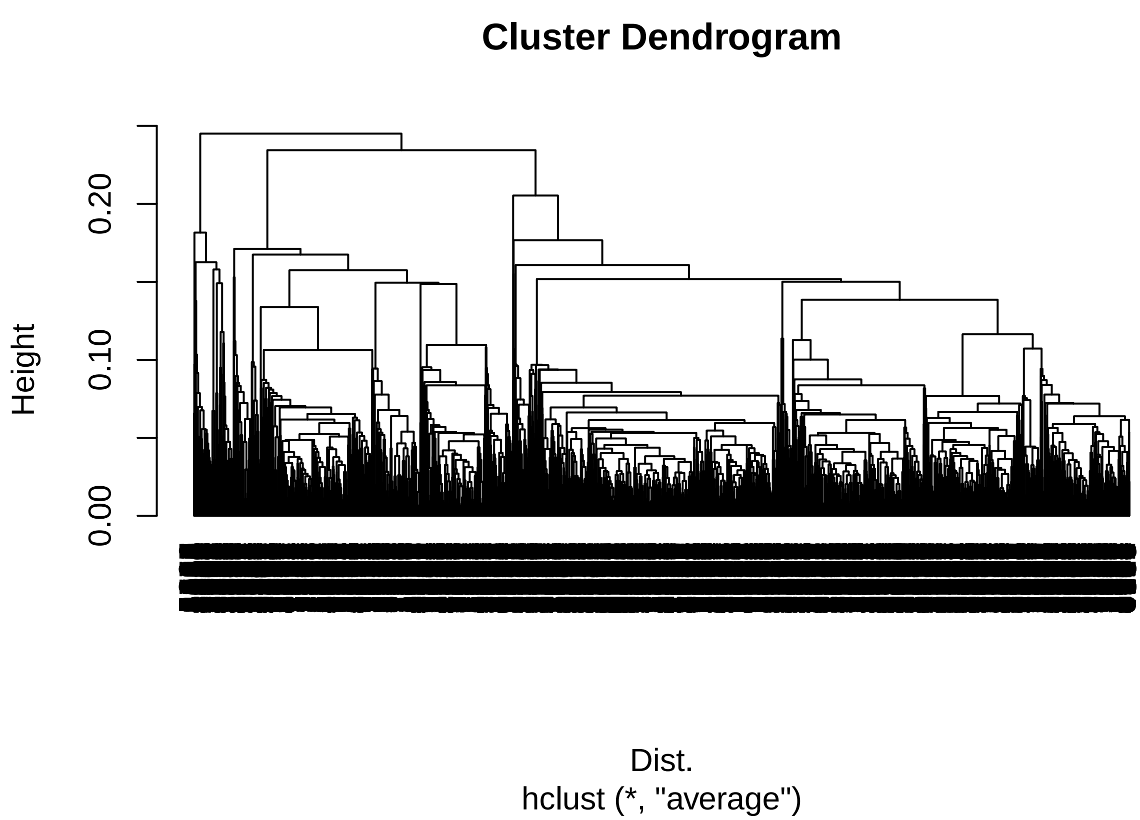
METHOD 1 : Hierarchial Clustering

First, we determine which of the hierarchial method to use.

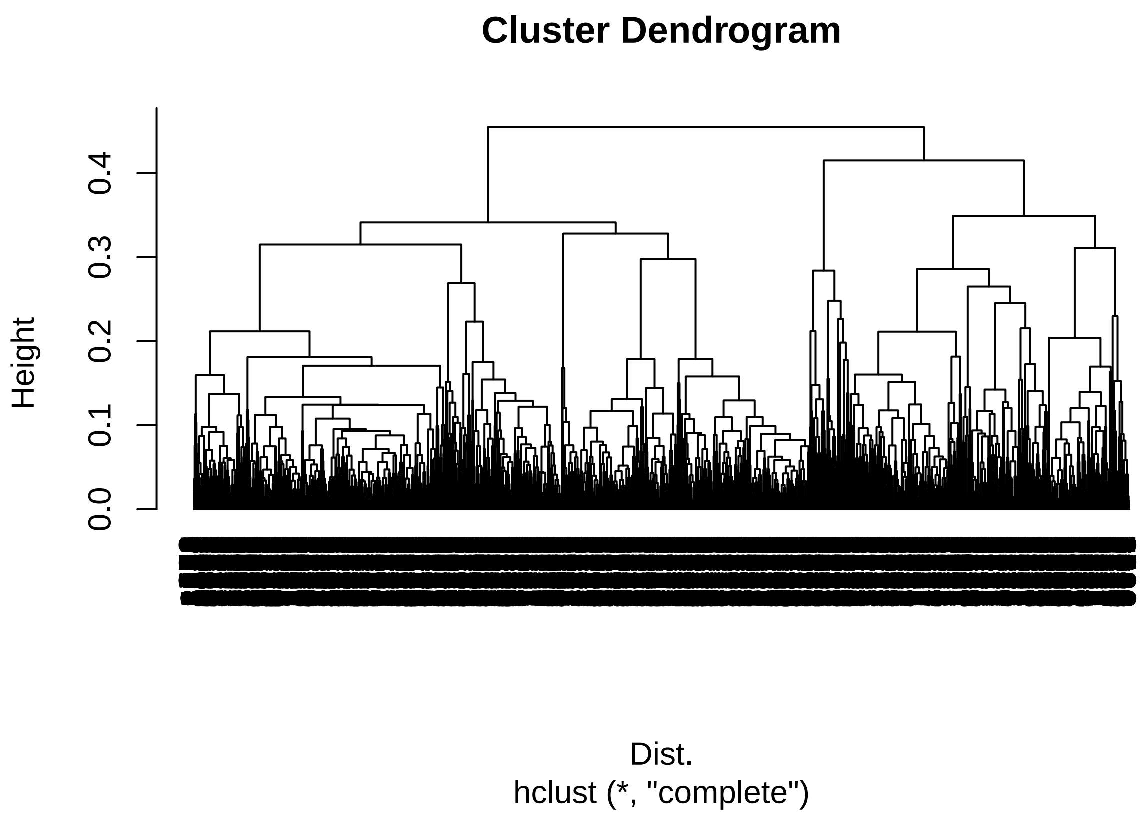
```
library(cluster)
fit.single <- hclust(Dist., method="single")
fit.average <- hclust(Dist., method="average")
fit.complete <- hclust(Dist., method="complete")
plot(fit.single, hang = -0.5)
```




```
plot(fit.average, hang = -0.5)
```



```r
plot(fit.complete, hang = -0.5)
```

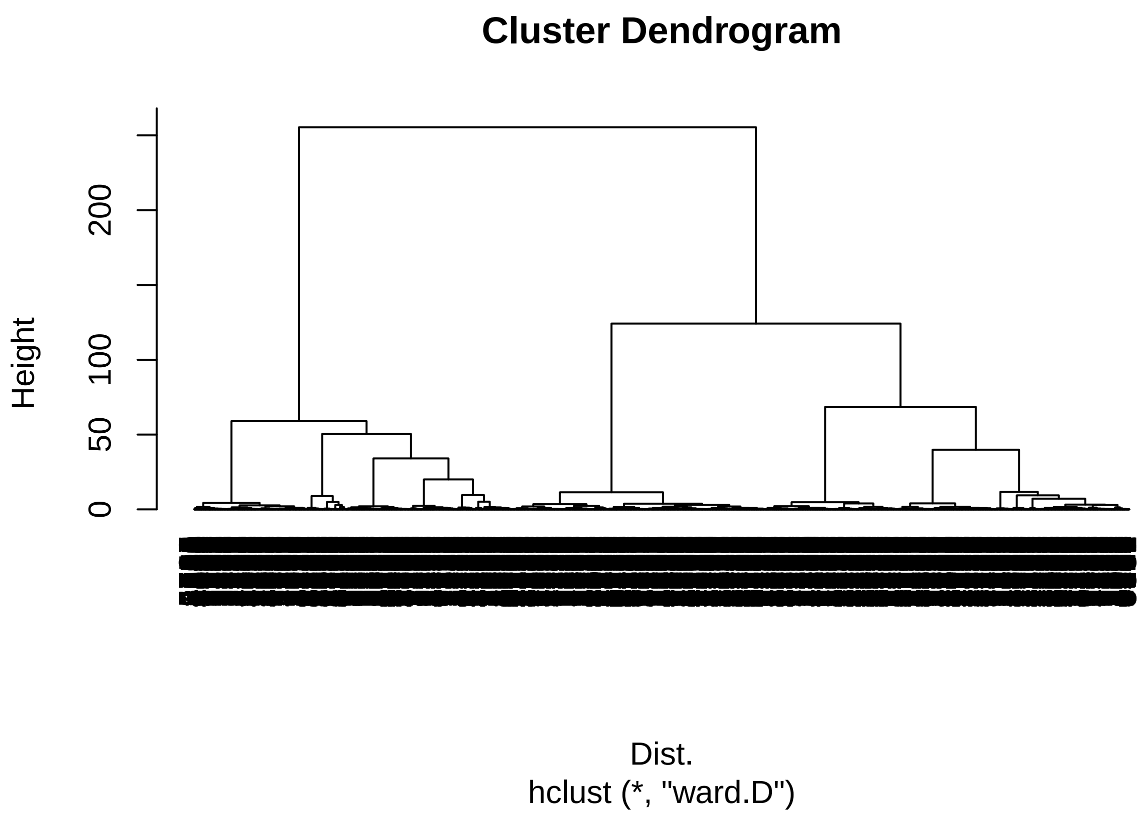


Comment

From the three cluster Dendrogram, we observe that the ward method gives a more clearer dendogram that the number of clusters can clearly be determined from it.

Using the ward method

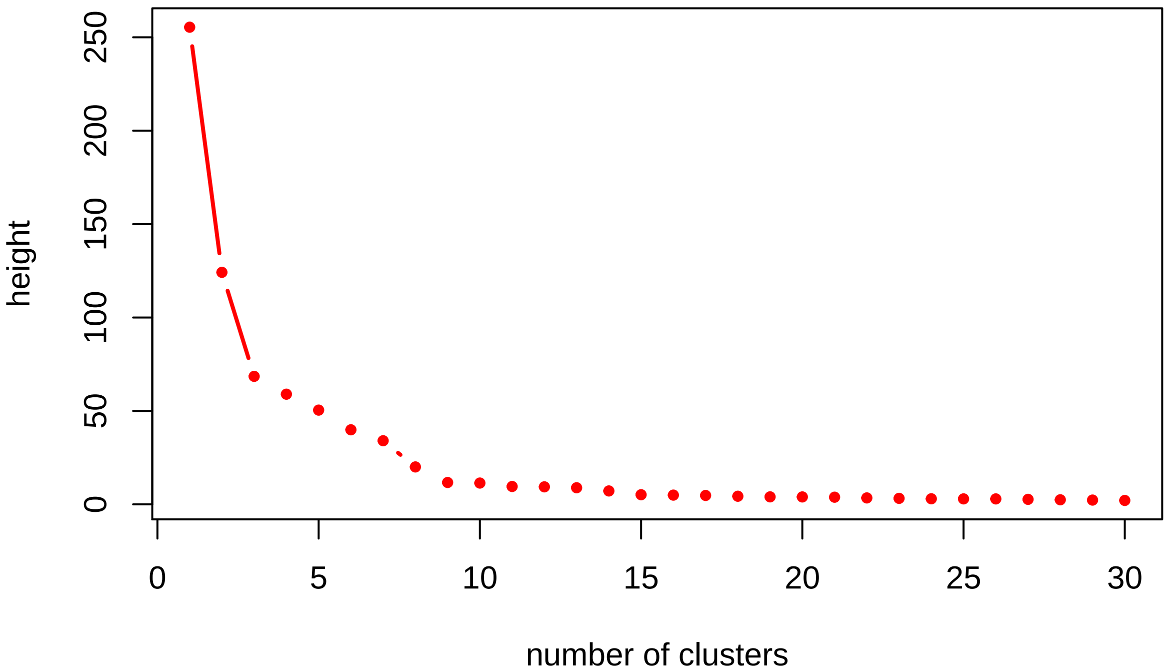
```r
fit.ward.d2 <- hclust(Dist., method="ward.D")
plot(fit.ward.d2, hang = -0.5)
```



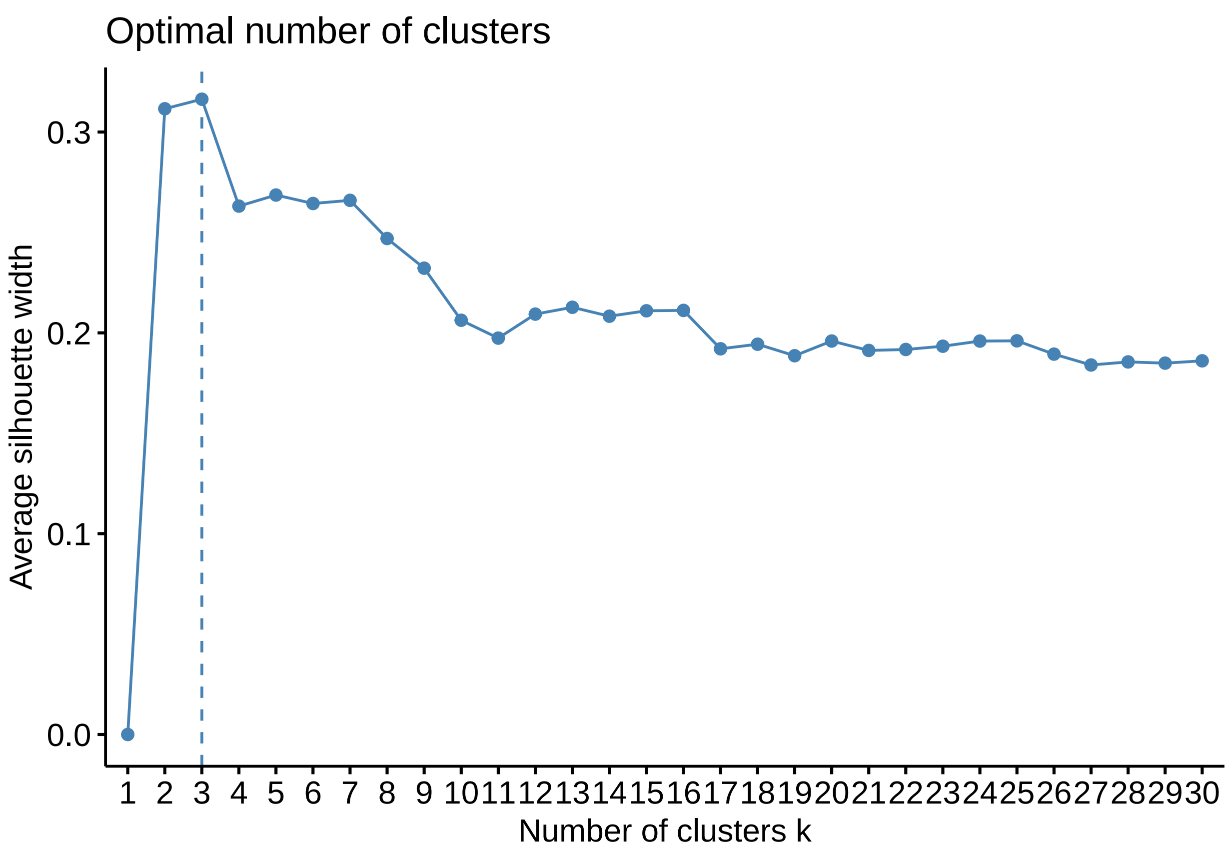
Determine Optimal Cluster

Screeplot of height in hierarchial clustering

```
K.max <- 30
height <- tail(fit.ward.d2$height, n=K.max)
n.cluster <- tail((nrow(dat.1)-1):1, n=K.max)
plot(n.cluster, height, type="b", pch=19, cex=.5, xlab="number of clusters",
  ylab="height", col="red", lwd=2)
```




```r
suppressMessages(library(factoextra))
fviz_nbclust(x=dat.1, FUNcluster = hcut, method = c("silhouette"), diss = NULL, k.max = 30)
```



CLUSTER MEMBERSHIPS

```
hclust.groups <- cutree(fit.ward.d2, k=2)
table(hclust.groups)
```

```
## hclust.groups
##    1    2
## 2032 3928
```

Comment

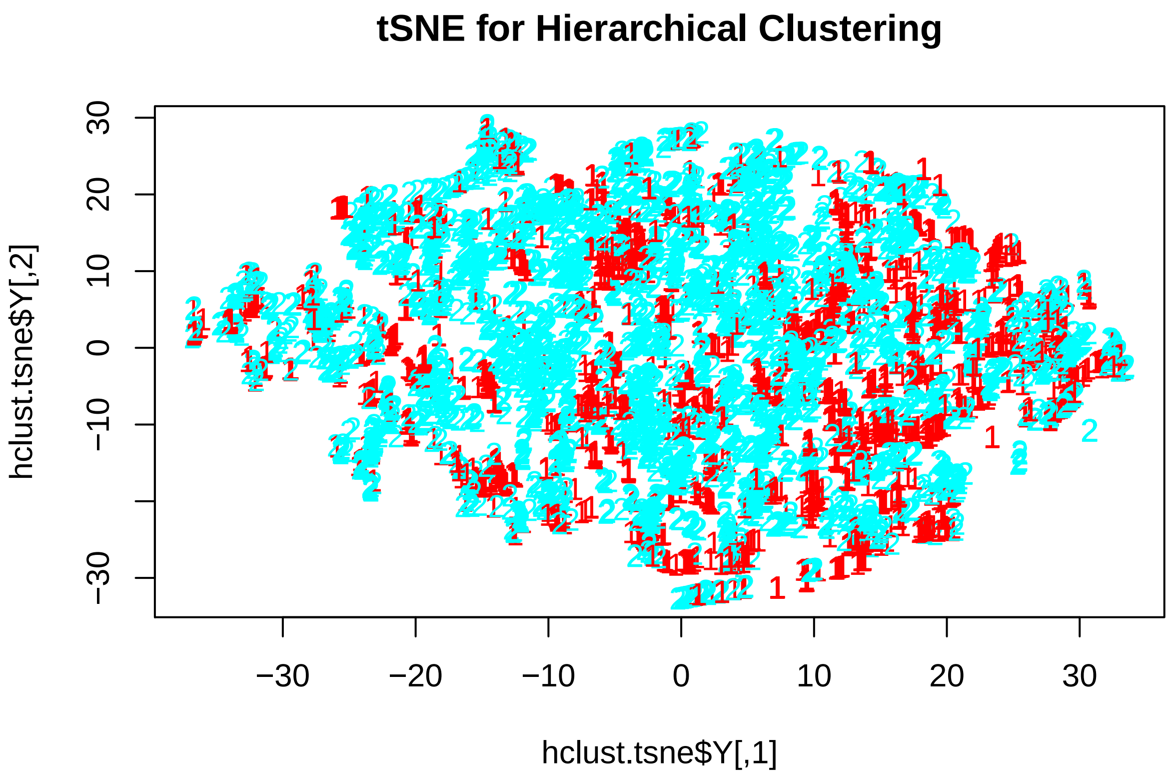
hclust.groups

1 2

2032 3928

Plotting the data dat.1

```
library(Rtsne)
colors = rainbow(length(unique(hclust.groups)))
names(colors) = unique(hclust.groups)
hclust.tsne <- Rtsne(dat.1, dims=2, perplexity=30, max_iter=500)
plot(hclust.tsne$Y, t="n", main = "tSNE for Hierarchical Clustering")
text(hclust.tsne$Y, labels = hclust.groups, col = colors[hclust.groups])
```

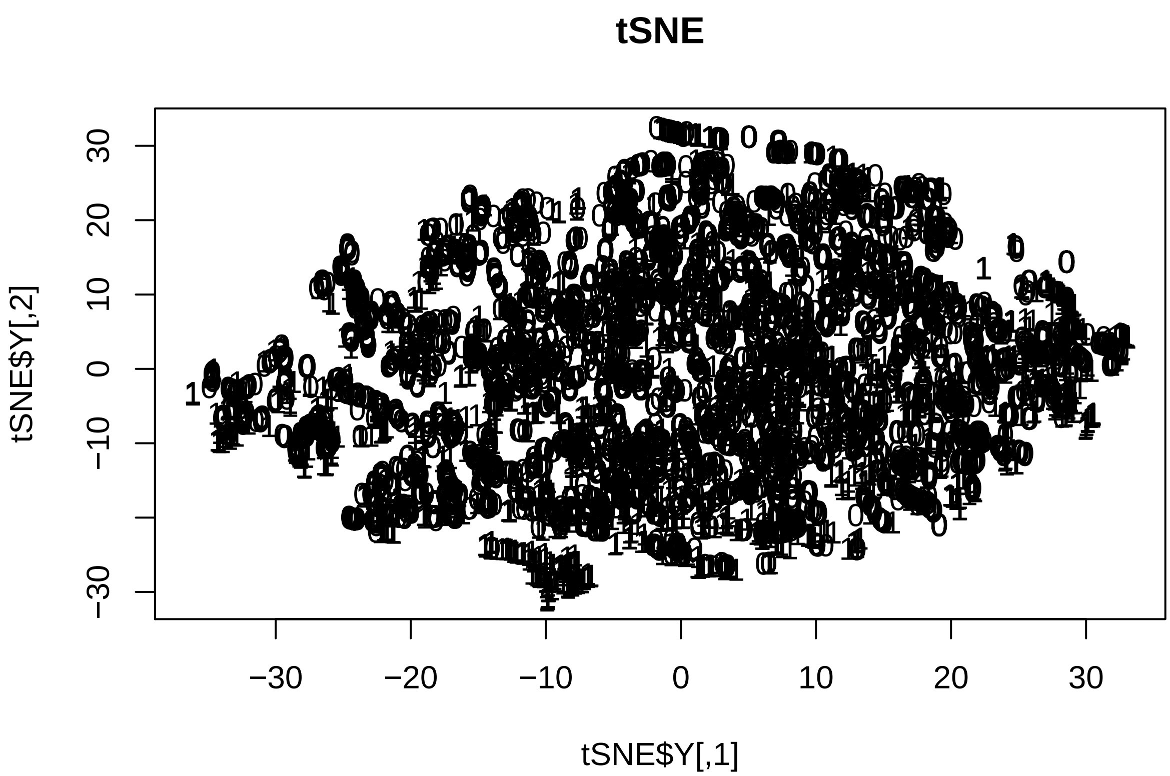


Plot the data using an MDS procedure (e.g., PCA or tSNE). Highlight both the ## clustering memberships and the BAD value with different color and symbols.

```
BAD <- dat$BAD
BAD <- as.vector(dat$BAD)
dat.2 <- (cbind(dat0,BAD))
dat.2 <- data.frame(dat.2)
library(Rtsne)
tSNE <- Rtsne(dat0, dims=2, perplexity=30, verbose=TRUE, max_iter = 500)
```

```
## Performing PCA
## Read the 5960 x 20 data matrix successfully!
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
## Done in 1.41 seconds (sparsity = 0.020005)!
## Learning embedding...
## Iteration 50: error is 91.834635 (50 iterations in 2.17 seconds)
## Iteration 100: error is 77.117264 (50 iterations in 1.58 seconds)
## Iteration 150: error is 73.217206 (50 iterations in 1.31 seconds)
## Iteration 200: error is 72.635922 (50 iterations in 1.31 seconds)
## Iteration 250: error is 72.488880 (50 iterations in 1.37 seconds)
## Iteration 300: error is 2.232922 (50 iterations in 1.58 seconds)
## Iteration 350: error is 1.798011 (50 iterations in 1.51 seconds)
## Iteration 400: error is 1.564351 (50 iterations in 1.53 seconds)
## Iteration 450: error is 1.416048 (50 iterations in 1.44 seconds)
## Iteration 500: error is 1.314290 (50 iterations in 1.47 seconds)
## Fitting performed in 15.27 seconds.
```

```
plot(tSNE$Y, t='n', main="tSNE")
text(tSNE$Y, labels=dat.2$BAD, col=fit.ward.d2$cluster)
```



METHOD 2: K-Means Cluster Analysis

```
library(cluster)
library(Rtsne)
K <- 2
```

```
fit.kmeans <- kmeans(dat0, K)
```

cluster memberships

```
kmeans.groups <- fit.kmeans$cluster
table(kmeans.groups)
```

```
## kmeans.groups
##    1    2
## 3674 2286
```

Comment

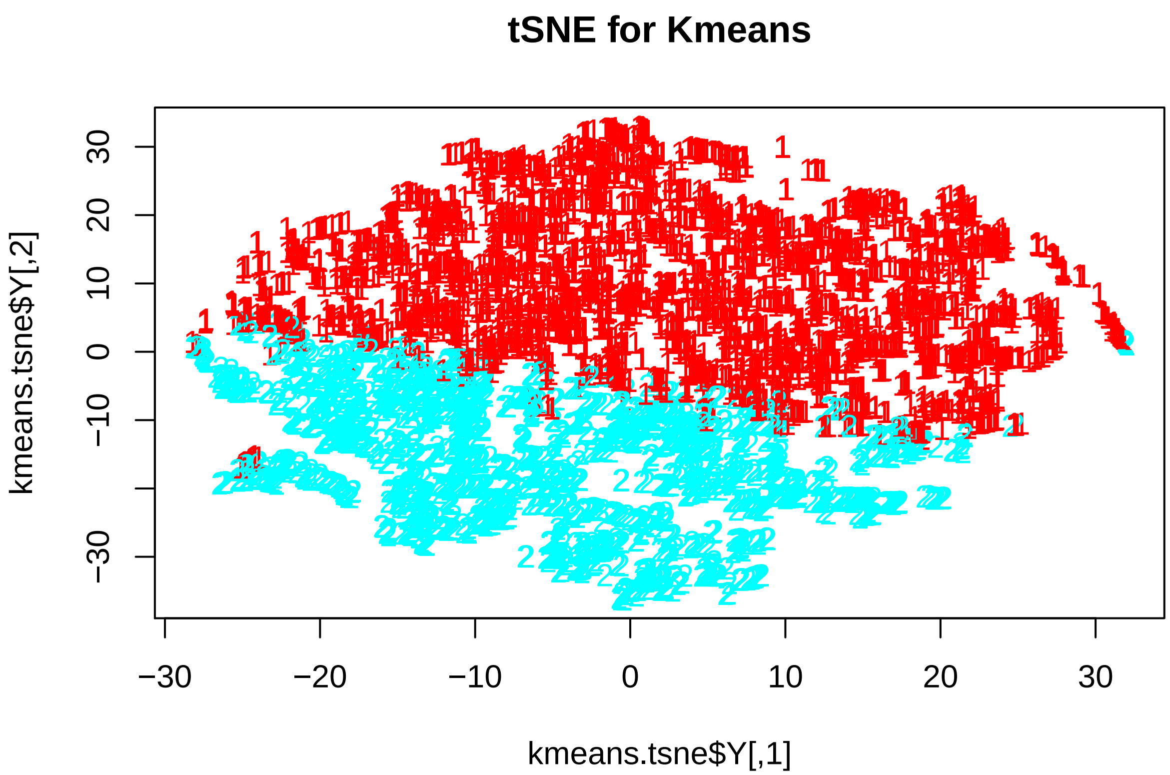
kmeans.groups

1 2

3674 2286

Plotting data

```
library(Rtsne)
colors = rainbow(length(unique(kmeans.groups)))
names(colors) = unique(kmeans.groups)
kmeans.tsne <- Rtsne(dat0, dims=2, perplexity=30, max_iter=500)
plot(kmeans.tsne$Y, t="n", main = "tSNE for Kmeans")
text(kmeans.tsne$Y, labels = kmeans.groups, col = colors[kmeans.groups])
```

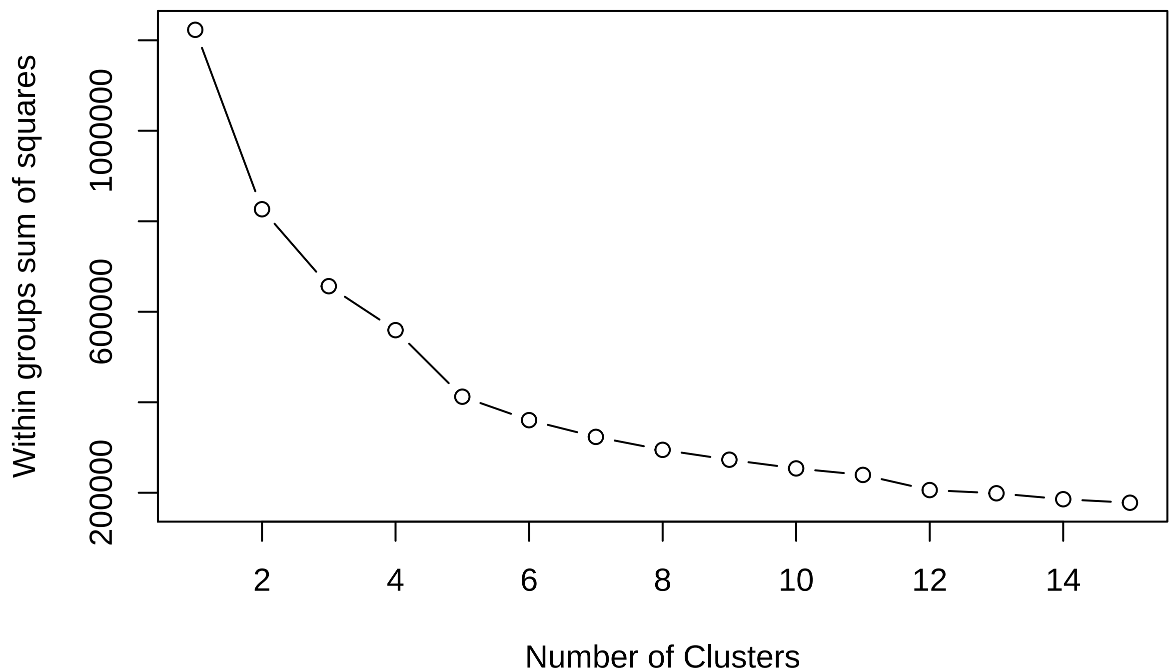


Determine Optimal Cluster

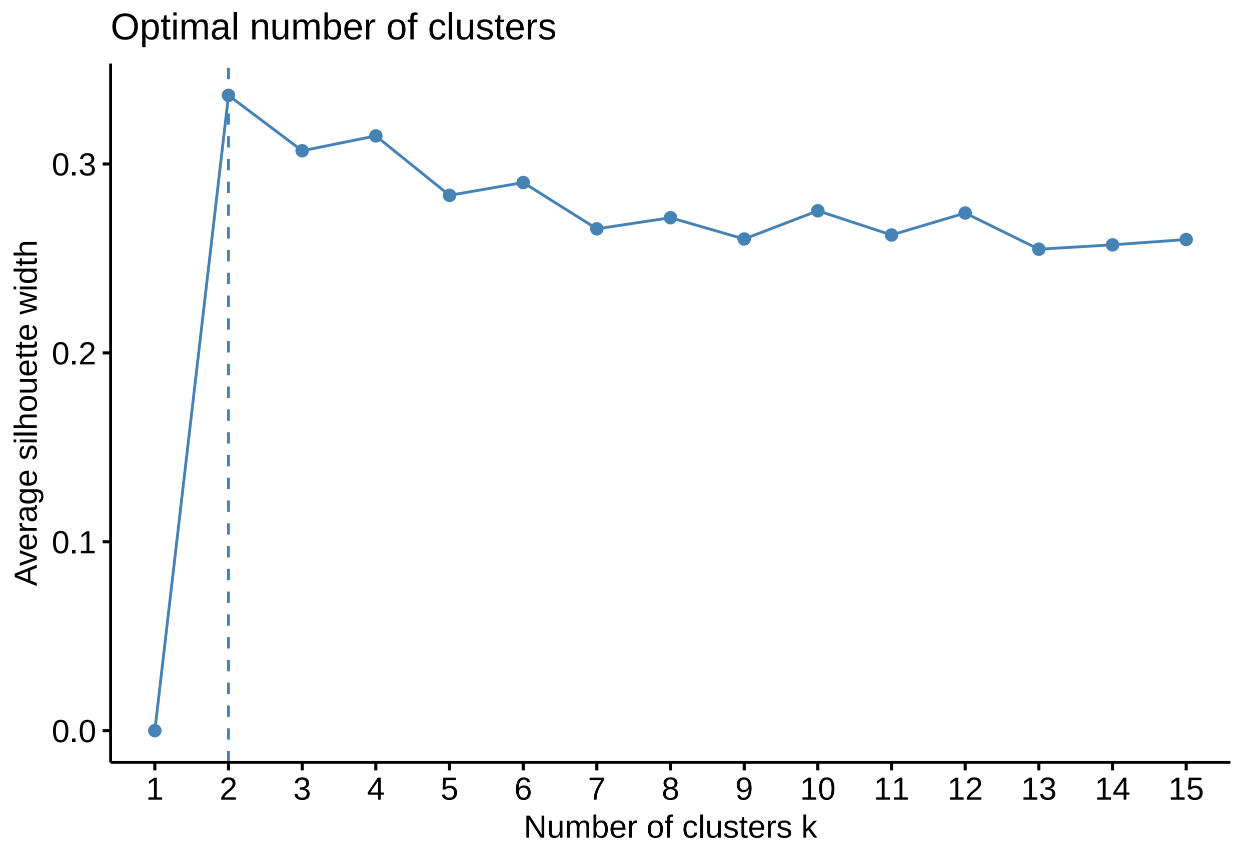
Scree plot of height IN Kmeans clustering

```
wss <- (nrow(dat0)-1)*sum(apply(dat0,2,var))
K.max <- 15
for (K in 2:K.max) wss[K] <- sum(kmeans(dat0, centers=K)$withinss)
```

```r
plot(1:K.max, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")
```




```r
suppressMessages(library(factoextra))
fviz_nbclust(x=dat0, FUNcluster = kmeans, method = c("silhouette"), diss = NULL, k.max = 15)
```



Comment

We observe from the scree plot that the graph begins to decrease slowly when K = 2, which suggest two clusters in the data.

Also, the silhouette method confirms that there are two clusters.

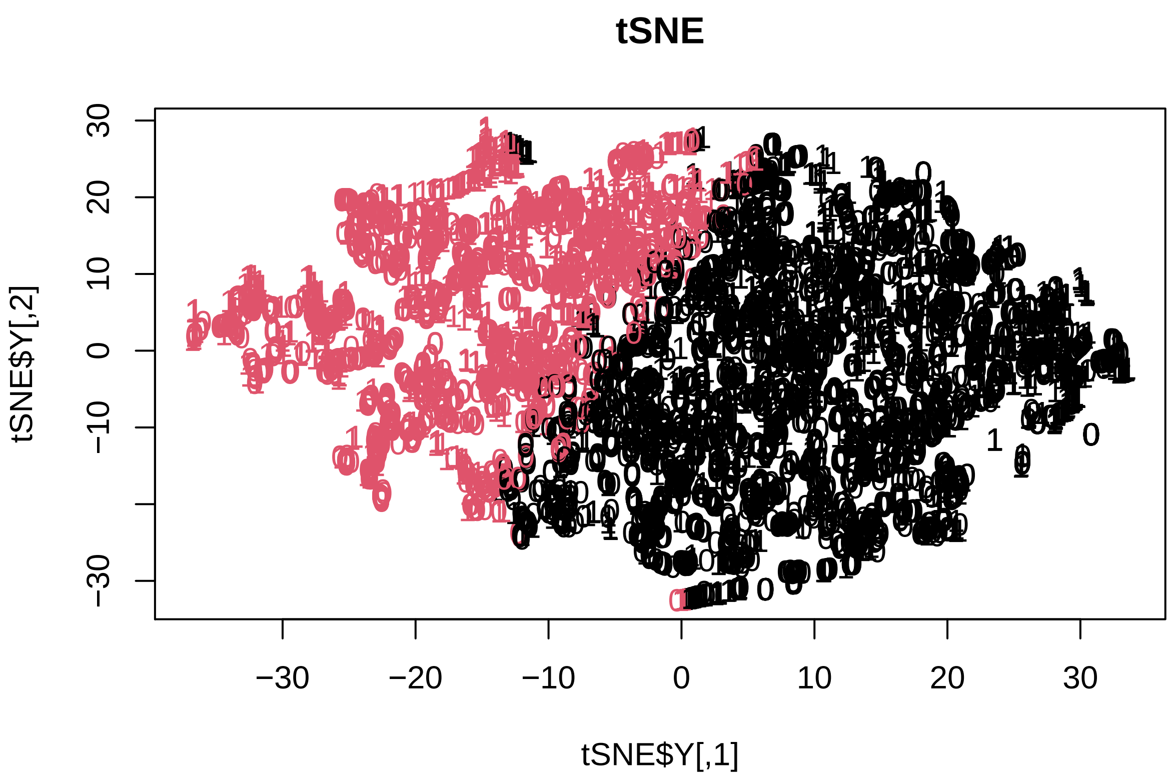
Plot the data using an MDS procedure (e.g., PCA or tSNE). Highlight both the clustering memberships and the BAD value with different color and symbols.

```
BAD <- dat$BAD
BAD <- as.vector(dat$BAD)
dat.3 <- (cbind(dat0,BAD))
dat.3 <- data.frame(dat.3)
library(Rtsne)
tSNE <- Rtsne(dat0, dims=2, perplexity=30, verbose=TRUE, max_iter = 500)
```

```
## Performing PCA
## Read the 5960 x 20 data matrix successfully!
```

```
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
## Done in 1.27 seconds (sparsity = 0.020005)!
## Learning embedding...
## Iteration 50: error is 91.830285 (50 iterations in 1.38 seconds)
## Iteration 100: error is 77.372182 (50 iterations in 1.32 seconds)
## Iteration 150: error is 74.179776 (50 iterations in 1.19 seconds)
## Iteration 200: error is 72.922391 (50 iterations in 1.19 seconds)
## Iteration 250: error is 72.584742 (50 iterations in 1.24 seconds)
## Iteration 300: error is 2.219963 (50 iterations in 1.16 seconds)
## Iteration 350: error is 1.789959 (50 iterations in 1.14 seconds)
## Iteration 400: error is 1.559274 (50 iterations in 1.17 seconds)
## Iteration 450: error is 1.411966 (50 iterations in 1.18 seconds)
## Iteration 500: error is 1.311419 (50 iterations in 1.18 seconds)
## Fitting performed in 12.16 seconds.
```

```r
plot(tSNE$Y, t='n', main="tSNE")
text(tSNE$Y, labels=dat.3$BAD, col=fit.kmeans$cluster)
```



Comments

We observe from the two graphs of tSNE that the hierarchical clustering better cluster the data into the two clusters than Kmeans.

Comparing Hierarchical Clustering and Kmeans Clustering

```
library(clusteval)
jaccard <- cluster_similarity(hclust.groups, kmeans.groups,
           similarity = "jaccard")
rand <- cluster_similarity(hclust.groups, kmeans.groups, similarity = "rand")
matrix(c("Jaccard", jaccard, "Rand", rand), byrow = T, ncol = 2)
```

```
##      [,1]      [,2]
## [1,] "Jaccard" "0.366788613397245"
## [2,] "Rand"    "0.500781965353855"
```

Comment

[,1] [,2]

[1,] "Jaccard" "0.366788613397245"

[2,] "Rand" "0.500781965353855"

Comment

We observe that the values are close or fairly similar.

(v)

Post Hoc Analysis

We consider the result from the hierarchical clustering method to perform post hoc analysis. We characterize each cluster, and explore the association of the cluster with the 12 predictors and the outcome 'BAD'.

Performing a t test for the differences in means

```
aggregate(dat[, c(1,2,3,4,7,9,13)], list(hclust.groups), mean, na.rm =T)
```

```
##   Group.1       BAD     LOAN  MORTDUE    VALUE      YOJ    DELINQ  DEBTINC
## 1       1 0.2185039 9.461746 10.87125 11.36399 2.015710 0.4768701 33.34093
## 2       2 0.1896640 9.780385 11.01978 11.42175 1.909102 0.4368635 34.77807
```

Comment

We observe that there is a statistically significant

difference in the mean value of DEBTINC: Debt-to-income ratio for the two ## groups. Cluster 2 turns to have high Debt-to-income ratio than Cluster 1.

```
cond1 <- hclust.groups == 1
cond2 <- hclust.groups == 2
var.test(dat$DEBTINC[cond1], dat$DEBTINC[cond2], alternative ="two.sided")
```

```
##
##  F test to compare two variances
##
## data:  dat$DEBTINC[cond1] and dat$DEBTINC[cond2]
## F = 1.3652, num df = 2031, denom df = 3927, p-value = 4.441e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.266178 1.473372
## sample estimates:
## ratio of variances
##            1.36522
```

Comment

F test to compare two variances

data: dat$DEBTINC[cond1]anddat$**DEBTINC[cond2]**

F = 1.3652, num df = 2031, denom df = 3927, p-value = 4.441e-16

alternative hypothesis: true ratio of variances is not equal to 1

95 percent confidence interval:

1.266178 1.473372

sample estimates:

ratio of variances

1.36522

```
t.test(dat$DEBTINC[cond1], dat$DEBTINC[cond2], alternative = "two.sided",
var.equal = T)
```

```
##
##  Two Sample t-test
##
## data:  dat$DEBTINC[cond1] and dat$DEBTINC[cond2]
## t = -5.4297, df = 5958, p-value = 5.867e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.956016 -0.918275
## sample estimates:
## mean of x mean of y
##  33.34093  34.77807
```

Comment

Two Sample t-test

data: dat$DEBTINC[cond1]anddat$DEBTINC[cond2]

t = -5.4297, df = 5958, p-value = 5.867e-08

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-1.956016 -0.918275

sample estimates:

mean of x mean of y

33.34093 34.77807

Difference in CLNO

```r
dat.4 <- table(dat$CLNO , hclust.groups)
dat.4 <- as.data.frame(dat.4)
dat.4$Var1 <- as.numeric(dat.4$Var1)
dat.4$hclust.groups <- as.numeric(dat.4$hclust.groups)
cond1 <- (dat.4$Var1 <= 21) & (dat.4$hclust.groups == 1)
cond2 <- (dat.4$Var1 <= 21) & (dat.4$hclust.groups == 2)
lessq1 <- sum(dat.4$Freq[cond1])
lessq2 <- sum(dat.4$Freq[cond2])
cond11 <- (dat.4$Var1 > 21) & (dat.4$hclust.groups == 1)
cond22 <- (dat.4$Var1 > 21) & (dat.4$hclust.groups == 2)
grt1 <- sum(dat.4$Freq[cond11])
grt2 <- sum(dat.4$Freq[cond22])
matrix(c("CLNO", "Cluster 1", "Cluster 2", "<= 21", lessq1, lessq2, "> 21", grt1, grt2),byrow = T, ncol
```

```
##      [,1]    [,2]        [,3]
## [1,] "CLNO"  "Cluster 1" "Cluster 2"
## [2,] "<= 21" "1209"      "1861"
## [3,] "> 21"  "823"       "2067"
```

Comment

[,1] [,2] [,3]

[1,] "CLNO" "Cluster 1" "Cluster 2"

[2,] "<= 21" "1209" "1861"

[3,] "> 21" "823" "2067"

We observe from the table above that 1209 individuals in Cluster 1

have number of credit lines (CLNO) less than or equal to 21, whereas,

2067 individuals in Cluster 2 have number of credit lines (CLNO)

greater than 21.

Relationship between Clusters and Predictors

```
table(dat$JOB , hclust.groups)
```

```
##          hclust.groups
##              1    2
##   Mgr      195  572
##   Office   328  620
##   Other    784 1604
##   ProfExe  429  847
##   Sales     12   97
##   Self     120   73
##   Unknown  164  115
```

Comment

hclust.groups

1 2

Mgr 195 572

Office 328 620

Other 784 1604

ProfExe 429 847

Sales 12 97

Self 120 73

Unknown 164 115

Comment

We observe that other Jobs are the main reason given by

applicants in the applicants in the two clusters.

```
table(dat$REASON , hclust.groups)
```

```
##          hclust.groups
##              1    2
##   DebtCon    0 3928
##   HomeImp 1780    0
##   Unknown  252    0
```

Comment

hclust.groups

1 2

DebtCon 0 3928

HomeImp 1780 0

Unknown 252 0

Comment

Debt consolidation is the main reason for applicants in cluster 2 while

HomeImp is the main reason for applicants in cluster 1.

Association between cluster and the outcome "BAD"

```r
table(dat$BAD, hclust.groups)
```

```
##    hclust.groups
##        1    2
##   0 1588 3183
##   1  444  745
```

Comment

hclust.groups

1 2

0 1588 3183

1 444 745

We observe that value 0 has the highest number in both cluster which

means when the homeowner repaid the home equity line of credit.