Assignment 2

Yizhe Qu

March 10, 2019

Q1

```
library(mlbench)
```

```
## Warning: package 'mlbench' was built under R version 3.5.2
```

```
data(HouseVotes84)
hv <- HouseVotes84
head(hv)
```

```
##        Class   V1 V2 V3   V4   V5 V6 V7 V8 V9 V10  V11  V12 V13 V14 V15
## 1 republican    n  y  n    y    y  y  n  n  n   y <NA>    y   y   y   n
## 2 republican    n  y  n    y    y  y  n  n  n   n    n    y   y   y   n
## 3   democrat <NA>  y  y <NA>    y  y  n  n  n   n    y    n   y   y   n
## 4   democrat    n  y  y    n <NA>  y  n  n  n   n    y    n   y   n   n
## 5   democrat    y  y  y    n    y  y  n  n  n   n    y <NA>   y   y   y
## 6   democrat    n  y  y    n    y  y  n  n  n   n    n    n   y   y   y
##    V16
## 1    y
## 2 <NA>
## 3    n
## 4    y
## 5    y
## 6    y
```

a.

```
# Cleaning NAs
hv2 <- data.frame(lapply(hv, as.character), stringsAsFactors=FALSE)
hv2[is.na(hv2)] <- 'NoVote'
hv <- data.frame(lapply(hv2, as.factor))
head(hv)
```

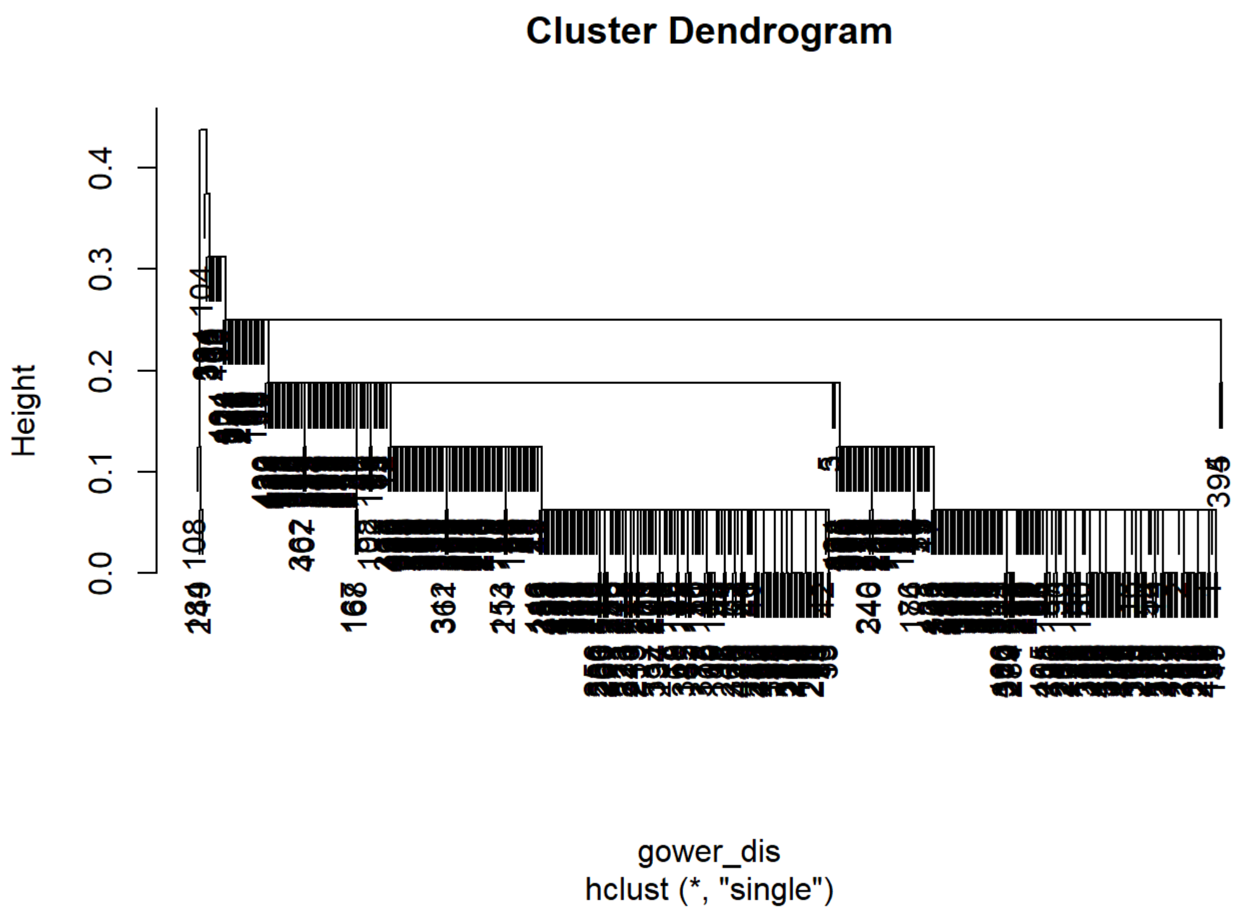
```
##        Class     V1 V2 V3     V4     V5 V6 V7 V8 V9 V10    V11    V12 V13
## 1 republican      n  y  n      y      y  y  n  n  n   y NoVote      y   y
## 2 republican      n  y  n      y      y  y  n  n  n   n      n      y   y
## 3   democrat NoVote  y  y NoVote      y  y  n  n  n   n      y      n   y
## 4   democrat      n  y  y      n NoVote  y  n  n  n   n      y      n   y
## 5   democrat      y  y  y      n      y  y  n  n  n   n      y NoVote   y
## 6   democrat      n  y  y      n      y  y  n  n  n   n      n      n   y
##   V14 V15    V16
## 1   y   n      y
## 2   y   n NoVote
## 3   y   n      n
## 4   n   n      y
## 5   y   y      y
## 6   y   y      y
```

b.

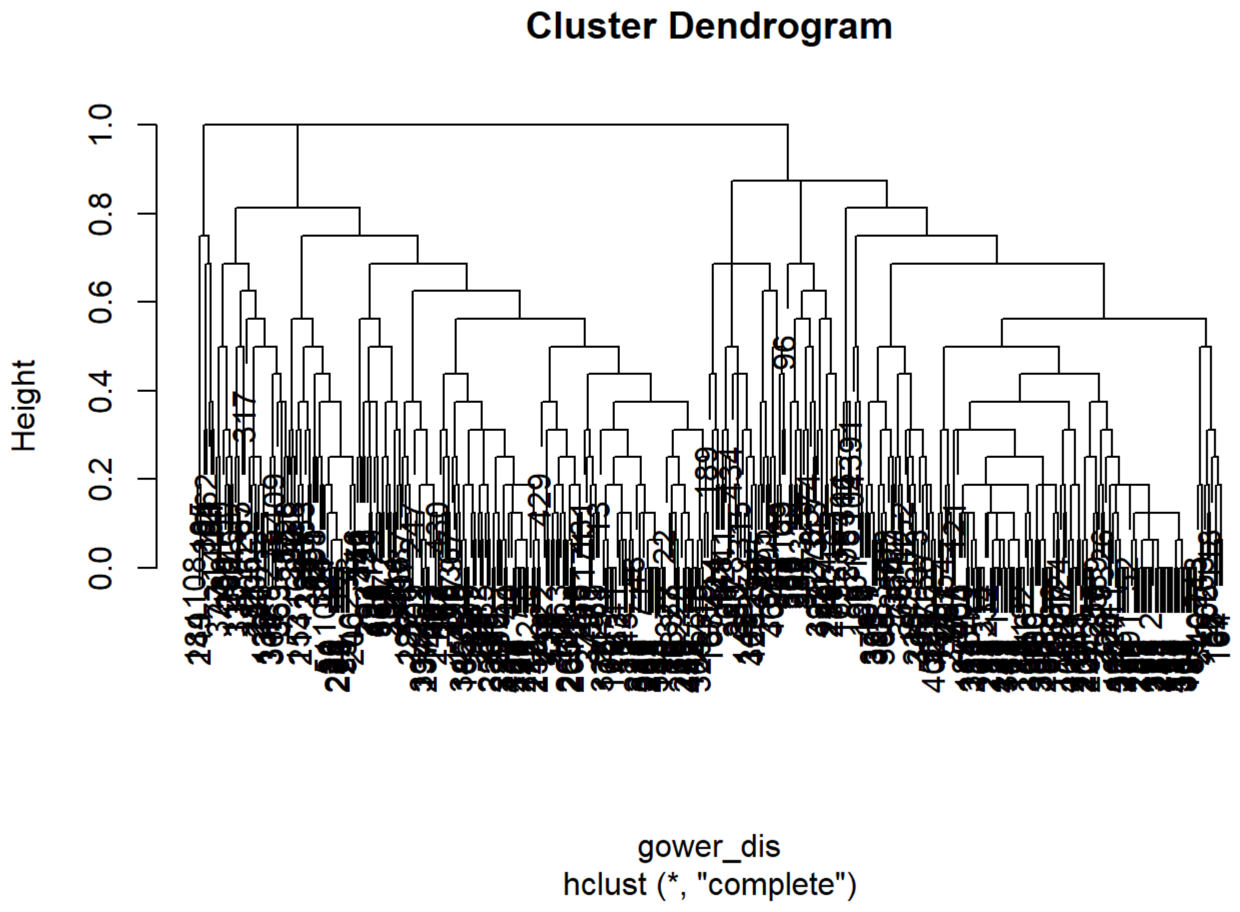
```
library(cluster)
hv_d <- hv[-1]
gower_dis <- daisy(hv_d, metric='gower')
dis_matrix <- as.matrix(gower_dis)
```

c.

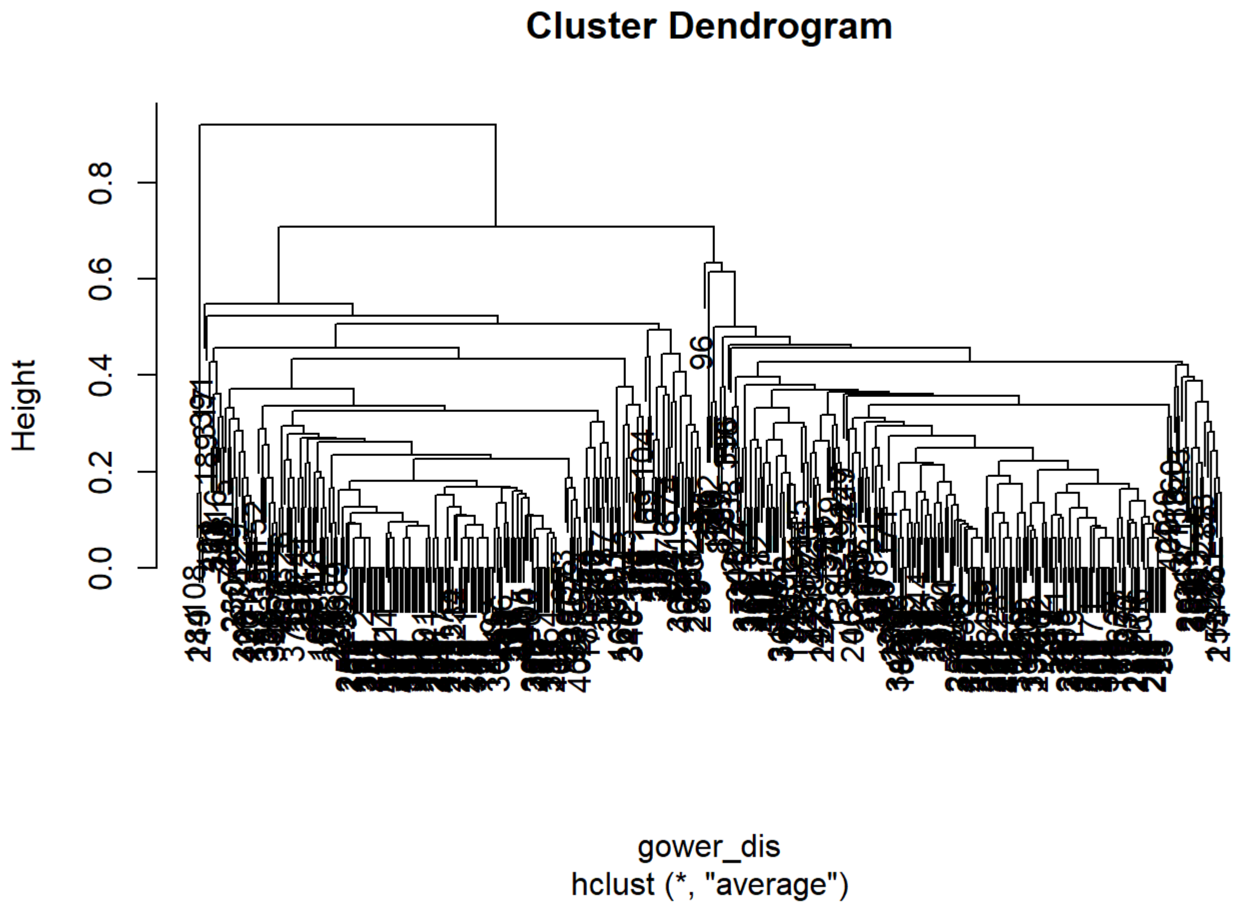
```
# Single Linkage Method
clus1 <- hclust(gower_dis, method='single')
plot(clus1)
```



```
# Complete Linkage Method
clus2 <- hclust(gower_dis, method='complete')
plot(clus2)
```



```
# Average Linkage Method
clus3 <- hclust(gower_dis, method='average')
plot(clus3)
```



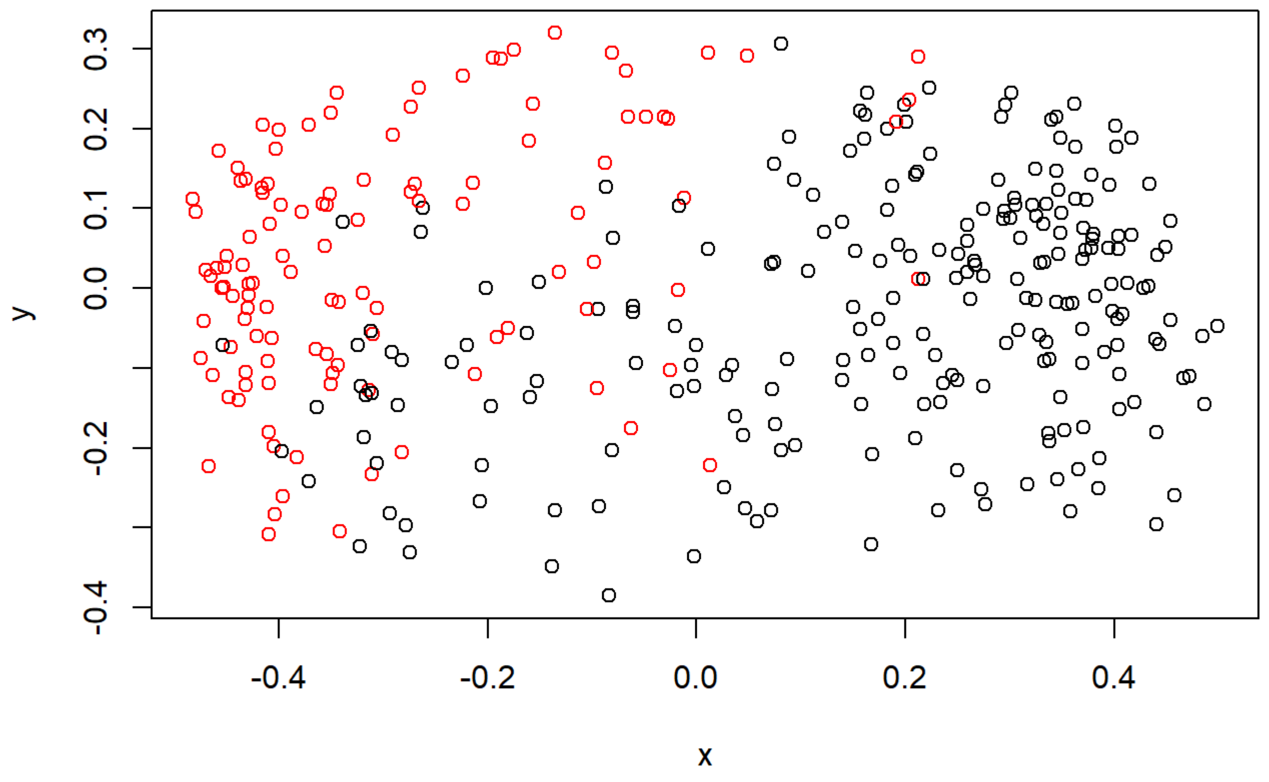
Cluster 3 - ‘average’ method gives the clearest grouping result, which suggests 2 groups

```
party_aff <- cutree(clus3,2)
table(hv$Class, party_aff)
```

```
##             party_aff
##                1   2
##   democrat   266   1
##   republican 166   2
```

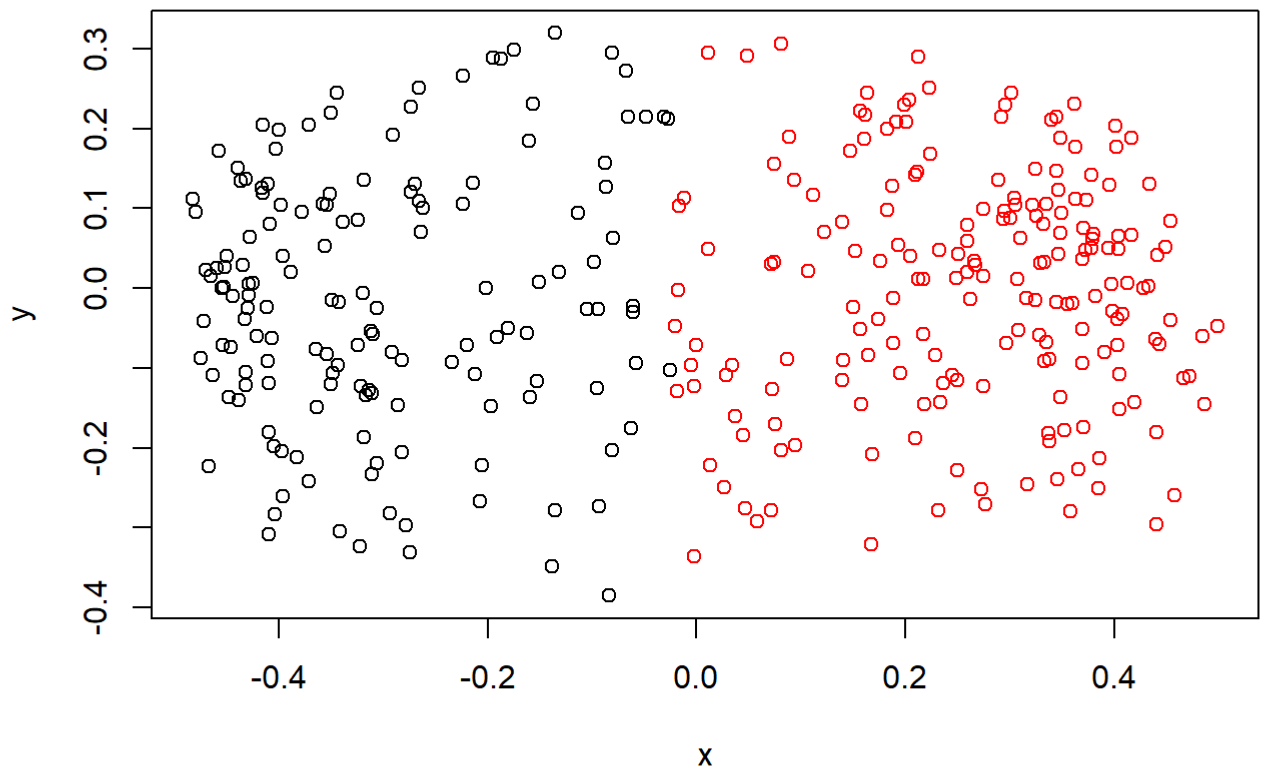
d.

```
mds <- cmdscale(dis_matrix,eig=TRUE, k=2)
x <- mds$points[,1]
y <- mds$points[,2]

# empty plot
plot(x,y, type = 'n')
# add points
points(x,y, col= hv$Class, type = "p", xlab = "", ylab = "", asp = 1)
```



e.

```
# Performing K-means
k_mean <- kmeans(mds$points, 2)
plot(x,y, col=k_mean$cluster)
```



```
table(party_aff,k_mean$cluster)
```

```
##
## party_aff   1   2
##         1 200 232
##         2   1   2
```

f.

```
# Mixture Models
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 3.5.2
```

```
## Package 'mclust' version 5.4.2
## Type 'citation("mclust")' for citing this R package in publications.
```

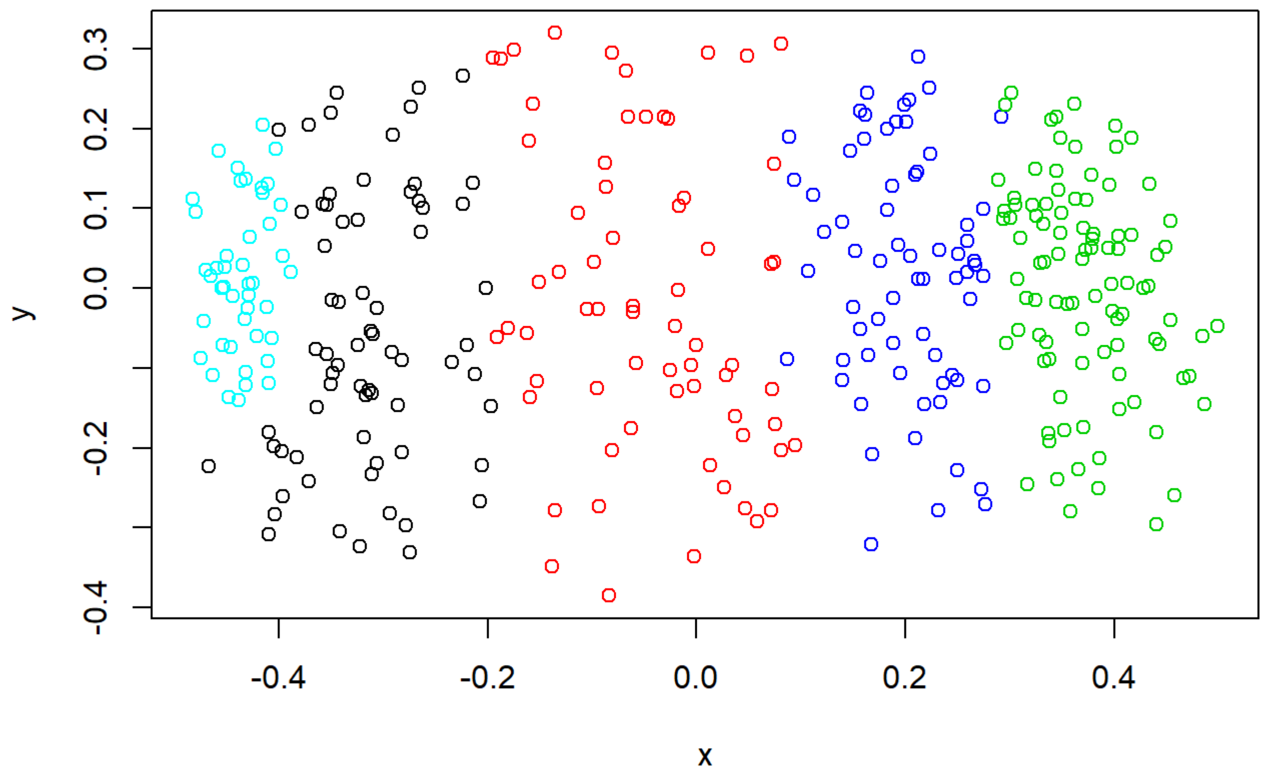
```
mhv <- Mclust(mds$points)
mhv$BIC
```

```
## Bayesian Information Criterion (BIC):
##         EII       VII      EEI      VEI      EVI      VVI      EEE
## 1 -138.3982 -138.3982 128.6049 128.6049 128.6049 128.6049 122.5295
## 2  355.7648  349.7271 352.4105 346.3489 347.4166 368.6731 346.8456
## 3  362.9855  400.5381 428.4158 451.7854 423.2005 480.7041 432.5525
## 4  386.6359  406.6416 452.1234 474.5390 443.1983 474.0827 454.7219
## 5  387.2951  433.2811 463.2730 500.4131 452.4908 495.2303 477.4899
## 6  416.4601  431.7960 466.0254 489.7436 451.9888 486.2804 472.8307
## 7  409.6636  417.0000 449.9723 470.7878 442.6724 472.7371 458.4763
## 8  402.5593  442.4085 431.7587 459.3562 423.8979 453.8031 455.0491
## 9  384.4001  448.1724 449.4872 475.8503 416.4475 425.8652 447.6013
##        EVE      VEE      VVE      EEV      VEV      EVV      VVV
## 1 122.5295 122.5295 122.5295 122.5295 122.5295 122.5295 122.5295
## 2 341.6965 340.8421 363.8768 340.8374 346.6227 335.7557 360.1159
## 3 427.5874 454.6219 478.8831 423.0056 450.5796 417.7866 471.5715
## 4 439.7113 481.8702 467.4849 433.0711 469.0756 426.2810 457.3049
## 5 463.9331 498.8761 486.8870 451.7624 469.3496 440.8429 480.5732
## 6 445.2878 483.7543 479.6194 435.8881 399.7563 440.3431 463.1684
## 7 444.1552 471.0763 467.6288 436.2342 427.8358 417.3385 439.9679
## 8 429.6832 459.1995 448.6888 436.3932 415.5953 395.1761 401.6932
## 9 420.2819 470.2834 408.7210 417.6591 407.7259 374.7703 378.3597
##
## Top 3 models based on the BIC criterion:
##    VEI,5    VEE,5    VVI,5
## 500.4131 498.8761 495.2303
```

```
table(hv$Class, mhv$classification)
```

```
##
##                1   2   3   4   5
##   democrat    26  39 138  63   1
##   republican  45  27   0   4  92
```

```
plot(x,y, col=mhv$classification)
```



5 groups are suggested by BIC

Q2

```
cov_mat <- ability.cov$cov
cov_mat
```

```
##         general picture  blocks   maze reading   vocab
## general  24.641   5.991  33.520  6.023  20.755  29.701
## picture   5.991   6.700  18.137  1.782   4.936   7.204
## blocks   33.520  18.137 149.831 19.424  31.430  50.753
## maze      6.023   1.782  19.424 12.711   4.757   9.075
## reading  20.755   4.936  31.430  4.757  52.604  66.762
## vocab    29.701   7.204  50.753  9.075  66.762 135.292
```

a.

```
facar1 <- factanal(covmat=cov_mat, factors=1, rotation='none',n.obs=112)
facar1
```

```
##
## Call:
## factanal(factors = 1, covmat = cov_mat, n.obs = 112, rotation = "none")
##
## Uniquenesses:
## general picture  blocks    maze reading   vocab
##   0.535   0.853   0.748   0.910   0.232   0.280
##
## Loadings:
##         Factor1
## general 0.682
## picture 0.384
## blocks  0.502
## maze    0.300
## reading 0.877
## vocab   0.849
##
##                Factor1
## SS loadings      2.443
## Proportion Var   0.407
##
## Test of the hypothesis that 1 factor is sufficient.
## The chi square statistic is 75.18 on 9 degrees of freedom.
## The p-value is 1.46e-12
```

No, since:
1) the uniqueness values for picture, blocks and maze are very high, which indicate that those variables are not being captured with the current factor
2) the cumulative proportion of variance explained by this factor is only 0.407 (pretty low value)
3) the p-value < 0.05, so reject the null hypothesis that only 1 factor is sufficient to describe the data

b.

```
cor_mat <- cov2cor(cov_mat)
facar2 <- factanal(covmat=cor_mat, factors=1, rotation='none',n.obs=112)
facar2
```

```
## 
## Call:
## factanal(factors = 1, covmat = cor_mat, n.obs = 112, rotation = "none")
## 
## Uniquenesses:
## general picture  blocks    maze reading   vocab 
##   0.535   0.853   0.748   0.910   0.232   0.280 
## 
## Loadings:
##         Factor1
## general 0.682  
## picture 0.384  
## blocks  0.502  
## maze    0.300  
## reading 0.877  
## vocab   0.849  
## 
##                Factor1
## SS loadings      2.443
## Proportion Var   0.407
## 
## Test of the hypothesis that 1 factor is sufficient.
## The chi square statistic is 75.18 on 9 degrees of freedom.
## The p-value is 1.46e-12
```

There doesn’t seem a difference in this model to the previous one

c.

```
facar3 <- factanal(covmat=cov_mat, factors=2, rotation='none', n.obs=112)
facar3
```

```
##
## Call:
## factanal(factors = 2, covmat = cov_mat, n.obs = 112, rotation = "none")
##
## Uniquenesses:
## general picture  blocks    maze reading   vocab
##   0.455   0.589   0.218   0.769   0.052   0.334
##
## Loadings:
##         Factor1 Factor2
## general  0.648   0.354
## picture  0.347   0.538
## blocks   0.471   0.748
## maze     0.253   0.408
## reading  0.964  -0.135
## vocab    0.815
##
##                Factor1 Factor2
## SS loadings      2.420   1.162
## Proportion Var   0.403   0.194
## Cumulative Var   0.403   0.597
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 6.11 on 4 degrees of freedom.
## The p-value is 0.191
```

Yes, the p-value > 0.05, so fail to reject the null hypothesis that 2 factors are sufficient to describe the data
Factor 1 has significant influence on the variables: “general”, “reading”, and “vocab”
Factor 2 has significant influence on the variable: “blocks”

d.

```
facar4 <- factanal(covmat=cov_mat, factors=2, rotation='varimax', n.obs=112)
facar4
```

```
##
## Call:
## factanal(factors = 2, covmat = cov_mat, n.obs = 112, rotation = "varimax")
##
## Uniquenesses:
## general picture  blocks    maze reading   vocab 
##   0.455   0.589   0.218   0.769   0.052   0.334 
##
## Loadings:
##         Factor1 Factor2
## general 0.499   0.543  
## picture 0.156   0.622  
## blocks  0.206   0.860  
## maze    0.109   0.468  
## reading 0.956   0.182  
## vocab   0.785   0.225  
##
##                Factor1 Factor2
## SS loadings      1.858   1.724
## Proportion Var   0.310   0.287
## Cumulative Var   0.310   0.597
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 6.11 on 4 degrees of freedom.
## The p-value is 0.191
```

The uniquenesses of the variables and the p-value haven’t changed, while each individual loadings have changed
Factor 1 now has significant influence on the variables: “reading” and “vocab”
Factor 2 now has significant influence on the variable: “blocks”
Yes, it is easier to interpret than before since the “varimax” rotation is used so that each variable loads heavily on only one factor

e.

```
facar5 <- factanal(covmat=cov_mat, factors=2, rotation='promax', n.obs=112)
facar5
```

```
##
## Call:
## factanal(factors = 2, covmat = cov_mat, n.obs = 112, rotation = "promax")
##
## Uniquenesses:
## general picture  blocks    maze reading   vocab
##   0.455   0.589   0.218   0.769   0.052   0.334
##
## Loadings:
##         Factor1 Factor2
## general  0.364   0.470
## picture          0.671
## blocks           0.932
## maze             0.508
## reading  1.023
## vocab    0.811
##
##                Factor1 Factor2
## SS loadings      1.853   1.807
## Proportion Var   0.309   0.301
## Cumulative Var   0.309   0.610
##
## Factor Correlations:
##         Factor1 Factor2
## Factor1   1.000   0.557
## Factor2   0.557   1.000
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 6.11 on 4 degrees of freedom.
## The p-value is 0.191
```

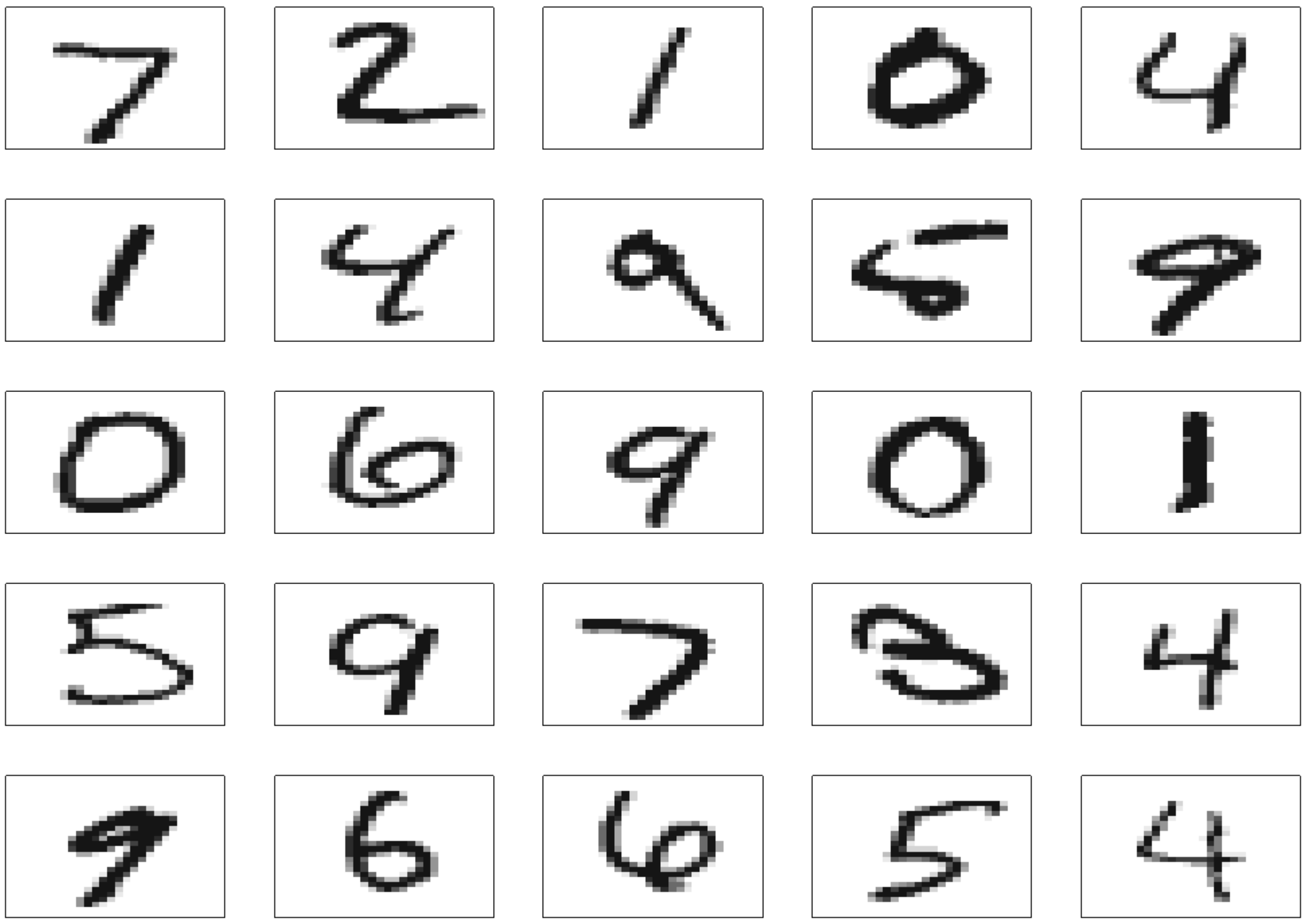
Factor Correlations are added to the output
We assume non-orthogonal rotation (oblique rotation), so that there will be a correlation between Factors
Factor 1 loads heavily on the variables: “reading” and “vocab”
Factor 2 loads heavily on the variable: “blocks”
The correlation between Factor 1 and Factro 2 is 0.557

Q3

```
#download.file("http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz","t10k-images-idx3-ub
yte.gz")
#download.file("http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz","t10k-labels-idx1-ub
yte.gz")
# install.packages("R.utils")
#R.utils::gunzip("t10k-images-idx3-ubyte.gz")
#R.utils::gunzip("t10k-labels-idx1-ubyte.gz")
# helper function for visualization
show_digit = function(arr784, col = gray(12:1 / 12), ...) {
  image(matrix(as.matrix(arr784[-785]), nrow = 28)[, 28:1], col = col, ...)
}

# load image files
load_image_file = function(filename) {
  ret = list()
  f = file(filename, 'rb')
  readBin(f, 'integer', n = 1, size = 4, endian = 'big')
  n    = readBin(f, 'integer', n = 1, size = 4, endian = 'big')
  nrow = readBin(f, 'integer', n = 1, size = 4, endian = 'big')
  ncol = readBin(f, 'integer', n = 1, size = 4, endian = 'big')
  x = readBin(f, 'integer', n = n * nrow * ncol, size = 1, signed = FALSE)
  close(f)
  data.frame(matrix(x, ncol = nrow * ncol, byrow = TRUE))
}

# load label files
load_label_file = function(filename) {
  f = file(filename, 'rb')
  readBin(f, 'integer', n = 1, size = 4, endian = 'big')
  n = readBin(f, 'integer', n = 1, size = 4, endian = 'big')
  y = readBin(f, 'integer', n = n, size = 1, signed = FALSE)
  close(f)
  y
}

# load images
test  = load_image_file("t10k-images-idx3-ubyte")

# load labels

test$y  = as.factor(load_label_file("t10k-labels-idx1-ubyte"))
```

a.

```
par(mfrow=c(5,5))
par(mar=c(1,1,1,1))
for(i in 1:25){
    show_digit(test[i, ],xaxt="n", yaxt="n")
    }
```



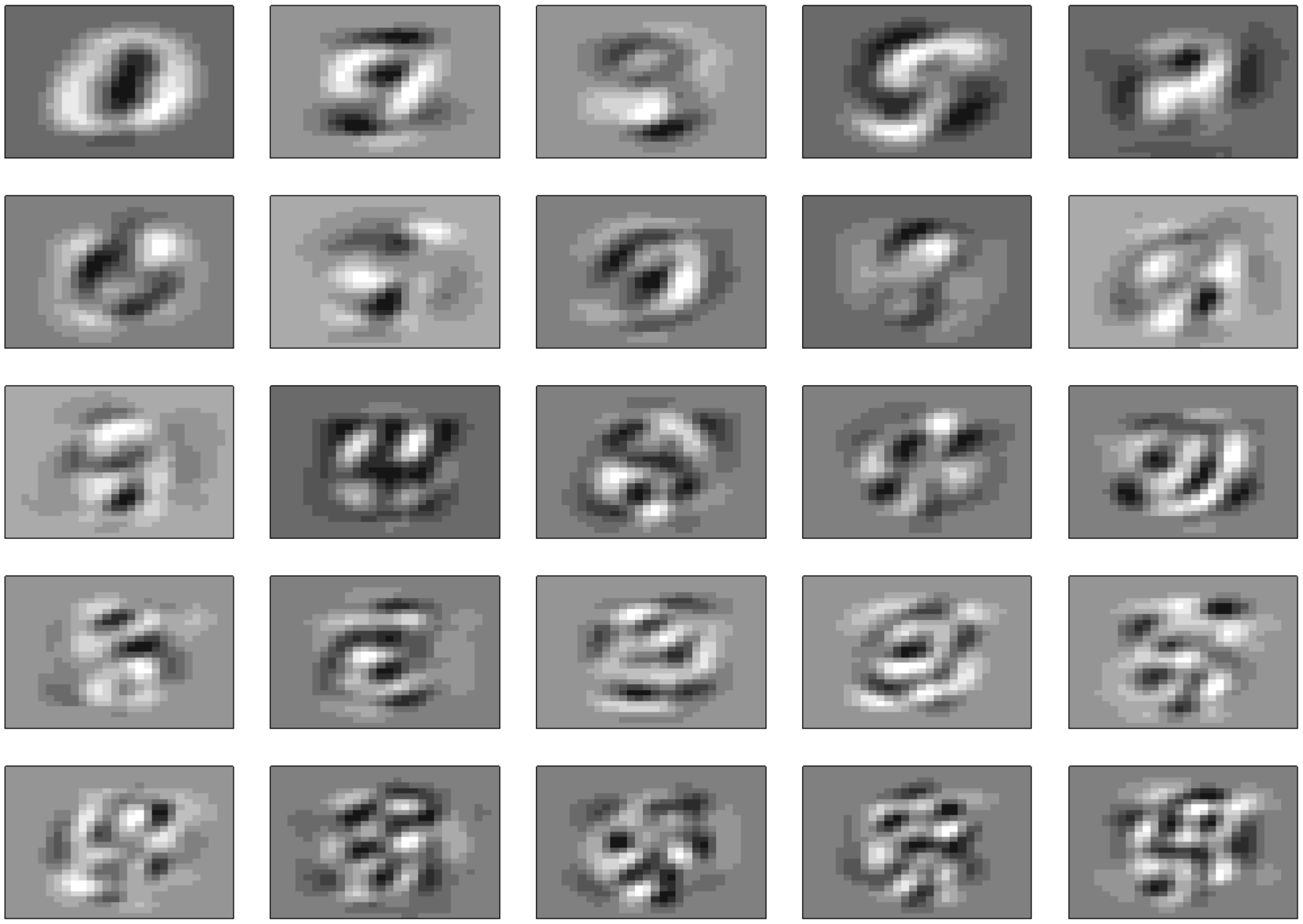
b)

```
pr <- prcomp(test[-785])
```

The maximum number of permittable components is 784

c.

```
par(mfrow=c(5,5), omi=c(0,0,0,0), mai=c(0.1,0.1,0.1,0.1))

for(i in 1:25){
    show_digit(matrix(pr$rotation[,i]),xaxt="n", yaxt="n")
}
```



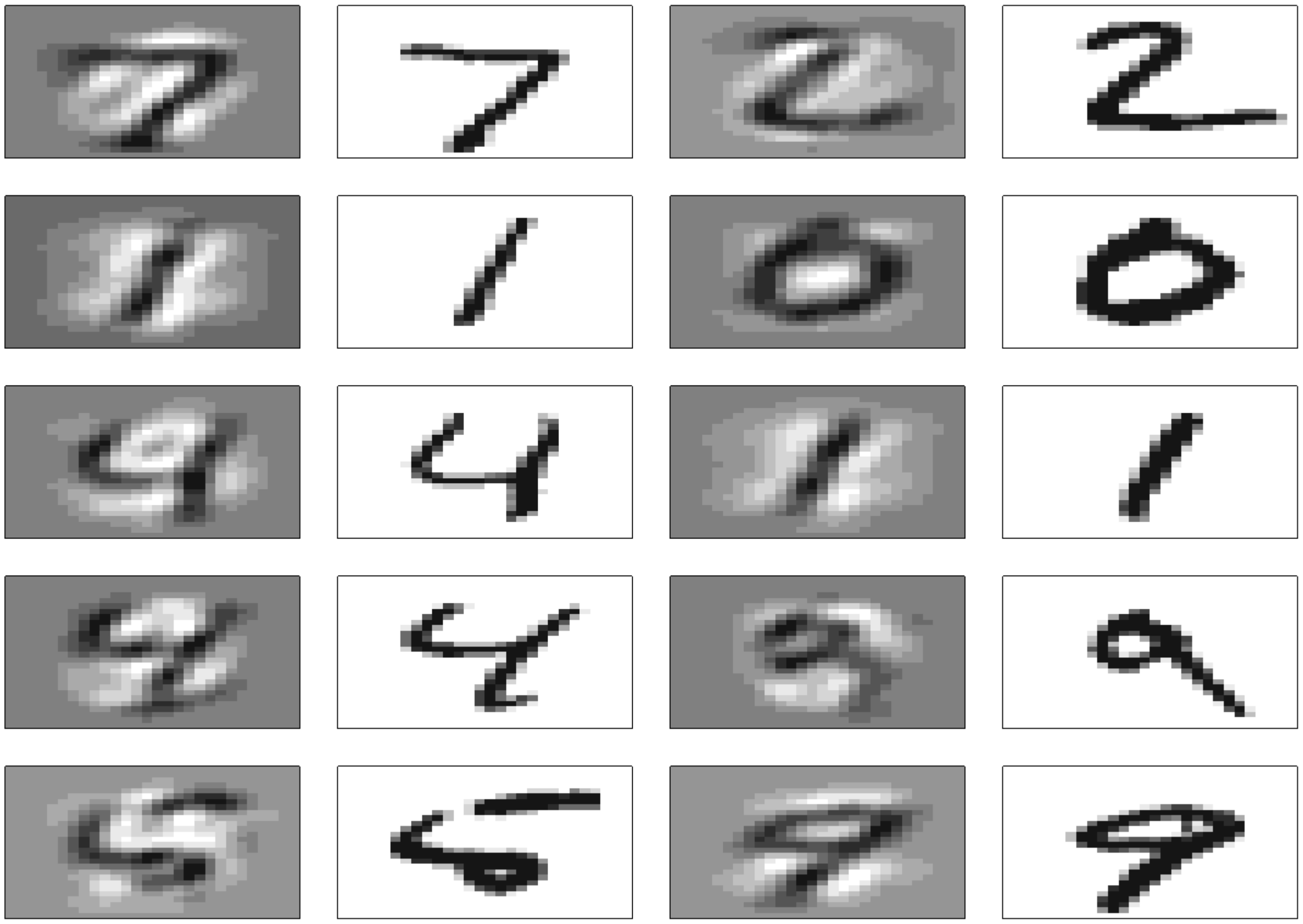
```
mx_transformed <- pr$x
vars_transformed <- apply(mx_transformed, 2, var)
cum_var_25 <- sum(vars_transformed[1:25])/sum(vars_transformed)
cum_var_25
```

```
## [1] 0.7018987
```

The percentage of the original variation in the pixels explained by the first 25 PCs is 0.7019

d.

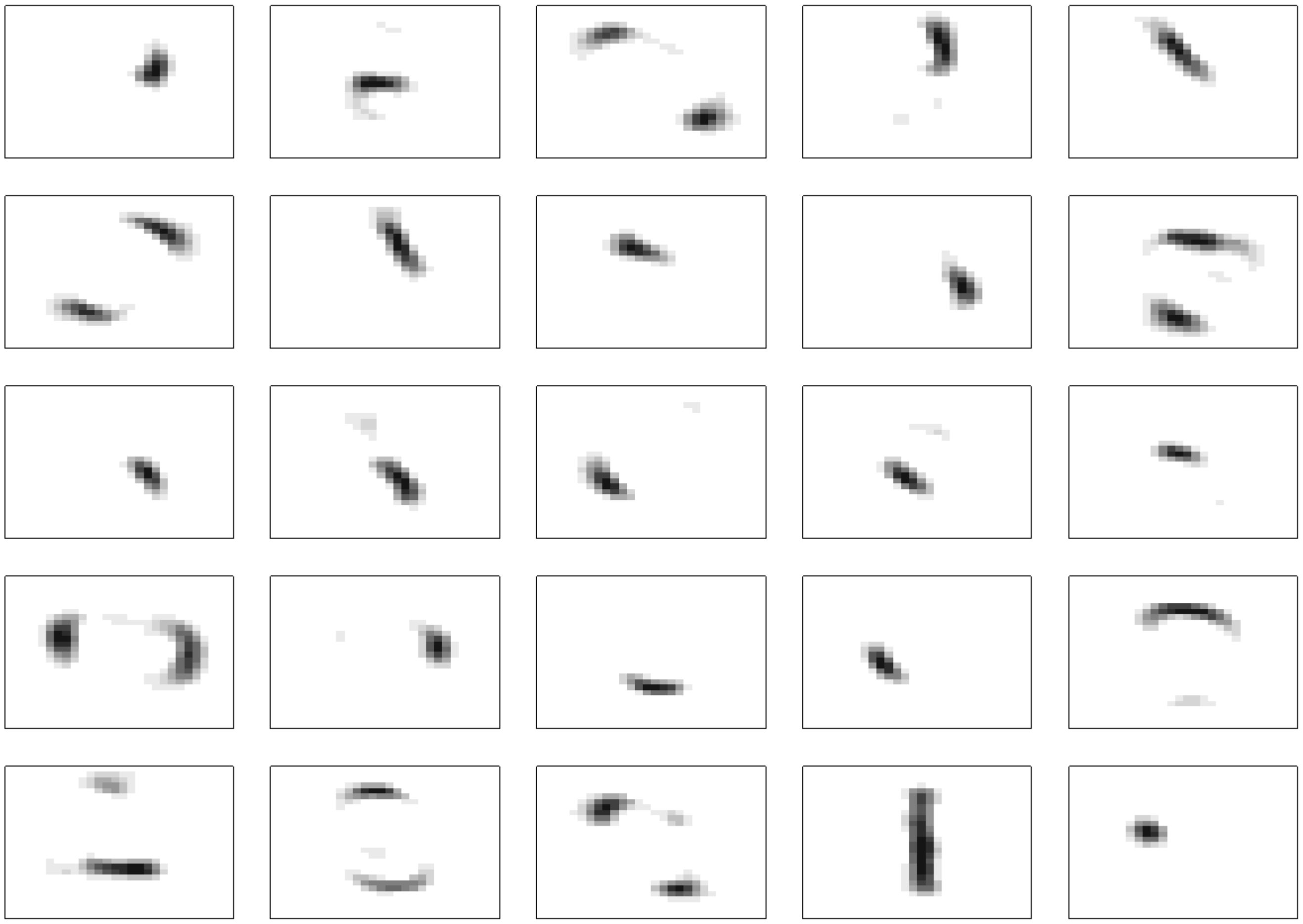
```
par(mfrow=c(5,4), omi=c(0,0,0,0), mai=c(0.1,0.1,0.1,0.1))

i <- 25
for(x in 1:10){
  reconst <- (t(pr$rotation[,1:i] %*% t(pr$x[,1:i])))
  # Plot first 10 digits from reconstructions
  show_digit(t(matrix(reconst[x,])), xaxt="n", yaxt="n")
  # Plot original first 10 digits
  show_digit(test[x, ],xaxt="n", yaxt="n")
}
```



e.

```r
load("C:/Users/yizhe/Desktop/MDS/Term5/data_573/assignments/nmfres.Rdata")
```

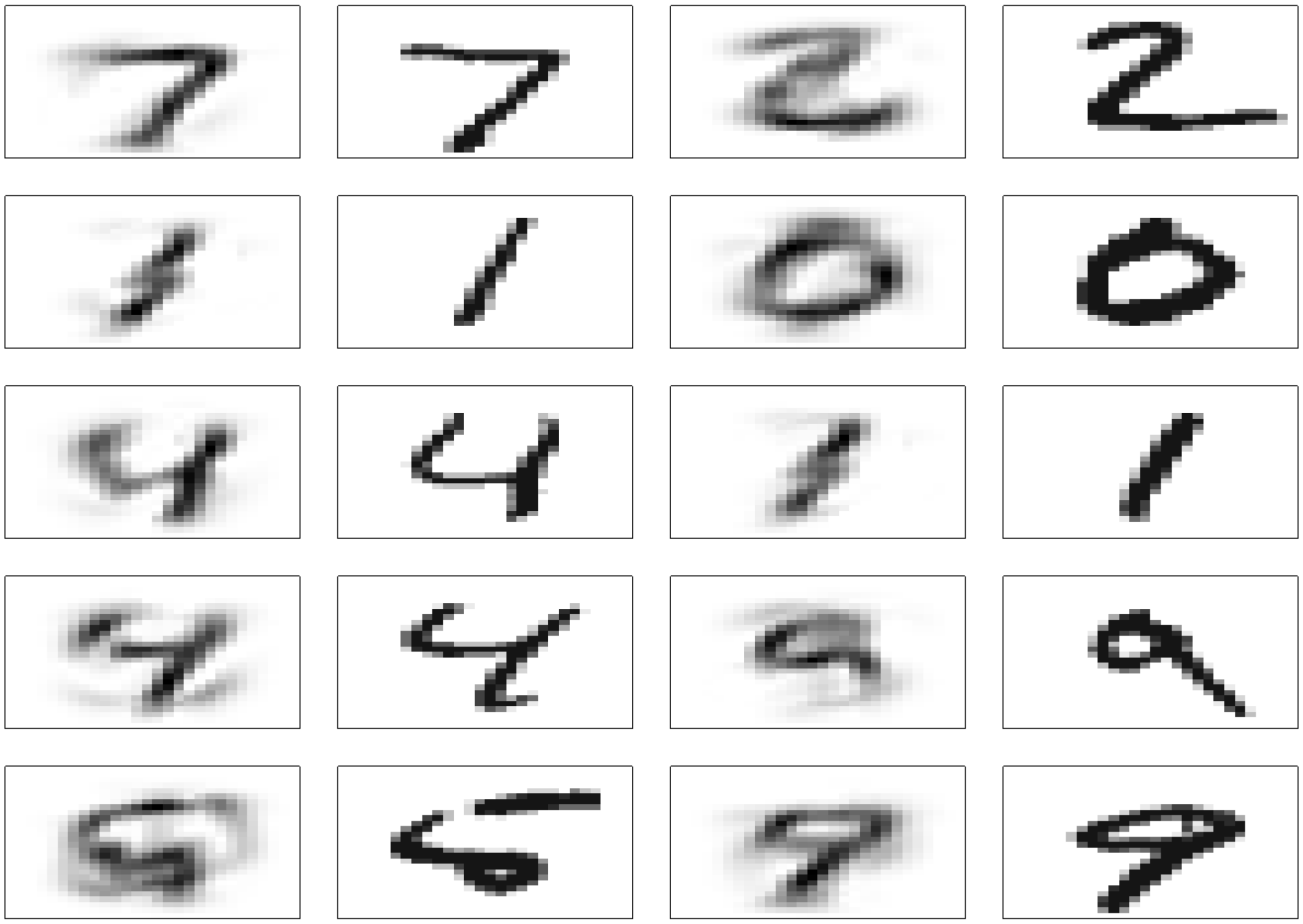
```r
par(mfrow=c(5,5), omi=c(0,0,0,0), mai=c(0.1,0.1,0.1,0.1))

for(i in 1:25){
    show_digit(t(matrix(nmfres$h[i,])), xaxt="n", yaxt="n")
}
```



The eigenvectors in PCA can have either postivie or negative coefficients. So in the images, each eigenvector will focus on different parts by scoring positive or negative on those different features. However, in NMF, the coefficients in the linear combination must be non-negative, and the reuslt is a additive combination of baiss parts to reconstruct the whole.

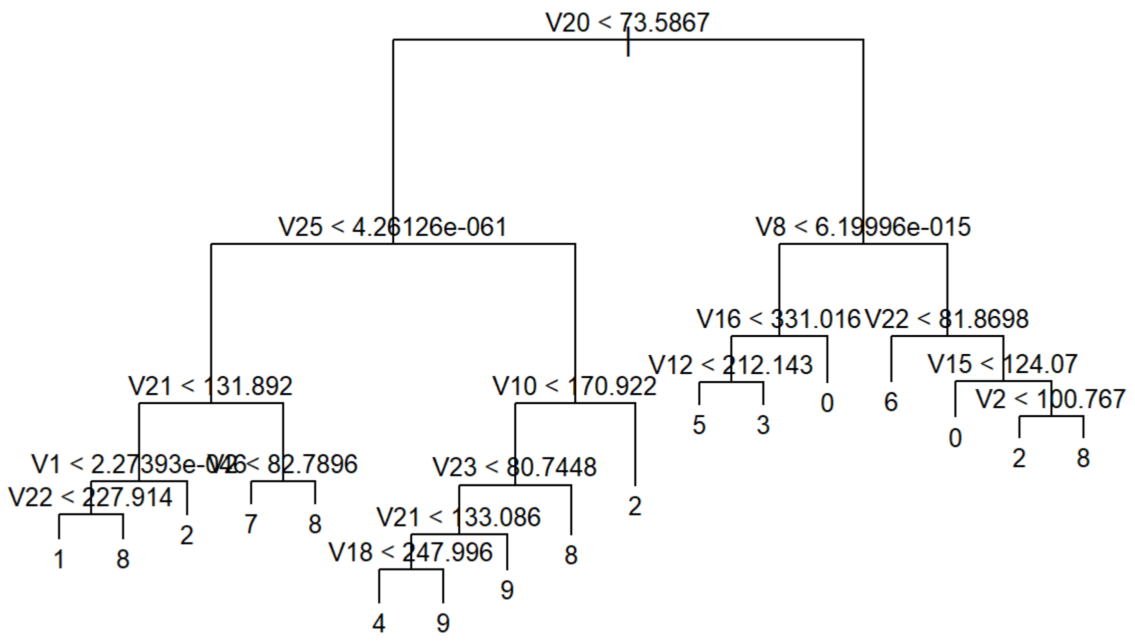
f.

```r
par(mfrow=c(5,4), omi=c(0,0,0,0), mai=c(0.1,0.1,0.1,0.1))

i = 25
for(j in 1:10){
  # Plot first 10 digits from reconstructions
  recon <- nmfres$w[j,1:i]%*%nmfres$h[1:i,]
  recon <- as.vector(recon)
  recon<- matrix(recon, nrow = 28, ncol=28, byrow=TRUE)
  image( -t(recon), col=gray((0:255)/255), xaxt="n", yaxt="n")
  # Plot original first 10 digits
  show_digit(test[j, ],xaxt="n", yaxt="n")
}
```



g.

```
preds <- nmfres$w
preds <- as.data.frame(preds)
preds <-cbind(preds, test[,785])
colnames(preds)[26]<- "labels"
preds$labels<- as.factor(preds$labels)

library(tree)
tree_fit <- tree(labels~., data=preds)
plot(tree_fit)
text(tree_fit,pretty=0,cex=0.8)
```



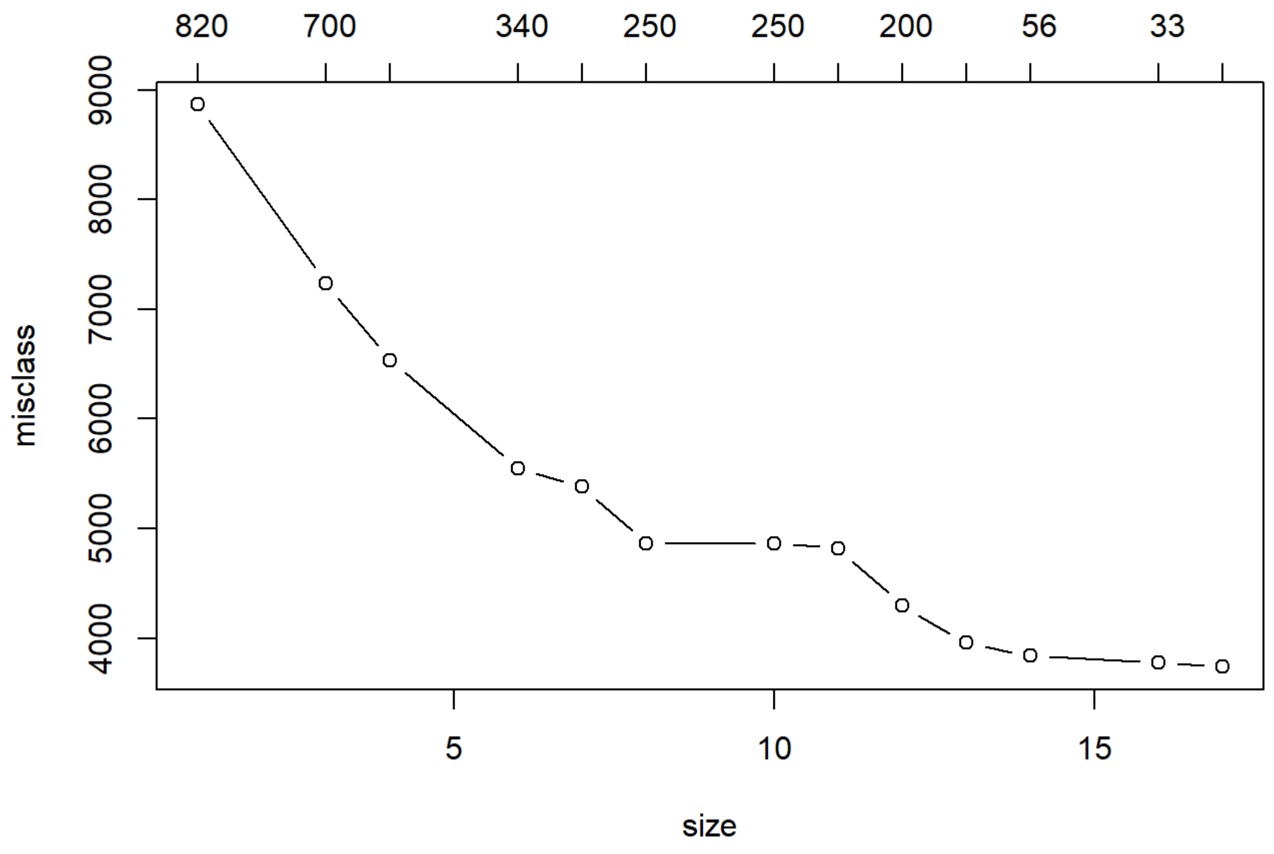
```
summary(tree_fit)
```

```
##
## Classification tree:
## tree(formula = labels ~ ., data = preds)
## Variables actually used in tree construction:
##  [1] "V20" "V25" "V21" "V1"  "V22" "V2"  "V10" "V23" "V18" "V8"  "V16"
## [12] "V12" "V15"
## Number of terminal nodes:  17
## Residual mean deviance:  2.287 = 22830 / 9983
## Misclassification error rate: 0.3606 = 3606 / 10000
```

```
tree_prune <- cv.tree(tree_fit, FUN = prune.misclass)
tree_prune
```

```
## $size
##  [1] 17 16 14 13 12 11 10  8  7  6  4  3  1
## 
## $dev
##  [1] 3739 3775 3838 3956 4296 4817 4865 4865 5381 5549 6534 7235 8865
## 
## $k
##  [1]  -Inf  33.0  55.5 102.0 195.0 242.0 248.0 248.5 297.0 344.0 427.5
## [12] 703.0 816.0
## 
## $method
## [1] "misclass"
## 
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(tree_prune, type="b")
```



Based on the pruned tree, it shows no nodes are removed.

And the misclassification rate is:

```
min(tree_prune$dev)/10000
```

```
## [1] 0.3739
```