Exercise 10.7

Section 23 - Group 6 Project Groups (Bosan Hsu, Fan Liu, Jimeng Yin, Michael Liu, Richard Wang, Zhuoqian Zhang

(a) Load in the data using read.csv(). You will need to select header=F.

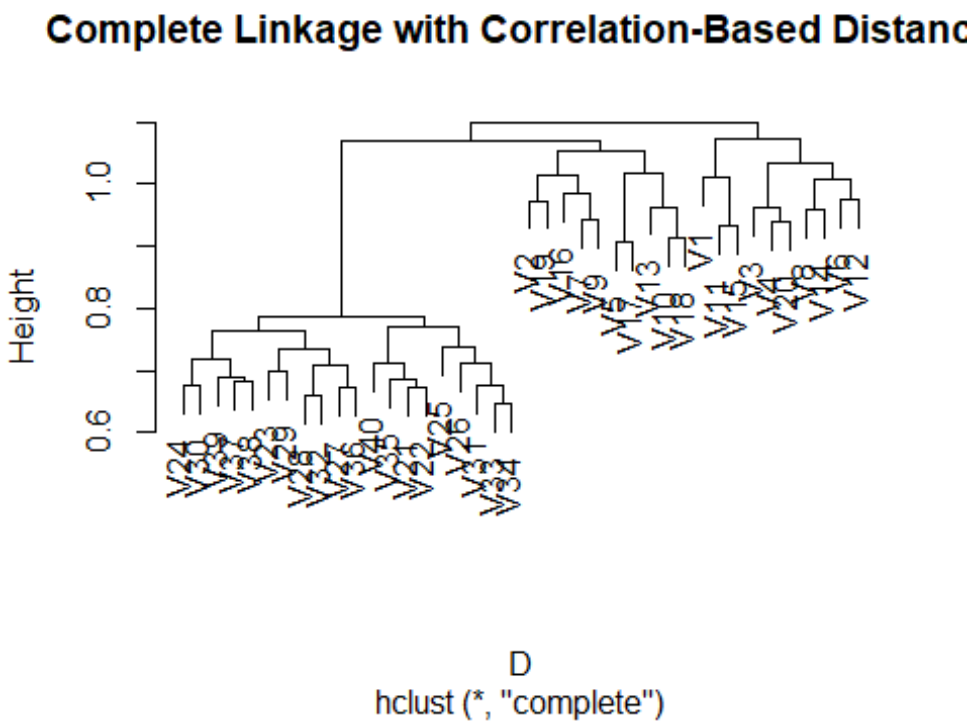
```
data = read.csv("C:\\Users\\LXQMI\\Downloads\\Ch10Ex11.csv", header = F
ALSE)
head(data, 5)
```

```
##           V1         V2         V3         V4         V5         V6
         V7
## 1 -0.9619334  0.4418028 -0.9750051  1.4175040  0.8188148  0.3162937
-0.02496682
## 2 -0.2925257 -1.1392670  0.1958370 -1.2811210 -0.2514393  2.5119970
-0.92220620
## 3  0.2587882 -0.9728448  0.5884858 -0.8002581 -1.8203980 -2.0589240
-0.06476437
## 4 -1.1521320 -2.2131680 -0.8615249  0.6309253  0.9517719 -1.1657240
-0.39155860
## 5  0.1957828  0.5933059  0.2829921  0.2471472  1.9786680 -0.8710180
-0.98971500
##            V8          V9        V10        V11        V12         V
13
## 1 -0.06396600  0.03149702 -0.3503106 -0.7227299 -0.2819547  1.337515
00
## 2  0.05954277 -1.40964500 -0.6567122 -0.1157652  0.8259783  0.346449
60
## 3  1.59212400 -0.17311700 -0.1210874 -0.1875790 -1.5001630 -1.228737
00
## 4  1.06361900 -0.35000900 -1.4890580 -0.2432189 -0.4330340 -0.038791
28
## 5 -1.03225300 -1.10965400 -0.3851423  1.6509570 -1.7449090 -0.378885
30
##           V14        V15        V16        V17        V18        V19
## 1  0.70197980  1.0076160 -0.4653828  0.6385951  0.2867807 -0.2270782
## 2 -0.56954860 -0.1315365  0.6902290 -0.9090382  1.3026420 -1.6726950
## 3  0.85598900  1.2498550 -0.8980815  0.8702058 -0.2252529  0.4502892
## 4 -0.05789677 -1.3977620 -0.1561871 -2.7359820  0.7756169  0.6141562
## 5 -0.67982610 -2.1315840 -0.2301718  0.4661243 -1.8004490  0.6262904
##           V20        V21        V22        V23        V24         V2
5
## 1 -0.22004520 -1.2425730 -0.1085056 -1.8642620 -0.5005122 -1.3250080
0
## 2 -0.52550400  0.7979700 -0.6897930  0.8995305  0.4285812 -0.6761141
0
## 3  0.55144040  0.1462943  0.1297400  1.3042290 -1.6619080 -1.6303760
```

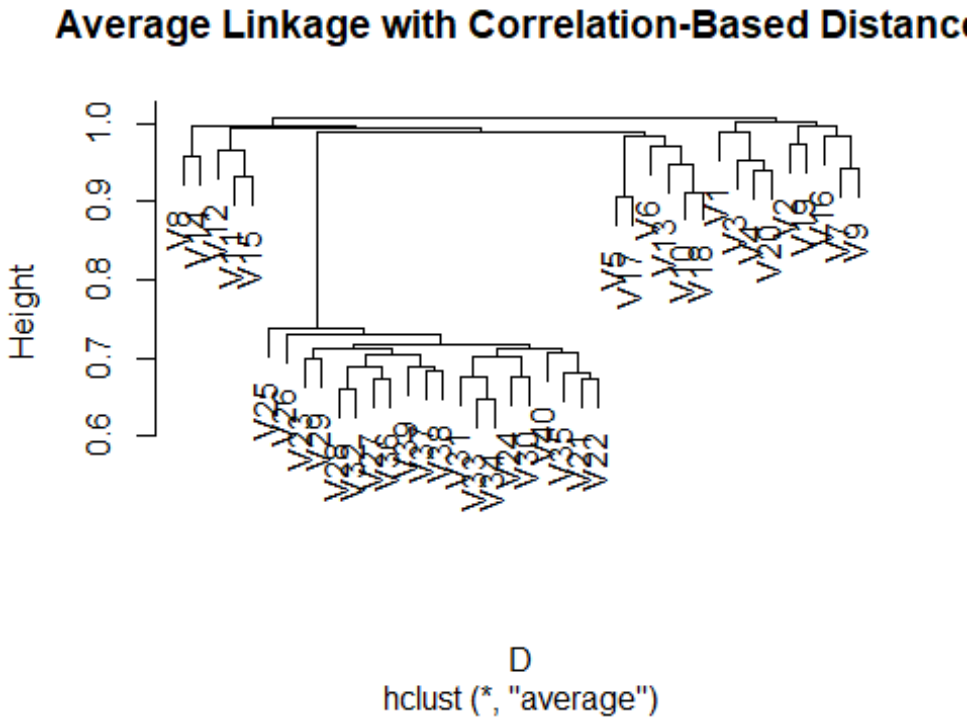
```
0
## 4  2.01919400  1.0811390 -1.0766180 -0.2434181  0.5134822 -0.5128578
0
## 5 -0.09772305 -0.2997108 -0.5295591 -2.0235670 -0.5108402  0.0460027
4
##           V26        V27        V28         V29         V30        V
31
## 1  1.06341100 -0.2963712 -0.1216457  0.08516605  0.62417640 -0.50959
15
## 2 -0.53409490 -1.7325070 -1.6034470 -1.08362000  0.03342185  1.70070
80
## 3 -0.07742528  1.3061820  0.7926002  1.55946500 -0.68851160 -0.61547
20
## 4  2.55167600 -2.3143010 -1.2764700 -1.22927100  1.43439600 -0.28427
74
## 5  1.26803000 -0.7439868  0.2231319  0.85846280  0.27472610 -0.69299
84
##            V32         V33        V34        V35        V36
V37
## 1 -0.216725500 -0.05550597 -0.4844491 -0.5215811  1.9491350  1.32433
500
## 2  0.007289556  0.09906234  0.5638533 -0.2572752 -0.5817805 -0.16988
710
## 3  0.009999363  0.94581000 -0.3185212 -0.1178895  0.6213662 -0.07076
396
## 4  0.198945600 -0.09183320  0.3496279 -0.2989097  1.5136960  0.67118
470
## 5 -0.845707200 -0.17749680 -0.1664908  1.4831550 -1.6879460 -0.14142
960
##          V38         V39        V40
## 1  0.4681471  1.06110000  1.6559700
## 2 -0.5423036  0.31293890 -1.2843770
## 3  0.4016818 -0.01622713 -0.5265532
## 4  0.0108553 -1.04368900  1.6252750
## 5  0.2007785 -0.67594210  2.2206110
```

(b) Apply hierarchical clustering to the samples using correlation-based distance, and plot the dendrogram. Do the genes separate the samples into the two groups? Do your results depend on the type of linkage used?

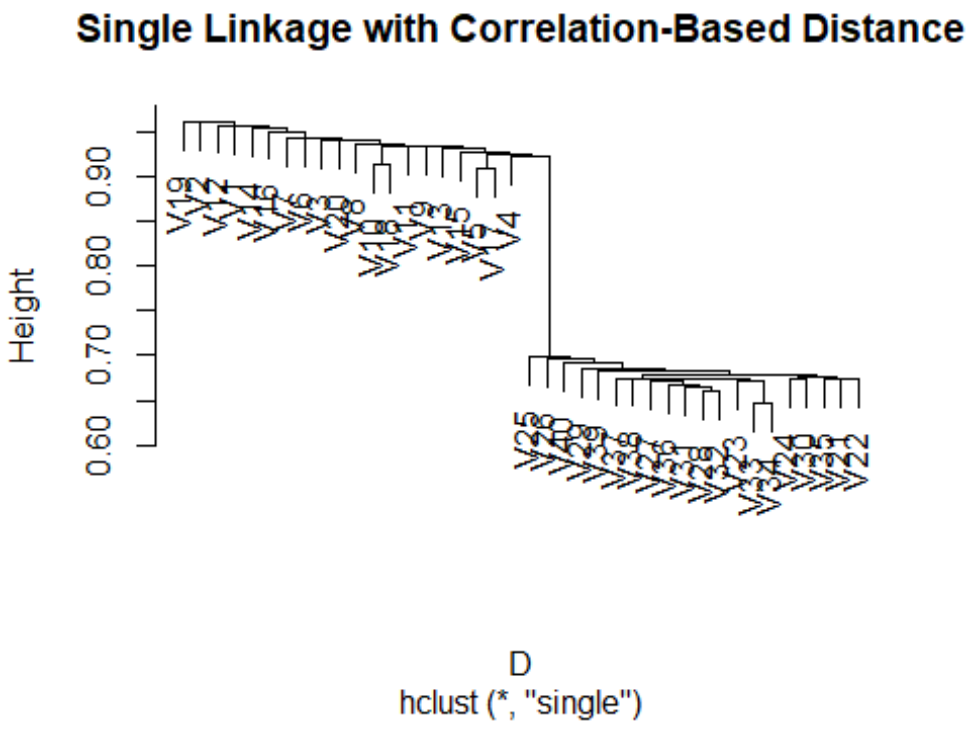
```
D = as.dist(1 - cor(data))
plot(hclust(D, method = "complete"), main = "Complete Linkage with Corr
elation-Based Distance")
```




```
plot(hclust(D, method = "average"), main = "Average Linkage with Correl
ation-Based Distance")
```



```
plot(hclust(D, method = "single"), main = "Single Linkage with Correlat
ion-Based Distance")
```




```
table(predicted = cutree(hclust(D, method = "complete"), k=2), truth=c
(rep(1,20), rep(2,20)))
```

```
##          truth
## predicted  1  2
##         1 10  0
##         2 10 20
```

```
table(predicted = cutree(hclust(D, method = "single"), k=2), truth=c(re
p(1,20), rep(2,20)))
```

```
##          truth
## predicted  1  2
##         1 19 20
##         2  1  0
```

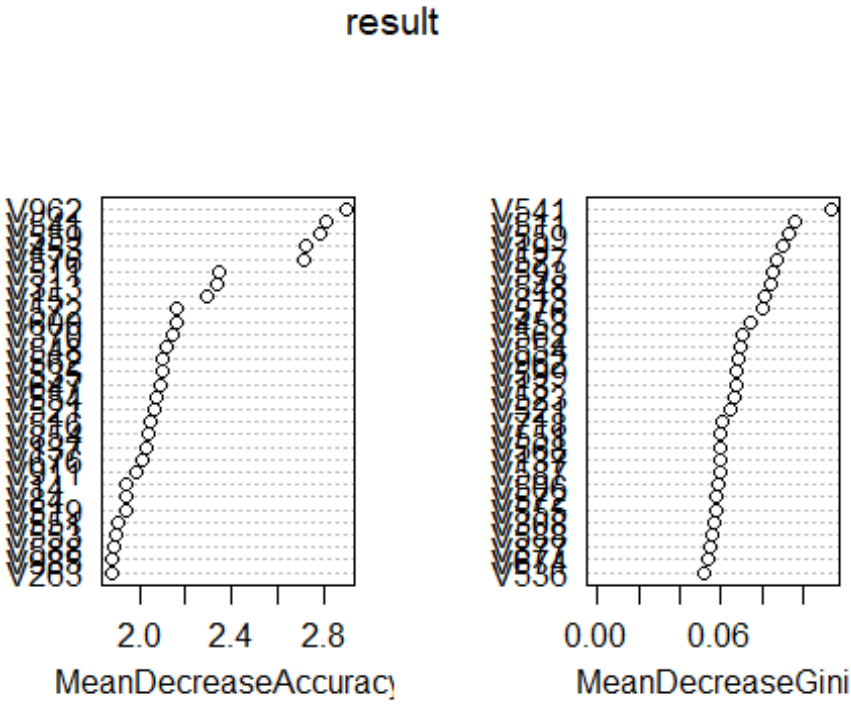
```
table(predicted = cutree(hclust(D, method = "average"), k=2), truth=c(r
ep(1,20), rep(2,20)))
```

```
##          truth
## predicted  1  2
##         1  9  0
##         2 11 20
```

##Complete linkage separate the samples into the two groups.The results depend on the type of linkage used.

(c) Your collaborator wants to know which genes differ the most across the two groups. Suggest a way to answer this question, and apply it here.

```
predicted = cutree(hclust(D, method = "complete"), k=2)
DF = t(data)
DF_1 = DF[predicted==1,]
DF_1 = cbind(DF_1, predict_cluster="one")
DF_2 = DF[predicted==2,]
DF_2 = cbind(DF_2, predict_cluster="two")
DF_labeled = rbind(DF_1, DF_2)
View((DF_labeled))

library(randomForest)

## Warning: 程辑包'randomForest'是用 R 版本 4.3.2 来建造的

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

class(DF_labeled)

## [1] "matrix" "array"

DF_labeled = as.data.frame(DF_labeled)
DF_labeled$predict_cluster = as.factor(DF_labeled$predict_cluster)
result = randomForest(predict_cluster~., data = DF_labeled, mtry = 5, n
tree = 1000, importance = TRUE)
varImpPlot(result)
```



##We first use complete linkage to perform hierarchical clustering. Then we used to cut the hierarchical clustering dendrogram into two clusters. We build a random forest model for classification and create a variable importance plot to show the importance of each predictor variable in the random forest model.From the variable importance plot, Variables that appear at the top of both plots are typically considered to differ the most important for the model.