ISLR Chapter 10
Ex. 10 & 11
Tatiana Romanchishina

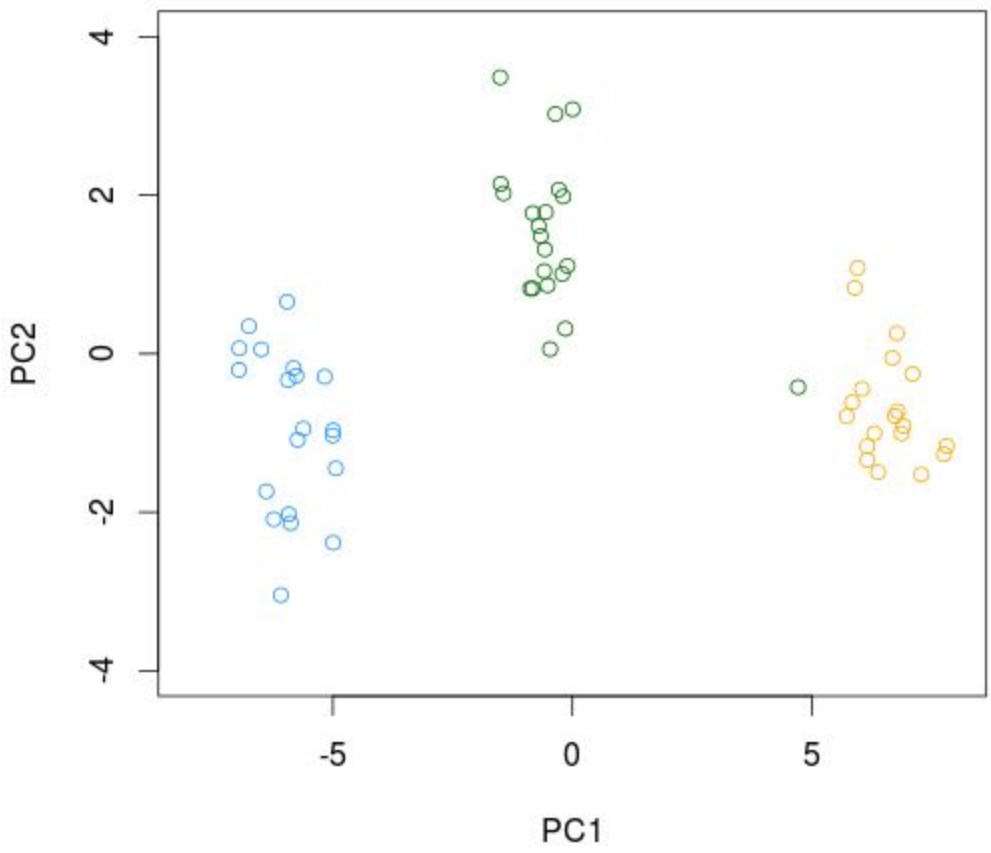
10. In this problem, you will generate simulated data, and then perform PCA and K-means clustering on the data.

(a) Generate a simulated data set with 20 observations in each of three classes (i.e. 60 observations total), and 50 variables. Hint: There are a number of functions in R that you can use to generate data. One example is the rnorm() function; runif() is another option. Be sure to add a mean shift to the observations in each class so that there are three distinct classes.

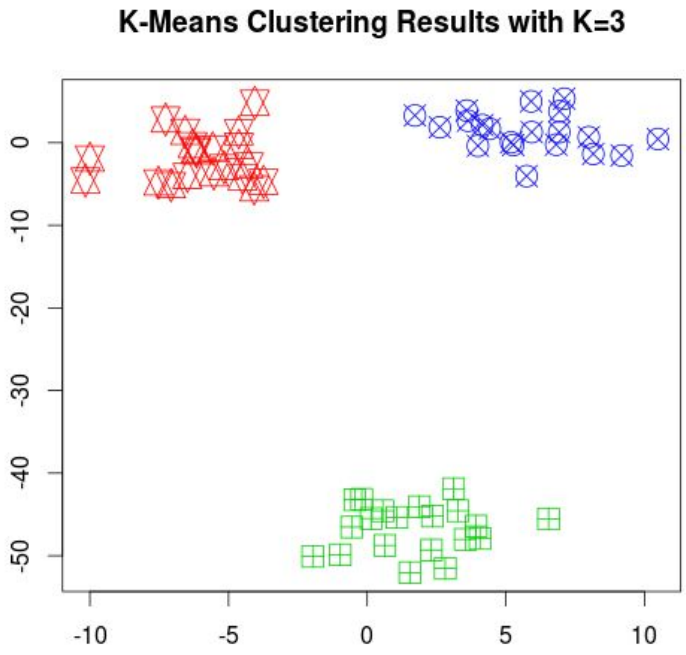
```
library(clusterGeneration)
tmp<-simClustDesign(numClust=3,
                    sepVal=.7,
                    numNonNoisy=50,
                    numNoisy = 0,
                    numReplicate=1,
                    clustszind = 1,
                    clustSizeEq = 20,
                    outputDatFlag = F)
```

(b) Perform PCA on the 60 observations and plot the first two principal component score vectors. Use a different color to indicate the observations in each of the three classes. If the three classes appear separated in this plot, then continue on to part (c). If not, then return to part (a) and modify the simulation so that there is greater separation between the three classes. Do not continue to part (c) until the three classes show at least some separation in the first two principal component score vectors.

As the following chart indicates, there are three distinct sub-groups in the data set:



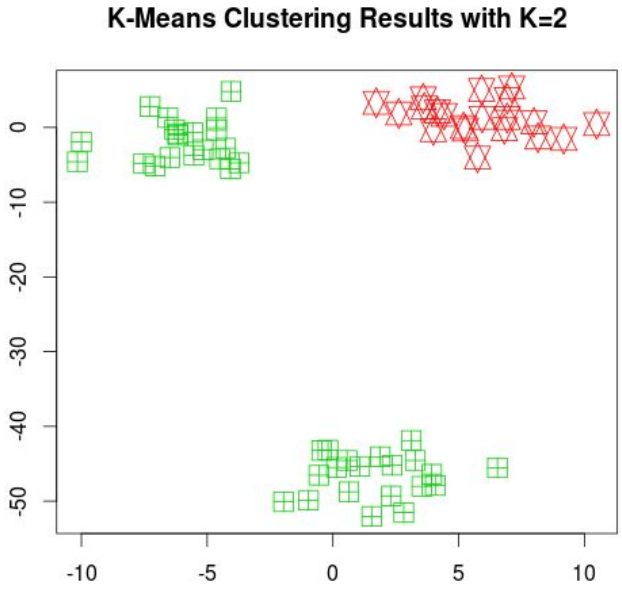
(c) Perform K-means clustering of the observations with K = 3. How well do the clusters that you obtained in K-means clustering compare to the true class labels? Hint: You can use the table() function in R to compare the true class labels to the class labels obtained by clustering. Be careful how you interpret the results: K-means clustering will arbitrarily number the clusters, so you cannot simply check whether the true class labels and clustering labels are the same.



(d) Perform K-means clustering with K = 2. Describe your results.

> km.out2$cluster

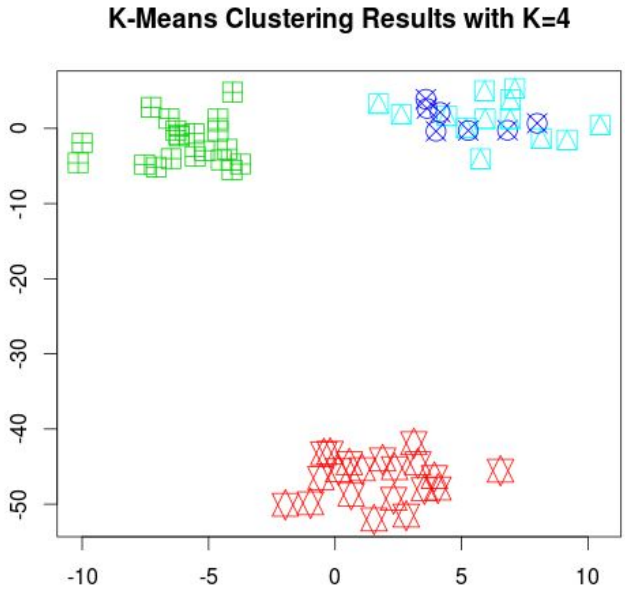
[1] 1 1 2 1 1 1 1 1 1 1 1 2 1 2 2 2 1 1 1 1 1 2 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1 1 2 1 2 2 1 2 1 1 2 1 2 1 2 2 2 1 1 1 2 2 2 1 1 1 1



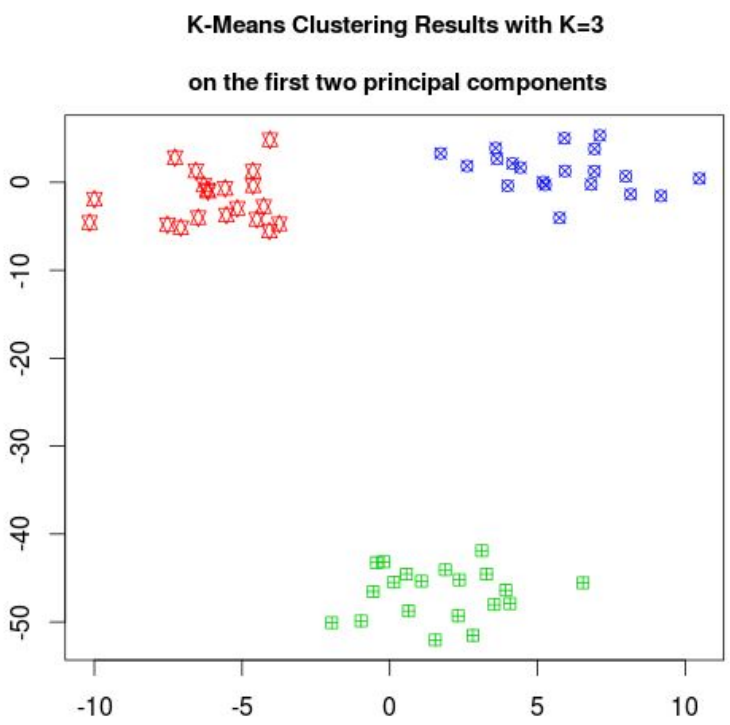
(e) Now perform K-means clustering with K = 4, and describe your results.

> km.out4$cluster

[1] 3 3 4 1 3 1 3 3 1 3 3 2 1 4 4 4 1 3 1 3 1 4 3 3 3 1 4 1 1 4 1 3 3 3 1 2 1 4 2 3 2 1 1 4 1 2 3 4 4 4 1 3 1 2 4 2 1 3 1 3



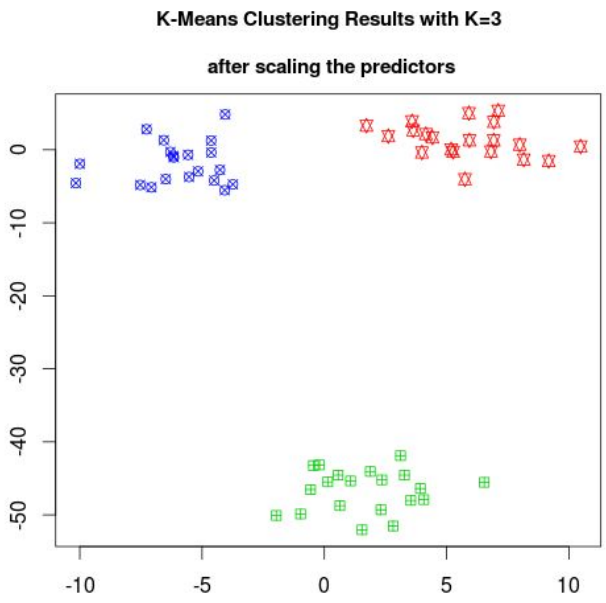
(f) Now perform K-means clustering with K = 3 on the first two principal component score vectors, rather than on the raw data. That is, perform K-means clustering on the 60 × 2 matrix of which the first column is the first principal component score vector, and the second column is the second principal component score vector. Comment on the results.



(g) Using the scale() function, perform K-means clustering with K = 3 on the data after scaling each variable to have standard deviation one. How do these results compare to those obtained in (b)? Explain.

> km.out.sc$cluster
[1] 3 3 1 2 3 2 3 3 2 3 3 1 2 1 1 1 2 3 2 3 2 1 3 3 3 2 1 2 2 1 2 3 3 3 2 1 2 1 1 3 1 2 2 1 2 1 3 1 1 1 2 3 2 1 1 1 2 3 2 3

The following chart shows the result of K-means clustering with k = 3 after scaling each predictor. The result is the same as in b).



11. On the book website, www.StatLearning.com, there is a gene expression data set (Ch10Ex11.csv) that consists of 40 tissue samples with measurements on 1,000 genes. The first 20 samples are from healthy patients, while the second 20 are from a diseased group.

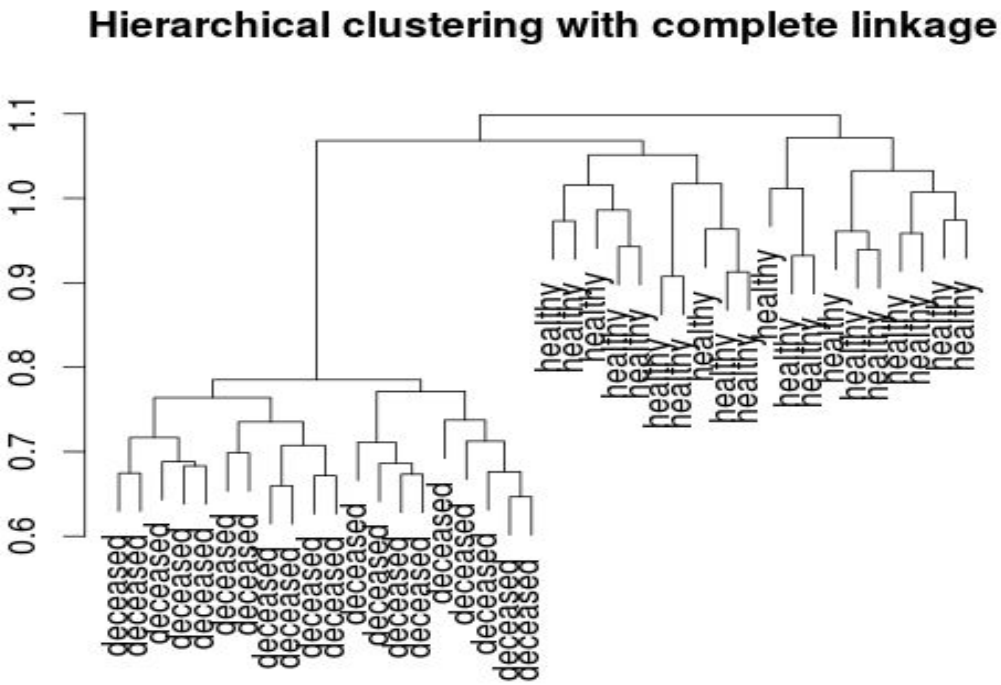
(a) Load in the data using read.csv() . You will need to select header=F .

```
data <- read.csv("Downloads/Ch10Ex11.csv", header=F)

str(data)
        'data.frame':    1000 obs. of  40 variables:
         $ V1 : num  -0.962 -0.293 0.259 -1.152 0.196 ...
         $ V2 : num  0.442 -1.139 -0.973 -2.213 0.593 ...
         $ V3 : num  -0.975 0.196 0.588 -0.862 0.283 ...
         $ V4 : num  1.418 -1.281 -0.8 0.631 0.247 …
        …

dim(data)
        [1] 1000   40
```

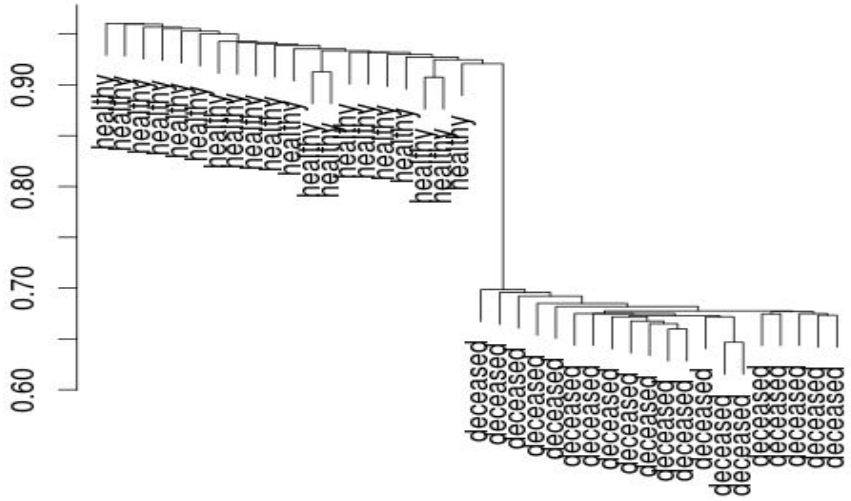
(b) Apply hierarchical clustering to the samples using correlation-based distance, and plot the dendrogram. Do the genes separate the samples into the two groups? Do your results depend on the type of linkage used?

The genes were separated into “healthy” and “deceased” with all three linkage methods, but the complete linkage produced a “cleaner” dendrogram.

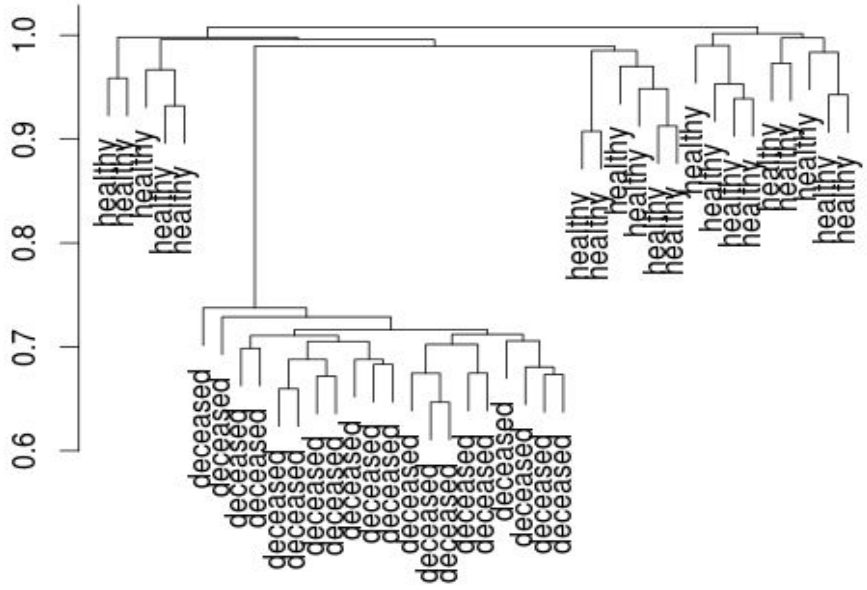
Single linkage yielded a lot of “trailing” clusters as expected, but complete and average linkage produced quite balanced plots:



Hierarchical clustering with single linkage



Hierarchical clustering with average linkage



(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.

I will use PCA to determine which genes differ the most. This could be done by first transposing the dataset, so that the genes are columns, instead of rows.
The rotation matrix provides the principal component loadings; each column of pr.out$rotation contains the corresponding principal component loading vector.