HW 10

Runze Wang

1. Question 12.6.2 on Page 548 in ISLRv2.

a

b

c Suppose that we cut the dendrogram obtained in (a) such that two clusters result. Which observations are in each cluster?

we have cluster (1,2) and (3,4)

d Suppose that we cut the dendrogram obtained in (b) such that two clusters result. Which observations are in each cluster?

we have cluster (1,2,3) and (4).

e It is mentioned in the chapter that at each fusion in the dendrogram, the position of the two clusters being fused can be swapped without changing the meaning of the dendrogram. Draw a dendrogram that is equivalent to the dendrogram in (a), for which two or more of the leaves are repositioned, but for which the meaning of the dendrogram is the same.

$$\begin{pmatrix} & 0.3 & 0.4 & 0.7 \\ 0.3 & & 0.5 & 0.8 \\ 0.4 & 0.5 & & 0.45 \\ 0.7 & 0.8 & 0.45 & \end{pmatrix}$$

dissimilarity matrix for 0.3 is minimum dissimilarity

$$\begin{pmatrix} & 0.5 & 0.8 \\ 0.5 & & 0.45 \\ 0.8 & 0.45 & \end{pmatrix}$$

dissimilarity matrix for 0.45 is minimum dissimilarity

$$\begin{pmatrix} & 0.8 \\ 0.8 & \end{pmatrix}$$

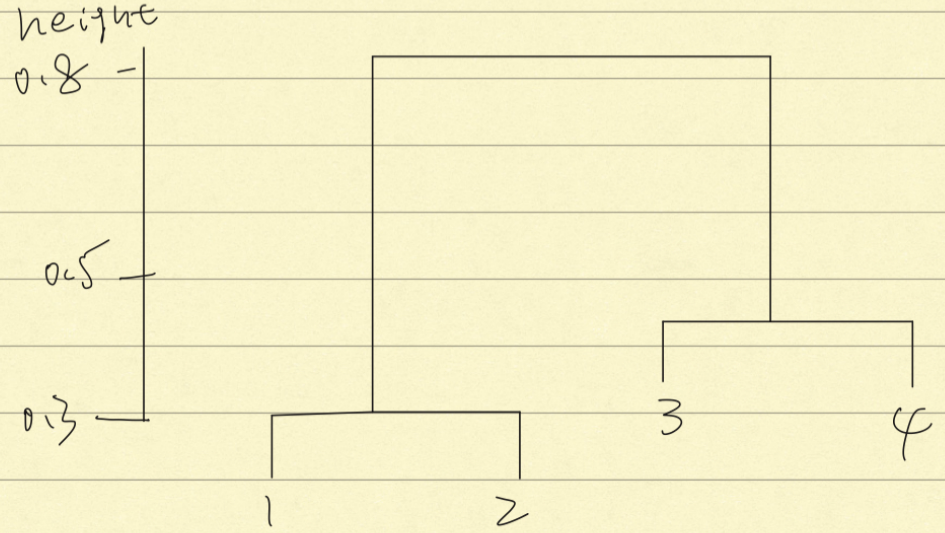


Figure 1: a

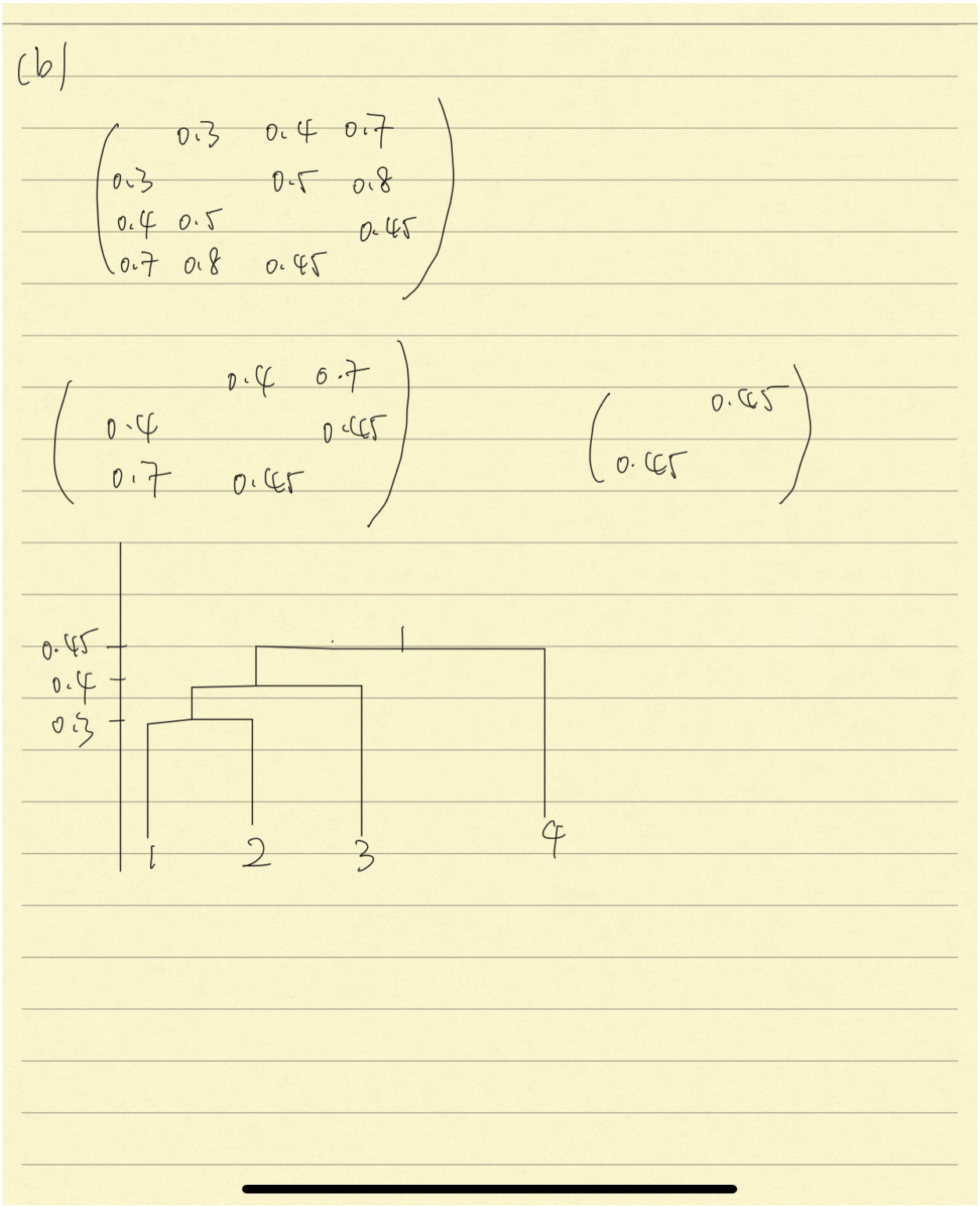

(b)
0.3 0.4 0.7
0.3 0.5 0.8
0.4 0.5 0.45
0.7 0.8 0.45
0.4 0.7
0.4 0.45
0.7 0.45
0.55
0.55
0.45
0.4
0.3
1
2
3
4

Figure 2: b

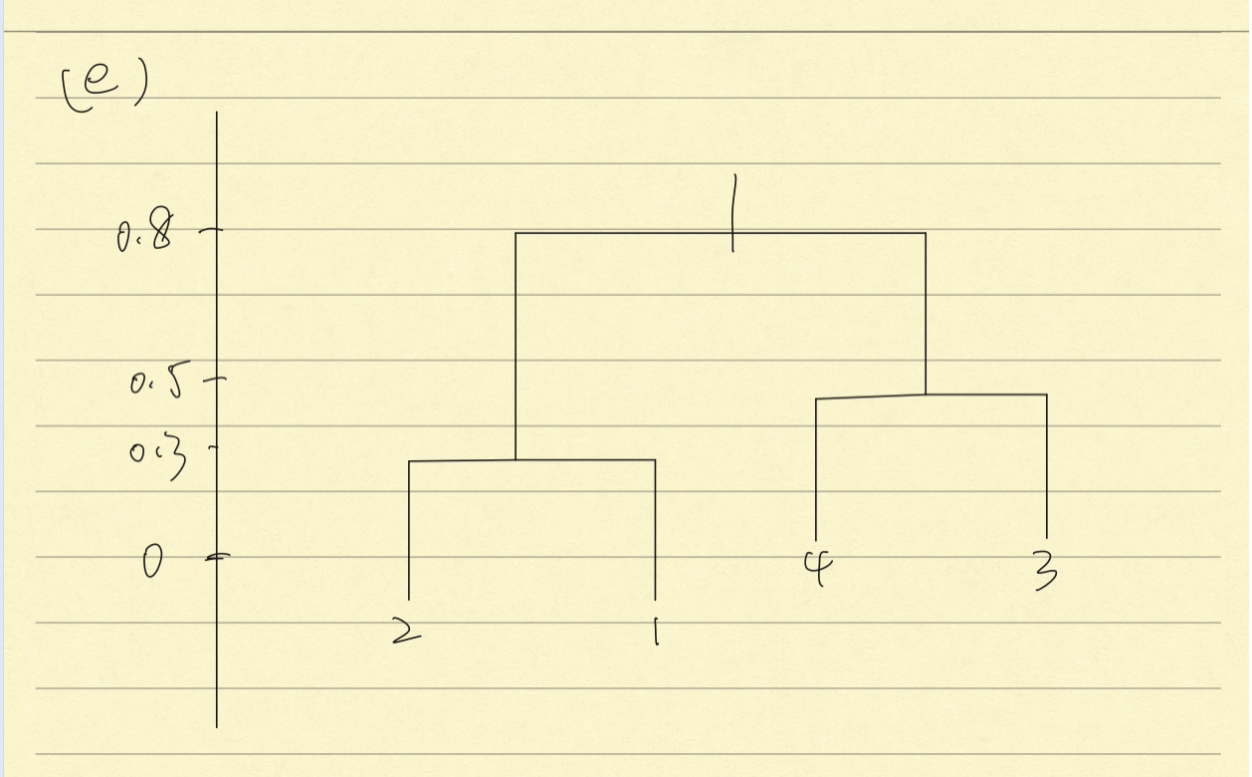
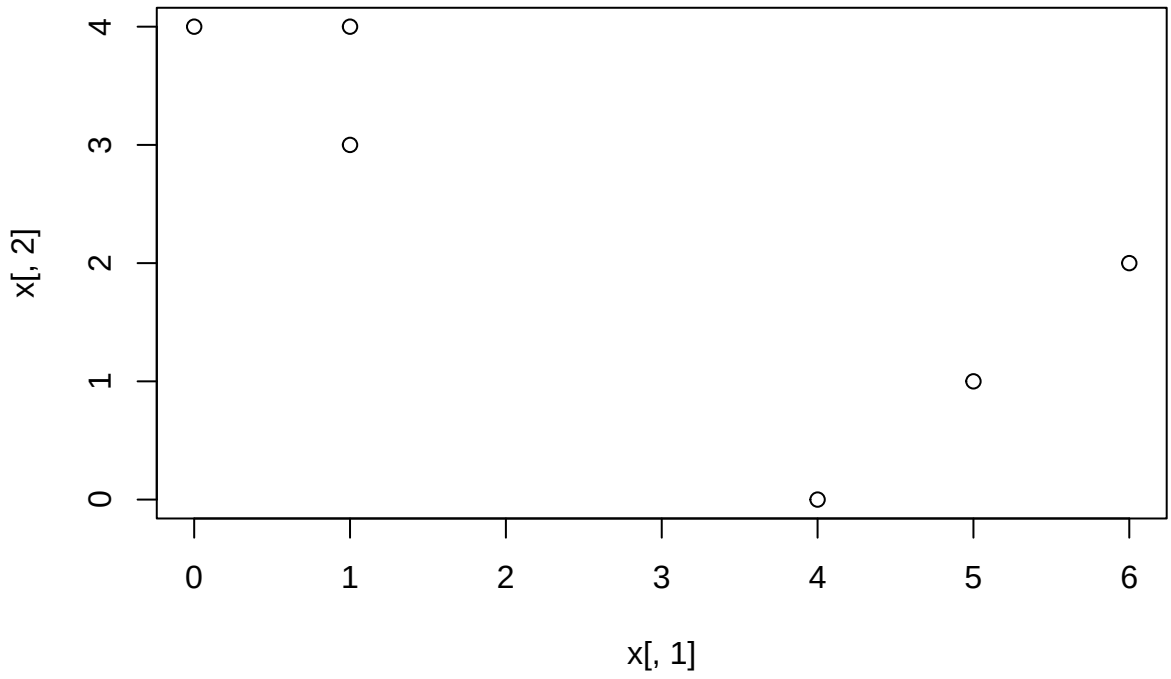


Figure 3: e

3. In this problem, you will perform K-means clustering manually, with K = 2, on a small example with n = 6 observations and p = 2 features. The observations are as follows.

a

```
x <- cbind(c(1, 1, 0, 5, 6, 4), c(4, 3, 4, 1, 2, 0))
plot(x[,1], x[,2])
```

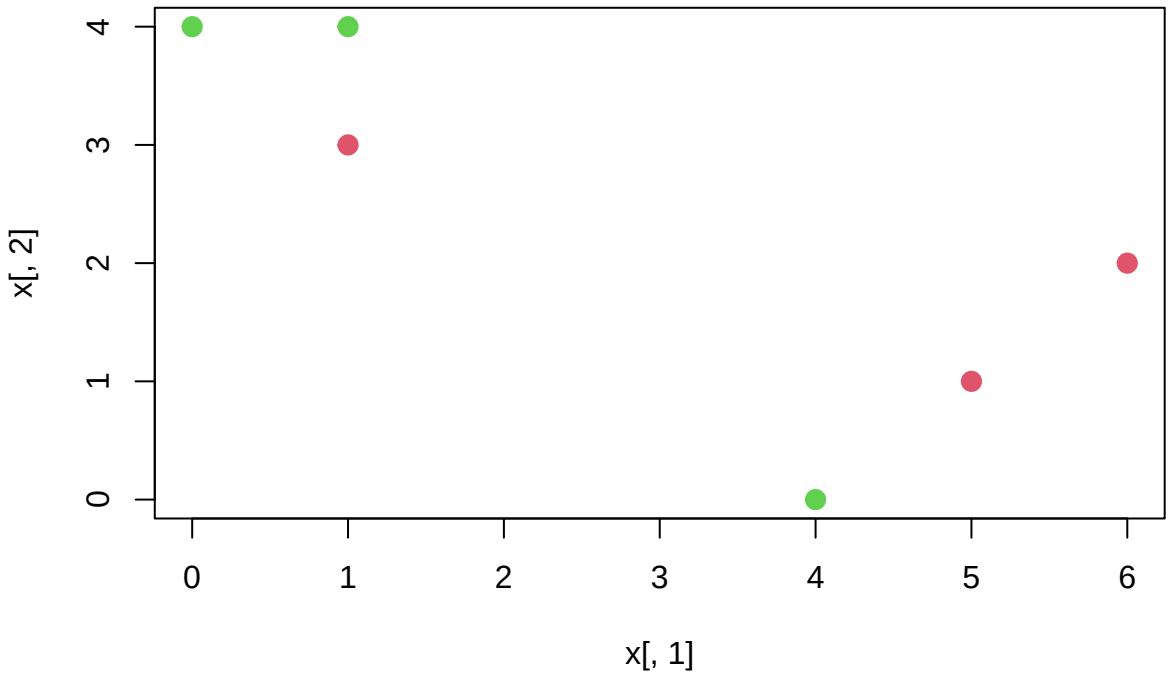


(b) Randomly assign a cluster label to each observation. You can use the sample() command in R to do this. Report the cluster labels for each observation.

```
set.seed(0)
label <- sample(x = 2, size = nrow(x), replace = TRUE)
label
```

```
## [1] 2 1 2 1 1 2
```

```
plot(x[, 1], x[, 2], col = (label + 1), pch = 20, cex = 2)
```

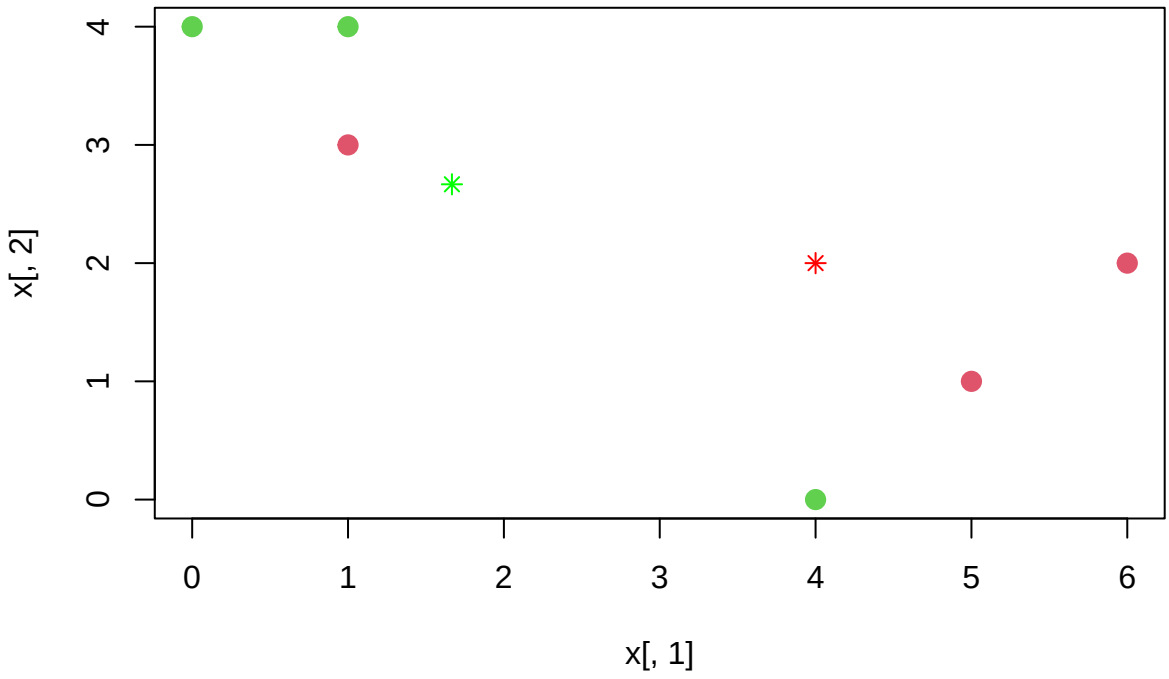


(c) Compute the centroid for each cluster.

```
centroids = aggregate(x, list(Cluster = label), mean)
centroids
```

```
##   Cluster       V1       V2
## 1       1 4.000000 2.000000
## 2       2 1.666667 2.666667
```

```
plot(x[, 1], x[, 2], col = (label + 1), pch = 20, cex = 2)

points(centroids[1,2:3], col = "red", pch = 8)
points(centroids[2,2:3], col = "green", pch = 8)
```



d Assign each observation to the centroid to which it is closest, in terms of Euclidean distance. Report the cluster labels for each observation.

```
distance <- function (x, y){
  return(sqrt((x[1] - y[1])^2 + (x[2] - y[2])^2))
}
c1 <- c(mean(x[label == 1, 1]), mean(x[label == 1, 2]))
c2 <- c(mean(x[label == 2, 1]), mean(x[label == 2, 2]))

distance(x[1,], c1)
```

```
## [1] 3.605551
```

```
distance(x[1,], c2)
```

```
## [1] 1.490712
```

```
label[1] <- 2
distance(x[2,], c1)
```

```
## [1] 3.162278
```

```
distance(x[2,], c2)
```

```
## [1] 0.745356
```

```
label[2] <- 2
distance(x[3,], c1)
```

```
## [1] 4.472136
```

```
distance(x[3,], c2)
```

```
## [1] 2.134375
```

```
label[3] <- 2
distance(x[4,], c1)
```

```
## [1] 1.414214
```

```
distance(x[4,], c2)
```

```
## [1] 3.72678
```

```
label[4] <- 2
distance(x[5,], c1)
```

```
## [1] 2
```

```
distance(x[5,], c2)
```

```
## [1] 4.384315
```

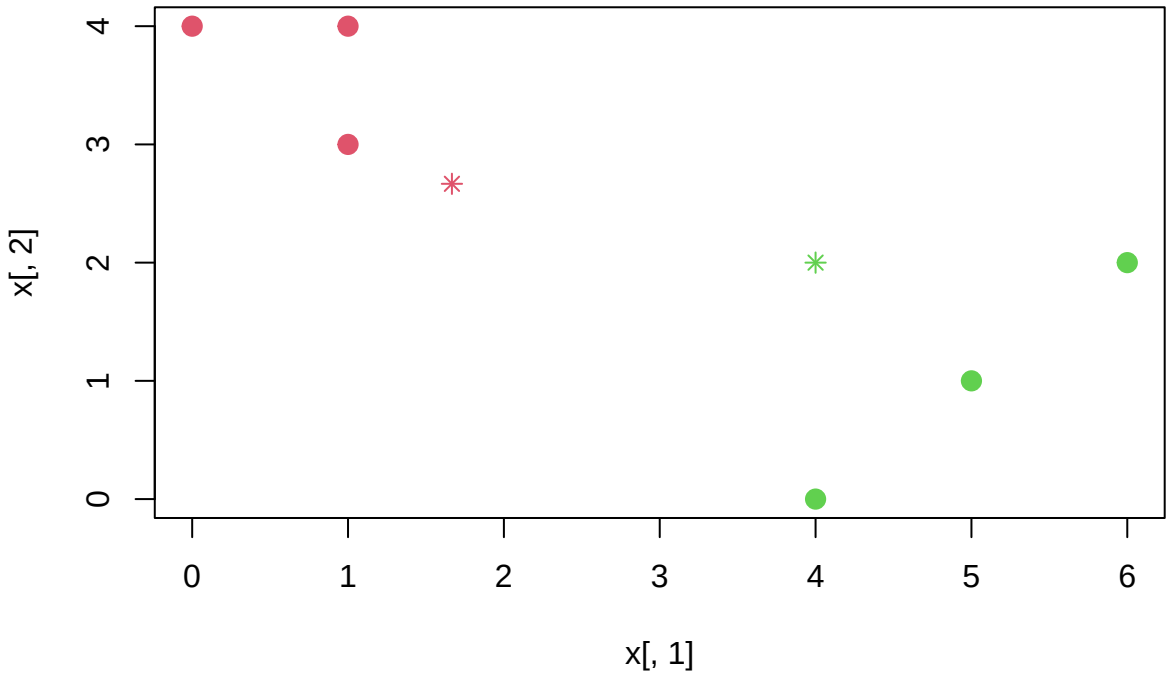
```
label[5] <- 1
distance(x[6,], c1)
```

```
## [1] 2
```

```
distance(x[6,], c2)
```

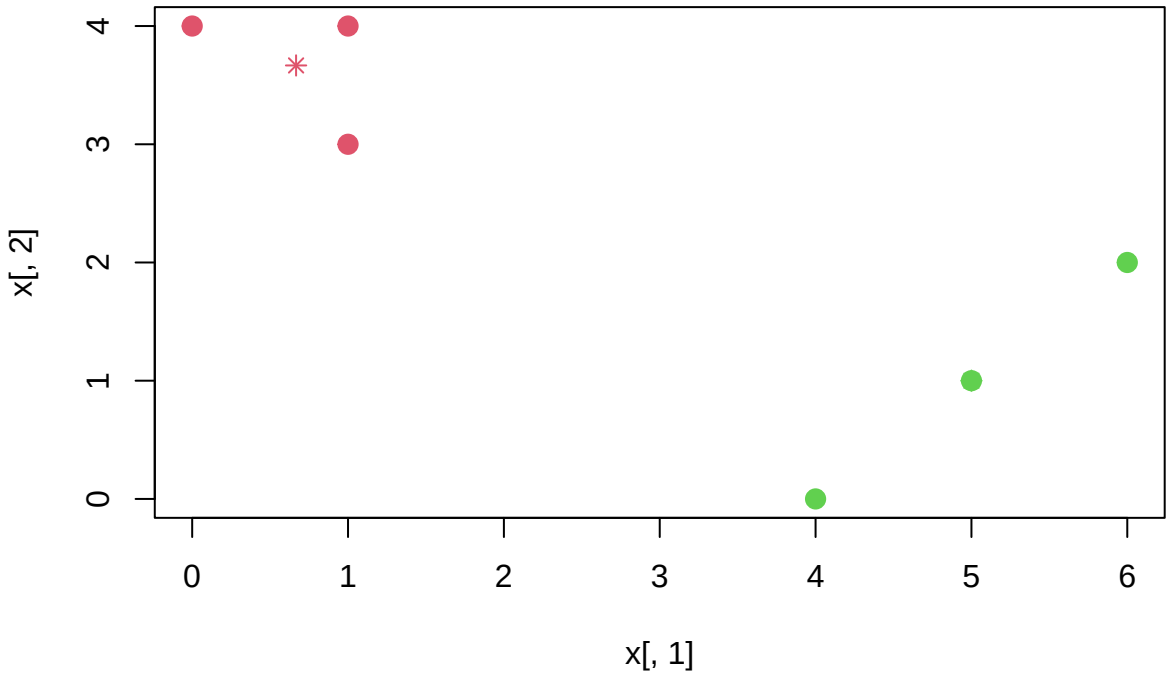
```
## [1] 3.543382
```

```
label[6] <- 1

labels <- c(1, 1, 1, 2, 2, 2)
plot(x[, 1], x[, 2], col = (labels + 1), pch = 20, cex = 2)
points(c1[1], c1[2], col = 3, pch = 8)
points(c2[1], c2[2], col = 2, pch = 8)
```



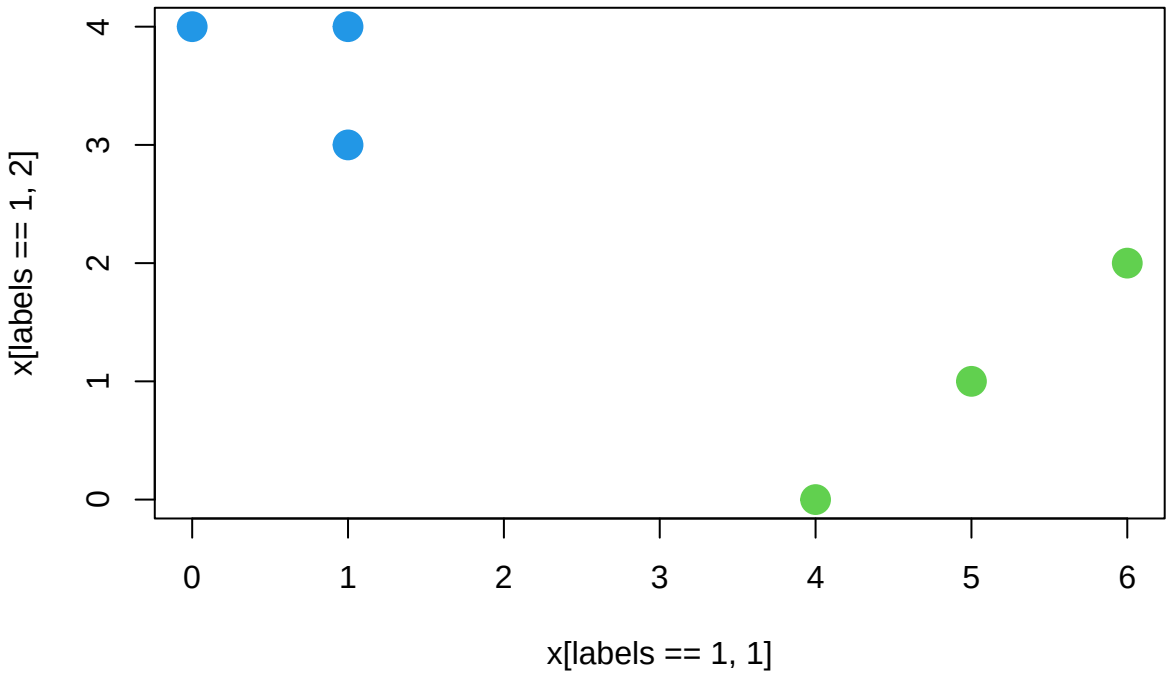
(e) Repeat (c) and (d) until the answers obtained stop changing.

```
c1 <- c(mean(x[labels == 1, 1]), mean(x[labels == 1, 2]))
c2 <- c(mean(x[labels == 2, 1]), mean(x[labels == 2, 2]))
plot(x[,1], x[,2], col=(labels + 1), pch = 20, cex = 2)
points(c1[1], c1[2], col = 2, pch = 8)
points(c2[1], c2[2], col = 3, pch = 8)
```



In your plot from (a), color the observations according to the cluster labels obtained.

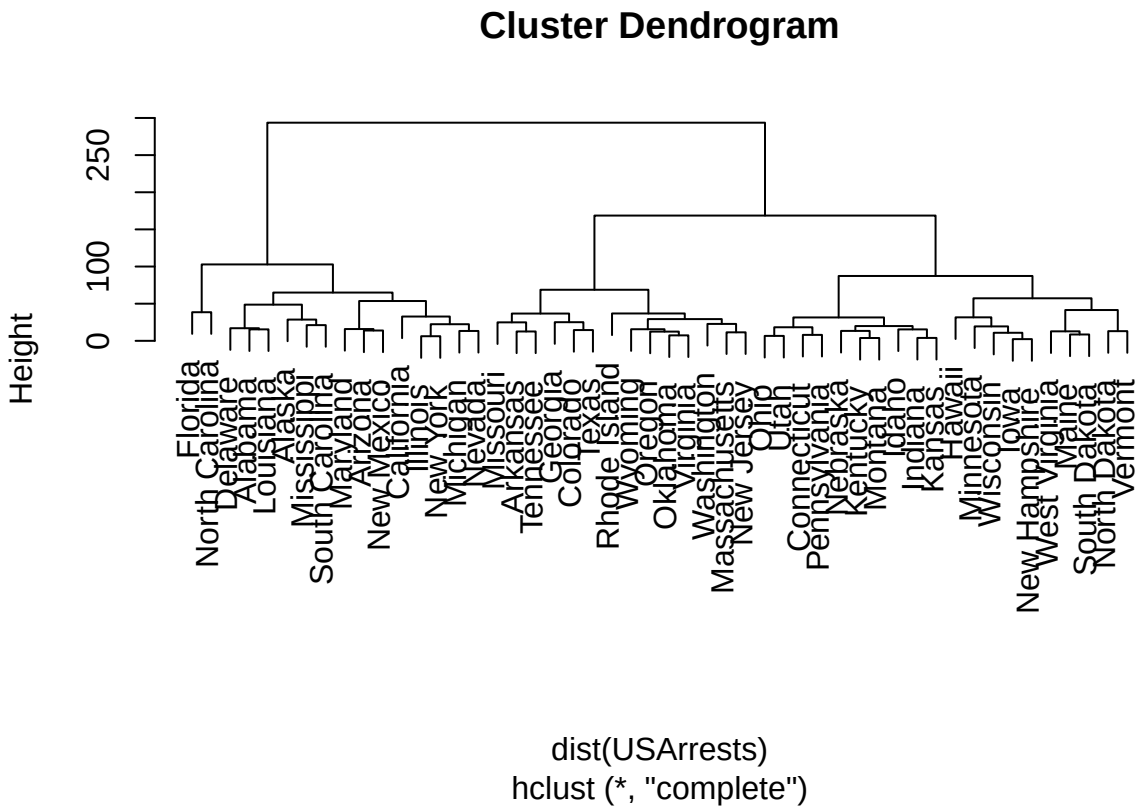
```
plot(x[labels == 1, 1], x[labels == 1, 2], col = 4, pch = 20, cex = 3,
     xlim = c(0, 6), ylim = c(0, 4))
points(x[labels == 2, 1], x[labels == 2, 2],
       col = 3, pch = 20, cex = 3)
```


x[labels == 1, 2]
0
1
2
3
4
x[labels == 1, 1]
0
1
2
3
4
5
6

3 9. Consider the USArrests data. We will now perform hierarchical clustering on the states.

(a) Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states.

```
set.seed(100)
data(USArrests)
complete_cluster = hclust(dist(USArrests), method="complete")
plot(complete_cluster)
```



b) Cut the dendrogram at a height that results in three distinct clusters. Which states belong to which clusters?

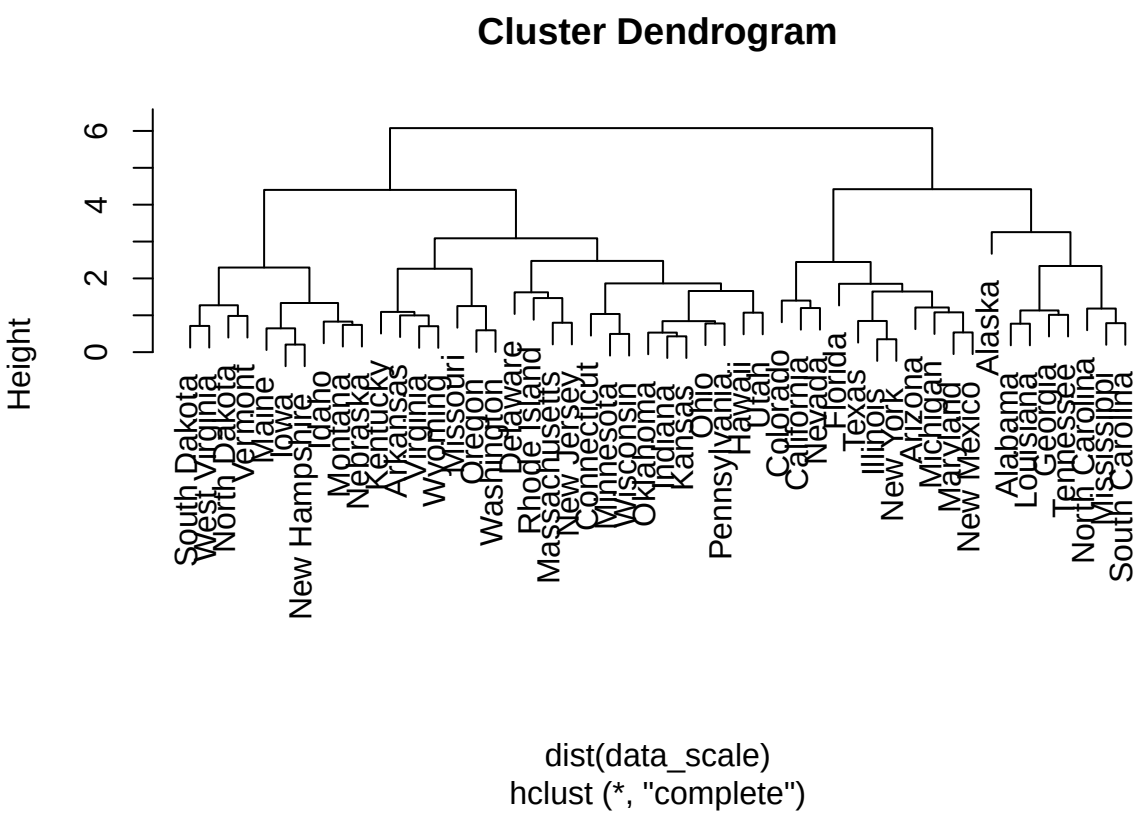
```
clustersb = cutree(complete_cluster, 3)
clustersb
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              1              1              2              1
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              2              3              1              1              2
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              3              3              1              3              3
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              3              3              1              3              1
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
```

```
##              2              1              3              1              2
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              3              3              1              3              2
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              1              1              1              3              3
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##              2              2              3              2              1
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              3              2              2              3              3
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              2              2              3              3              2
```

(c) Hierarchically cluster the states using complete linkage and Euclidean distance, after scaling the variables to have standard deviation one

```
data_scale = scale(USArrests)
complete_cluster_scale = hclust(dist(data_scale), method="complete")
plot(complete_cluster_scale)
```



(d) What efect does scaling the variables have on the hierarchical clustering obtained? In your opinion, should the variables be scaled before the inter-observation dissimilarities are computed? Provide a justifcation for your answer.

```
clustersd = cutree(complete_cluster_scale, 3)
clustersd
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              1              2              3              2
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              2              3              3              2              1
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              3              3              2              3              3
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              3              3              1              3              2
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              3              2              3              1              3
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              3              3              2              3              3
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              2              2              1              3              3
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##              3              3              3              3              1
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              3              1              2              3              3
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              3              3              3              3              3
```

```
table(clustersb)
```

```
## clustersb
##  1  2  3
## 16 14 20
```

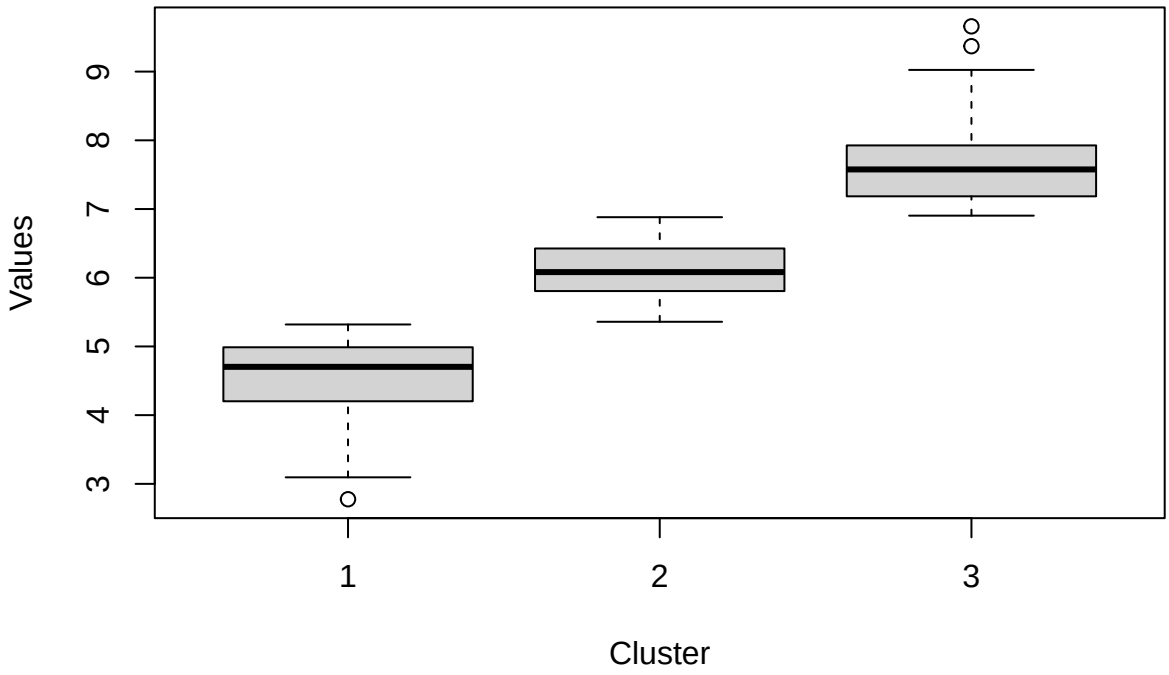
```
table(clustersd)
```

```
## clustersd
##  1  2  3
##  8 11 31
```

Our result will change after scaling. If the unit of the variable is not the same, we should standardize it.

4. Conduct K-means algorithm without using built-in function kmeans() on the following one-dimension data x. Output the corresponding centroids in each cluster when K=3. Visualize your clustering results using separate box-plots for each cluster.

```r
kmeans_algorithm <- function(D, k, epsilon) {
  t <- 0

  mu <- matrix(NA, nrow = k, ncol = ncol(D))
  for (i in 1:k) {
    mu[i, ] <- D[sample(1:nrow(D), 1), ]
  }

  cluster_assignment <- rep(0, nrow(D))

  repeat {
    t <- t + 1
    new_mu <- matrix(NA, nrow = k, ncol = ncol(D))

    for (j in 1:nrow(D)) {
      distances <- apply(mu, 1, function(centroid) sum((D[j, ] - centroid)^2))
      cluster_assignment[j] <- which.min(distances)
    }

    for (i in 1:k) {
  cluster_points <- D[cluster_assignment == i, , drop = FALSE]
  if (nrow(cluster_points) > 0) {
    new_mu[i, ] <- colMeans(cluster_points)
  } else {

    new_mu[i, ] <- D[sample(1:nrow(D), 1), ]
  }
}

    if (sum(sqrt(apply((mu - new_mu)^2, 1, sum))) <= epsilon) {
      break
    }
    mu <- new_mu
  }

  list(centroids = mu, assignment = cluster_assignment)
}

set.seed(0)
x <- c(rnorm(100, mean = 5, sd = 1), rnorm(100, mean = 7, sd = 1), rnorm(100, mean = 6, sd = 1))
D <- matrix(x, ncol = 1)

result <- kmeans_algorithm(D, k = 3, epsilon = 1e-4)

centroids <- result$centroids

boxplot(D ~ result$assignment, xlab = "Cluster", ylab = "Values", main = "Boxplot of Clusters")
```


Boxplot of Clusters
Values
3
4
5
6
7
8
9
1
2
3
Cluster