HW7_Unsupervised_Learning_Shuting_Li

Shuting

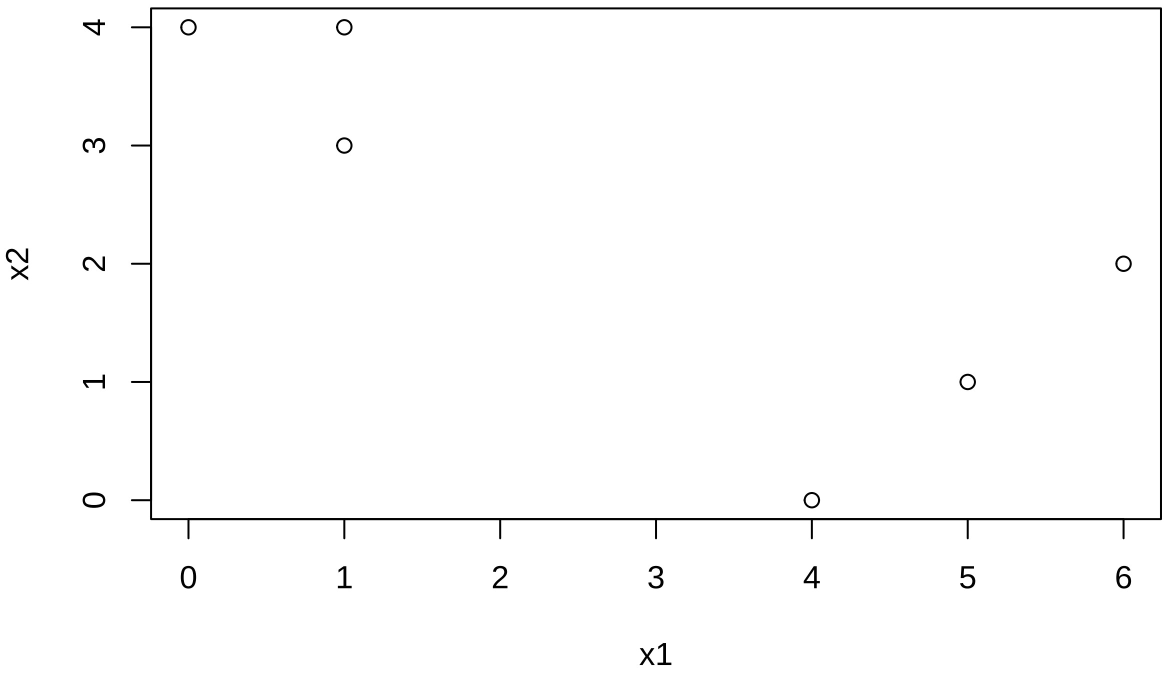
3/28/2022

12.3

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)

```r
x <- cbind(x1 <- c(1, 1, 0, 5, 6, 4), x2 <- c(4, 3, 4, 1, 2, 0))
plot(x1,x2)
```



Answer:

(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.

```r
set.seed(111)
labels <- sample(1:2, nrow(x), replace = TRUE)
labels
```

Answer:

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

(c)

Compute the centroid for each cluster.

```
(centroid1 = c(mean(x[labels==1, 1]), mean(x[labels==1, 2]))) ##label==1
```

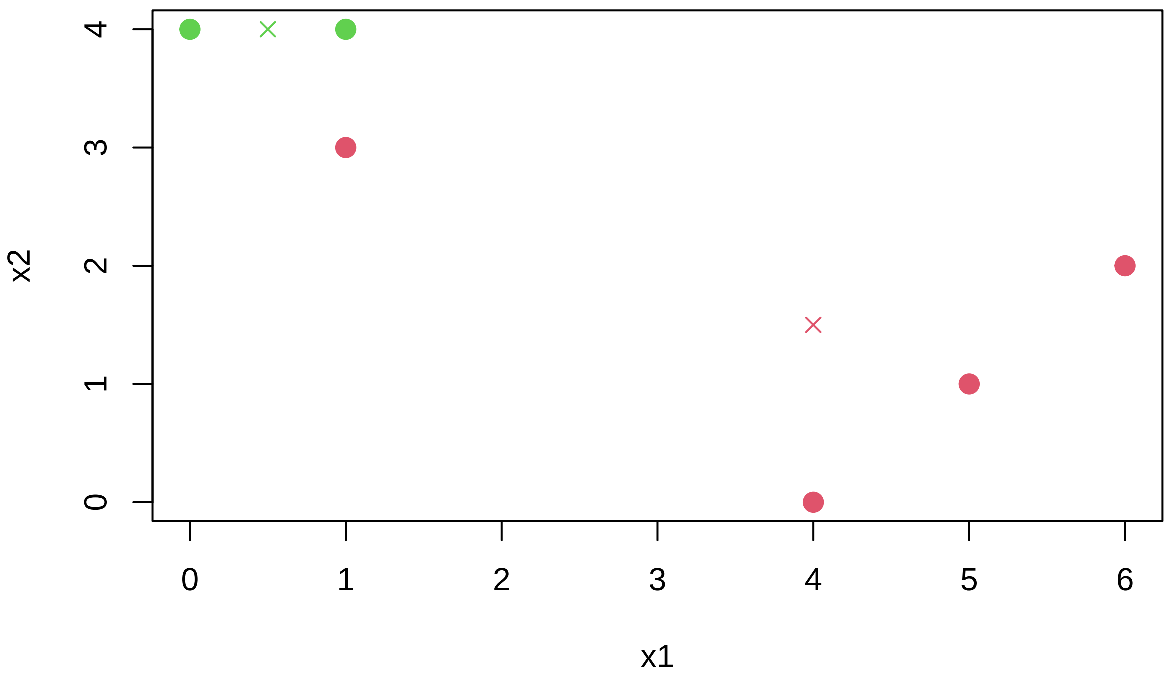
Answer:

```
## [1] 4.0 1.5
```

```
(centroid2 = c(mean(x[labels==2, 1]), mean(x[labels==2, 2]))) ##label=2
```

```
## [1] 0.5 4.0
```

```
plot(x1, x2, col=(labels+1), pch=20, cex=2)
points(centroid1[1], centroid1[2], col=2, pch=4)##label==1, red
points(centroid2[1], centroid2[2], col=3, pch=4)##label=2, green
```



(d)

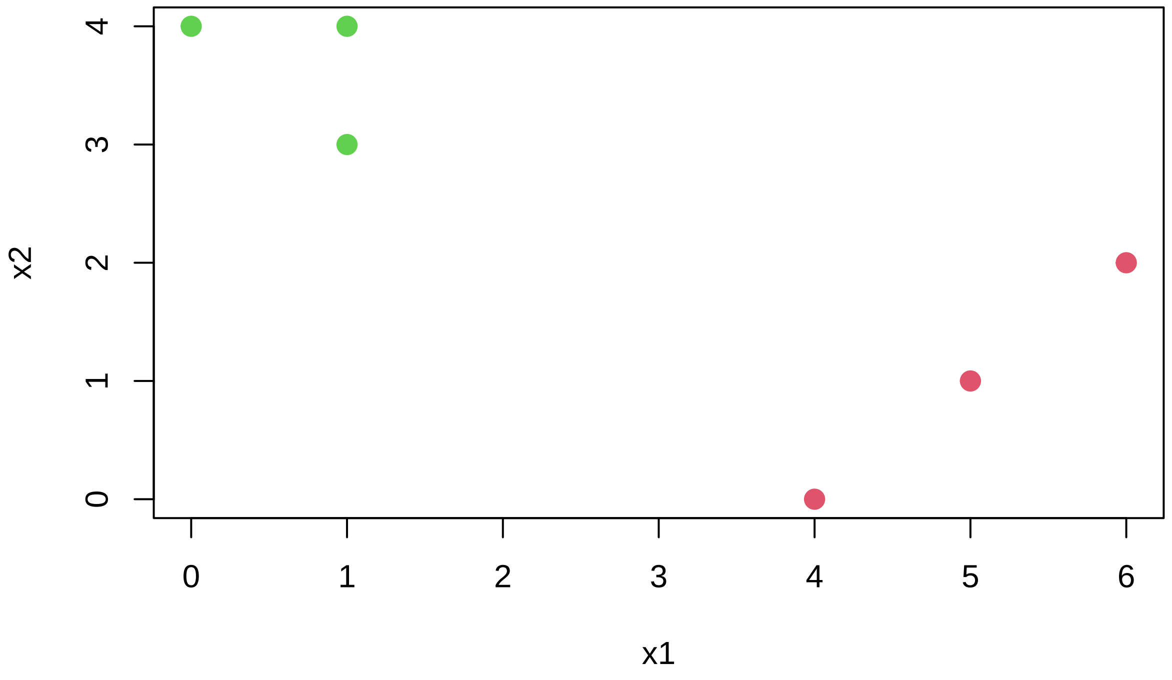
Assign each observation to the centroid to which it is closest, in terms of Euclidean distance. Report the cluster labels for each observation.

```
labels_assign <- NULL
dis1 <- NULL
dis2 <- NULL
for (i in 1:nrow(x)){
  dis1[i] <- sqrt((x[i,1]-centroid1[1])^2+(x[i,2]-centroid1[2])^2)
  dis2[i] <- sqrt((x[i,1]-centroid2[1])^2+(x[i,2]-centroid2[2])^2)
  if (dis1[i]<dis2[i]){labels_assign[i] <- 1}
  else {labels_assign[i] <- 2}
  i <- i+1
}
labels_assign
```

Answer:

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

```
plot(x1, x2, col=(labels_assign+1), pch=20, cex=2)
```



(e)

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

```

##make (d) as a function
labelsCompute <- function(x,centroid1,centroid2){
  labels_assign <- NULL
  dis1 <- NULL
  dis2 <- NULL
  for (i in 1:nrow(x)){
    dis1[i] <- sqrt((x[i,1]-centroid1[1])^2+(x[i,2]-centroid1[2])^2)
    dis2[i] <- sqrt((x[i,1]-centroid2[1])^2+(x[i,2]-centroid2[2])^2)
    if (dis1[i]<dis2[i]){labels_assign[i] <- 1}
    else {labels_assign[i] <- 2}
    i <- i+1
  }
  return(labels_assign)
}
##loop
labels_med <- labels
if (!all(labels_med==labels_assign)){
  labels_med <- labels_assign
  centroid1 <- c(mean(x[labels_assign==1, 1]), mean(x[labels_assign==1, 2]))
  centroid2 <- c(mean(x[labels_assign==2, 1]), mean(x[labels_assign==2, 2]))
  labels_assign <- labelsCompute(x,centroid1,centroid2)
}
labels_assign
```

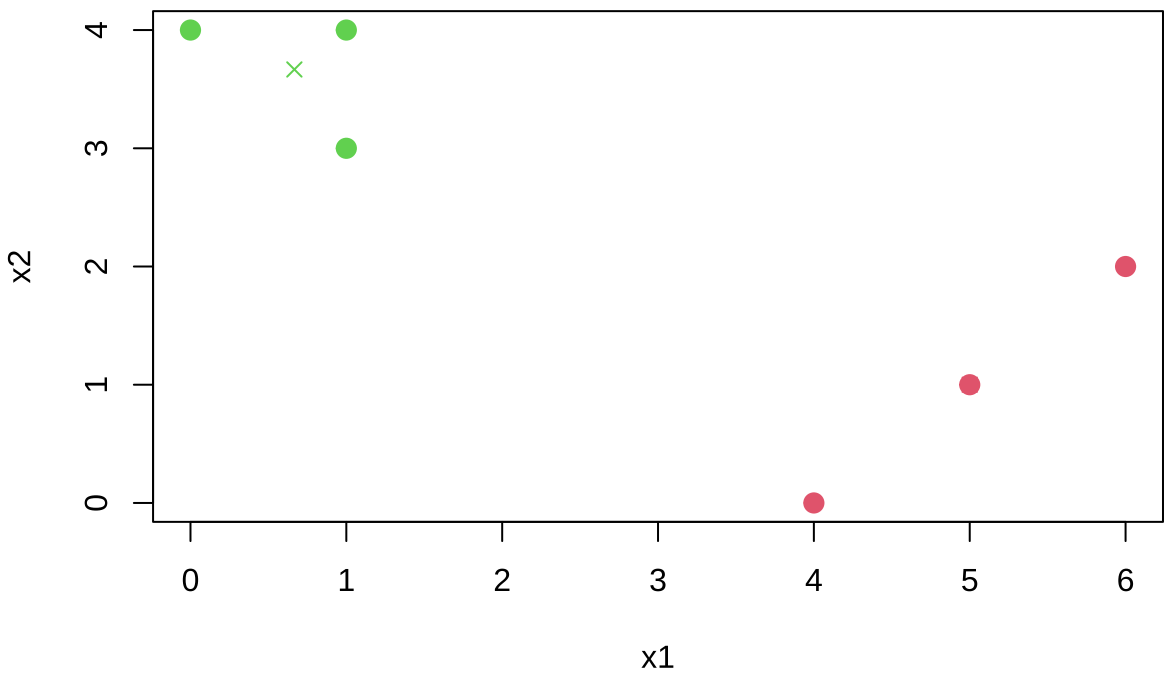
Answer:

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

(f)

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

```
plot(x1,x2,col=(labels_assign+1),pch=20,cex=2)
points(centroid1[1], centroid1[2], col=2, pch=4)##label==1, red
points(centroid2[1], centroid2[2], col=3, pch=4)##label=2, green
```



Answer:
## 12.5 In words, describe the results that you would expect if you performed K-means clustering of the eight shoppers in Figure 12.16, on the basis of their sock and computer purchases, with K = 2. Give three answers, one for each of the variable scalings displayed. Explain.

Answer: For no scaling situation, the clustering result will be critically decided by number of socks. Shopper (3,4,5,6) will be in a group, (1,2,7,8) with 11 socks will in another group.

For scaling situation, the clustering result is decided by socks and computers. Shopper (1,2,3,4) with no computer purchase will be in a group, (5,6,7,8) will be in another group.

For purchase money situation, the clustering result is decided by computers. Shopper (1,2,3,4) with no computer purchase will be in a group, (5,6,7,8) will be in another group.

12.8

In Section 12.2.3, a formula for calculating PVE was given in Equa- tion 12.10. We also saw that the PVE can be obtained using the sdev output of the prcomp() function.

On the USArrests data, calculate PVE in two ways:

(a)

Using the sdev output of the prcomp() function, as was done in Section 12.2.3.

```
pr.var <- prcomp(USArrests, center=T, scale=T)$sdev^2
(pve <- pr.var / sum(pr.var))
```

Answer:

```
## [1] 0.62006039 0.24744129 0.08914080 0.04335752
```

(b)

By applying Equation 12.10 directly. That is, use the prcomp() function to compute the principal component loadings. Then, use those loadings in Equation 12.10 to obtain the PVE.

```
loadings <- prcomp(USArrests, center=T, scale=T)$rotation
pve2 <- NULL
xscale <- apply(USArrests, 2, scale)

for (i in 1:dim(USArrests)[2]){
  z_med <- sweep(xscale, MARGIN=2, loadings[,i], "*")
  z <- apply(z_med,1,sum)
  var_z <- sum(z^2)
  pve2[i] <- var_z/sum(xscale^2) ##add all x^2
}
pve2
```

Answer:

```
## [1] 0.62006039 0.24744129 0.08914080 0.04335752
```

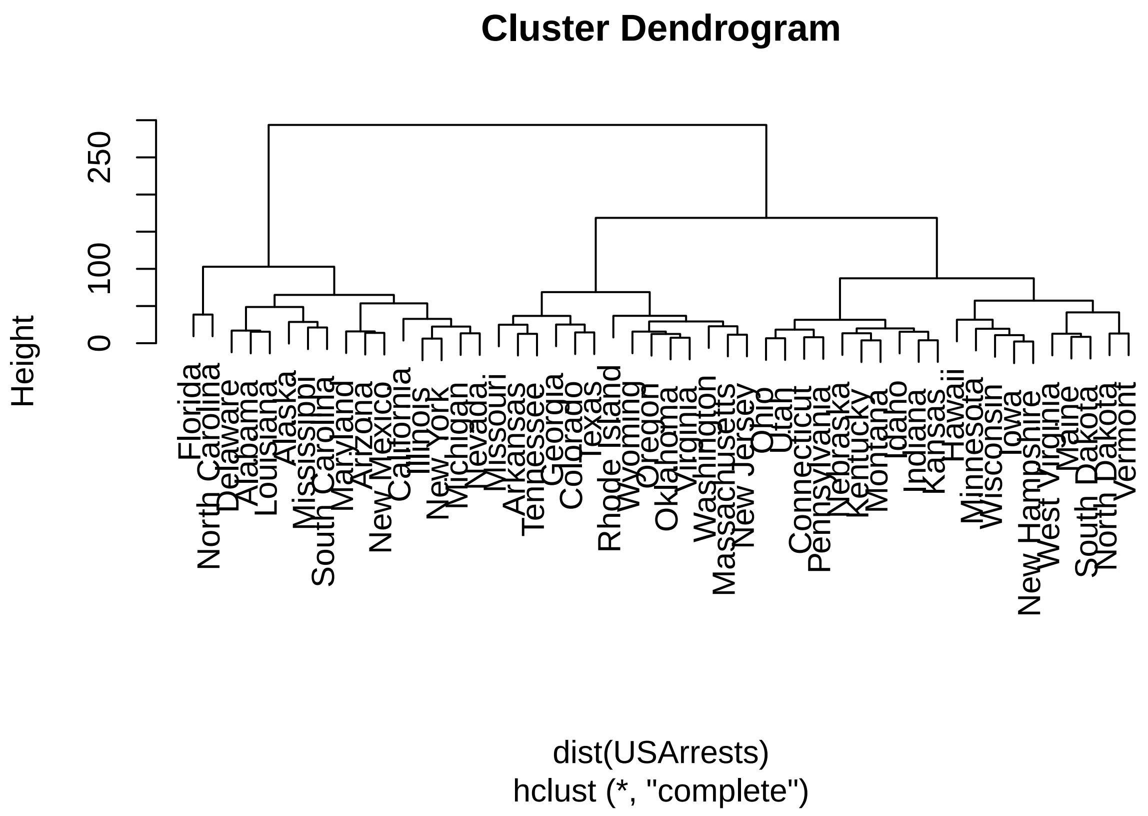
12.9

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(222)
hc.complete = hclust(dist(USArrests), method="complete")
plot(hc.complete)
```



Answer:

(b)

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

```
cutree(hc.complete, 3)
```

Answer:

```
##        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
```

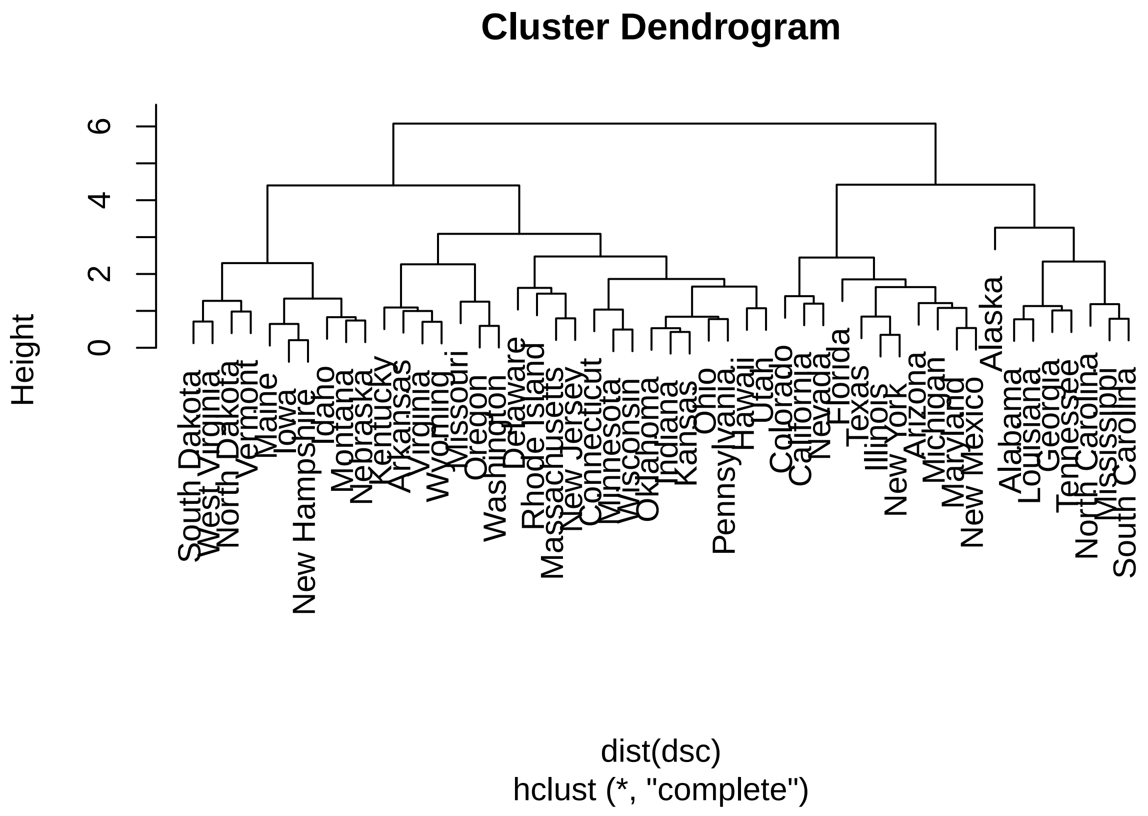
```r
table(cutree(hc.complete, 3))
```

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

(c)

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

```r
dsc = scale(USArrests)
hc.s.complete = hclust(dist(dsc), method="complete")
plot(hc.s.complete)
```



Answer:

```r
cutree(hc.s.complete, 3)
```

```
##         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(cutree(hc.s.complete, 3))
```

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

(d)

What effect 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 justification for your answer.

```
table(cutree(hc.s.complete, 3), cutree(hc.complete, 3))
```

Answer:

```
##
##      1  2  3
##   1  6  2  0
##   2  9  2  0
##   3  1 10 20
```

Scaling the variables will affect the height of clustering trees. The variables should be scaled before dissimilarities are computed because the variables have different units.

12.10

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.

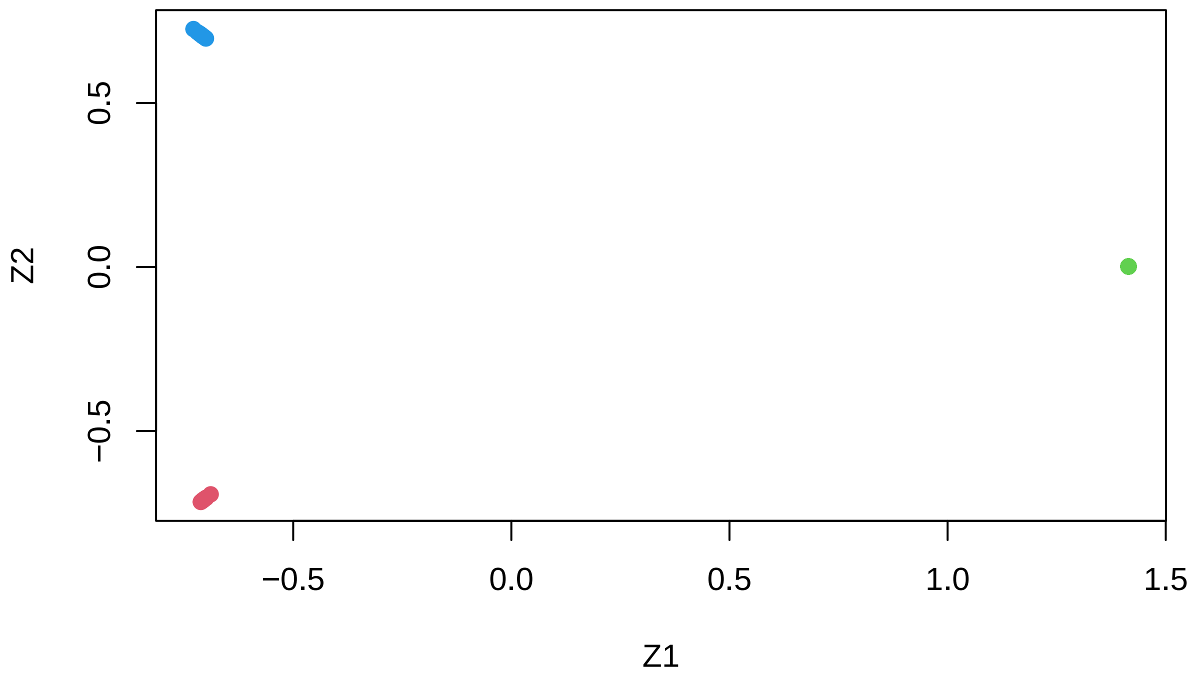
```
set.seed(100)
x1 <- matrix(rnorm(20*50,0,0.01),20,50)
x2 <- matrix(rnorm(20*50,0,0.01),20,50)
x3 <- matrix(rnorm(20*50,0,0.01),20,50)
x <- rbind(x1,x2,x3)
x[1:20,2] <- 1
x[21:40,1:2] <- 2
x[41:60,1] <- 1
color <- rep(1:3, each=20)
```

Answer:

(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.

```
##?
pca.out <- prcomp(x)
plot(pca.out$x[,1:2], col=(color+1), xlab="Z1", ylab="Z2", pch=19)
```



Answer:

(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?

```
km.out <- kmeans(x, 3, nstart = 20)
table(color, km.out$cluster)
```

Answer:

```
##
## color  1  2  3
##     1  0  0 20
##     2 20  0  0
##     3  0 20  0
```

(d)

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

```r
km.out2 <- kmeans(x, 2, nstart = 20)
table(color, km.out2$cluster)
```

Answer:

```
##
## color  1  2
##     1  0 20
##     2 20  0
##     3  0 20
```

(e)

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

```r
km.out4 <- kmeans(x, 4, nstart = 20)
table(color, km.out4$cluster)
```

Answer:

```
##
## color  1  2  3  4
##     1  0 12  0  8
##     2  0  0 20  0
##     3 20  0  0  0
```

(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.

```r
km.out.pc <- kmeans(pca.out$x[, 1:2], 3, nstart = 20)
table(color, km.out.pc$cluster)
```

Answer:

```
##
## color  1  2  3
##     1 20  0  0
##     2  0  0 20
##     3  0 20  0
```

Observations be perfectly divided by first two principal components.

(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.

```r
km.out.scale <- kmeans(scale(x), 3, nstart = 20)
table(color, km.out.scale$cluster)
```

Answer:

```
##
## color  1  2  3
##     1  9 10  1
##     2  5  1 14
##     3  4 16  0
```

The cluster result seems worse, because scaling affects the distance between the observations.