Math642_HW2_FyonaSun

Fyona Sun

1/22/2020

Lab1

```
data("USArrests")
names(USArrests)
```

```
## [1] "Murder"   "Assault"  "UrbanPop" "Rape"
```

```
apply(USArrests,2,mean)
```

```
##   Murder  Assault UrbanPop     Rape
##    7.788  170.760   65.540   21.232
```

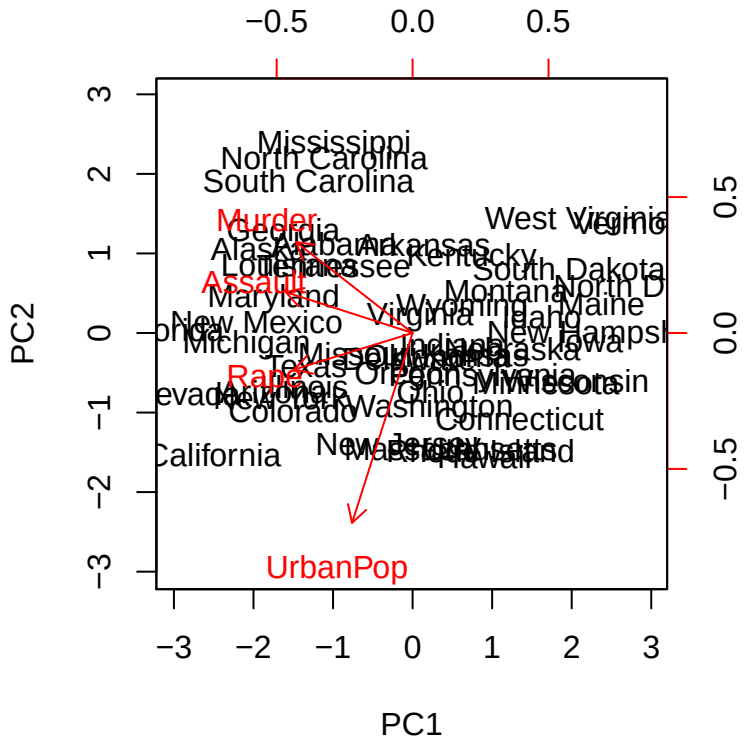
```
apply(USArrests,2,var)
```

```
##     Murder    Assault   UrbanPop       Rape
##   18.97047 6945.16571  209.51878   87.72916
```

```
model1=prcomp(USArrests,scale=TRUE)
#orthogonal unit vectors
model1$rotation
```

```
##                 PC1        PC2        PC3         PC4
## Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
## Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
## UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
## Rape     -0.5434321 -0.1673186  0.8177779  0.08902432
```

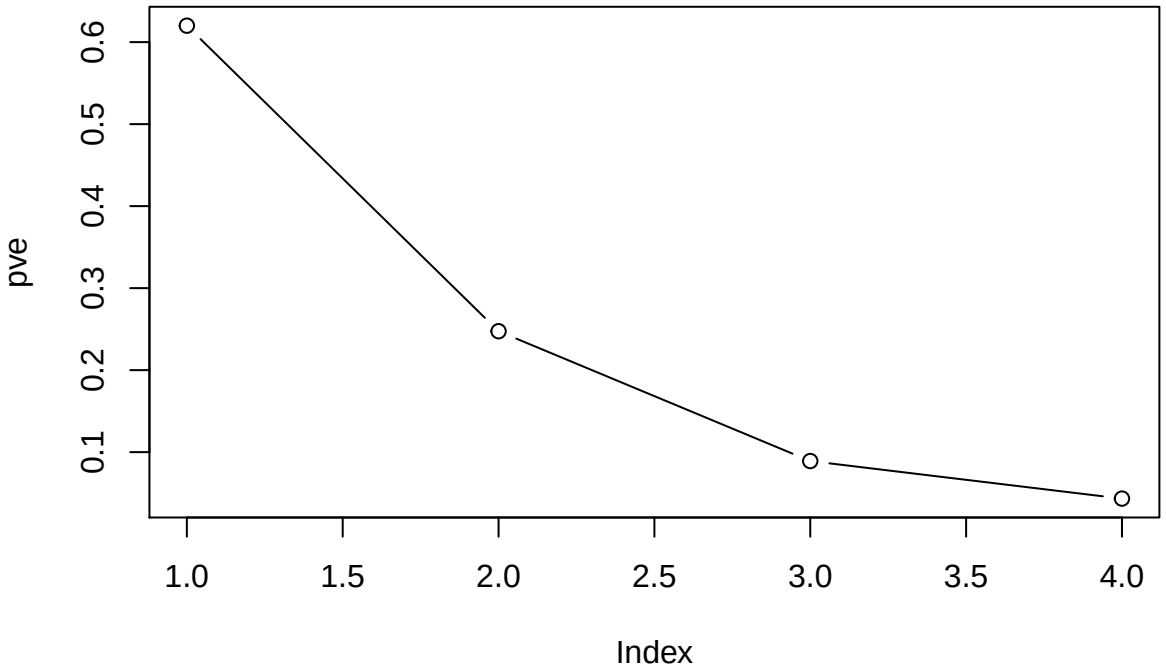
```
biplot(model1,scale=0)
```



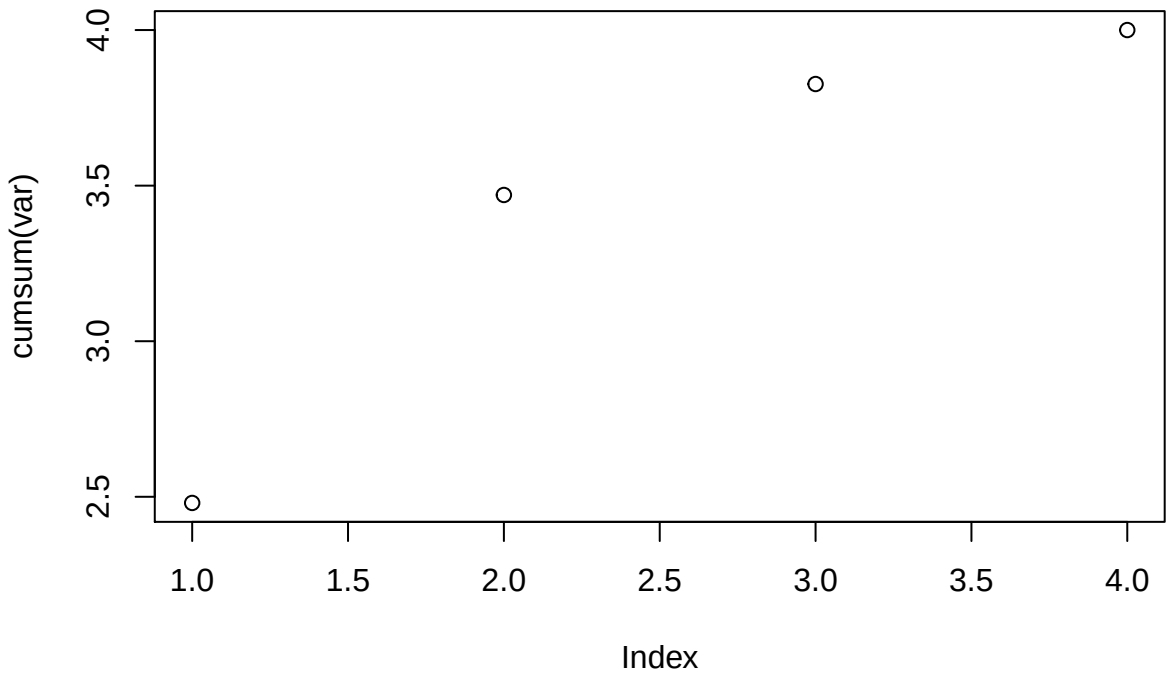

```
model1$sdev
```

```
## [1] 1.5748783 0.9948694 0.5971291 0.4164494
```

```
var=model1$sdev^2
pve=var/sum(var)
plot(pve,type='b')
```

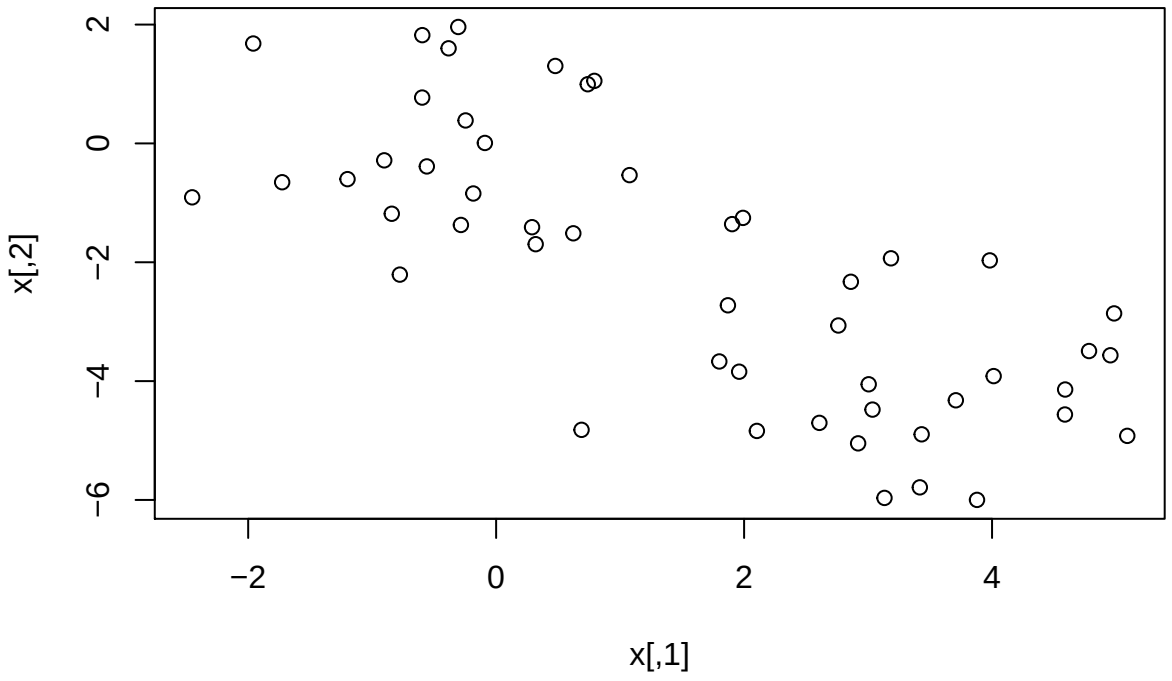


```r
plot(cumsum(var))
```



Lab2

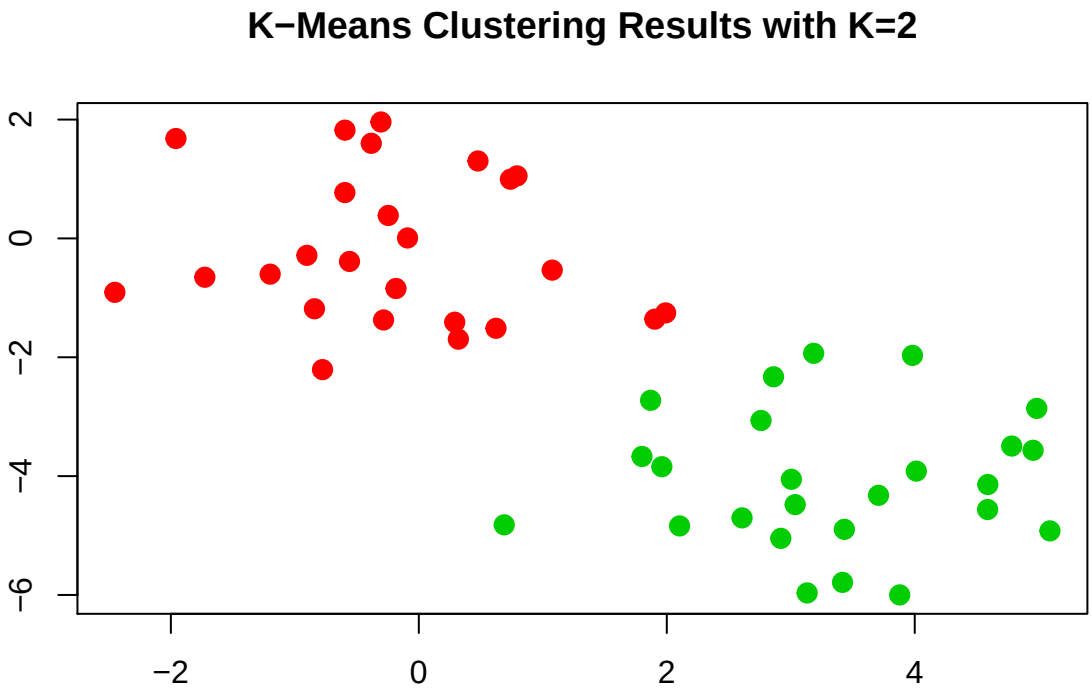
```r
set.seed(2)
x=matrix(rnorm(50*2),ncol=2)
x[1:25,1]=x[1:25,1]+3
x[1:25,2]=x[1:25,2]-4
plot(x)
```



```
kmodel=kmeans(x,2,nstart = 20)
kmodel$cluster
```

```
##  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
## [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
plot(x, col=(kmodel$cluster +1), main="K-Means Clustering Results with K=2", xlab="", ylab="", pch=20, c
```



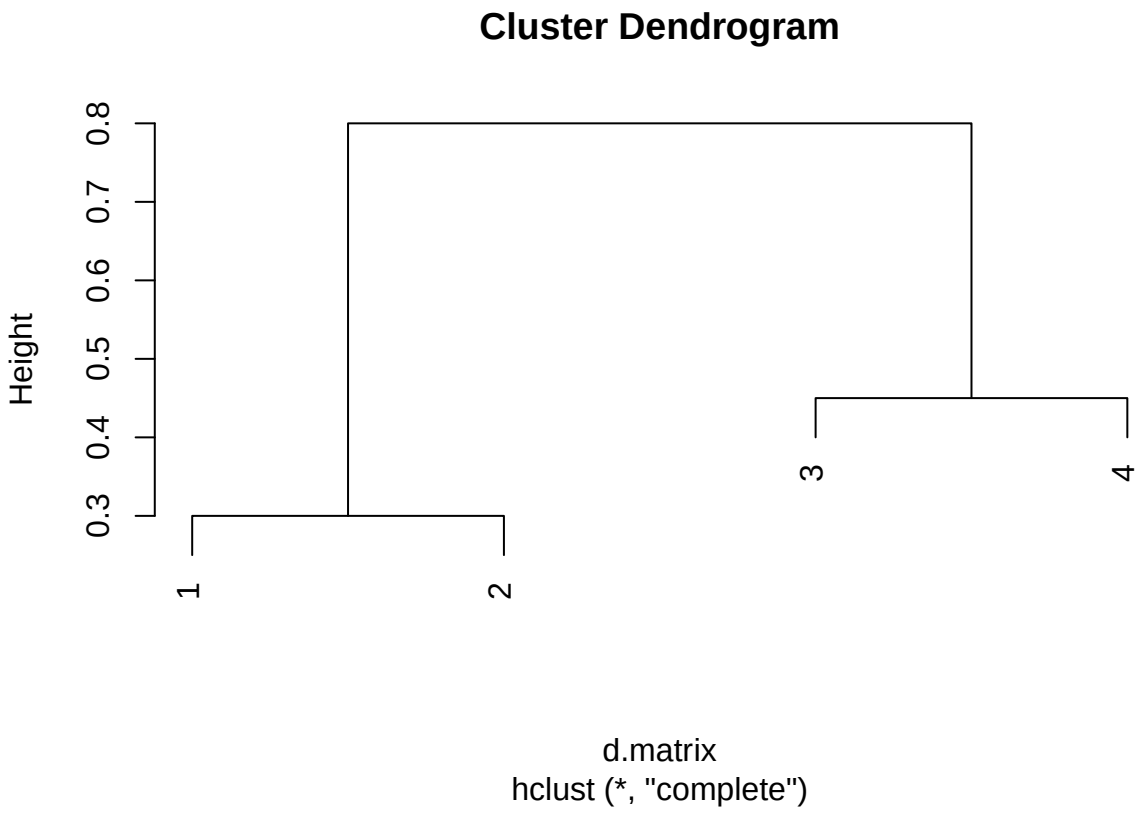
Lab3

```
hcmodel1=hclust(dist(x),method = 'complete')
hcmodel2=hclust(dist(x),method = 'average')
hcmodel3=hclust(dist(x),method = 'single')
```

Question 10.2

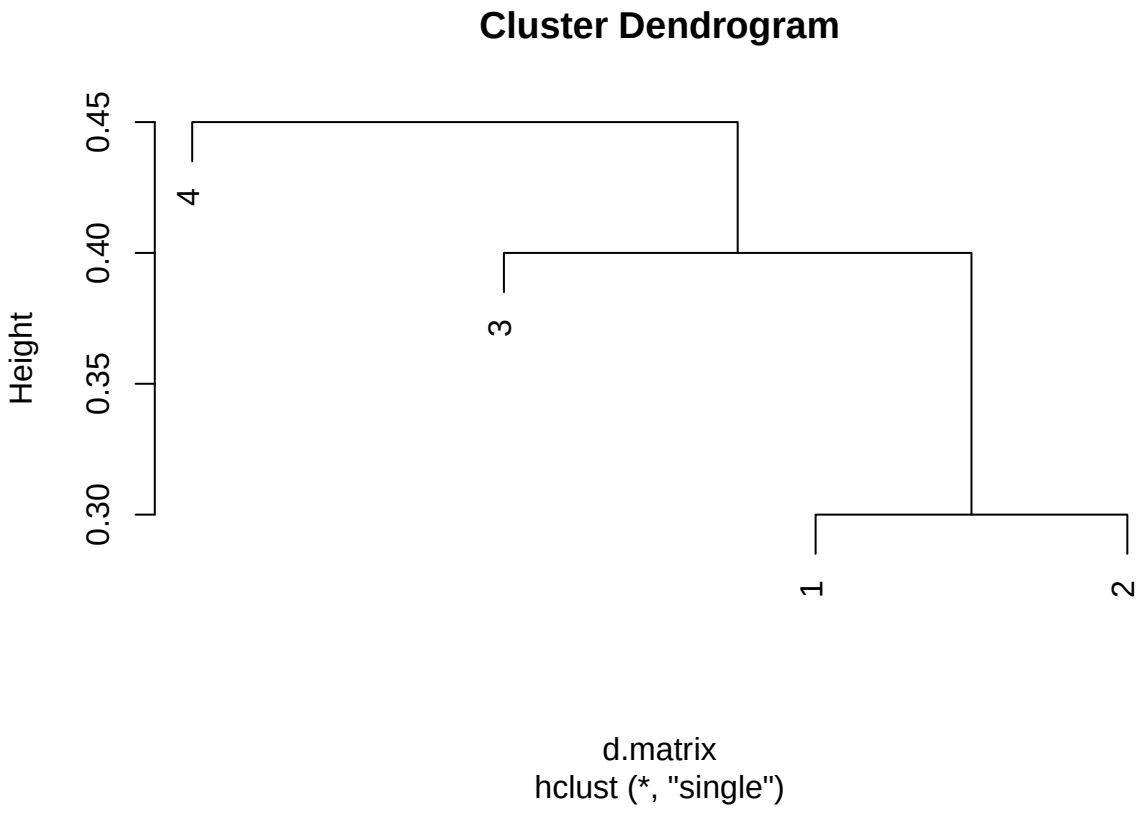
(a) On the basis of this dissimilarity matrix, sketch the dendrogram that results from hierarchically clustering these four observations using complete linkage. Be sure to indicate on the plot the height at which each fusion occurs, as well as the observations corresponding to each leaf in the dendrogram.

```
d.matrix = as.dist(matrix(c(0, 0.3, 0.4, 0.7,
                      0.3, 0, 0.5, 0.8,
                      0.4, 0.5, 0.0, 0.45,
                      0.7, 0.8, 0.45, 0.0), nrow = 4))
plot(hclust(d.matrix, method = "complete"))
```



(b) Repeat (a), this time using single linkage clustering.

```
plot(hclust(d.matrix, method = "single"))
```



(c) Sup-

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

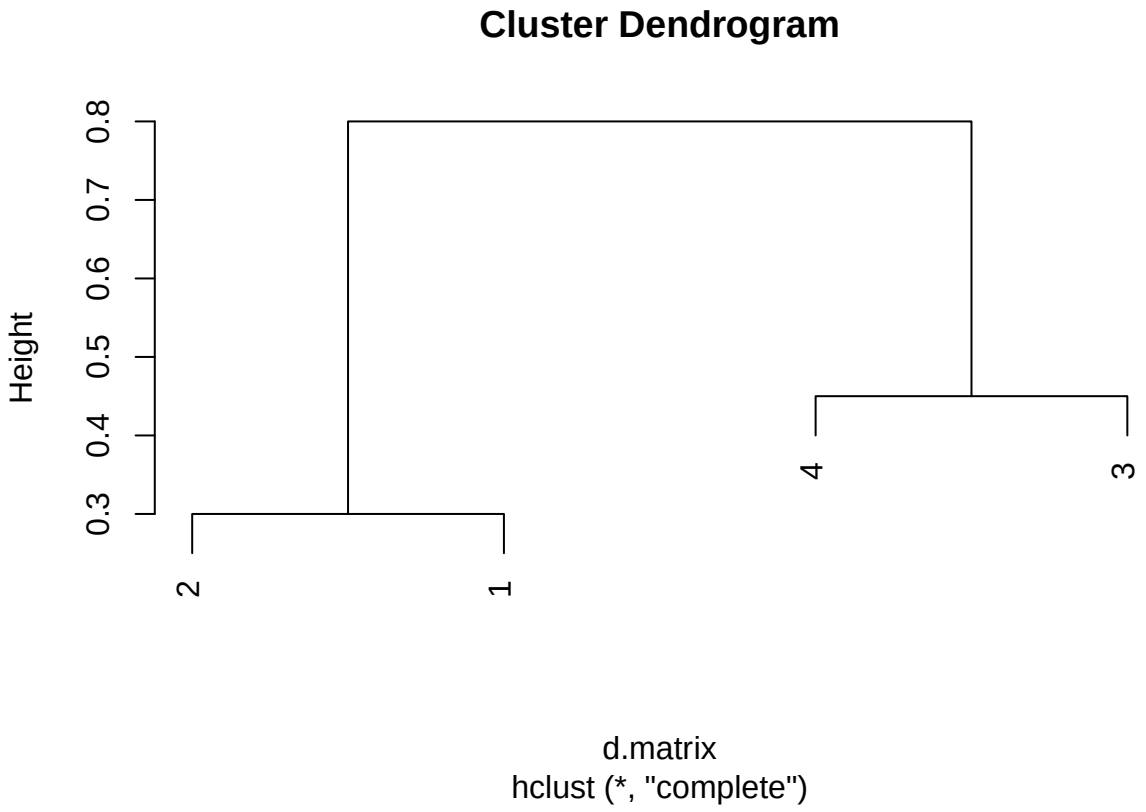
The first cluster contains 1 and 2, and the second cluster contains 3 and 4.

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

The first cluster contains 4 and the second cluster contains 3 and (1,2).

(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.

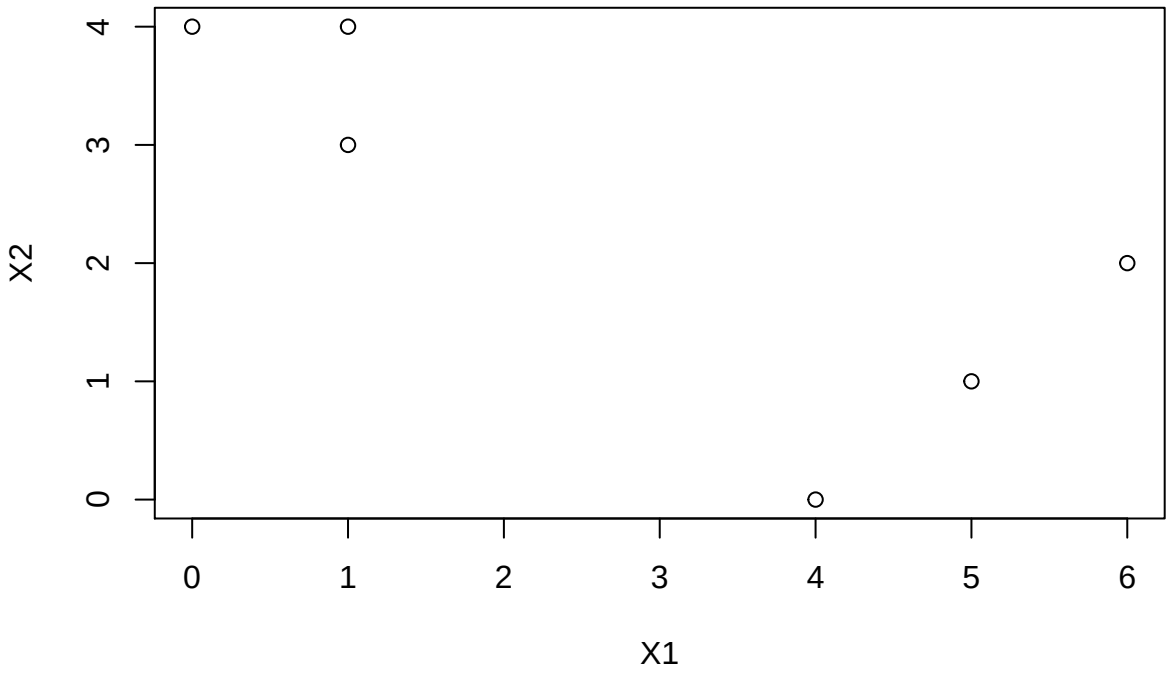
```
plot(hclust(d.matrix, method = "complete"), labels = c(2,1,4,3))
```



##Question 10.3

(a) Plot the observations.

```
x <- cbind(c(1, 1, 0, 5, 6, 4), c(4, 3, 4, 1, 2, 0))
plot(x[,1], x[,2], xlab = 'X1', ylab='X2')
```

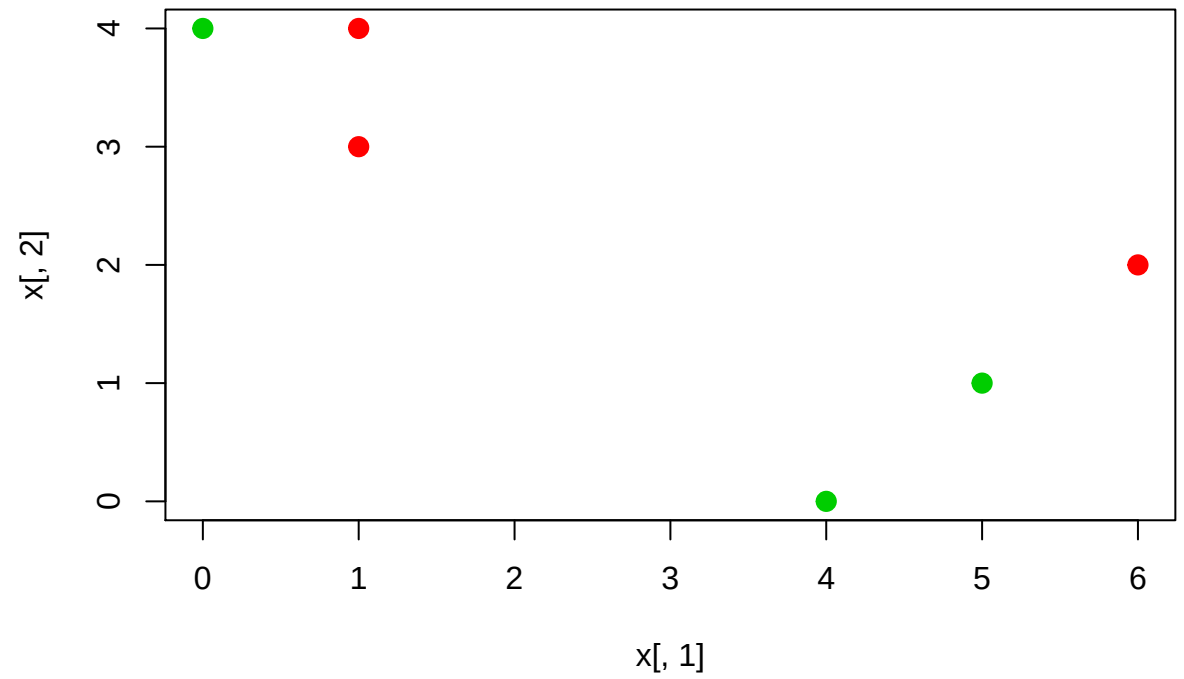


(b) Randomly assign a cluster label to each observation. You can use the sample() command in R to do this. Report the cluster labels for each observation.

```
set.seed(1)
labels <- sample(2, nrow(x), replace = T)
labels
```

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

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

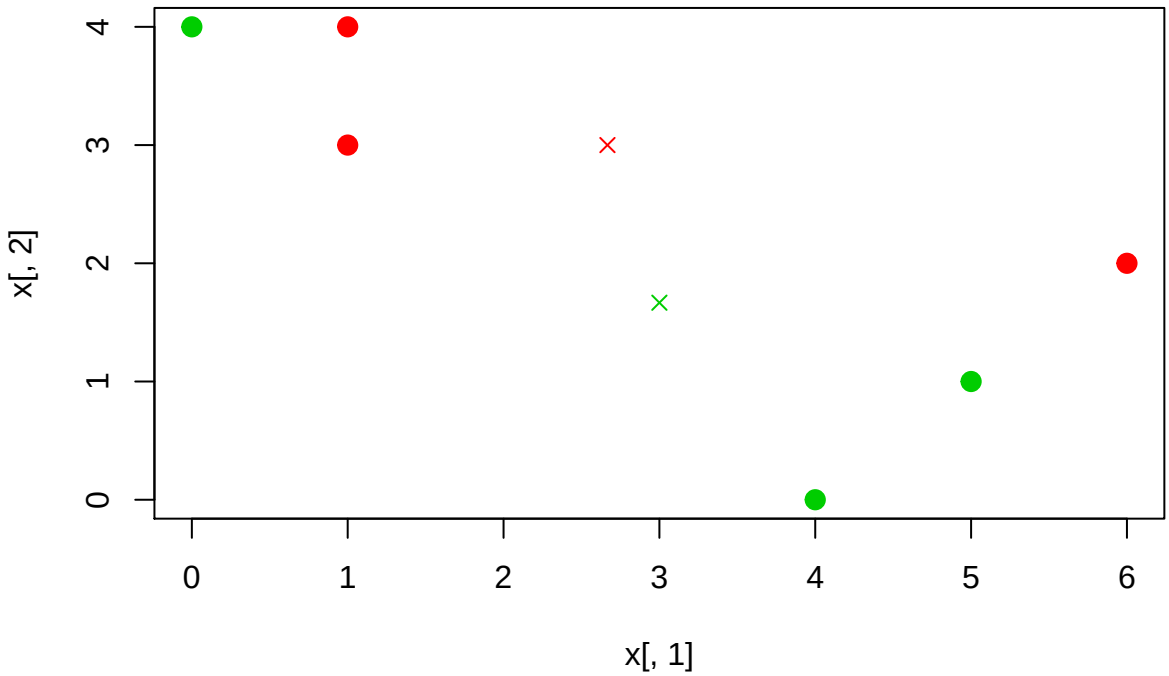


(c) Compute the centroid for each cluster.

```
cbind(x,labels)
```

```
##          labels
## [1,] 1 4      1
## [2,] 1 3      1
## [3,] 0 4      2
## [4,] 5 1      2
## [5,] 6 2      1
## [6,] 4 0      2
```

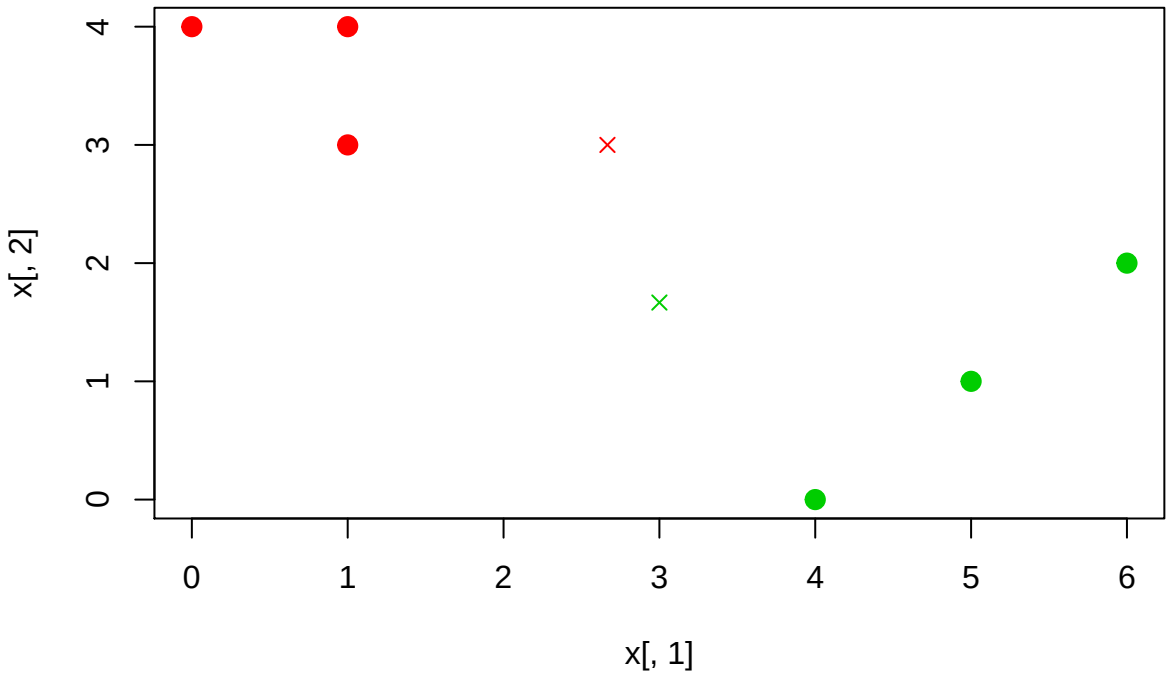
```
centroid1 <- c(mean(x[labels == 1, 1]), mean(x[labels == 1, 2]))
centroid2 <- c(mean(x[labels == 2, 1]), mean(x[labels == 2, 2]))
plot(x[,1], x[,2], col=(labels + 1), pch = 20, cex = 2)
points(centroid1[1], centroid1[2], col = 2, pch = 4)
points(centroid2[1], centroid2[2], col = 3, pch = 4)
```



$x_{11} = 1/3*(1+1+6) = 8/3$ and $x_{12} = 1/3*(2+3+4) = 3$ $x_{21} = 1/3*(0+4+5) = 11/4$ and $x_{22} = 1/3*(0+1+4) = 5/3$

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

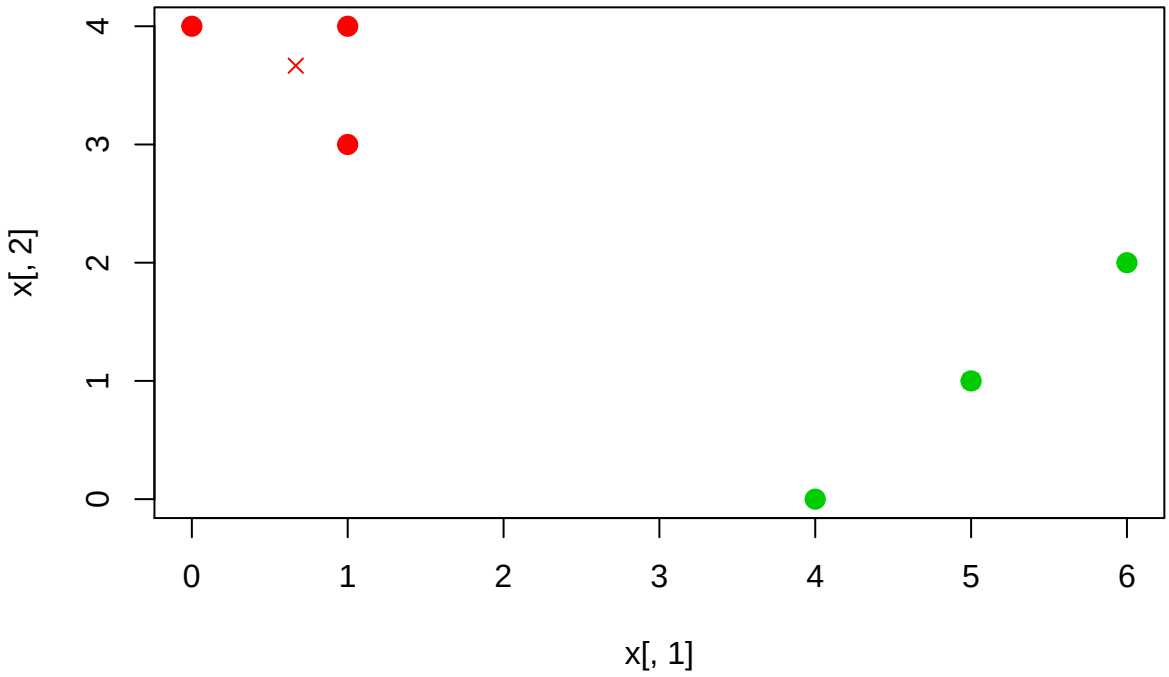
```
labels <- c(1, 1, 1, 2, 2, 2)
plot(x[, 1], x[, 2], col = (labels + 1), pch = 20, cex = 2)
points(centroid1[1], centroid1[2], col = 2, pch = 4)
points(centroid2[1], centroid2[2], col = 3, pch = 4)
```



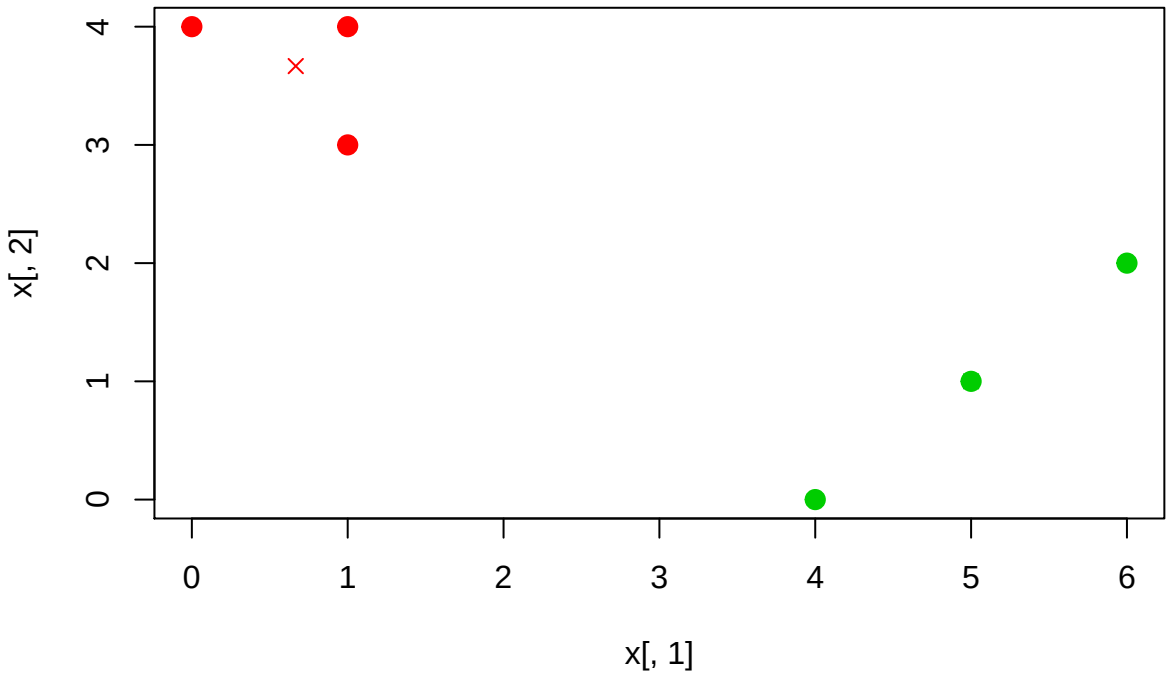
(e)

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

```
centroid1 <- c(mean(x[labels == 1, 1]), mean(x[labels == 1, 2]))
centroid2 <- c(mean(x[labels == 2, 1]), mean(x[labels == 2, 2]))
plot(x[,1], x[,2], col=(labels + 1), pch = 20, cex = 2)
points(centroid1[1], centroid1[2], col = 2, pch = 4)
points(centroid2[1], centroid2[2], col = 3, pch = 4)
```



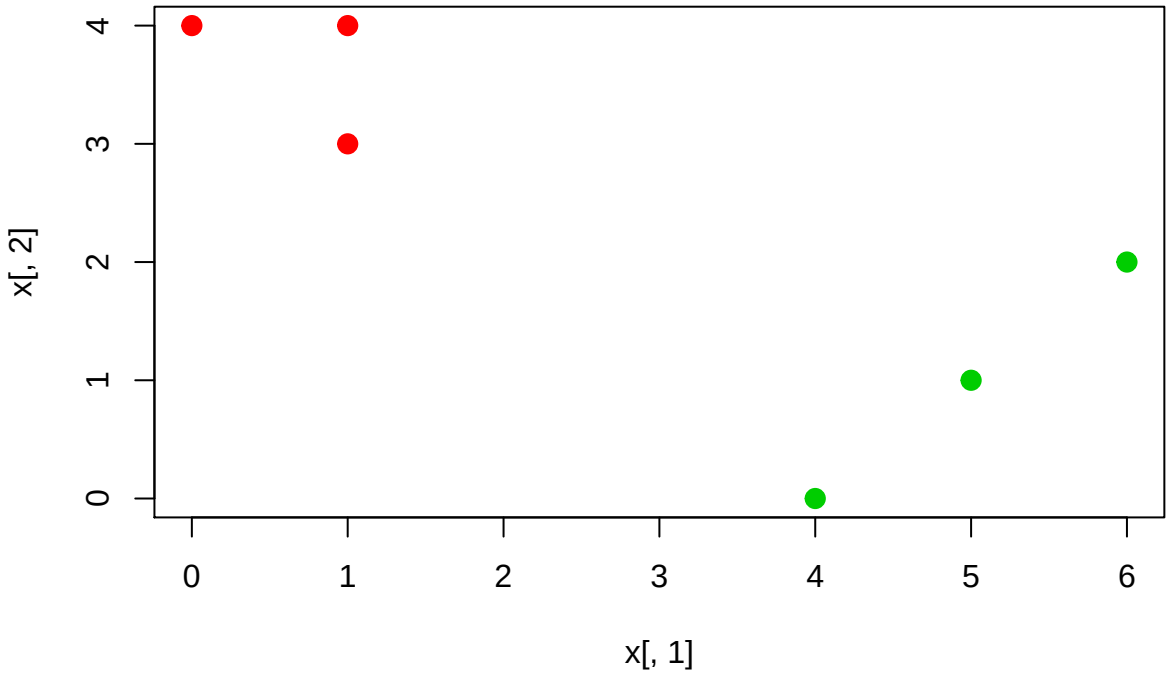

```
labels <- c(1, 1, 1, 2, 2, 2)
plot(x[, 1], x[, 2], col = (labels + 1), pch = 20, cex = 2)
points(centroid1[1], centroid1[2], col = 2, pch = 4)
points(centroid2[1], centroid2[2], col = 3, pch = 4)
```



$x_{11} = 1/3*(0+1+1) = 2/3$ and $x_{12} = 1/3*(3+4+4) = 11/3$ $x_{21} = 1/3*(4+5+6) = 5$ and $x_{22} = 1/3*(0+1+2) = 1$

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

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



Question 10.8

(a) Using the sdev output of the prcomp() function, as was done in Section 10.2.3.

```
data("USArrests")
model1=prcomp(USArrests,scale=TRUE)
```

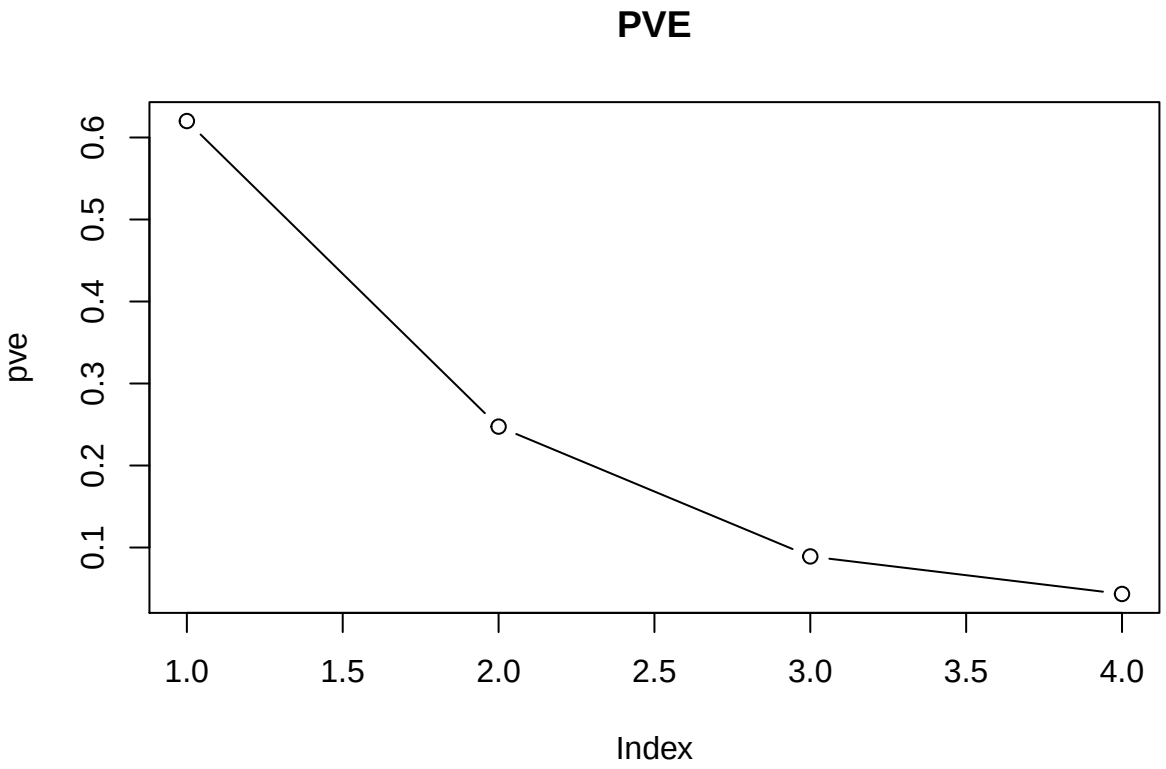
```
model1$sdev
```

```
## [1] 1.5748783 0.9948694 0.5971291 0.4164494
```

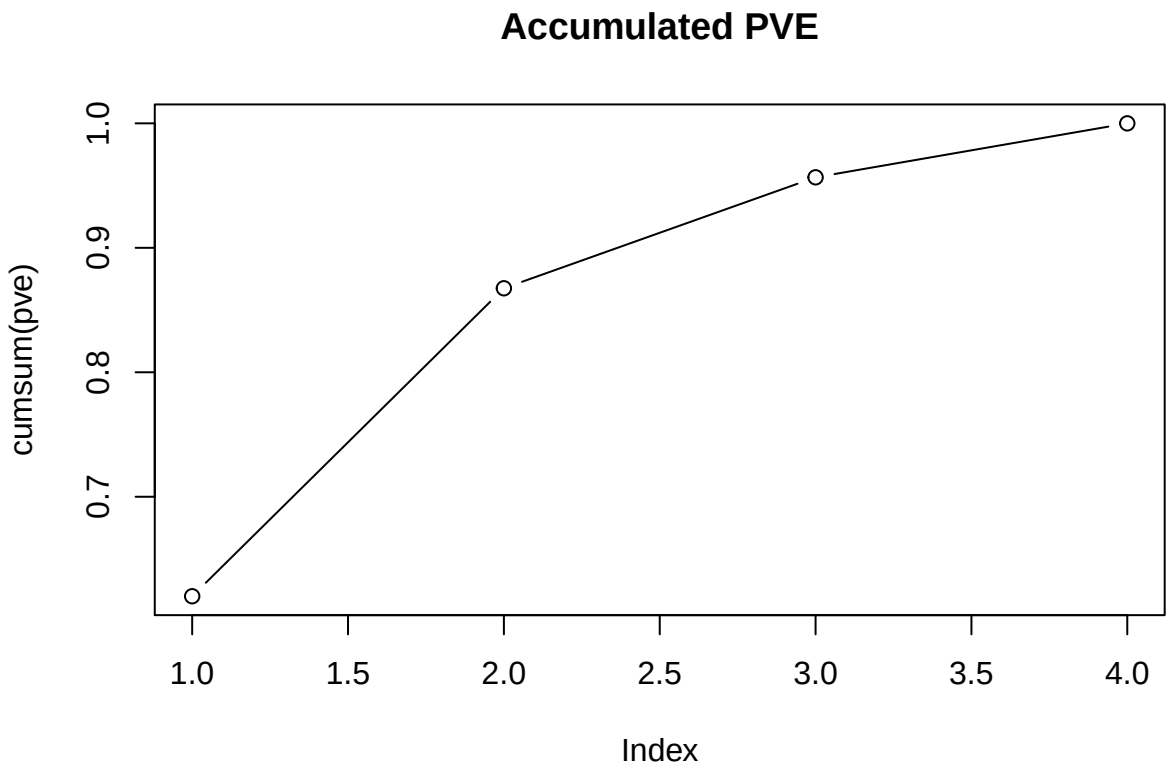
```
var=model1$sdev^2
pve=var/sum(var)
print(pve)
```

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

```
plot(pve,type='b', main='PVE')
```




```
plot(cumsum(pve),type ='b', main='Accumulated PVE')
```

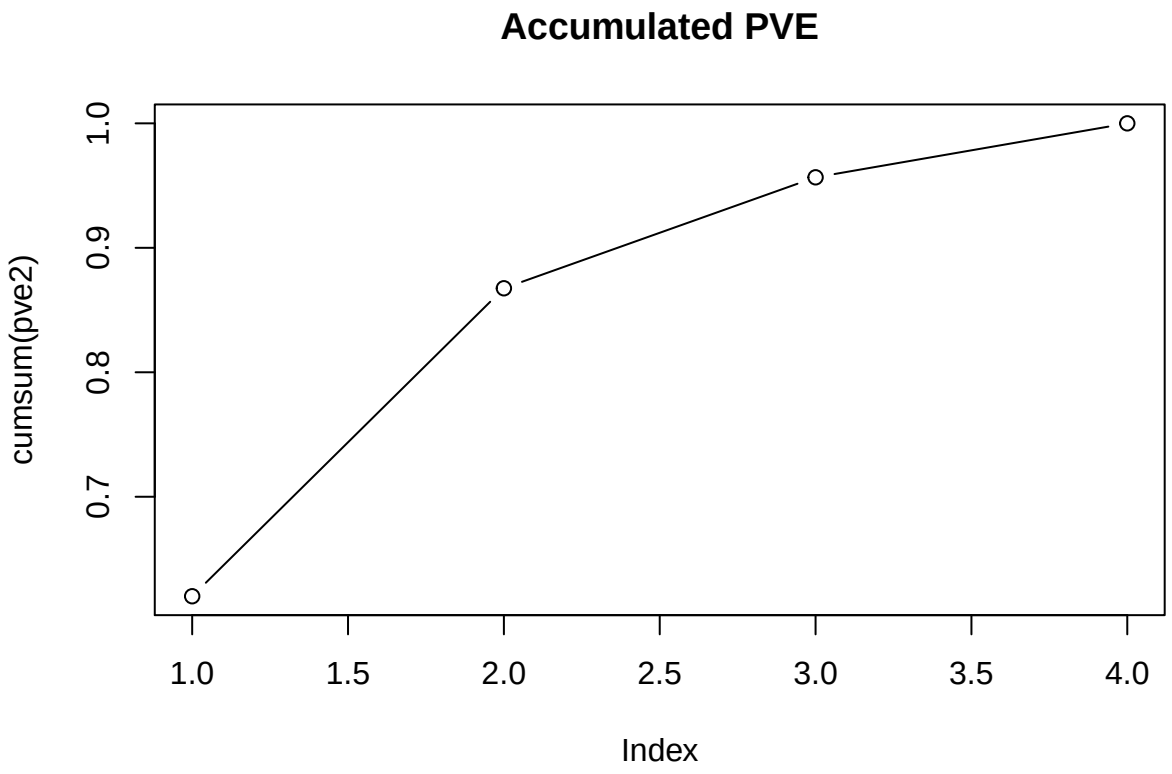


(b) By applying Equation 10.8 directly. That is, use the prcomp() function to compute the principal component loadings. Then, use those loadings in Equation 10.8 to obtain the PVE.

```
pca<- model1$rotation
sumvar<- sum(apply(as.matrix(scale(USArrests))^2,2,sum))
pve2<- apply((as.matrix(scale(USArrests)) %*% pca)^2, 2, sum) / sumvar
print(pve2)
```

```
##        PC1        PC2        PC3        PC4
## 0.62006039 0.24744129 0.08914080 0.04335752
```

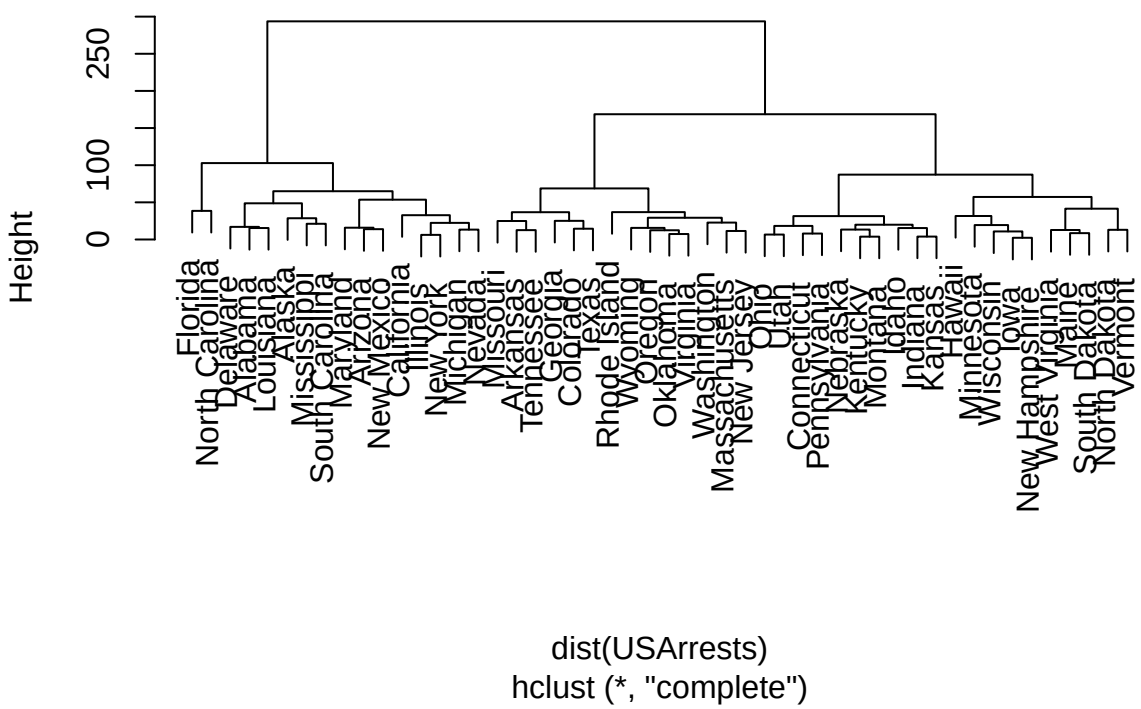
```
plot(cumsum(pve2),type ='b', main='Accumulated PVE')
```



Question 10.9

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

```
hcmodel.complete <- hclust(dist(USArrests), method = "complete")
plot(hcmodel.complete)
```



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

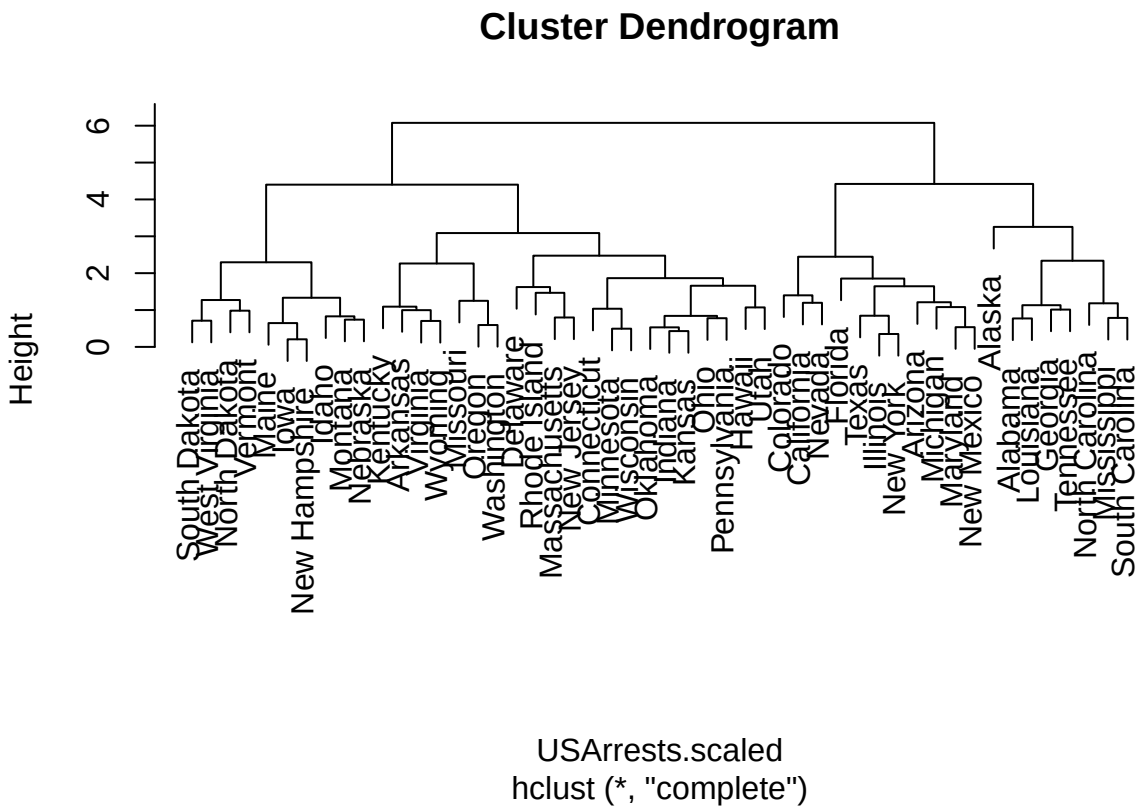
```
cutree(hcmodel.complete, 3)
```

```
##        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.

```
USArrests.scaled<- dist(scale(USArrests))
hcmodel.complete.scaled <- hclust(USArrests.scaled, method = "complete")
```

```
plot(hcmodel.complete.scaled)
```



(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.

```
cutree(hcmodel.complete.scaled, 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
```

```
cross.table<- table(cutree(hcmodel.complete, 3), cutree(hcmodel.complete.scaled, 3))
cross.table
```

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

Compare the result of hcmodel.complete and hcmodel.complete.scaled, there are 28/50 states are classified into the same category. There is no state that classified into category 1 or 2 by the scaled model that have been classified into category 3 by the non-scaled model. Scaling impacts the branch lengths, and the height of the tree. The height of the un-scaled model is 293.622751 while the height of the scaled tree is 6.0766416. In this case , scaling is more appropriate because the variables all have different unites. Therefore, it is imporant to scale the data before clustering.