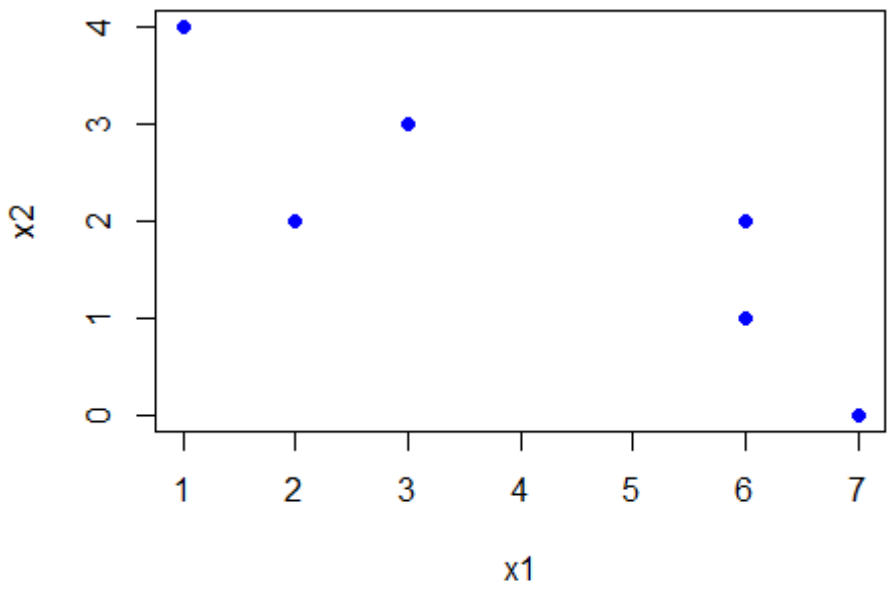
HW5-MATH4323

anthonycastillo ID:1670011

2022-11-01

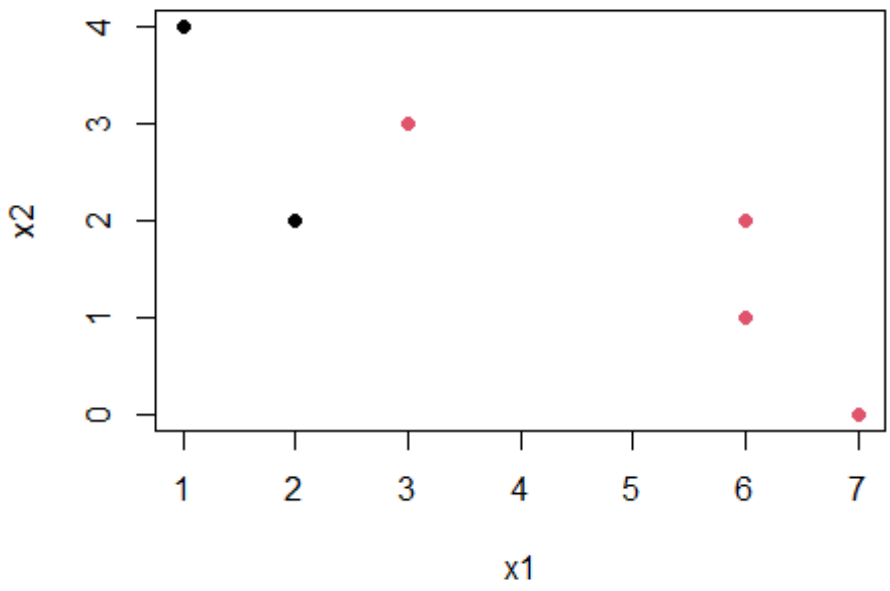
Question 1 (a):

```
x1 <- c(1,2,3,6,6,7)
x2 <- c(4,2,3,2,1,0)
plot(x1,x2, xlab = "x1", ylab = "x2", pch = 19, col = "blue")
```



Question 1 (b):

```
RNGkind(sample.kind = "default")
set.seed(2)
labels <- sample(2,6, replace = T)
labels

## [1] 1 1 2 2 2 2

plot(x1,x2, pch = 19, col = labels ,xlab = "x1",ylab = "x2")
```

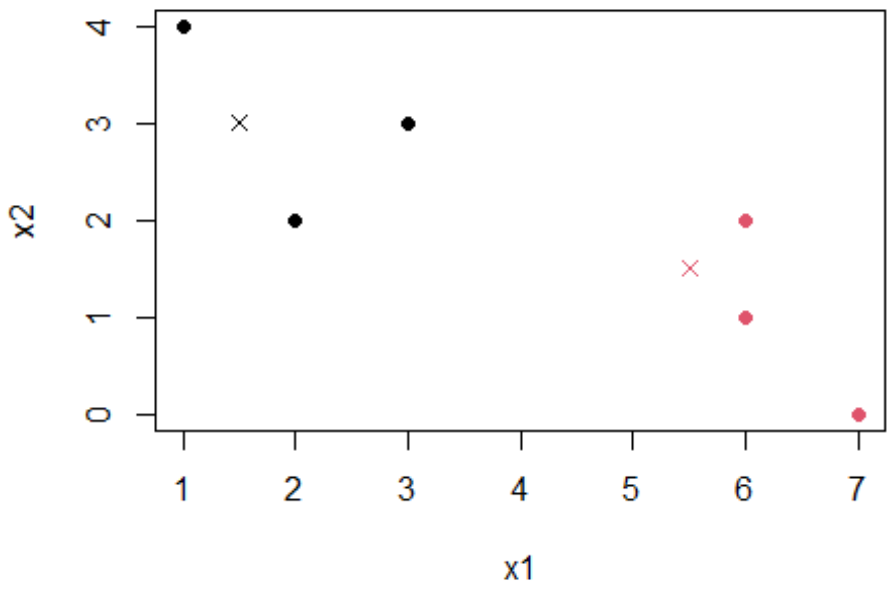


Question 1 (c):

```
cluster1x <- sum(x1[1:2])/2
cluster1y <- sum(x2[1:2])/2
cluster2x <- sum(x1[3:6])/4
cluster2y <- sum(x2[3:6])/4
```

Question 1 (d):

```
clusterassign <- c(1,1,1,2,2,2)
plot(x1,x2, pch = 19, col = clusterassign,xlab = "x1",ylab = "x2")
points(cluster1x,cluster1y, col = 1, pch = 4)
points(cluster2x,cluster2y, col = 2, pch = 4)
```



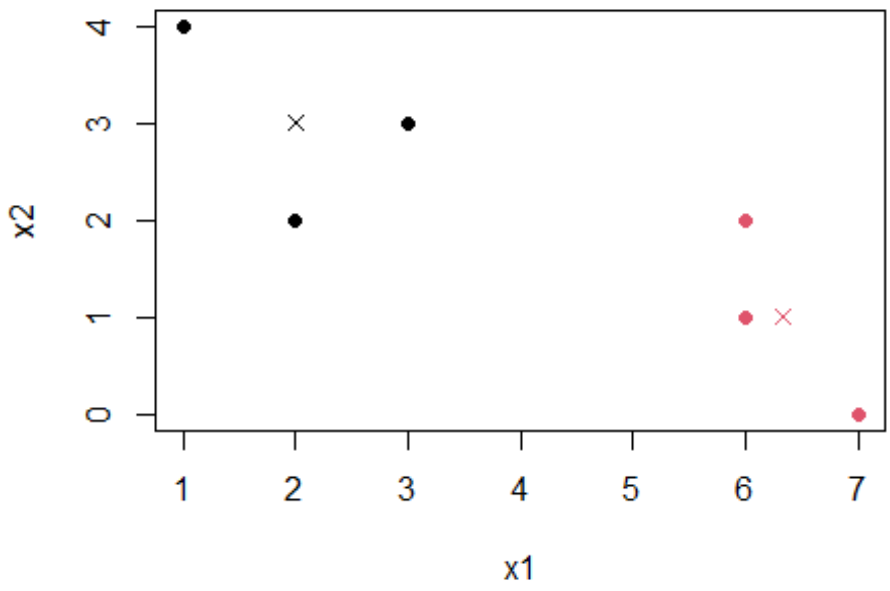
assign centroid (1.5,3) to cluster 1
assign centroid (5.5,1.5) to cluster 2
Question 1 (e):

```
up.cluster1x <- sum(x1[1:3])/3
up.cluster1y <- sum(x2[1:3])/3
up.cluster2x <- sum(x1[4:6])/3
up.cluster2y <- sum(x2[4:6])/3
```

The observations stayed the same after one iteration

Question 1 (f):

```
clusterlabel <- c(1,1,1,2,2,2)
plot(x1,x2, pch = 19, col = clusterlabel, xlab = "x1",ylab = "x2")
points(up.cluster1x,up.cluster1y, col = 1, pch = 4)
points(up.cluster2x,up.cluster2y, col = 2, pch = 4)
```



Question 2 (a):

```
library(MASS)
newBoston <- Boston[,-4]
```

Question 2 (b):

```
newBoston <- newBoston[,-13]
```

Question 2 (c):

```
boston.pca <- prcomp(newBoston, scale = TRUE)
#summary(boston.pca)
boston.pca$sdev
```

```
##  [1] 2.4752210 1.1586541 1.0861790 0.9138194 0.8152738 0.7330805 0.6296169
##  [8] 0.5263720 0.4693245 0.4314643 0.4114793 0.2542551
```

Question 2 (d):

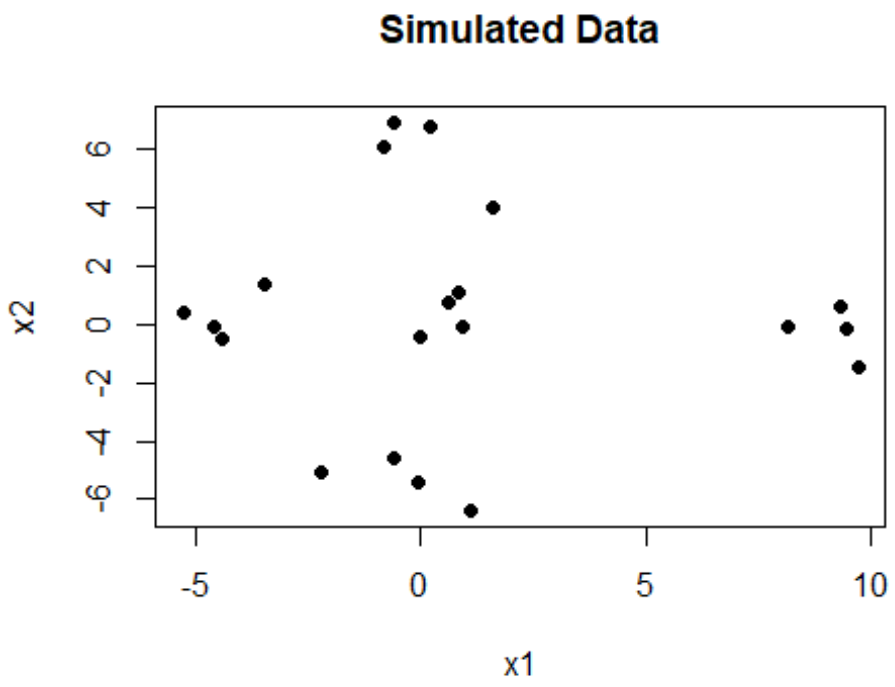
```
boston.var <- boston.pca$sdev^2
print(boston.var)
```

```
##  [1] 6.12671880 1.34247929 1.17978483 0.83506595 0.66467141 0.53740698
##  [7] 0.39641745 0.27706753 0.22026544 0.18616148 0.16931520 0.06464564
```

Question 2 (e):

Question 3 (a):

```
set.seed(1)
x<- matrix(rnorm(4*5*2),ncol=2)

#split in 5 clusters
x[1:4,2] <- x[1:4,2]+6
x[5:8,1] <- x[5:8,1]+9
x[9:12,1] <- x[9:12,1]-5
x[13:16,2] <- x[13:16,2]-5
scale(x)

##              [,1]        [,2]
##  [1,] -0.34704999  1.78509343
##  [2,] -0.17317981  1.74877257
##  [3,] -0.39194493  1.56096639
##  [4,]  0.12979826  1.01315401
##  [5,]  1.78978617  0.11314939
##  [6,]  1.54296816 -0.06626494
##  [7,]  1.82368062 -0.09271886
##  [8,]  1.87753006 -0.44173956
##  [9,] -1.16215973 -0.17827939
## [10,] -1.35128414  0.05956455
## [11,] -0.96126721  0.30925869
## [12,] -1.20206741 -0.07864934
## [13,] -0.34593108 -1.27558824
## [14,] -0.68793335 -1.39276660
## [15,]  0.02884762 -1.74398970
## [16,] -0.22223900 -1.48863486
## [17,] -0.21606985 -0.15602094
## [18,] -0.01002052 -0.06711022
## [19,] -0.03633723  0.24060575
## [20,] -0.08512663  0.15119788
## attr(,"scaled:center")
## [1] 0.9905239 0.1935285
## attr(,"scaled:scale")
## [1] 4.659207 3.767561

plot(x, pch=19, main = "Simulated Data",
     ylab = "x2", xlab = "x1")
```

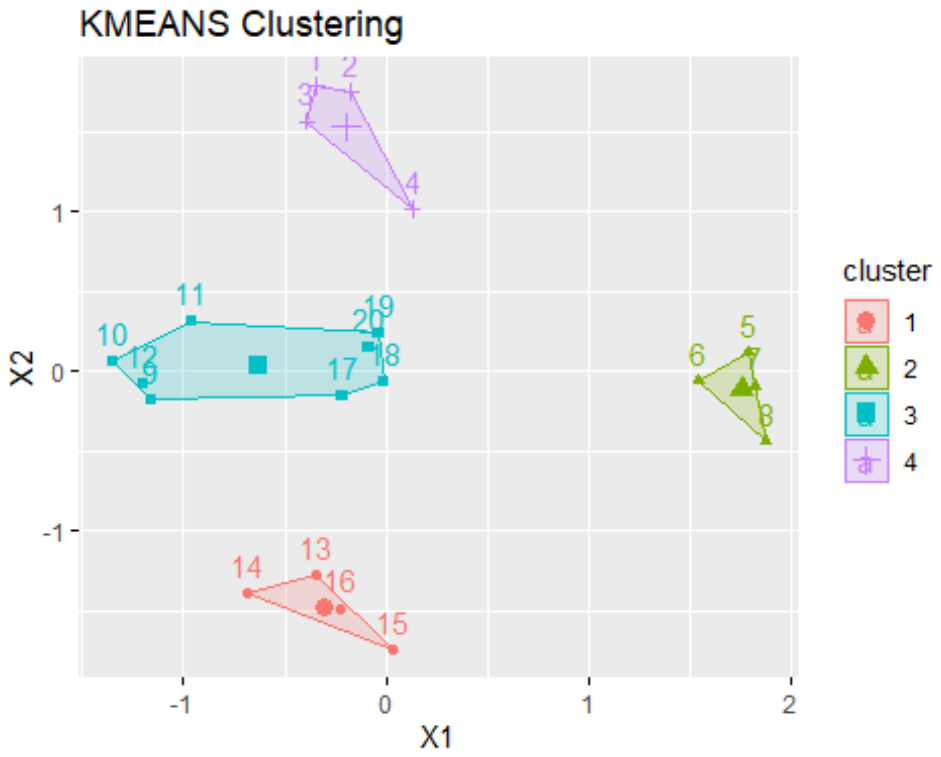


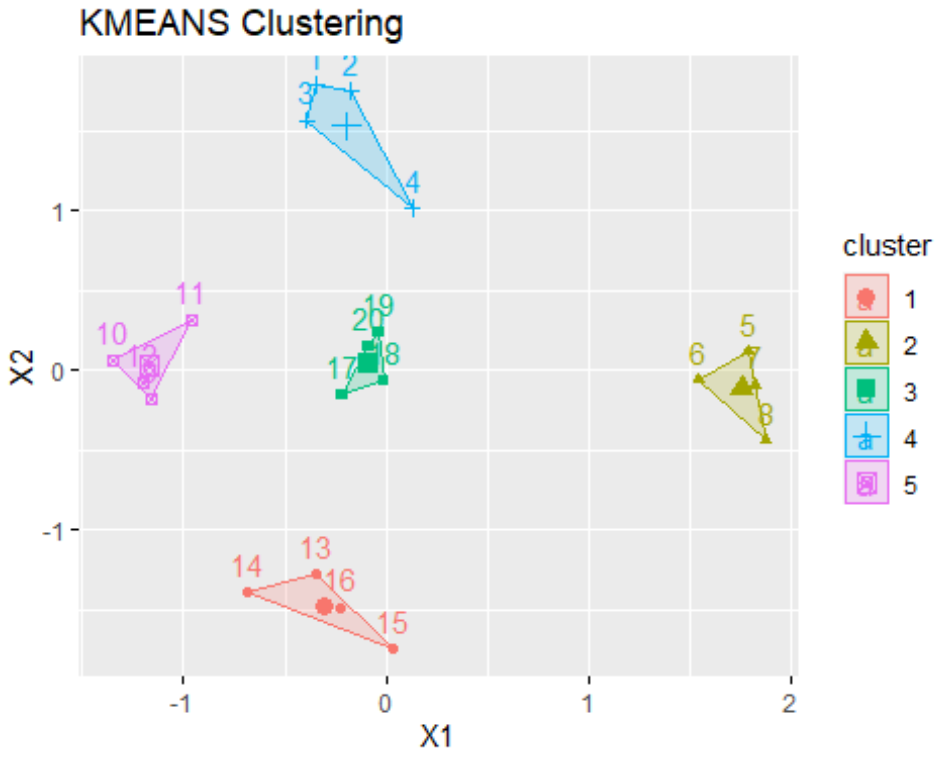
Question 3 (b):

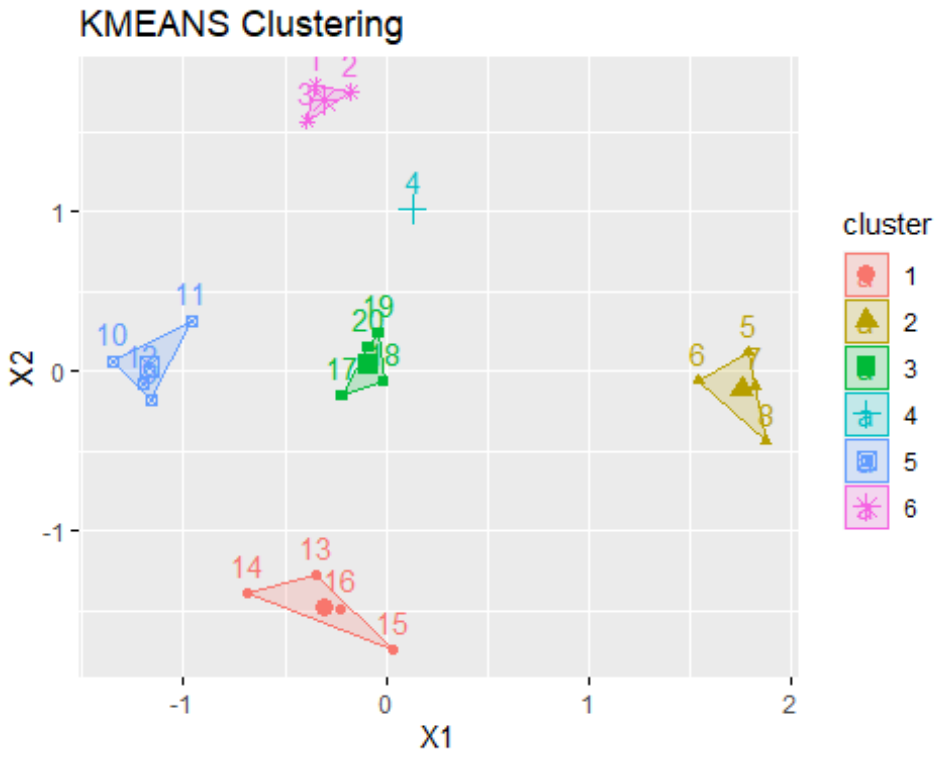
```
x <- data.frame(x)
library(factoextra)

## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

sim.wss <- numeric(6)
for(i in 4:6){
  sim.wss[i] <- eclust(x,
                       FUNcluster = "kmeans",
                       k=i,
                       nstart=50,
                       graph = 0)$tot.withinss
}
print(sim.wss[4:6])

## [1] 76.70080 25.83768 17.77594
```

going from k=4 to k=5 had a bigger drop in total WSS
Question 3 (c):

```r
for(i in 4:6){
  eclust(x,FUNcluster = "kmeans",
         k=i,
         nstart = 50)
}
```


KMEANS Clustering
X2
X1
cluster
1
2
3
4

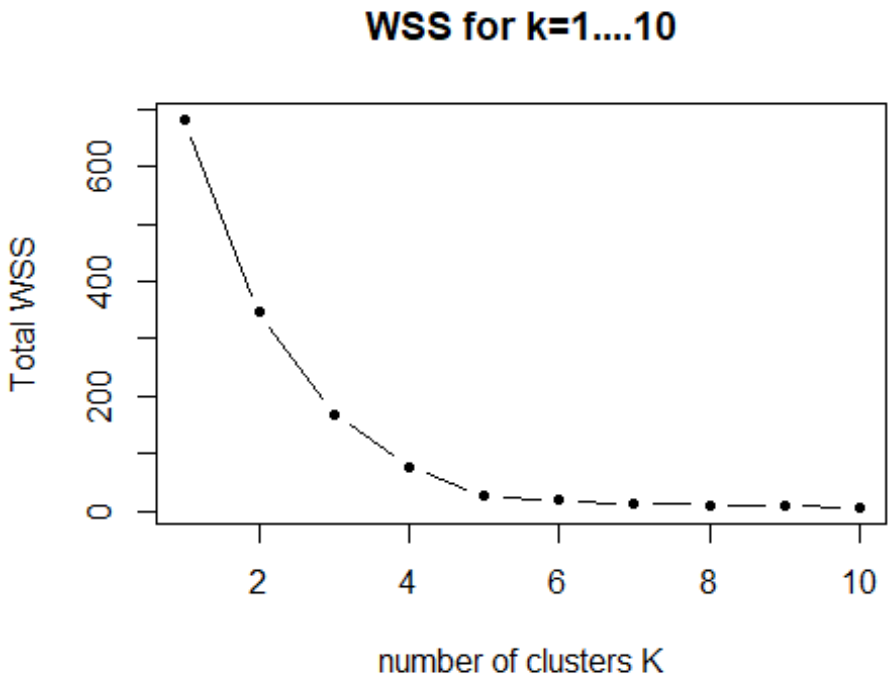

KMEANS Clustering
X2
X1
cluster
1
2
3
4
5



The clusters got smaller when increasing the K-value meaning the total distance between each value in each cluster got smaller

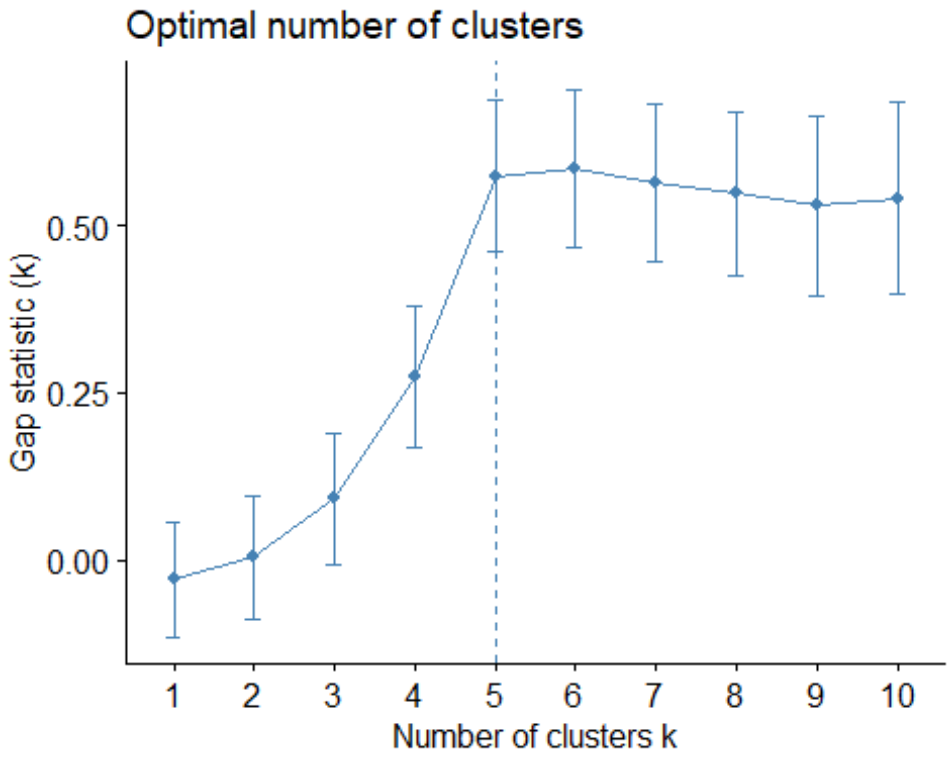
Question 3 (d):

```
k.max <- 10
sim.wss <- numeric(k.max)
for(i in 1:k.max){
  sim.wss[i] <- eclust(x,
                       FUNcluster = "kmeans",
                       k = i,
                       nstart = 50,
                       graph = 0)$tot.withinss
}
plot(sim.wss, pch = 20, type = "b",
     main = "WSS for k=1....10",
     ylab = "Total WSS",
     xlab = "number of clusters K")
```



The optimal k value seems to be k=5
Question 3 (e):

```
sim.gap <- fviz_nbclust(x,
                        kmeans,
                        nstart = 50,
                        nboot = 50,
                        method = "gap_stat")
plot(sim.gap)
```



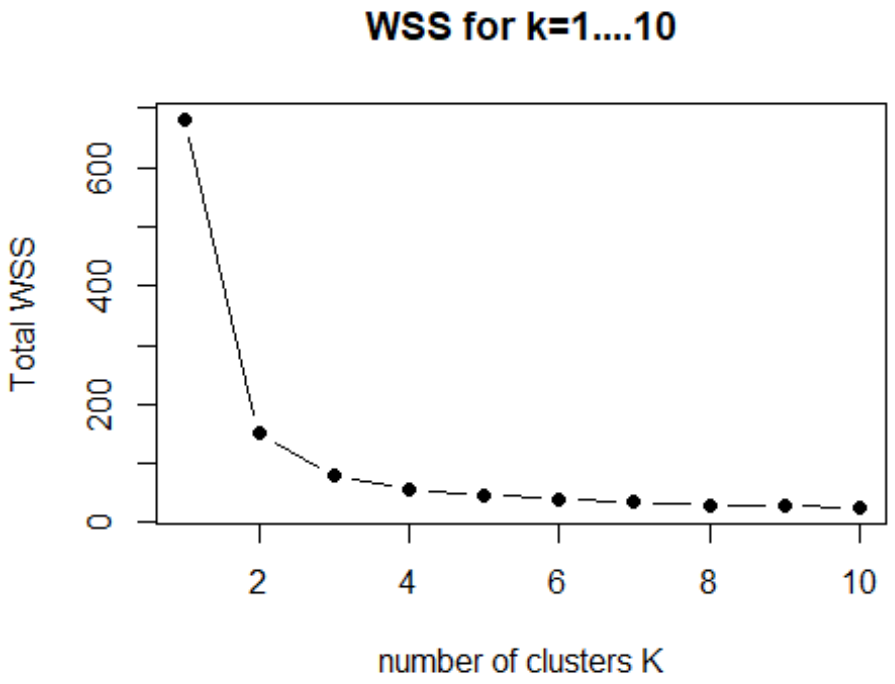
The optimal k values is 5 which also match the number of separated clusters in the simulated data

Question 4 (a):

```
newIris <- iris[-5]
k.max <- 10
iris.wss <- numeric(k.max)
for(i in 1:k.max){
  iris.wss[i] <- eclust(newIris,
                        FUNcluster = "kmeans",
                        k = i,
                        nstart = 50,
                        graph = 0)$tot.withinss
}
print(iris.wss)
```

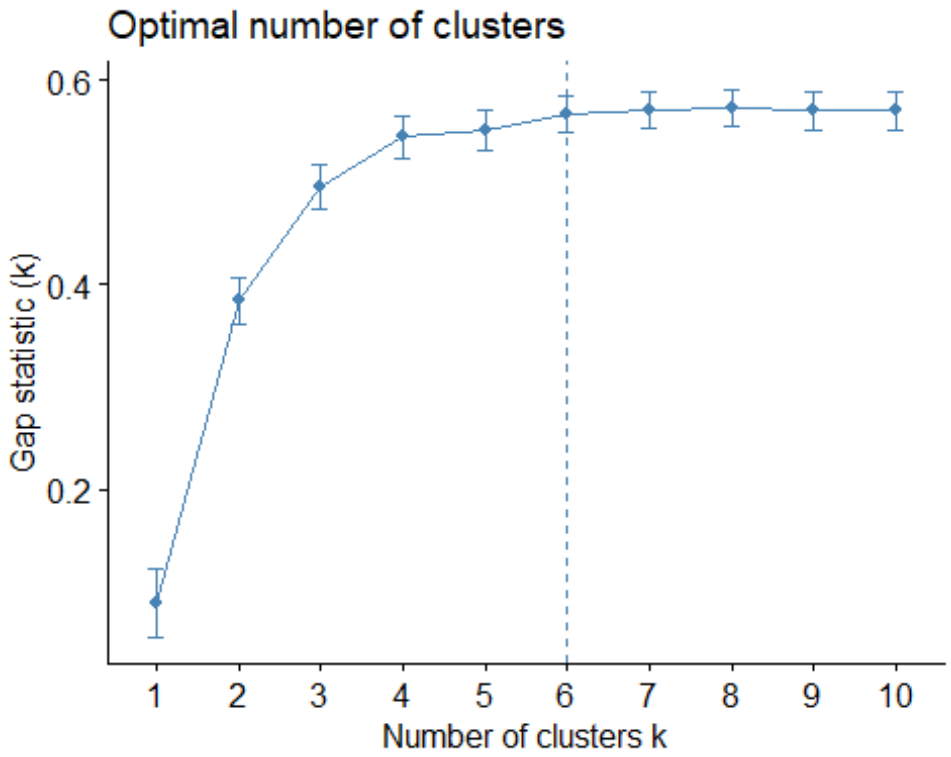
```
##  [1] 681.37060 152.34795  78.85144  57.22847  46.44618  39.03999  34.29823
##  [8]  29.98894  27.78609  25.83405
```

```
plot(iris.wss, pch = 19, type = "b",
     main = "WSS for k=1....10",
     ylab = "Total WSS",
     xlab = "number of clusters K")
```



Question 4 (b):

```
iris.gap <- fviz_nbclust(newIris,
                         kmeans,
                         nstart = 50,
                         nboot = 50,
                         method = "gap_stat")
plot(iris.gap)
```



The optimal K value seems to be k=6
Question 4 (c):

```
iris.clust <- eclust(newIris,
                     FUNcluster = "kmeans",
                     nstart = 50,
                    nboot  = 50,
                     graph = 0)
table(iris$Species[iris.clust$cluster == 6])

##
##     setosa versicolor  virginica
##         28          0          0

table(iris$Species[iris.clust$cluster == 5])

##
##     setosa versicolor  virginica
##         22          0          0

table(iris$Species[iris.clust$cluster == 4])

##
##     setosa versicolor  virginica
##          0         24          1

table(iris$Species[iris.clust$cluster == 3])
```

```
##
##     setosa versicolor  virginica
##          0          0         12
```

```
table(iris$Species[iris.clust$cluster == 2])
```

```
##
##     setosa versicolor  virginica
##          0         26         13
```

```
table(iris$Species[iris.clust$cluster == 1])
```

```
##
##     setosa versicolor  virginica
##          0          0         24
```

Question 4 (d):

```
scale(newIris)
```

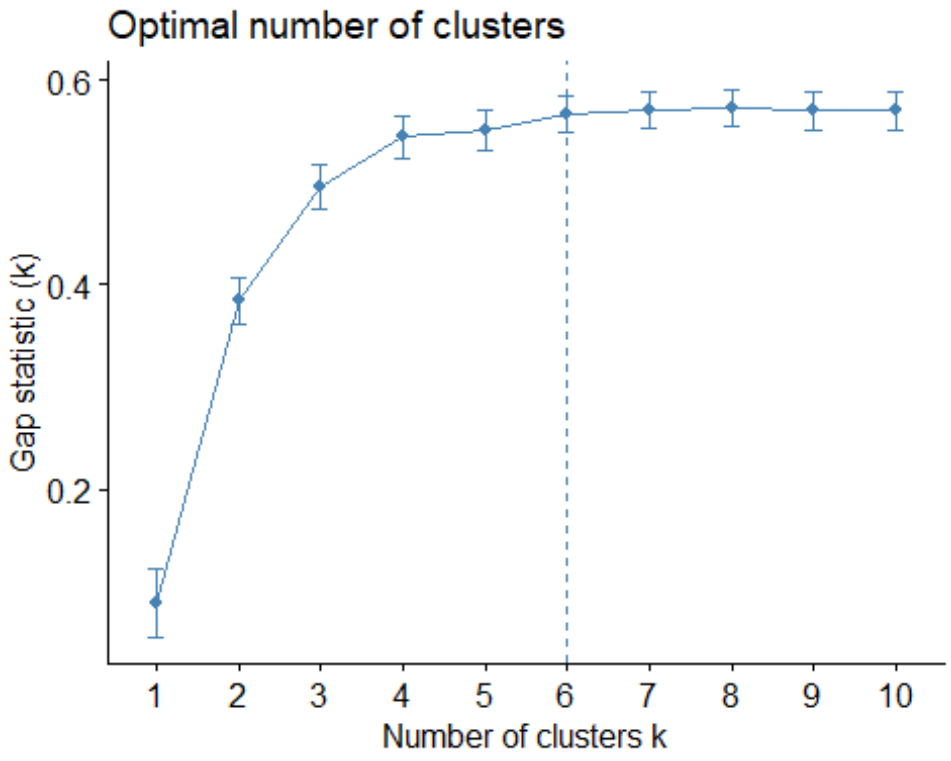
```
##        Sepal.Length Sepal.Width Petal.Length   Petal.Width
##   [1,]  -0.89767388  1.01560199  -1.33575163 -1.3110521482
##   [2,]  -1.13920048 -0.13153881  -1.33575163 -1.3110521482
##   [3,]  -1.38072709  0.32731751  -1.39239929 -1.3110521482
##   [4,]  -1.50149039  0.09788935  -1.27910398 -1.3110521482
##   [5,]  -1.01843718  1.24503015  -1.33575163 -1.3110521482
##   [6,]  -0.53538397  1.93331463  -1.16580868 -1.0486667950
##   [7,]  -1.50149039  0.78617383  -1.33575163 -1.1798594716
##   [8,]  -1.01843718  0.78617383  -1.27910398 -1.3110521482
##   [9,]  -1.74301699 -0.36096697  -1.33575163 -1.3110521482
##  [10,]  -1.13920048  0.09788935  -1.27910398 -1.4422448248
##  [11,]  -0.53538397  1.47445831  -1.27910398 -1.3110521482
##  [12,]  -1.25996379  0.78617383  -1.22245633 -1.3110521482
##  [13,]  -1.25996379 -0.13153881  -1.33575163 -1.4422448248
##  [14,]  -1.86378030 -0.13153881  -1.50569459 -1.4422448248
##  [15,]  -0.05233076  2.16274279  -1.44904694 -1.3110521482
##  [16,]  -0.17309407  3.08045544  -1.27910398 -1.0486667950
##  [17,]  -0.53538397  1.93331463  -1.39239929 -1.0486667950
##  [18,]  -0.89767388  1.01560199  -1.33575163 -1.1798594716
##  [19,]  -0.17309407  1.70388647  -1.16580868 -1.1798594716
##  [20,]  -0.89767388  1.70388647  -1.27910398 -1.1798594716
##  [21,]  -0.53538397  0.78617383  -1.16580868 -1.3110521482
##  [22,]  -0.89767388  1.47445831  -1.27910398 -1.0486667950
##  [23,]  -1.50149039  1.24503015  -1.56234224 -1.3110521482
##  [24,]  -0.89767388  0.55674567  -1.16580868 -0.9174741184
##  [25,]  -1.25996379  0.78617383  -1.05251337 -1.3110521482
##  [26,]  -1.01843718 -0.13153881  -1.22245633 -1.3110521482
##  [27,]  -1.01843718  0.78617383  -1.22245633 -1.0486667950
##  [28,]  -0.77691058  1.01560199  -1.27910398 -1.3110521482
##  [29,]  -0.77691058  0.78617383  -1.33575163 -1.3110521482
```

```
##  [30,]  -1.38072709  0.32731751  -1.22245633 -1.3110521482
##  [31,]  -1.25996379  0.09788935  -1.22245633 -1.3110521482
##  [32,]  -0.53538397  0.78617383  -1.27910398 -1.0486667950
##  [33,]  -0.77691058  2.39217095  -1.27910398 -1.4422448248
##  [34,]  -0.41462067  2.62159911  -1.33575163 -1.3110521482
##  [35,]  -1.13920048  0.09788935  -1.27910398 -1.3110521482
##  [36,]  -1.01843718  0.32731751  -1.44904694 -1.3110521482
##  [37,]  -0.41462067  1.01560199  -1.39239929 -1.3110521482
##  [38,]  -1.13920048  1.24503015  -1.33575163 -1.4422448248
##  [39,]  -1.74301699 -0.13153881  -1.39239929 -1.3110521482
##  [40,]  -0.89767388  0.78617383  -1.27910398 -1.3110521482
##  [41,]  -1.01843718  1.01560199  -1.39239929 -1.1798594716
##  [42,]  -1.62225369 -1.73753594  -1.39239929 -1.1798594716
##  [43,]  -1.74301699  0.32731751  -1.39239929 -1.3110521482
##  [44,]  -1.01843718  1.01560199  -1.22245633 -0.7862814418
##  [45,]  -0.89767388  1.70388647  -1.05251337 -1.0486667950
##  [46,]  -1.25996379 -0.13153881  -1.33575163 -1.1798594716
##  [47,]  -0.89767388  1.70388647  -1.22245633 -1.3110521482
##  [48,]  -1.50149039  0.32731751  -1.33575163 -1.3110521482
##  [49,]  -0.65614727  1.47445831  -1.27910398 -1.3110521482
##  [50,]  -1.01843718  0.55674567  -1.33575163 -1.3110521482
##  [51,]   1.39682886  0.32731751   0.53362088  0.2632599711
##  [52,]   0.67224905  0.32731751   0.42032558  0.3944526477
##  [53,]   1.27606556  0.09788935   0.64691619  0.3944526477
##  [54,]  -0.41462067 -1.73753594   0.13708732  0.1320672944
##  [55,]   0.79301235 -0.59039513   0.47697323  0.3944526477
##  [56,]  -0.17309407 -0.59039513   0.42032558  0.1320672944
##  [57,]   0.55148575  0.55674567   0.53362088  0.5256453243
##  [58,]  -1.13920048 -1.50810778  -0.25944625 -0.2615107354
##  [59,]   0.91377565 -0.36096697   0.47697323  0.1320672944
##  [60,]  -0.77691058 -0.81982329   0.08043967  0.2632599711
##  [61,]  -1.01843718 -2.42582042  -0.14615094 -0.2615107354
##  [62,]   0.06843254 -0.13153881   0.25038262  0.3944526477
##  [63,]   0.18919584 -1.96696410   0.13708732 -0.2615107354
##  [64,]   0.30995914 -0.36096697   0.53362088  0.2632599711
##  [65,]  -0.29385737 -0.36096697  -0.08950329  0.1320672944
##  [66,]   1.03453895  0.09788935   0.36367793  0.2632599711
##  [67,]  -0.29385737 -0.13153881   0.42032558  0.3944526477
##  [68,]  -0.05233076 -0.81982329   0.19373497 -0.2615107354
##  [69,]   0.43072244 -1.96696410   0.42032558  0.3944526477
##  [70,]  -0.29385737 -1.27867961   0.08043967 -0.1303180588
##  [71,]   0.06843254  0.32731751   0.59026853  0.7880306775
##  [72,]   0.30995914 -0.59039513   0.13708732  0.1320672944
##  [73,]   0.55148575 -1.27867961   0.64691619  0.3944526477
##  [74,]   0.30995914 -0.59039513   0.53362088  0.0008746178
##  [75,]   0.67224905 -0.36096697   0.30703027  0.1320672944
##  [76,]   0.91377565 -0.13153881   0.36367793  0.2632599711
##  [77,]   1.15530226 -0.59039513   0.59026853  0.2632599711
##  [78,]   1.03453895 -0.13153881   0.70356384  0.6568380009
##  [79,]   0.18919584 -0.36096697   0.42032558  0.3944526477
```

```
##  [80,]  -0.17309407 -1.04925145  -0.14615094 -0.2615107354
##  [81,]  -0.41462067 -1.50810778   0.02379201 -0.1303180588
##  [82,]  -0.41462067 -1.50810778  -0.03285564 -0.2615107354
##  [83,]  -0.05233076 -0.81982329   0.08043967  0.0008746178
##  [84,]   0.18919584 -0.81982329   0.76021149  0.5256453243
##  [85,]  -0.53538397 -0.13153881   0.42032558  0.3944526477
##  [86,]   0.18919584  0.78617383   0.42032558  0.5256453243
##  [87,]   1.03453895  0.09788935   0.53362088  0.3944526477
##  [88,]   0.55148575 -1.73753594   0.36367793  0.1320672944
##  [89,]  -0.29385737 -0.13153881   0.19373497  0.1320672944
##  [90,]  -0.41462067 -1.27867961   0.13708732  0.1320672944
##  [91,]  -0.41462067 -1.04925145   0.36367793  0.0008746178
##  [92,]   0.30995914 -0.13153881   0.47697323  0.2632599711
##  [93,]  -0.05233076 -1.04925145   0.13708732  0.0008746178
##  [94,]  -1.01843718 -1.73753594  -0.25944625 -0.2615107354
##  [95,]  -0.29385737 -0.81982329   0.25038262  0.1320672944
##  [96,]  -0.17309407 -0.13153881   0.25038262  0.0008746178
##  [97,]  -0.17309407 -0.36096697   0.25038262  0.1320672944
##  [98,]   0.43072244 -0.36096697   0.30703027  0.1320672944
##  [99,]  -0.89767388 -1.27867961  -0.42938920 -0.1303180588
## [100,]  -0.17309407 -0.59039513   0.19373497  0.1320672944
## [101,]   0.55148575  0.55674567   1.27004036  1.7063794137
## [102,]  -0.05233076 -0.81982329   0.76021149  0.9192233541
## [103,]   1.51759216 -0.13153881   1.21339271  1.1816087073
## [104,]   0.55148575 -0.36096697   1.04344975  0.7880306775
## [105,]   0.79301235 -0.13153881   1.15674505  1.3128013839
## [106,]   2.12140867 -0.13153881   1.60992627  1.1816087073
## [107,]  -1.13920048 -1.27867961   0.42032558  0.6568380009
## [108,]   1.75911877 -0.36096697   1.43998331  0.7880306775
## [109,]   1.03453895 -1.27867961   1.15674505  0.7880306775
## [110,]   1.63835547  1.24503015   1.32668801  1.7063794137
## [111,]   0.79301235  0.32731751   0.76021149  1.0504160307
## [112,]   0.67224905 -0.81982329   0.87350679  0.9192233541
## [113,]   1.15530226 -0.13153881   0.98680210  1.1816087073
## [114,]  -0.17309407 -1.27867961   0.70356384  1.0504160307
## [115,]  -0.05233076 -0.59039513   0.76021149  1.5751867371
## [116,]   0.67224905  0.32731751   0.87350679  1.4439940605
## [117,]   0.79301235 -0.13153881   0.98680210  0.7880306775
## [118,]   2.24217198  1.70388647   1.66657392  1.3128013839
## [119,]   2.24217198 -1.04925145   1.77986923  1.4439940605
## [120,]   0.18919584 -1.96696410   0.70356384  0.3944526477
## [121,]   1.27606556  0.32731751   1.10009740  1.4439940605
## [122,]  -0.29385737 -0.59039513   0.64691619  1.0504160307
## [123,]   2.24217198 -0.59039513   1.66657392  1.0504160307
## [124,]   0.55148575 -0.81982329   0.64691619  0.7880306775
## [125,]   1.03453895  0.55674567   1.10009740  1.1816087073
## [126,]   1.63835547  0.32731751   1.27004036  0.7880306775
## [127,]   0.43072244 -0.59039513   0.59026853  0.7880306775
## [128,]   0.30995914 -0.13153881   0.64691619  0.7880306775
## [129,]   0.67224905 -0.59039513   1.04344975  1.1816087073
```

```
## [130,]   1.63835547 -0.13153881   1.15674505  0.5256453243
## [131,]   1.87988207 -0.59039513   1.32668801  0.9192233541
## [132,]   2.48369858  1.70388647   1.49663097  1.0504160307
## [133,]   0.67224905 -0.59039513   1.04344975  1.3128013839
## [134,]   0.55148575 -0.59039513   0.76021149  0.3944526477
## [135,]   0.30995914 -1.04925145   1.04344975  0.2632599711
## [136,]   2.24217198 -0.13153881   1.32668801  1.4439940605
## [137,]   0.55148575  0.78617383   1.04344975  1.5751867371
## [138,]   0.67224905  0.09788935   0.98680210  0.7880306775
## [139,]   0.18919584 -0.13153881   0.59026853  0.7880306775
## [140,]   1.27606556  0.09788935   0.93015445  1.1816087073
## [141,]   1.03453895  0.09788935   1.04344975  1.5751867371
## [142,]   1.27606556  0.09788935   0.76021149  1.4439940605
## [143,]  -0.05233076 -0.81982329   0.76021149  0.9192233541
## [144,]   1.15530226  0.32731751   1.21339271  1.4439940605
## [145,]   1.03453895  0.55674567   1.10009740  1.7063794137
## [146,]   1.03453895 -0.13153881   0.81685914  1.4439940605
## [147,]   0.55148575 -1.27867961   0.70356384  0.9192233541
## [148,]   0.79301235 -0.13153881   0.81685914  1.0504160307
## [149,]   0.43072244  0.78617383   0.93015445  1.4439940605
## [150,]   0.06843254 -0.13153881   0.76021149  0.7880306775
## attr(,"scaled:center")
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
## attr(,"scaled:scale")
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##    0.8280661    0.4358663    1.7652982    0.7622377
```

```
iris.gap <- fviz_nbclust(newIris,
                         kmeans,
                         nstart = 50,
                         nboot = 50,
                         method = "gap_stat")
plot(iris.gap)
```



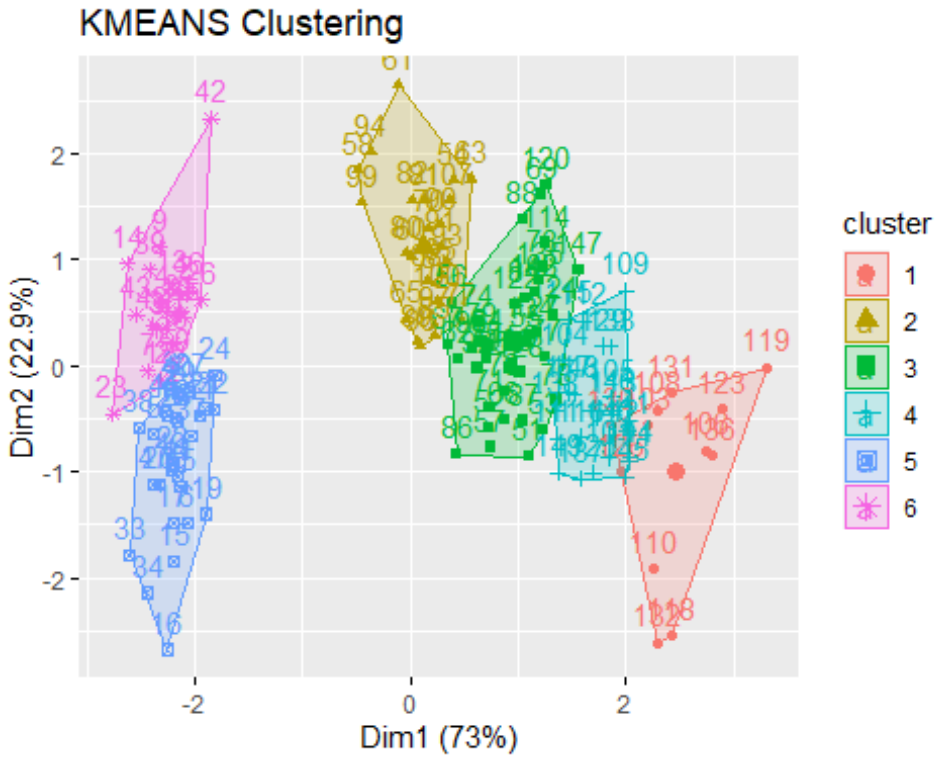
Question 4 (e):

```
iris.clust <- eclust(newIris,
                     FUNcluster = "kmeans",
                     nstart = 50,
                    nboot  = 50,
                     graph = 0)
table(iris$Species[iris.clust$cluster == 6])

##
##     setosa versicolor  virginica
##         28          0          0

table(iris$Species[iris.clust$cluster == 5])

##
##     setosa versicolor  virginica
##         22          0          0

table(iris$Species[iris.clust$cluster == 4])

##
##     setosa versicolor  virginica
##          0         24          1

table(iris$Species[iris.clust$cluster == 3])
```

```
##
##     setosa versicolor  virginica
##          0          0         12

table(iris$Species[iris.clust$cluster == 2])

##
##     setosa versicolor  virginica
##          0         26         13

table(iris$Species[iris.clust$cluster == 1])

##
##     setosa versicolor  virginica
##          0          0         24
```

Question 4 (f):

```
iris.optimal <- eclust(newIris,
                       FUNcluster = "kmeans",
                       k = 6,
                       nstart = 50)
```

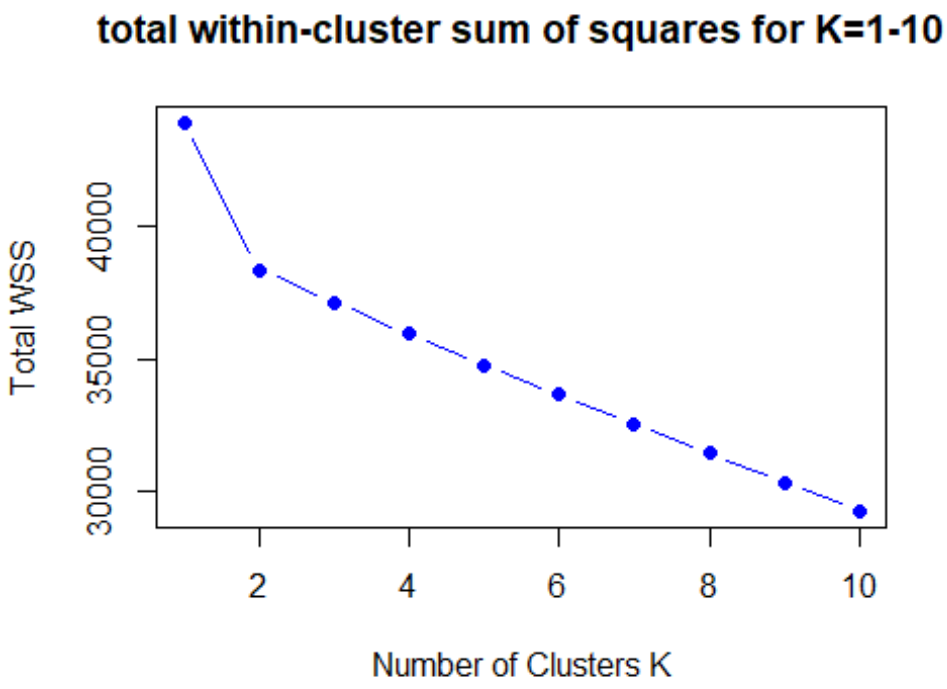


Question 5 (a):

```
temp <- read.csv("E:/fall22/math4323/Ch10Ex11.csv", header = F)
data<-t(temp)
data <- data.frame(data)
```

Question 5 (b):

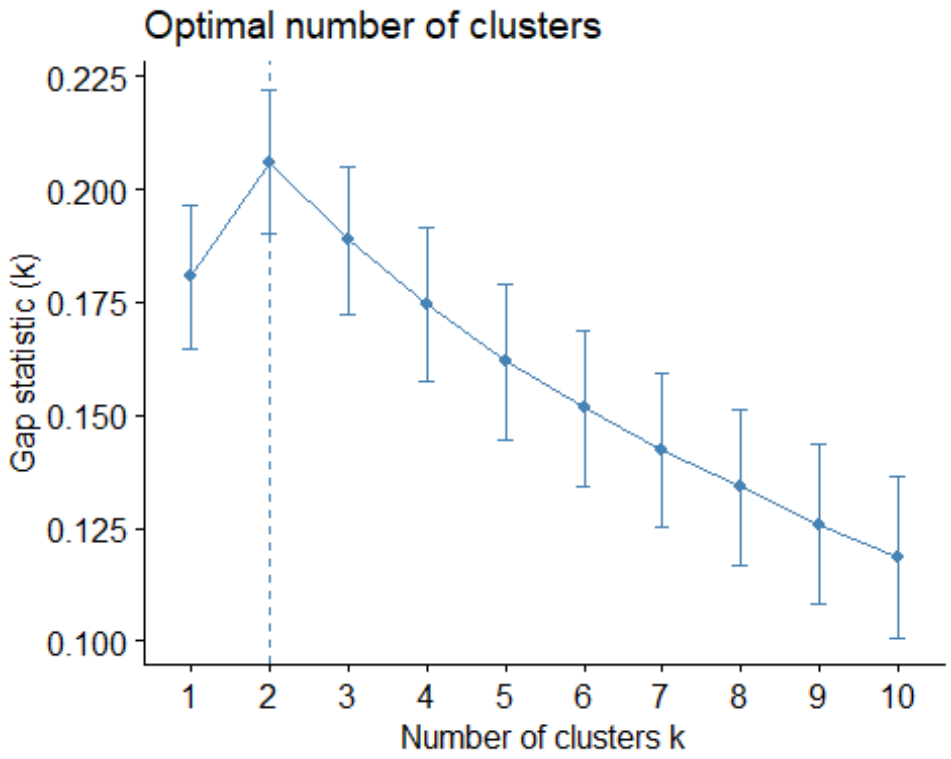
```
library(factoextra)
k.max <- 10
wss <- numeric(k.max)
for(i in 1:k.max){
  wss[i] <- eclust(data,
                   FUNcluster = "kmeans",
                   k=i,
                   nstart=50,
                   graph = 0)$tot.withinss
}
plot(wss, type = "b",
     pch = 19, col = "blue",
     main = "total within-cluster sum of squares for K=1-10",
     xlab = "Number of Clusters K",
     ylab = "Total WSS")
```



Yes there is an elbow at K=2
Question 5 (c):

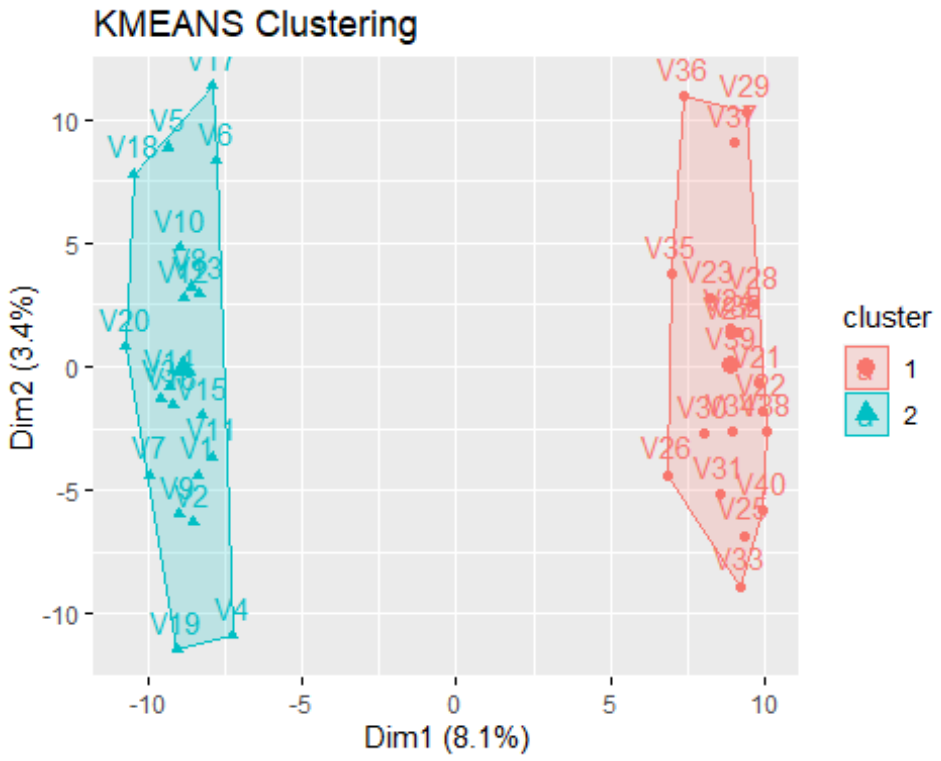
```
data.gap <- fviz_nbclust(data,
                         kmeans,
                         nstart = 50,
                         nboot = 20,
```

```
                          method = "gap_stat")
plot(data.gap)
```



The optimal k value is k=2
Question 5 (d):

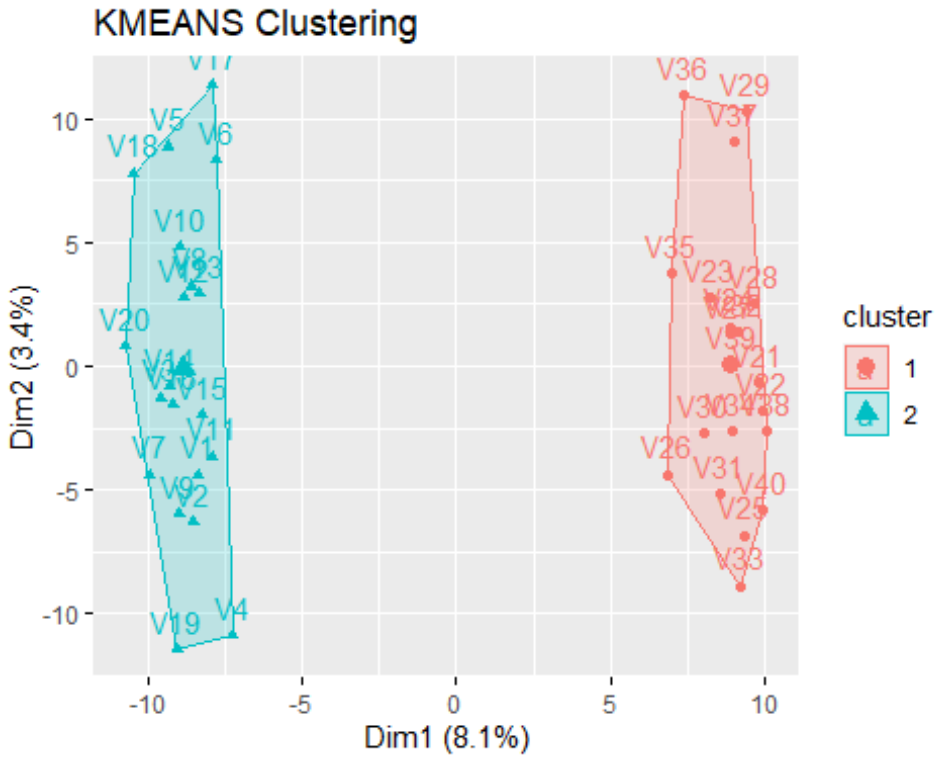
```
data.optimal <- eclust(data,
                       FUNcluster = "kmeans",
                       nboot = 20,
                       nstart = 50)
```




```
table(temp[1,data.optimal$cluster==2])

## , , V3 = -0.9750051, V4 = 1.417504, V5 = 0.8188148, V6 = 0.3162937, V7 = -
0.02496682, V8 = -0.063966, V9 = 0.03149702, V10 = -0.3503106, V11 = -
0.7227299, V12 = -0.2819547, V13 = 1.337515, V14 = 0.7019798, V15 = 1.007616,
V16 = -0.4653828, V17 = 0.6385951, V18 = 0.2867807, V19 = -0.2270782, V20 = -
0.2200452
##
##             V2
## V1           0.4418028
##   -0.9619334         1

table(temp[1,data.optimal$cluster==1])

## , , V23 = -1.864262, V24 = -0.5005122, V25 = -1.325008, V26 = 1.063411,
V27 = -0.2963712, V28 = -0.1216457, V29 = 0.08516605, V30 = 0.6241764, V31 =
-0.5095915, V32 = -0.2167255, V33 = -0.05550597, V34 = -0.4844491, V35 = -
0.5215811, V36 = 1.949135, V37 = 1.324335, V38 = 0.4681471, V39 = 1.0611, V40
= 1.65597
##
##            V22
## V21         -0.1085056
##   -1.242573          1
```

Yes the answer is close to part c
Question 5 (e):

```
data.optimal <- eclust(data,
                       FUNcluster = "kmeans",
                       k=2,
                       nstart = 50)
```



Yes one cluster only contain healthy and the other contains the diseased group