Cluster Analysis

Vazgen Tadevosyan

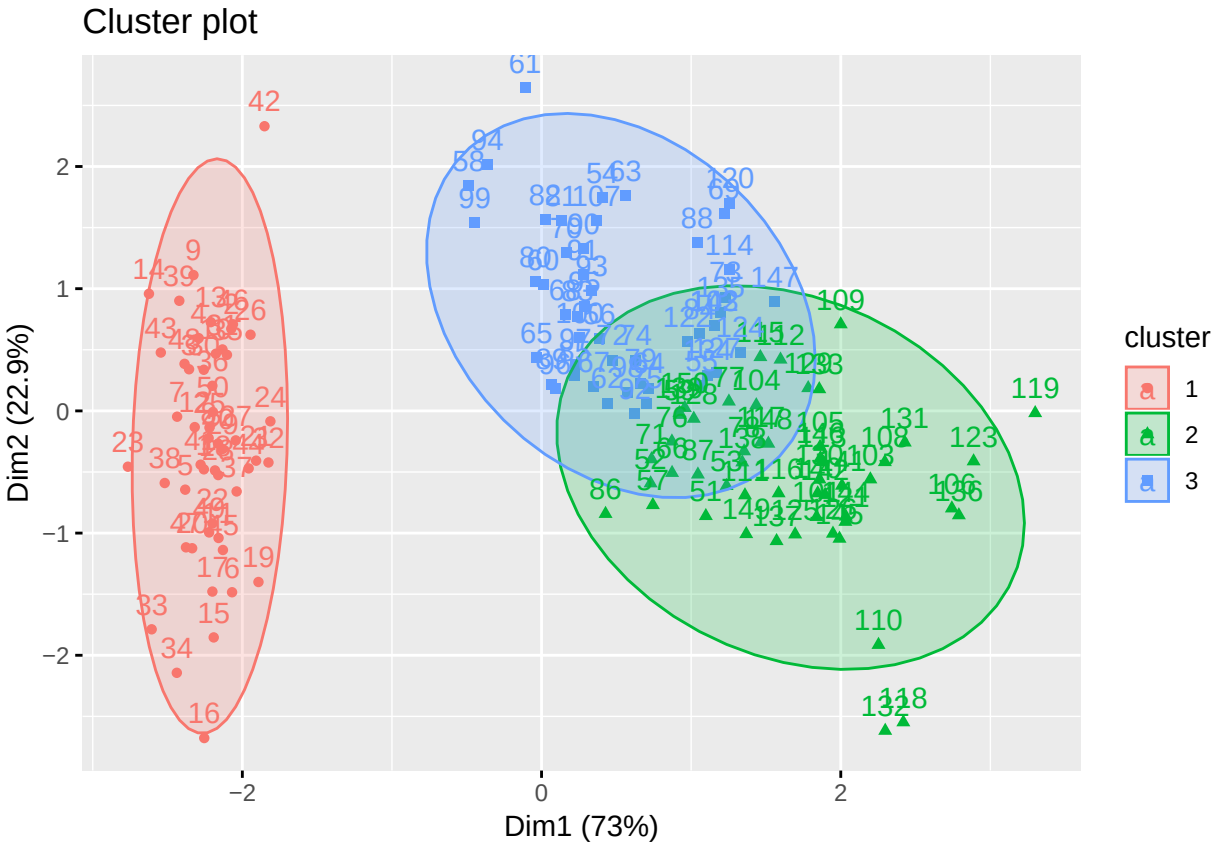
Problem 1 (7 pt.)

a. What are the differences among exclusive, overlapping and fuzzy clustering? Bring an example of fuzzy clustering with k=2. Use the function funny() from library {cluster} and data visualization techniques from package {factoextra} to show your results. Show the membership matrix. Which of your observations belongs to both clusters. Exclusive clustering is as the name suggests and stipulates that each data object can only exist in one cluster. Overlapping allows data objects to be grouped in 2 or more clusters. In Fuzzy clustering every data object belongs to every cluster.the major difference is that the data objects has a membership weight that is between 0 to 1 where 0 means it does not belong to a given cluster and 1 means it absolutely belongs to the cluster. Fuzzy clustering is also known as probabilistic clustering. Example, lets suppose we do on clustering on customers by dividing them into 2 groups and then label them as loyal or not loyal. So every customer have a certain probability of being each cluster for ex:(0.2 to be loyal, 0.8 to be unloyal).

```
data("iris")
iris.scaled <- scale(iris[, -5])
obj<-cluster::fanny(iris.scaled,k=3)
head(obj$membership)
```

```
##           [,1]       [,2]       [,3]
## [1,] 0.8385968 0.07317706 0.08822614
## [2,] 0.6704536 0.14154548 0.18800091
## [3,] 0.7708430 0.10070648 0.12845057
## [4,] 0.7086764 0.12665562 0.16466795
## [5,] 0.8047709 0.08938753 0.10584154
## [6,] 0.6200074 0.18062228 0.19937035
```

```
fviz_cluster(obj, ellipse.type = "norm")
```



From plot it is seen that there some observtion that could be in two clusters.

b. Suppose we have an example of a data set with 20 observations. We need to cluster the data using the K-means algorithm. After clustering using k=1, 2, 3, 4 and 5 we obtained only one non-empty cluster. How is it possible?

If it returns only one non empty cluster it means that other clusters are empty and the problem of empty clusters occurs when the initial center vectors are such that any two or more of them are either equal or very close to each other. In such a situation, after assignment of data to clusters, data elements will be assigned to only one of the clusters with nearly equal centers, and the others remain empty.Another case will be when all obserations have the same value or it can be because of bad starting centroids.

c. Suppose we have an example of a data set consisting of three natural circular clusters. These clusters have the same number of points and have the same distribution. The centers of clusters lie on a line, the clusters are located such that the center of the middle cluster is equally distant from the other two. Why will not *bisecting* K-means find the correct cluster? Because neither on horisontal nor on vertical axeses it is indivisible. If one???

Problem 2 (6 pt.)

Perform K-means clustering (manually:) using R), with K = 2, using data from the table with 2 features.

- **a. Plot the observations.**
- **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.**
- **c. Compute the centroid for each cluster.**
- **d. Assign each observation to the centroid to which it is closest, in terms of Euclidean**

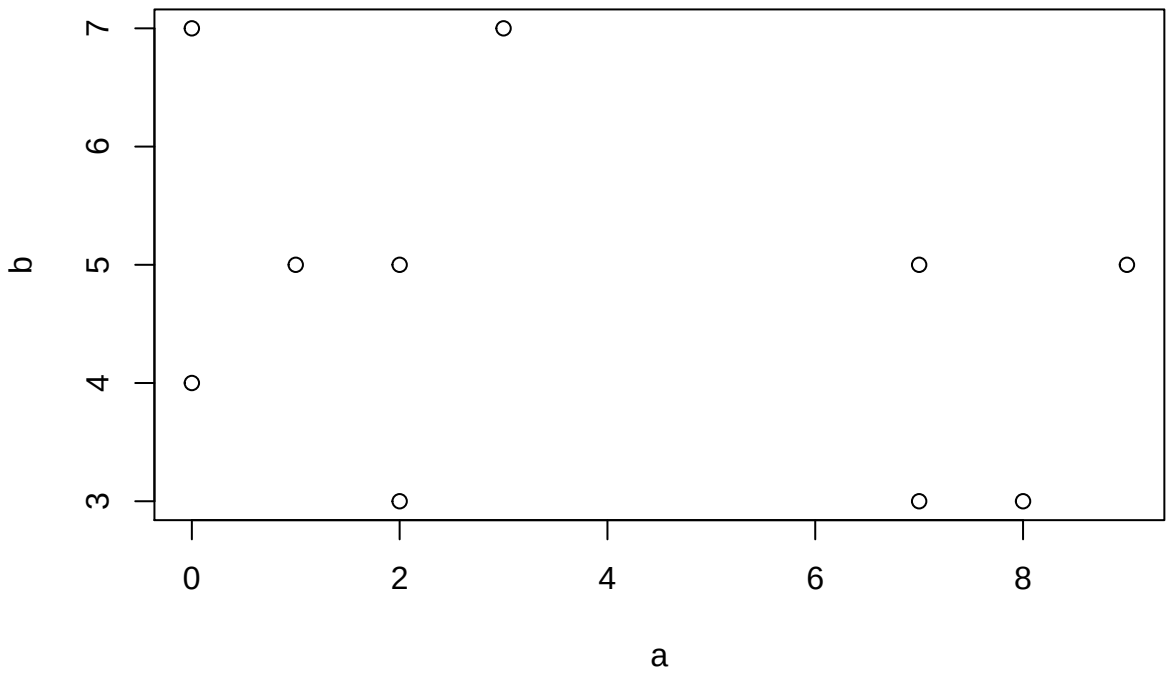
Observations	Var1	Var2
1	2	5
2	2	3
3	8	3
4	0	4
5	7	5
6	0	7
7	1	5
8	7	3
9	3	7
10	9	5

Figure 1:

distance. Report the cluster labels for each observation.

- **e. Repeat (c) and (d) until the answers obtained stop changing.**
- **f. In your plot from (a), color the observations according to the cluster labels obtained.**

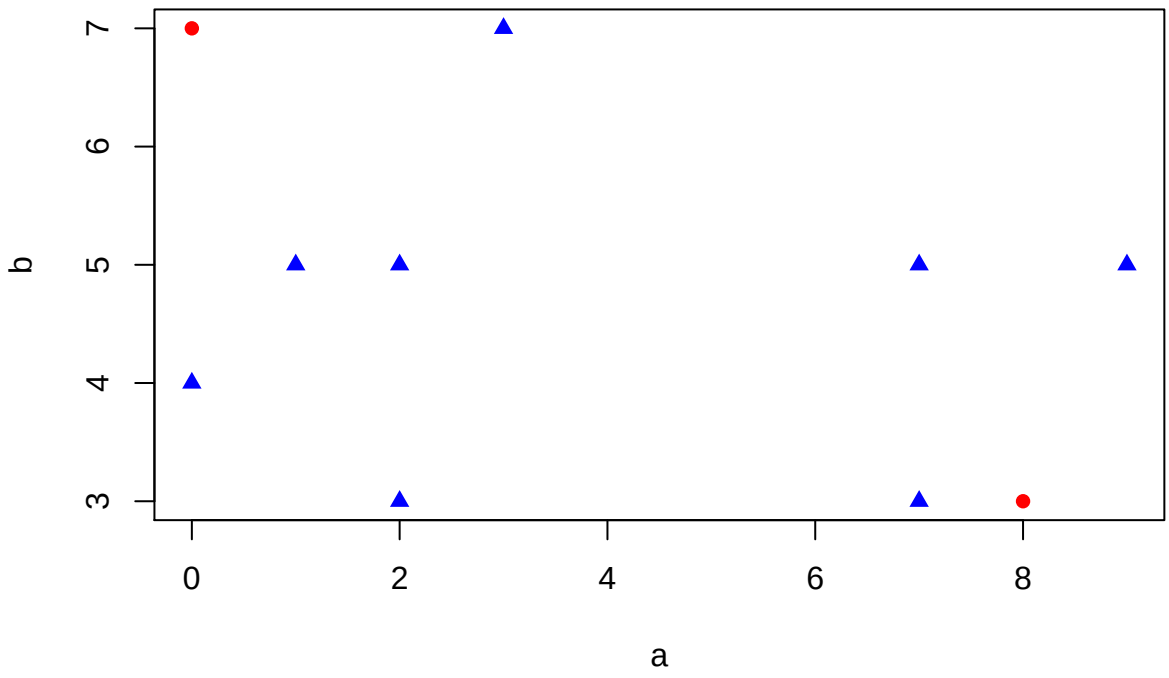
```
n=10
a <- c(2,2,8,0,7,0,1,7,3,9)
b <- c(5,3,3,4,5,7,5,3,7,5)
df <- data.frame(a, b)
plot(a,b)
```




```
clusters = sample(1:2, n, replace = T)
clusters
```

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

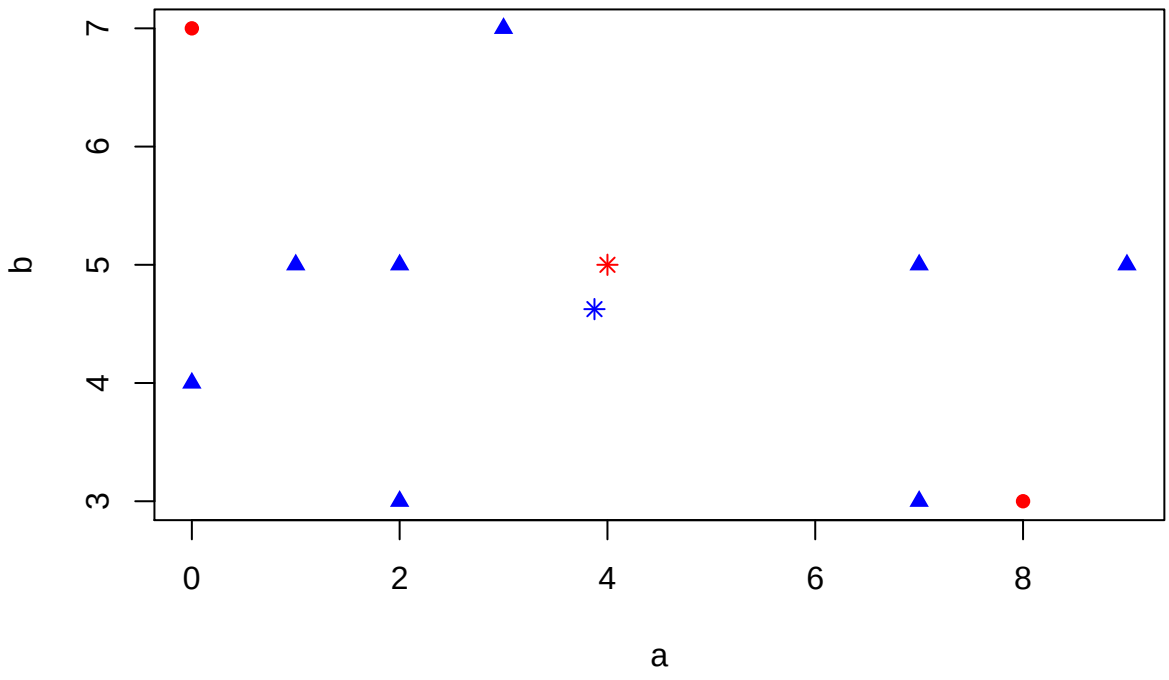
```
col = rep("red", 10)
col[clusters == 2] = "blue"
pch = rep(16, n)
pch[clusters == 2] = 17
plot(df, col = col, pch = pch)
```




```
centroids = aggregate(df, list(Cluster = clusters), mean)
centroids
```

```
##   Cluster     a     b
## 1       1 4.000 5.000
## 2       2 3.875 4.625
```

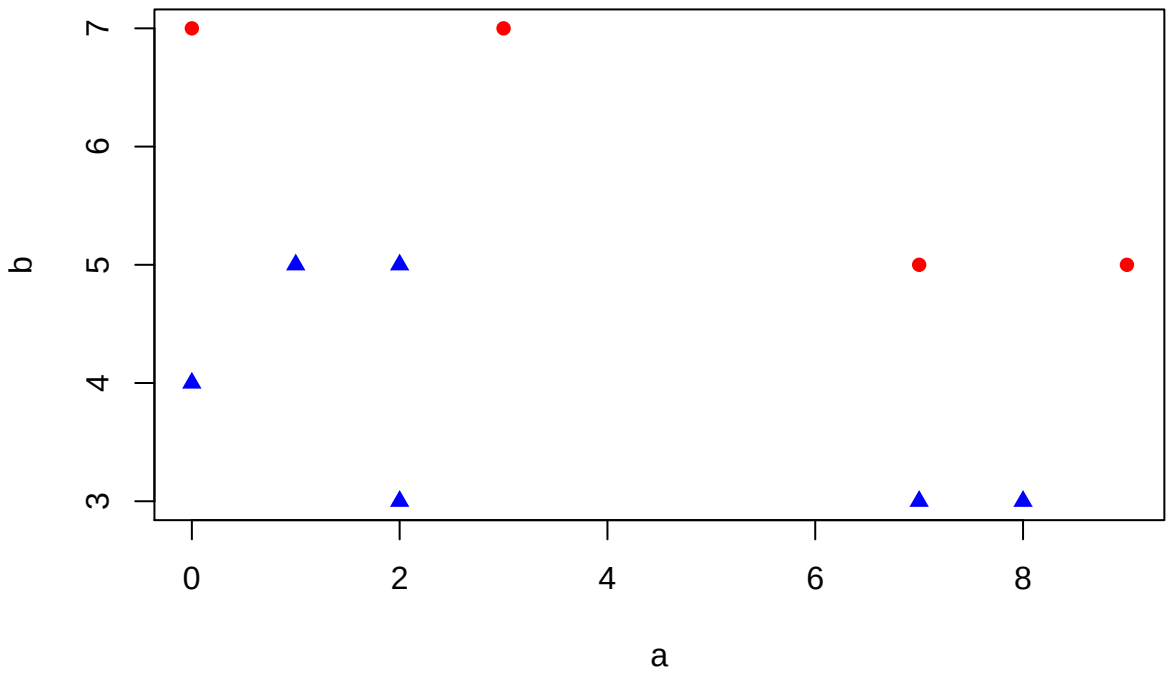
```
plot(df, col = col, pch = pch)
points(centroids[1,2:3], col = "red", pch = 8)
points(centroids[2,2:3], col = "blue", pch = 8)
```




```
library(class)
clusters = knn(centroids[,2:3], df, factor(centroids[,1]))
clusters
```

```
##  [1] 2 2 2 2 1 1 2 2 1 1
## Levels: 1 2
```

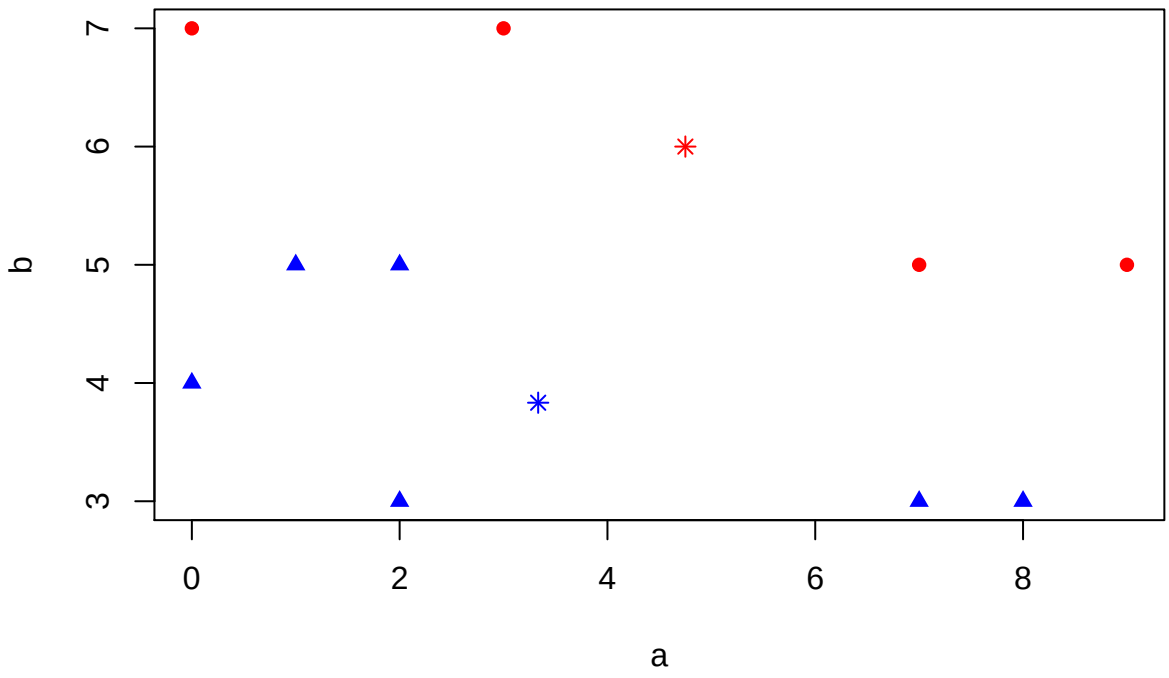
```
col = rep("red", n)
col[clusters == 2] = "blue"
pch = rep(16, n)
pch[clusters == 2] = 17
plot(df, col = col, pch = pch)
```




```
centroids = aggregate(df, list(Cluster = clusters), mean)
centroids
```

```
##   Cluster        a        b
## 1       1 4.750000 6.000000
## 2       2 3.333333 3.833333
```

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



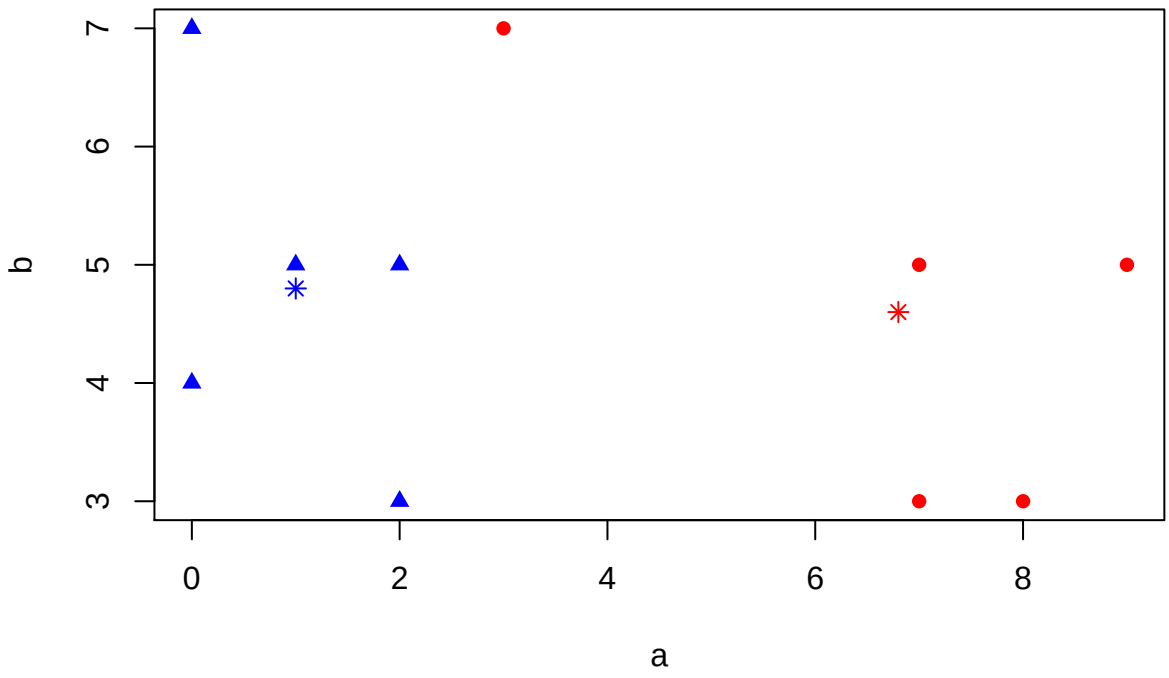

```
clusters = knn(centroids[,2:3], df, factor(centroids[,1]))
clusters
```

```
##  [1] 2 2 1 2 1 2 2 1 1 1
## Levels: 1 2
```

```
centroids = aggregate(df, list(Cluster = clusters), mean)
centroids
```

```
##   Cluster   a   b
## 1       1 6.8 4.6
## 2       2 1.0 4.8
```

```
col = rep("red", n)
col[clusters == 2] = "blue"
pch = rep(16, n)
pch[clusters == 2] = 17
plot(df, col = col, pch = pch)
points(centroids[1,2:3], col = "red", pch = 8)
points(centroids[2,2:3], col = "blue", pch = 8)
```



Problem 3 (7 pt.)

Use the data from the World Value Survey (Wave 6) to understand the disposition of our country among others based on some criteria. The description of the variables and the survey are given with a separate file. Here is the link to obtain more information: http://www.worldvaluessurvey.org/wvs.jsp. Choose the subset from Wave 6 data to perform the cluster analysis. *Note* that you need to use meaningful selections both of variables (based on some theme/problem) and countries.

- **a. Describe how and why you choose your subset of variables and observations. What is your goal?**
- **b. Use all (appropriate) tools/functions from our lecture to cluster (both nested and untested algorithms) the countries. Interpret them. b1. Is your hierarchical clustering stable regards to between clusters distance measures? b2.Compare the results obtained from two different k-means.**
- **c. Make the conclusion (also based on cluster centers).**

```
data<-readRDS("WVS.rds")


data<-data[ , which(names(data) %in% c("V2","V4","V5","V6","V7","V8","V9","V10","V26","V27"))]
dim(data)
```

```
## [1] 89565    10
```

```
colnames(data)<-c("Country","family","friend","leisure","politics","work","religion","happiness","sport
df<-data %>%
  group_by(Country) %>%
```

```
  summarise_all("mean")

df<-data %>%
  group_by(Country) %>%
  summarise_all("mean")%>%filter(Country %in% c(31, 51, 356, 392, 398, 417, 643, 840, 860,32,858,268,791
df$Country<-c("Azerbejan","Argentina","Armenia","Georgia","India","Japan","Kazaxstan","Kghrghrstan","Rus
head(df)
```

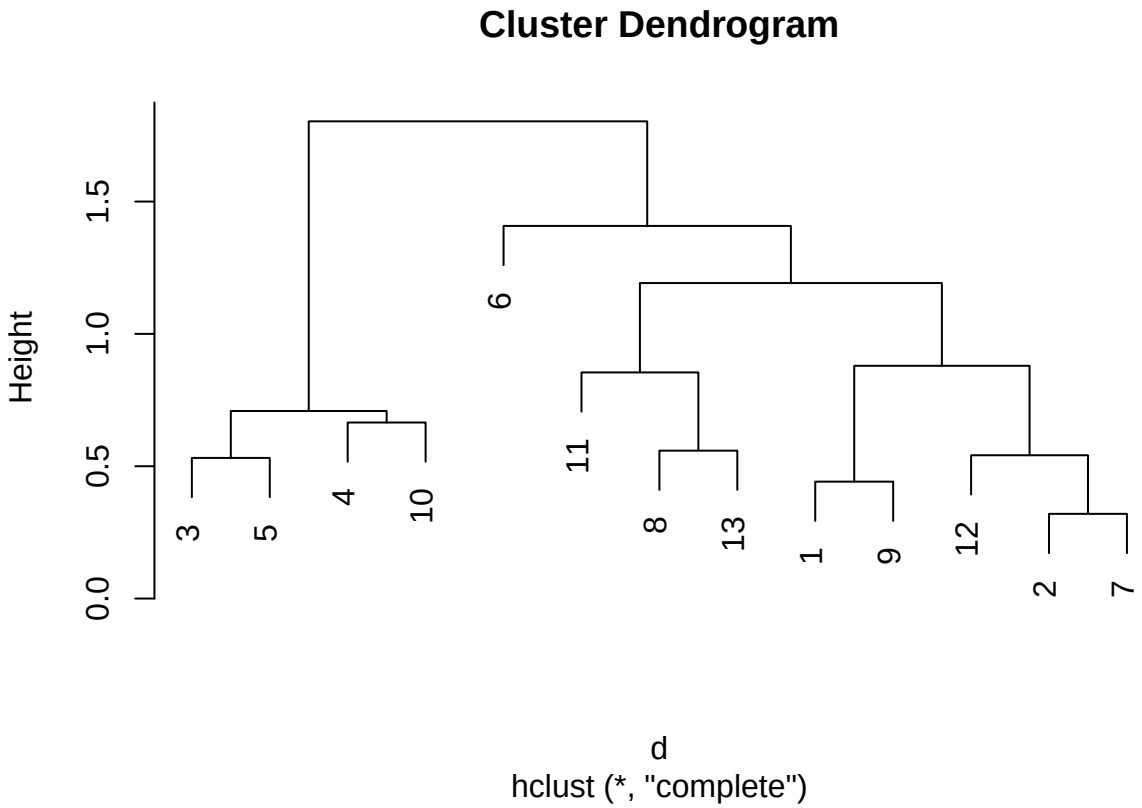
```
## # A tibble: 6 x 10
##   Country family friend leisure politics  work religion happiness  sport
##   <chr>    <dbl>  <dbl>   <dbl>    <dbl> <dbl>    <dbl>     <dbl>  <dbl>
## 1 Azerbe~   1.07   1.79    2.12     2.93  1.50     2.04      1.94 0.0289
## 2 Argent~   1.12   1.56    1.85     2.82  1.53     2.38      1.78 0.304
## 3 Armenia   1.00   1.75    1.89     2.89  1.30     1.53      1.90 0.0336
## 4 Georgia   1.02   1.29    2.01     2.73  1.43     1.19      2.14 0.0116
## 5 India     1.04   1.67    1.97     2.49  1.24     1.38      1.75 0.235
## 6 Japan     1.05   1.55    1.58     1.84  1.44     2.56      1.70 0.346
## # ... with 1 more variable: art <dbl>
```

The chosen variables were selected in a way not to have correleted ones. See in the documentation. Now lets do clusterinbg.

b.

```
df$Country<-NULL
d<-dist(df,method = "euclidian")

c1<-hclust(d,method="complete")
plot(c1)
```



We got plot and as you can see that we are very similar to our neighbors Turkey and Georgia. But interestingly Turkey is not in the same class with Azers. The one that is very different from the others is JAPONIA.

b1.

```
c1$height
```

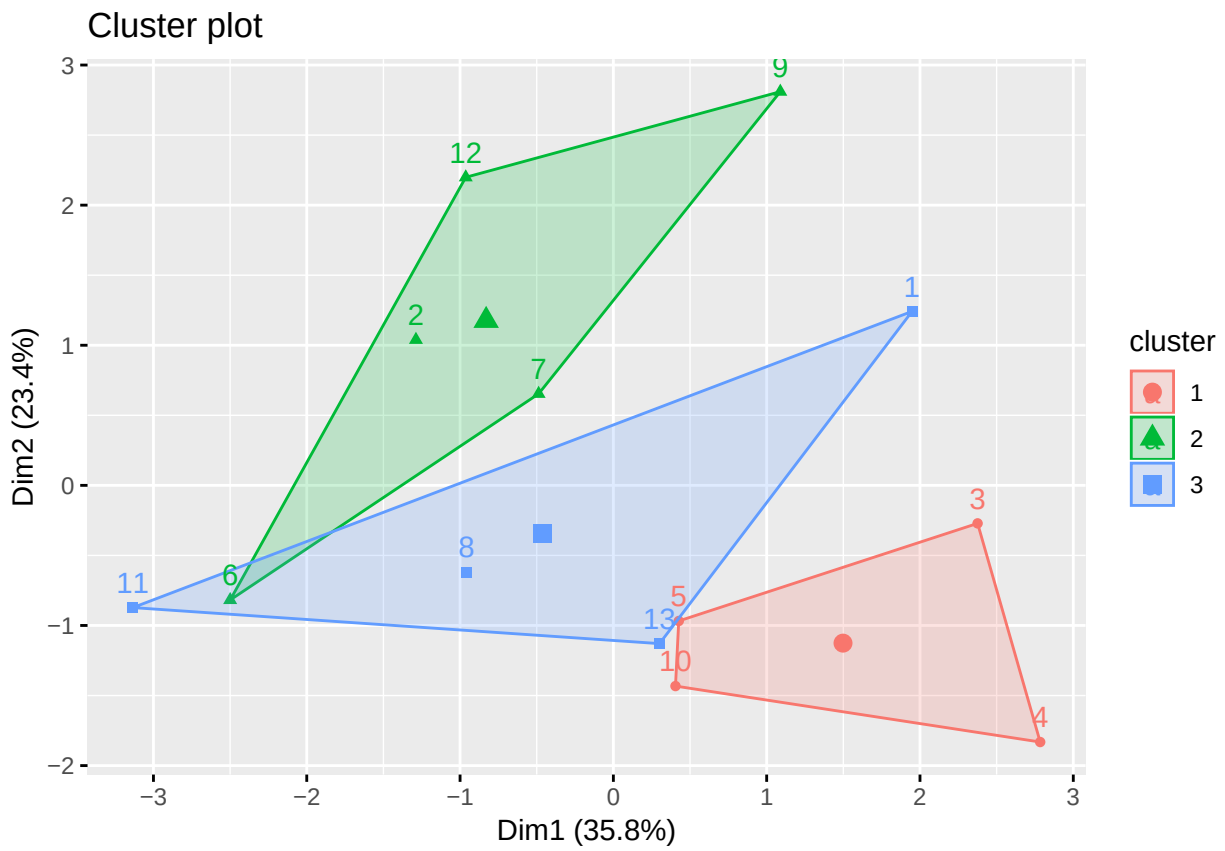
```
##  [1] 0.3200855 0.4417030 0.5313647 0.5413781 0.5588468 0.6651315 0.7086737
##  [8] 0.8543481 0.8794705 1.1921202 1.4078603 1.8030301
```

Yes it is stable.

```
#clust<-
df2<-  df[complete.cases(df),]
km<-kmeans(df2,3)
km$centers
```

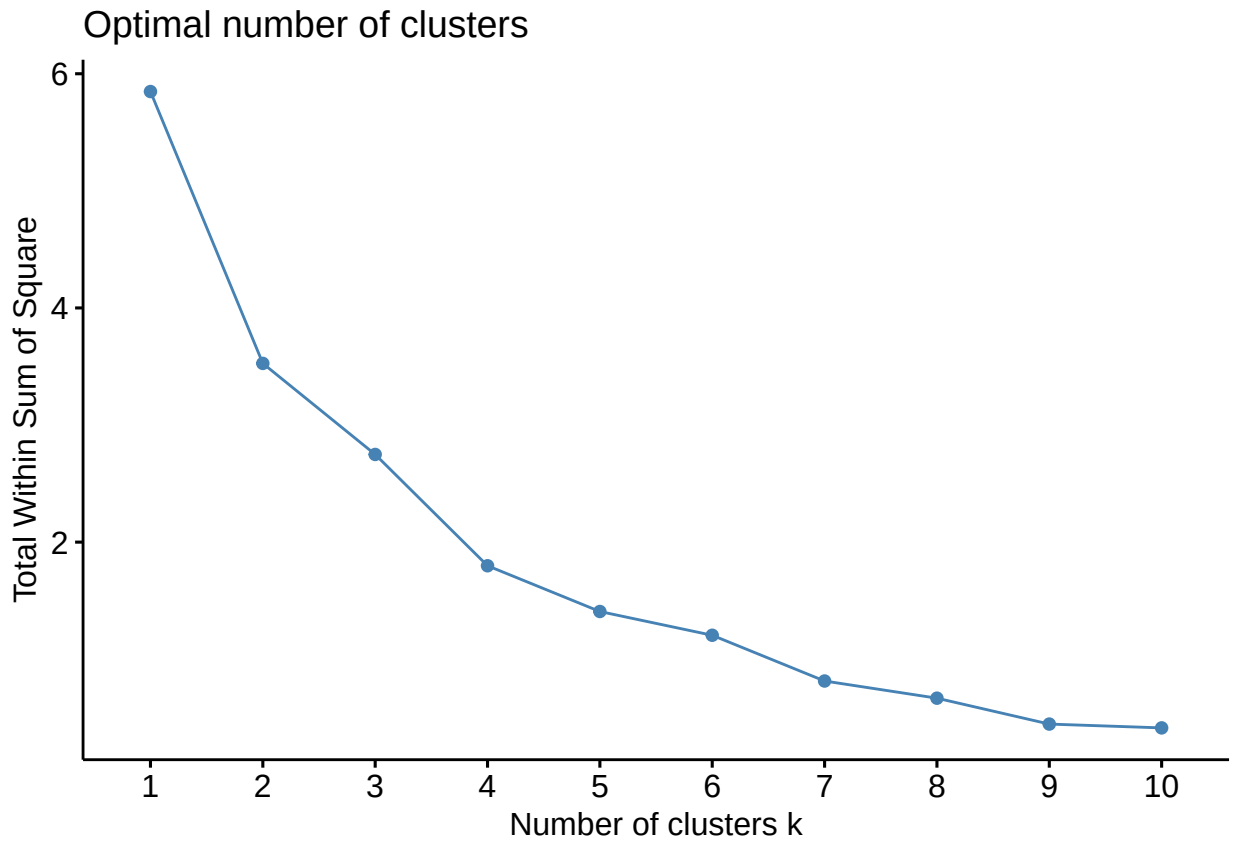
```
##     family   friend  leisure politics     work religion happiness
## 1 1.027295 1.533577 1.885971 2.650629 1.414663 1.383155  1.896325
## 2 1.106751 1.662921 1.755797 2.605196 1.485945 2.474721  1.807143
## 3 1.057103 1.656079 2.011174 2.576071 1.580723 1.938628  1.676740
##        sport       art
## 1 0.08453666 0.0831547
## 2 0.21710247 0.1552867
## 3 0.19952944 0.2077721
```

```
fviz_cluster(km,data=df)
```



Meaninful plot as Japan and USA are different from others.

```
library(factoextra)
set.seed(1)
fviz_nbclust(df,kmeans,method = "wss")
```

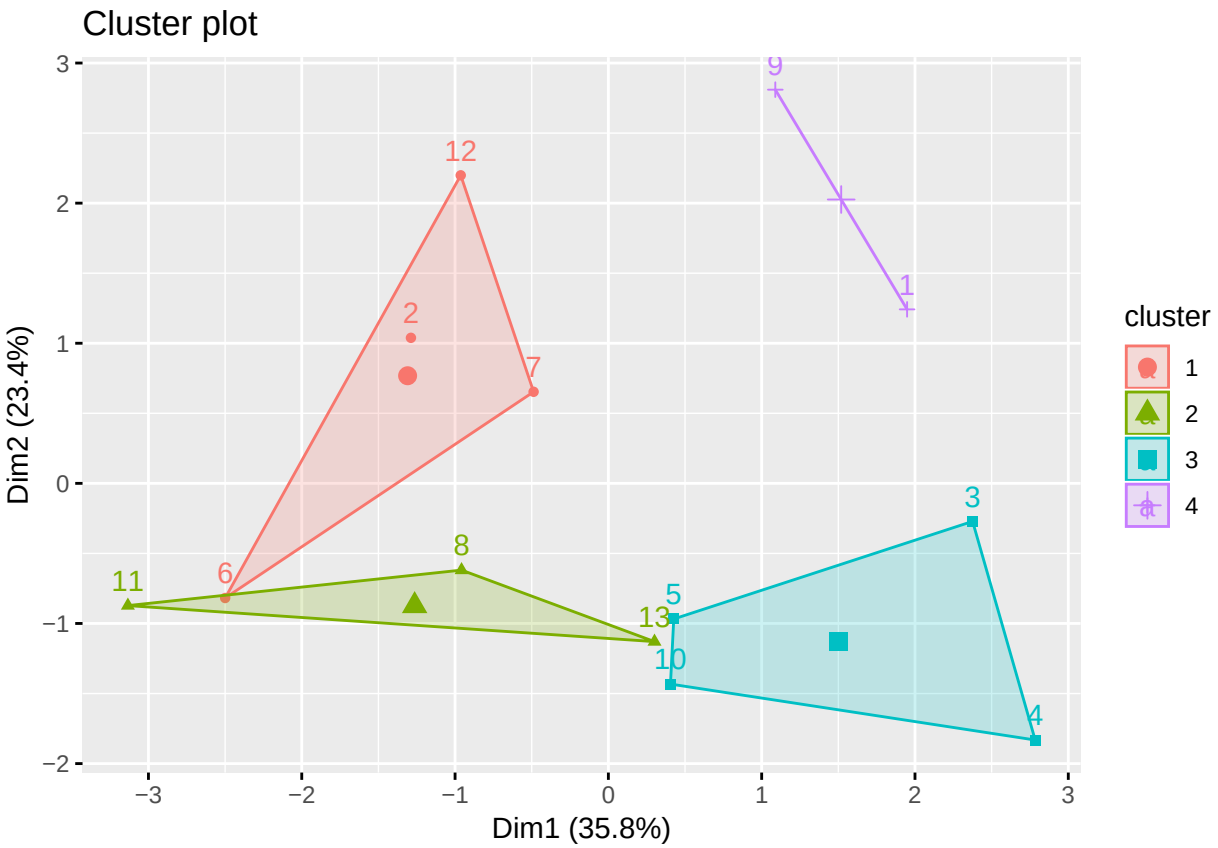


Optimanl number of clusters seems to be 4.

```
df2<-  df[complete.cases(df),]
km<-kmeans(df2,4)
km$centers
```

```
##     family   friend  leisure politics     work religion happiness
## 1 1.095339 1.615251 1.711146 2.545194 1.504831 2.492202  1.768628
## 2 1.051188 1.610967 1.973315 2.459379 1.608962 1.904864  1.588282
## 3 1.027295 1.533577 1.885971 2.650629 1.414663 1.383155  1.896325
## 4 1.113625 1.822509 2.029575 2.885674 1.453204 2.222360  1.951658
##        sport       art
## 1 0.25437809 0.1841084
## 2 0.25639188 0.2607288
## 3 0.08453666 0.0831547
## 4 0.04847106 0.0444511
```

```
fviz_cluster(km,data=df)
```



In this case it is very similar to dendogram plot because Japan was in only one in cluster. But now we countries that are closer to forein clusters'point than their centroids.