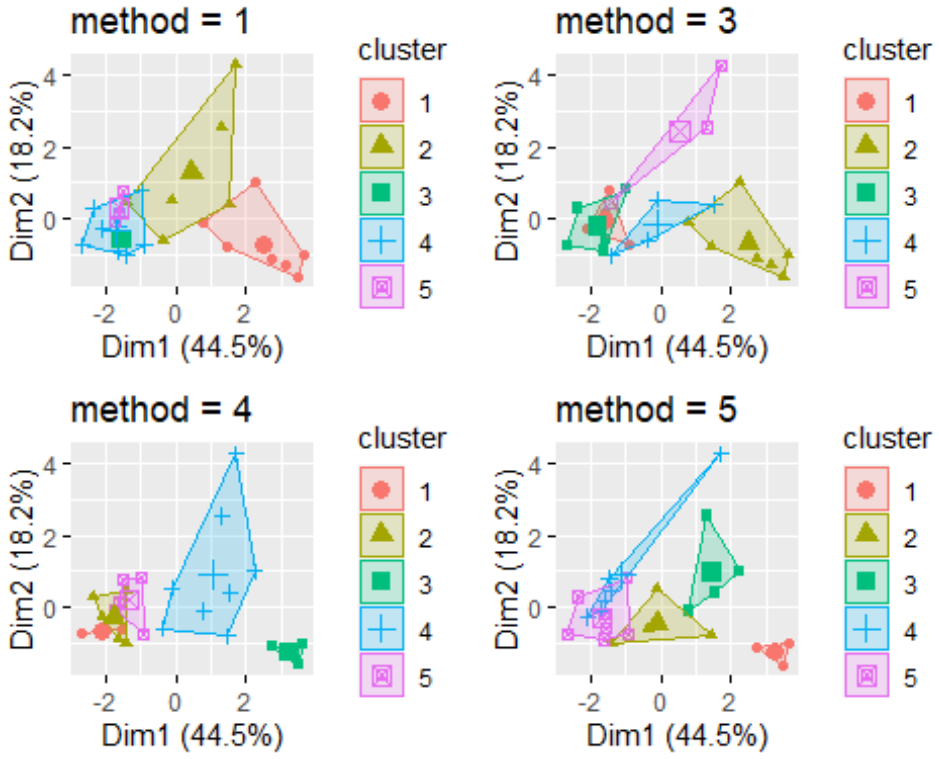
# Exercise-section27

Mehrab Atighi

12/22/2021

```
Data = data.frame(
  Country = c("Albania" , "Austria" , "Belgium",
              "Bulgaria" , "Czech" , "Denmark" ,
              "E.Germany" , "Finlad" , "France",
              "Greece" , "Hungary" , "Ireland" ,
              "Italy" , "Netherlands" , "Norway" ,
              "Poland" , "Portugal" , "Romania" ,
              "Spain" , "Sweden" , "Switzerland" ,
              "UK" , "USSR" , "W.Germany" , "Yogoslavia"),
  Red_Meat = c(10.1,8.9,13.5,7.8,9.7,10.6,8.4,
               9.5,18,10.2,5.3,13.9,9,9.5,9.4,
               6.9,6.2,6.2,7.1,9.9,13.1,17.4,
               9.3,11.4,4.4),
  White_Meat = c(1.4,14,9.3,6,11.4,10.8,11.6,
                 4.9,9.9,3,12.4,10,5.1,13.6,
                 4.7,10.2,3.7,6.3,3.4,7.8,
                 10.1,5.7,4.6,12.5,5),
  Eggs = c(0.5,4.3,4.1,1.6,2.8,3.7,3.7,2.7,
           3.3,2.8,2.9,4.7,2.9,3.6,2.7,2.7,
           1.1,1.5,3.1,3.5,3.1,4.7,2.1,4.1,
           1.2),
  Milk = c(8.9,19.9,17.5,8.3,12.5,25,11.1,
           33.7,19.5,17.6,9.7,25.8,13.7,23.4,
           23.3,19.3,4.9,11.1,8.6,24.7,23.8,
           20.6,16.6,18.8,9.5),
  Fish = c(0.2,2.1,4.5,1.2,2,9.9,5.4,5.8,5.7,
           5.9,0.3,2.2,3.4,2.5,9.7,3,14.2,1,
           7,7.5,2.3,4.3,3,3.4,0.6),
  Cereals = c(42.3,28,26.6,56.7,34.3,21.9,
              24.6,26.6,28.1,41.7,40.1,24,
              36.8,22.4,23,36.1,27,49.6,29.2,
              19.5,25.6,24.3,43.6,18.6,55.9),
  Strachy_Foods = c(0.6,3.6,5.7,1.1,5,4.8,6.5,
                    5.1,4.8,2.2,4,6.2,2.1,4.2,
                    4.6,5.9,5.9,3.1,5.7,3.7,
                    2.8,4.7,6.4,5.2,3),
  Nuts = c(5.5,1.3,2.1,3.7,1.1,0.7,0.8,1,2.4,
           7.8,5.4,1.6,4.3,1.8,1.6,2,4.7,5.3,
           5.9,1.4,2.4,3.4,3.4,1.5,5.7),
  Fruit_veg = c(1.7,4.3,4,4.2,4,2.4,3.6,1.4,
                6.5,6.5,4.2,2.9,6.7,3.7,2.7,
                6.6,7.9,2.8,7.2,2,4.9,3.3,2.9,
```

```
                3.8,3.2))
rownames(Data) = Data[,1]
Data[,1] = c()
library(factoextra)

## Loading required package: ggplot2

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

library(gridExtra)
model1 = kmeans(Data , 5 , 1000 , algorithm = "Hartigan-Wong")
model2 = kmeans(Data , 5 , 1000 , algorithm = "Lloyd")
model3 = kmeans(Data , 5 , 1000 , algorithm = "Forgy")
model4 = kmeans(Data , 5 , 1000 , algorithm = "MacQueen")


# plots to compare
p1 <- fviz_cluster(model1, geom = "point", data = Data) + ggtitle("method =
1")
p2 <- fviz_cluster(model2, geom = "point",  data = Data) + ggtitle("method =
3")
p3 <- fviz_cluster(model3, geom = "point",  data = Data) + ggtitle("method =
4")
p4 <- fviz_cluster(model4, geom = "point",  data = Data) + ggtitle("method =
5")
grid.arrange(p1, p2, p3, p4, nrow = 2)
```

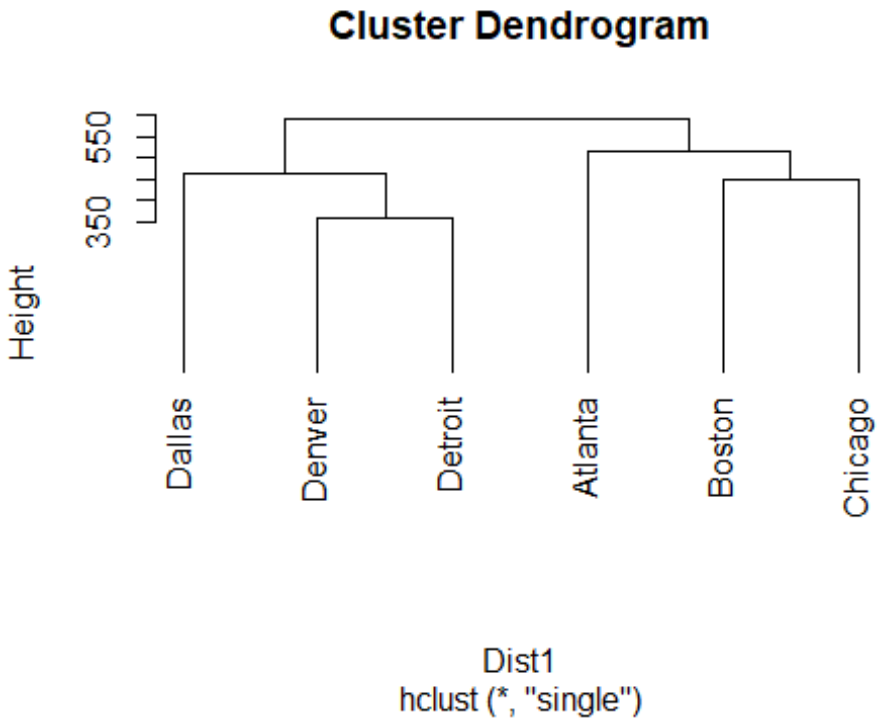


```r
# B)


Data2 = data.frame(
  City = c("Atlanta" , "Boston" , "Chicago" ,
           "Dallas" , "Denver" , "Detroit",
           "Hartford" , "Honolulu" , "Houston",
           "Kansas City" , "Los Angeles" ,
           "New Orleans", "New York",
           "Portland" , "Tucson" ,
           "Washington"),
  Murder = c(16.5,4.2,11.6,18.1,6.9,13.0,
             2.5,3.6,16.8,10.8,9.7,10.3,
             9.4,5.0,5.1,12.5),
  Rape = c(24.8,13.3,24.7,34.2,41.5,35.7,
           8.8,12.7,26.6,43.2,51.8,39.7,
           19.4,23.0,22.9,27.6),
  Robbery = c(106,122,340,184,173,477,
              68,42,289,255,286,266,
              522,157,85,524),
  Assault = c(147,90,242,293,191,220,
              103,28,186,226,355,283,
              267,144,148,217),
  Burglary = c(1112,982,808,1668,1534,
               1566,1017,1457,1509,1494,
               1902,1056,1674,1530,1206,
               1496),
  Larceny = c(905,669,609,901,1368,1183,
              724,1102,787,955,1386,
              1036,1392,1281,756,1003),
  AutoThef = c(494,954,645,605,780,
               788,468,637,697,765,862,
               776,848,488,483,793))



Data2 = Data2[1:6,]
rownames(Data2) = Data2 [1:6 , 1]
Data2[,1] = c()

Dist1 = dist(Data2, method = "euclidean",diag = TRUE , upper = TRUE)
```

```r
#single method
model1 = hclust(Dist1 , method = "single")
model1
```

```
##
## Call:
## hclust(d = Dist1, method = "single")
##
## Cluster method   : single
## Distance         : euclidean
## Number of objects: 6
```

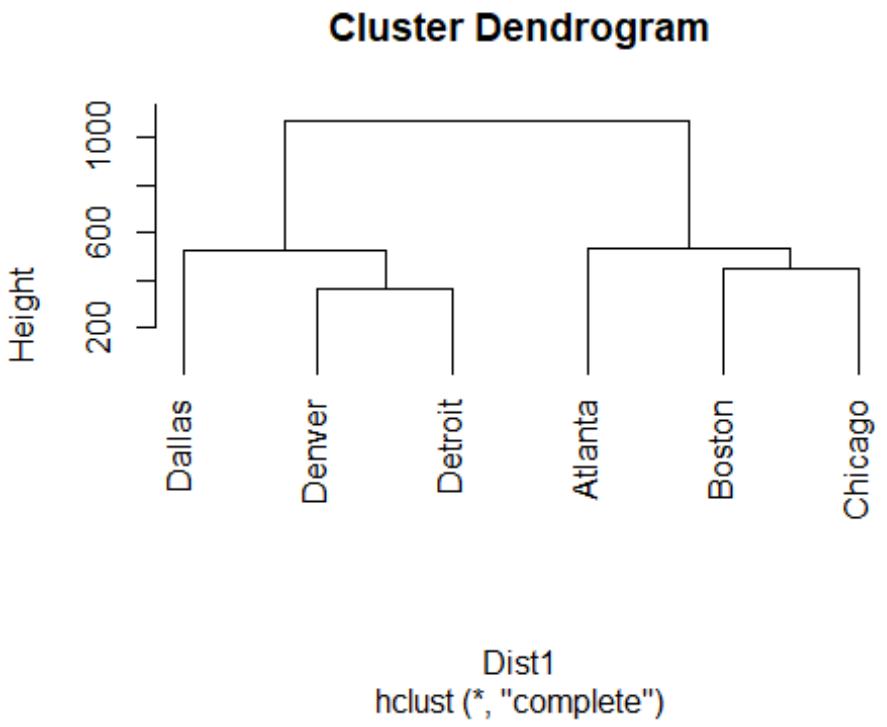
```r
plot( model1 , hang = -1  )
```



```r
#complete method:
model2 = hclust(Dist1 , method = "complete")
model2
```

```
##
## Call:
## hclust(d = Dist1, method = "complete")
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 6
```

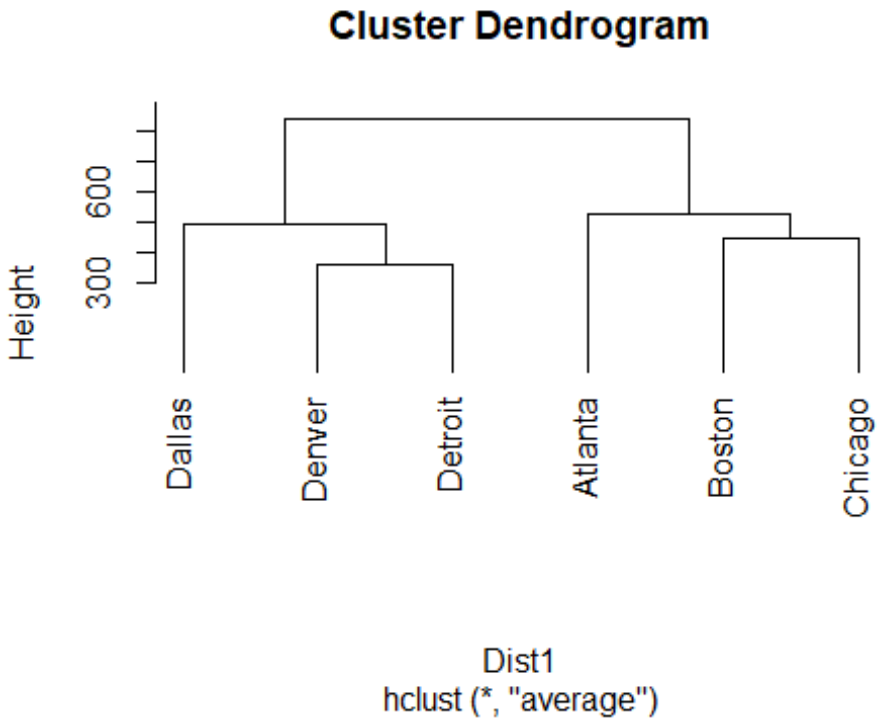
```r
plot( model2 , hang = -1  )
```




```r
#average method:
model3 = hclust(Dist1 , method = "average")
model3
```

```
##
## Call:
## hclust(d = Dist1, method = "average")
##
## Cluster method   : average
## Distance         : euclidean
## Number of objects: 6
```

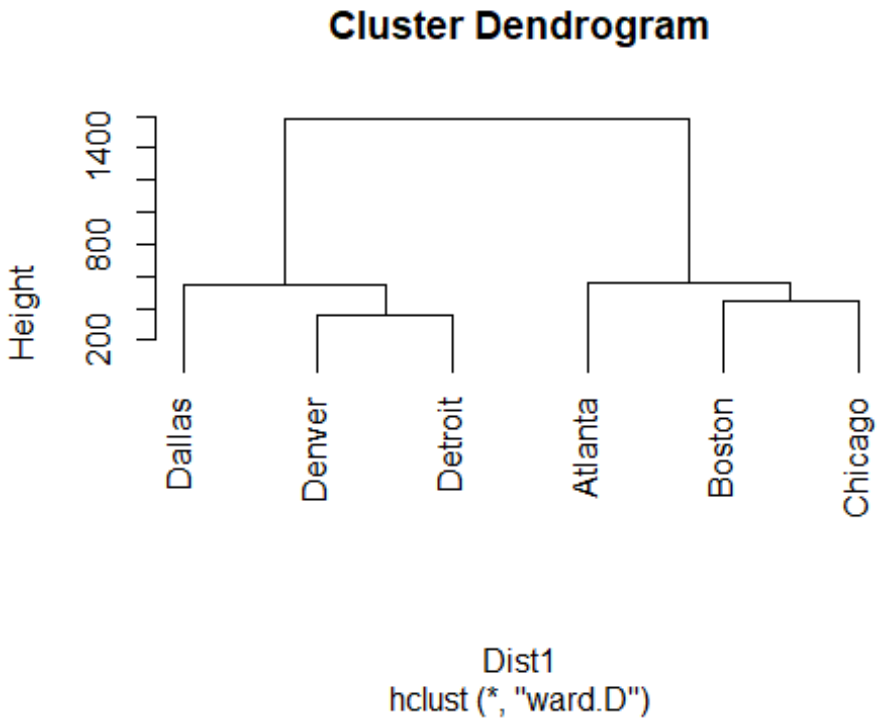
```
plot( model3 , hang = -1  )
```




```
#ward method:
model4 = hclust(Dist1 , method = "ward.D")
model4
```

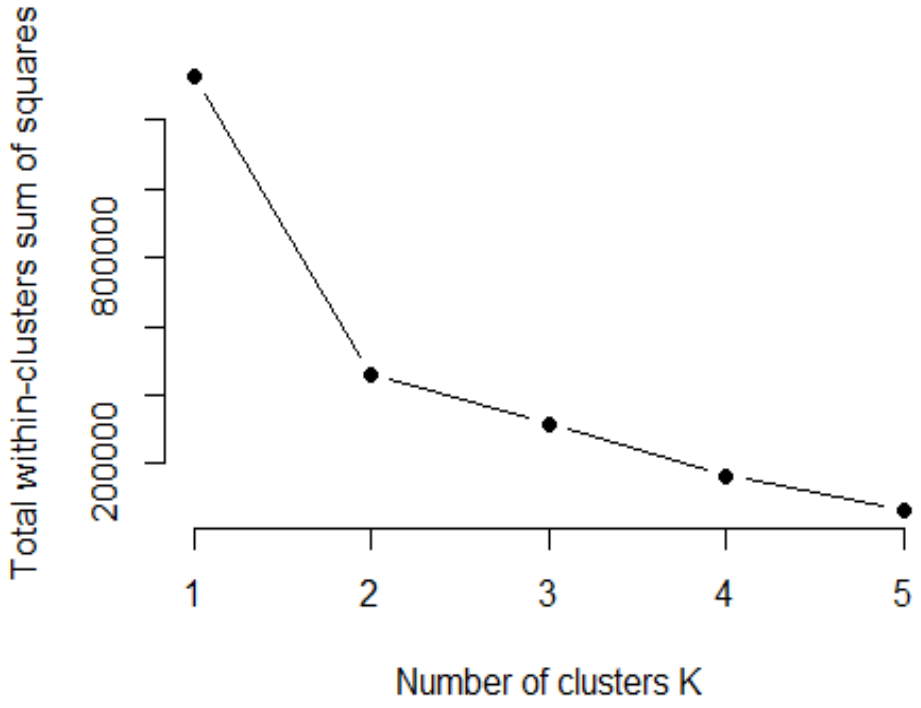
```
##
## Call:
## hclust(d = Dist1, method = "ward.D")
```

```
##
## Cluster method   : ward.D
## Distance         : euclidean
## Number of objects: 6

plot( model4 , hang = -1  )
```



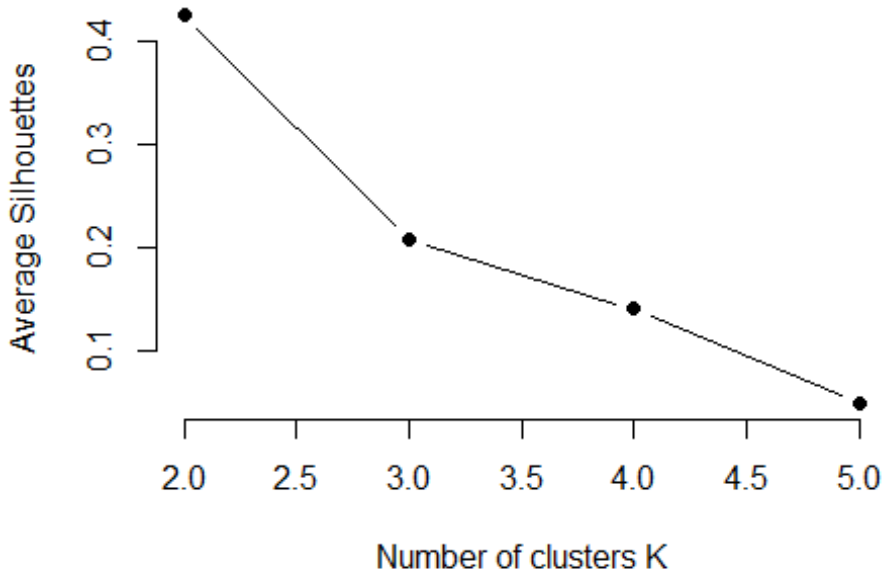

```
For Findig The Best Optimal K ( Number of the Clusters) we have 3 methods:
# Elbow Method :
set.seed(123)
# function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(Data2, k )$tot.withinss
}
# Compute and plot wss for k = 1 to k = 5
```

```r
k.values <- 1:5
# extract wss for 2-15 clusters
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()

wss_values <- map_dbl(k.values, wss)
plot(k.values, wss_values,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```




```r
#fviz_nbclust(Data2, kmeans, method = "wss")
```

```r
####
# function to compute average silhouette for k clusters
library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(Data2, centers = k)
  ss <- silhouette(km.res$cluster, dist(Data2))
  mean(ss[, 3])
}

# Compute and plot wss for k = 2 to k = 15
k.values <- 2:5

# extract avg silhouette for 2-15 clusters
avg_sil_values <- map_dbl(k.values, avg_sil)

plot(k.values, avg_sil_values,
     type = "b", pch = 19, frame = FALSE,
     xlab = "Number of clusters K",
     ylab = "Average Silhouettes")
```




```r
#fviz_nbclust(Data2, kmeans, method = "silhouette" )
```

```
####

# compute gap statistic

#gap_stat <- clusGap(Data2, FUN = kmeans,
                    #K.max = 10, B = 50)
# Print the result
#print(gap_stat, method = "firstmax")

#fviz_gap_stat(gap_stat)
```