Machine Learning and Data Mining II

LABWORK 2: Clustering

Groupwork:

Nguyễn Thị Quỳnh Anh **BI11-029**
Vũ Yến Linh **BI11-152**

I. AHC

Given the data X = {1, 2, 9, 12, 20} in 1-D space.

1. Apply the AHC clustering using Single Linkage / Complete Linkage for the X dataset.

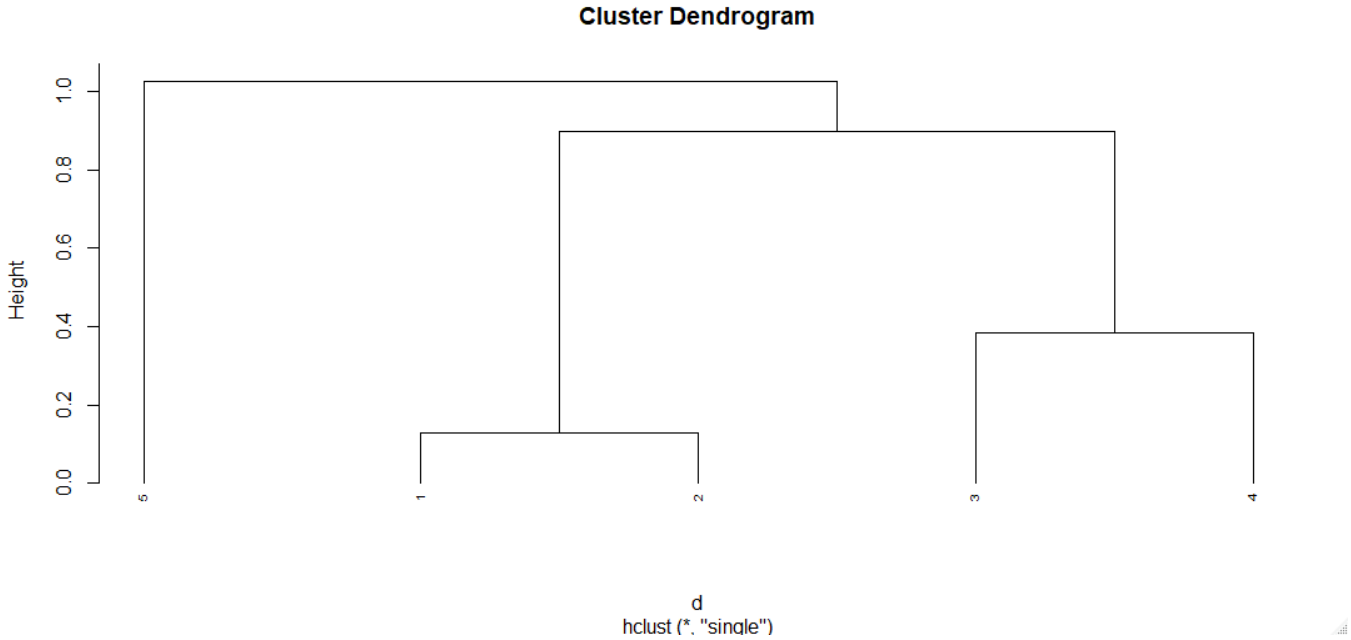
- Using R studio

```
> X <- c(1,2,9,12,20)
> X <- na.omit(X)
> X <- scale(X)
> d <- dist(X, method ="euclidean")
> hc1 <- hclust(d, method = "single")
> hc2 <- hclust(d, method = "complete")
> plot(hc1, cex = 0.6, hang = -1)
>
```

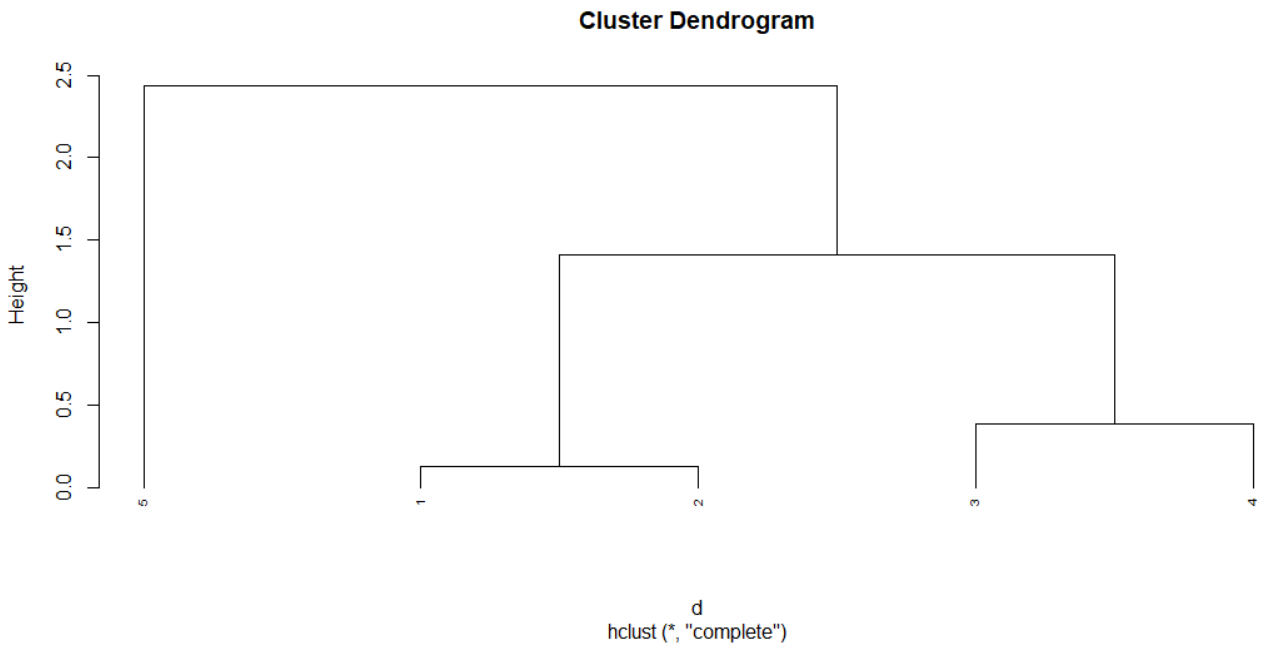
	V1
1	-1.00115255
2	-0.87279966
3	0.02567058
4	0.41072925
5	1.43755238

2. Draw the clustering result using available functions (dendrogram() in Matlab or hclust() in R, etc.).

- Single Linkage



- Complete Linkage



3. Apply on 2 more datasets from UCI. Make a study of data features. Observe the dendrogram and comment on results.

- Dataset from UCI :

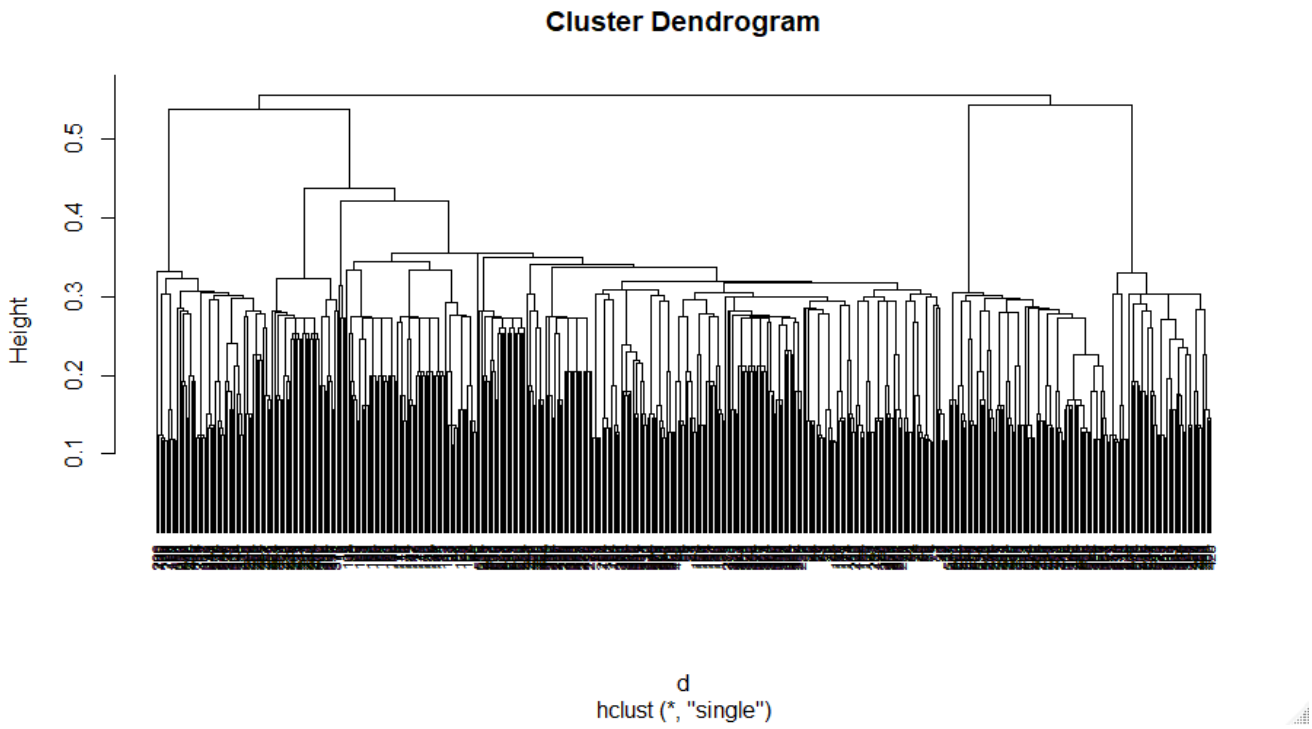
  Synchronous Machine Data Set:
  http://archive.ics.uci.edu/ml/datasets/Synchronous+Machine+Data+Set
  Energy efficiency Data Set:
  http://archive.ics.uci.edu/ml/datasets/Energy+efficiency

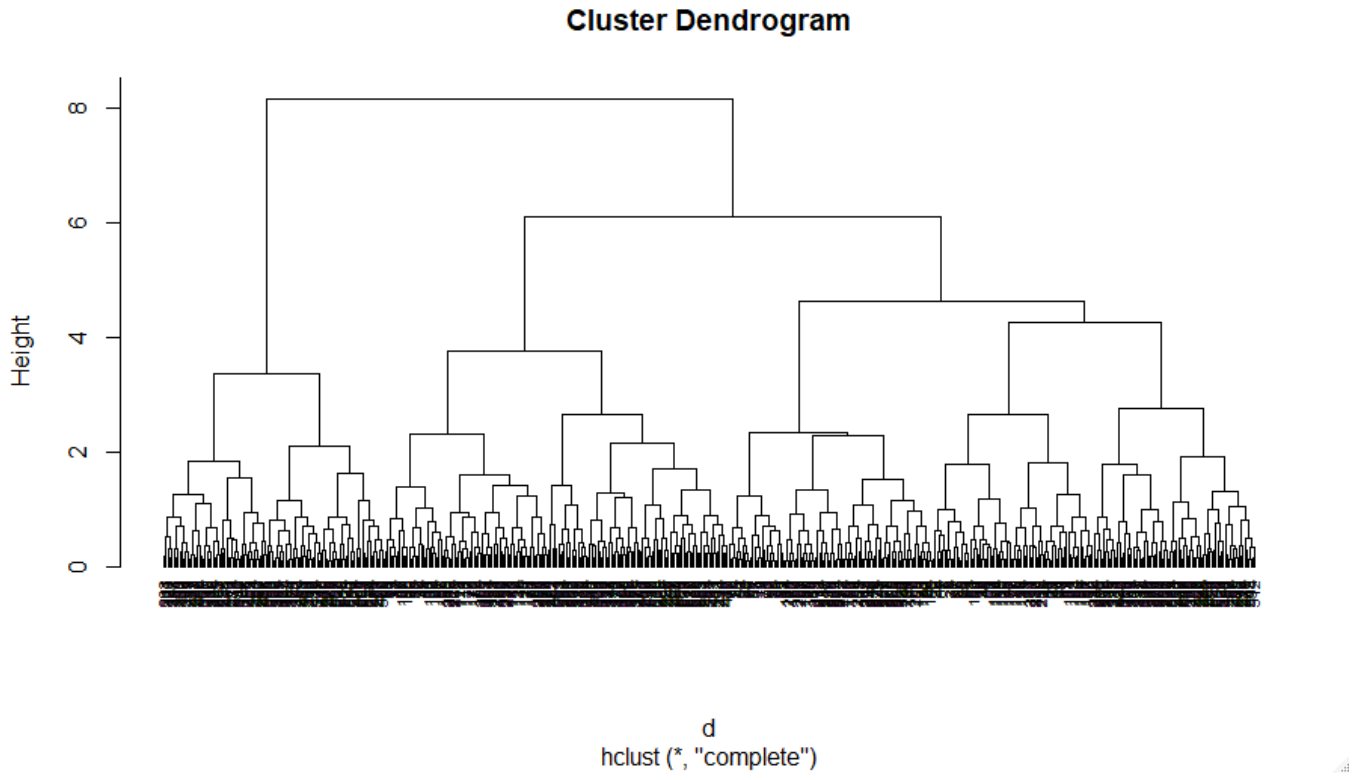
- For Synchronous Machine Dataset:

  Since we have already made a data analysis for the Synchronous dataset in the previous labwork, so now we will just make a summary for it.

```
> View(synchronous.machine)
> syn <- synchronous.machine
> summary(syn)
       Iy              PF               e               dIf              If       
 Min.   :3.0   Min.   :0.6500   Min.   :0.0000   Min.   :0.0370   Min.   :1.217  
 1st Qu.:3.7   1st Qu.:0.7400   1st Qu.:0.0800   1st Qu.:0.1890   1st Qu.:1.369  
 Median :4.5   Median :0.8200   Median :0.1800   Median :0.3450   Median :1.525  
 Mean   :4.5   Mean   :0.8253   Mean   :0.1747   Mean   :0.3507   Mean   :1.531  
 3rd Qu.:5.3   3rd Qu.:0.9200   3rd Qu.:0.2600   3rd Qu.:0.4860   3rd Qu.:1.666  
 Max.   :6.0   Max.   :1.0000   Max.   :0.3500   Max.   :0.7690   Max.   :1.949  
> cor(syn)
            Iy          PF           e         dIf          If
Iy   1.00000000 -0.04157389  0.04157389  0.4249449  0.4249449
PF  -0.04157389  1.00000000 -1.00000000 -0.8610135 -0.8610135
e    0.04157389 -1.00000000  1.00000000  0.8610135  0.8610135
dIf  0.42494491 -0.86101347  0.86101347  1.0000000  1.0000000
If   0.42494491 -0.86101347  0.86101347  1.0000000  1.0000000
> 
```

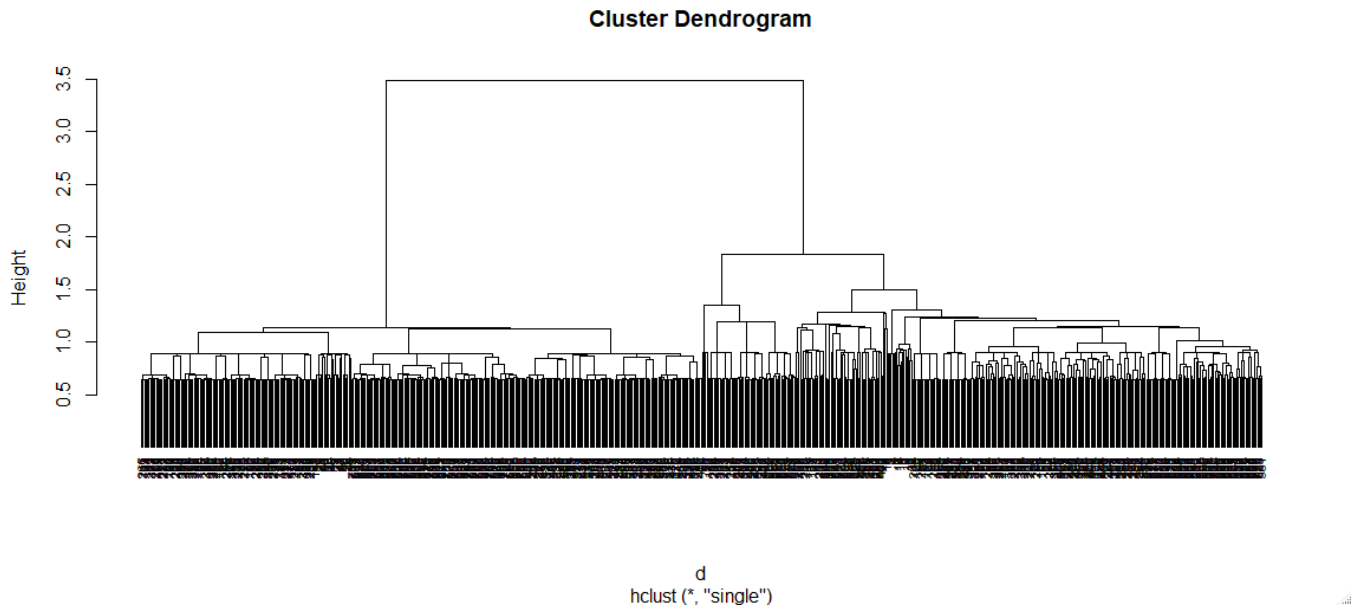
```
> syn <- na.omit(syn)
> syn <- scale(syn)
> d <- dist(syn, method ="euclidean")
> hc1 <- hclust(d, method = "single")
> hc2 <- hclust(d, method = "complete")
> plot(hc1, cex = 0.6, hang = -1)
> 
```

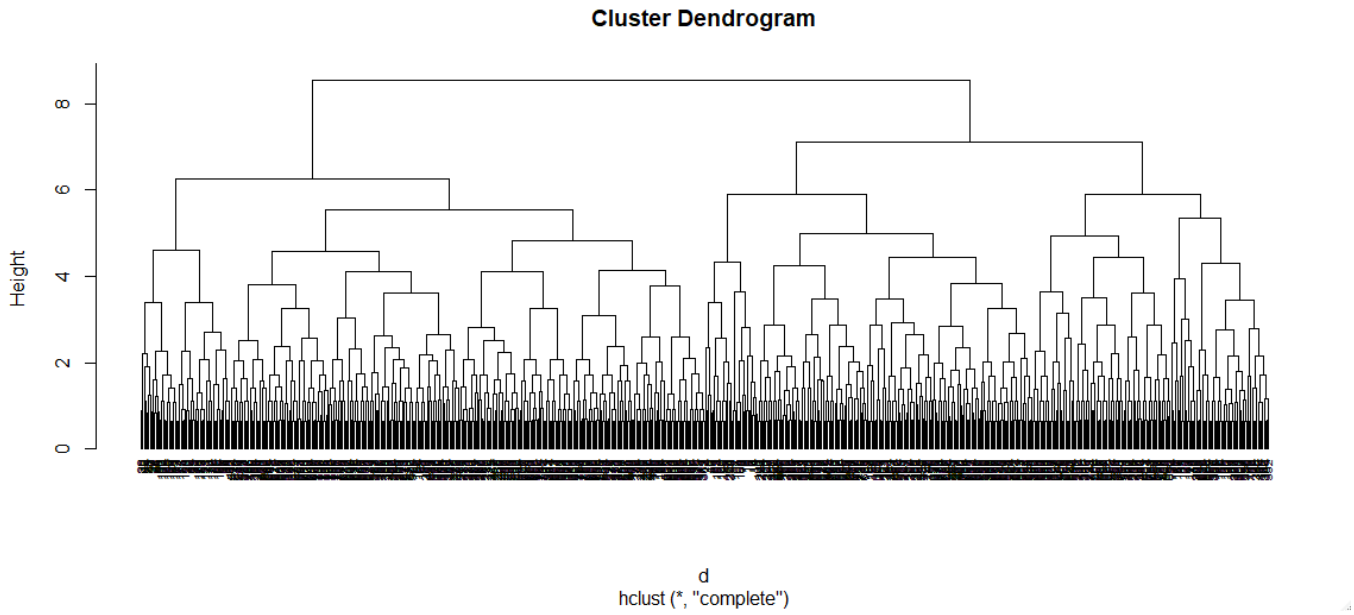




- For Energy efficiency Dataset:

```
> enb <- ENB2012_data
> summary(enb)
       X1               X2               X3               X4               X5               X6               X7               X8               Y1               Y2
 Min.   :0.6200   Min.   :514.5   Min.   :245.0   Min.   :110.2   Min.   :3.50   Min.   :2.00   Min.   :0.0000   Min.   :0.000   Min.   : 6.01   Min.   :10.90
 1st Qu.:0.6825   1st Qu.:606.4   1st Qu.:294.0   1st Qu.:140.9   1st Qu.:3.50   1st Qu.:2.75   1st Qu.:0.1000   1st Qu.:1.750   1st Qu.:12.99   1st Qu.:15.62
 Median :0.7500   Median :673.8   Median :318.5   Median :183.8   Median :5.25   Median :3.50   Median :0.2500   Median :3.000   Median :18.95   Median :22.08
 Mean   :0.7642   Mean   :671.7   Mean   :318.5   Mean   :176.6   Mean   :5.25   Mean   :3.50   Mean   :0.2344   Mean   :2.812   Mean   :22.31   Mean   :24.59
 3rd Qu.:0.8300   3rd Qu.:741.1   3rd Qu.:343.0   3rd Qu.:220.5   3rd Qu.:7.00   3rd Qu.:4.25   3rd Qu.:0.4000   3rd Qu.:4.000   3rd Qu.:31.67   3rd Qu.:33.13
 Max.   :0.9800   Max.   :808.5   Max.   :416.5   Max.   :220.5   Max.   :7.00   Max.   :5.00   Max.   :0.4000   Max.   :5.000   Max.   :43.10   Max.   :48.03
> var(enb)
             X1           X2          X3           X4          X5          X6           X7          X8           Y1           Y2
X1  1.118887e-02 -9.242069e+00   -0.9403911 -4.150839e+00   0.1533246  0.00000000 1.073424e-21 0.00000000   0.66416098   0.6383312
X2 -9.242069e+00  7.759164e+03  751.2907432  3.503937e+03 -132.3702738  0.00000000 5.473313e-19 0.00000000 -584.94150880 -563.9664689
X3 -9.403911e-01  7.512907e+02 1903.2698827 -5.759896e+02   21.4654498  0.00000000 0.000000e+00 0.00000000  200.58657888  177.2672425
X4 -4.150839e+00  3.503937e+03 -575.9895698  2.039963e+03  -76.9178618  0.00000000 -7.203513e-19 0.00000000 -392.76404384 -370.6168557
X5  1.533246e-01 -1.323703e+02   21.4654498 -7.691786e+01    3.0664928  0.00000000 0.000000e+00 0.00000000   15.71567080   14.9230052
X6  0.000000e+00  0.000000e+00    0.0000000  0.000000e+00    0.0000000  1.25162973 0.000000e+00 0.00000000   -0.02920078    0.1520860
X7  1.073424e-21  5.473313e-19    0.0000000 -7.203513e-19    0.0000000  0.00000000 1.774772e-02 0.04400261    0.36272731    0.2629852
X8  0.000000e+00  0.000000e+00    0.0000000  0.000000e+00    0.0000000  0.00000000 4.400261e-02 2.40547588    1.36727265    0.7454857
Y1  6.641610e-01 -5.849415e+02  200.5865789 -3.927640e+02   15.7156708 -0.02920078 3.627273e-01 1.36727265  101.81221616   93.6741331
Y2  6.383312e-01 -5.639665e+02  177.2672425 -3.706169e+02   14.9230052  0.15208605 2.629852e-01 0.74548566   93.67413308   90.5029827
> cov(enb)
             X1           X2          X3           X4          X5          X6           X7          X8           Y1           Y2
X1  1.118887e-02 -9.242069e+00   -0.9403911 -4.150839e+00   0.1533246  0.00000000 1.073424e-21 0.00000000   0.66416098   0.6383312
X2 -9.242069e+00  7.759164e+03  751.2907432  3.503937e+03 -132.3702738  0.00000000 5.473313e-19 0.00000000 -584.94150880 -563.9664689
X3 -9.403911e-01  7.512907e+02 1903.2698827 -5.759896e+02   21.4654498  0.00000000 0.000000e+00 0.00000000  200.58657888  177.2672425
X4 -4.150839e+00  3.503937e+03 -575.9895698  2.039963e+03  -76.9178618  0.00000000 -7.203513e-19 0.00000000 -392.76404384 -370.6168557
X5  1.533246e-01 -1.323703e+02   21.4654498 -7.691786e+01    3.0664928  0.00000000 0.000000e+00 0.00000000   15.71567080   14.9230052
X6  0.000000e+00  0.000000e+00    0.0000000  0.000000e+00    0.0000000  1.25162973 0.000000e+00 0.00000000   -0.02920078    0.1520860
X7  1.073424e-21  5.473313e-19    0.0000000 -7.203513e-19    0.0000000  0.00000000 1.774772e-02 0.04400261    0.36272731    0.2629852
X8  0.000000e+00  0.000000e+00    0.0000000  0.000000e+00    0.0000000  0.00000000 4.400261e-02 2.40547588    1.36727265    0.7454857
Y1  6.641610e-01 -5.849415e+02  200.5865789 -3.927640e+02   15.7156708 -0.02920078 3.627273e-01 1.36727265  101.81221616   93.6741331
Y2  6.383312e-01 -5.639665e+02  177.2672425 -3.706169e+02   14.9230052  0.15208605 2.629852e-01 0.74548566   93.67413308   90.5029827
> cor(enb)
             X1           X2          X3           X4          X5          X6           X7          X8           Y1          Y2
X1  1.000000e+00 -9.919015e-01 -0.2037817 -8.688234e-01  0.8277473  0.00000000  7.617400e-20 0.00000000  0.622271936  0.63433907
X2 -9.919015e-01  1.000000e+00  0.1955016  8.807195e-01 -0.8581477  0.00000000  4.664140e-20 0.00000000 -0.658119917 -0.67299893
X3 -2.037817e-01  1.955016e-01  1.0000000 -2.923165e-01  0.2809757  0.00000000  0.000000e+00 0.00000000  0.455671365  0.42711700
X4 -8.688234e-01  8.807195e-01 -0.2923165  1.000000e+00 -0.9725122  0.00000000 -1.197187e-19 0.00000000 -0.861828052 -0.86254660
X5  8.277473e-01 -8.581477e-01  0.2809757 -9.725122e-01  1.0000000  0.00000000  0.000000e+00 0.00000000  0.889430464  0.89578517
X6  0.000000e+00  0.000000e+00  0.0000000  0.000000e+00  0.0000000  1.00000000  0.000000e+00 0.00000000 -0.002586763  0.01428960
X7  7.617400e-20  4.664140e-20  0.0000000 -1.197187e-19  0.0000000  0.00000000  1.000000e+00 0.21296422  0.269841685  0.20750499
X8  0.000000e+00  0.000000e+00  0.0000000  0.000000e+00  0.0000000  0.00000000  2.129642e-01 1.00000000  0.087368460  0.05052512
Y1  6.222719e-01 -6.581199e-01  0.4556714 -8.618281e-01  0.8894305 -0.002586763  2.698417e-01 0.08736846  1.000000000  0.97586174
Y2  6.343391e-01 -6.729989e-01  0.4271170 -8.625466e-01  0.8957852  0.014289598  2.075050e-01 0.05052512  0.975861739  1.00000000
> 
```





4. Conclude on the advantages and drawbacks of AHC.

- Advantage:

- AHC outputs a hierarchy → easier to decide on the number of clusters (by looking at the dendrogram).
- Easy to implement, doesn't require any input.

● Drawbacks:

- For a n-element dataset, AHC must calculate the distance from one element to (n-1) remaining elements so the AHC has the complexity of O(n^2) → time complexity: not suitable for large databases.
- Initial seeds have a strong impact on the final results.
- The order of the data has an impact on the final results.
- Very sensitive to noise/outliers.

II. K-means:

Select two datasets from UCI:

- 1.Iris Dataset: http://archive.ics.uci.edu/ml/datasets/Iris
- 2.Wine Data Set: http://archive.ics.uci.edu/ml/datasets/Wine

1. Run experiments with k-means. Explain the experimental protocol.

For Iris Dataset:

- **Model kmeans_re:**

```
> # Loading data
> data(iris)
> # Structure
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
> # Loading package
> library(ClusterR)
> library(cluster)
> # Removing initial label of species from original dataset
> iris_1 <- iris[, -5]
> # Fitting K-Means clustering Model
> # to training dataset
> set.seed(240) # Setting seed
> kmeans.re <- kmeans(iris_1, centers = 3, nstart = 20)
> kmeans.re
K-means clustering with 3 clusters of sizes 50, 62, 38

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     5.901613    2.748387     4.393548    1.433871
3     6.850000    3.073684     5.742105    2.071053

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 3 2 2
 [56] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3
[111] 3 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 2 3 3 3 3 3 2 3 3 3 3 2 3 3 3 2 3 3 3 2 3 3 3 2 3 3 2

Within cluster sum of squares by cluster:
[1] 15.15100 39.82097 23.87947
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
```

The 3 clusters are made which are of 50, 62, and 38 sizes respectively. Within the cluster, the sum of squares is 88.4%.

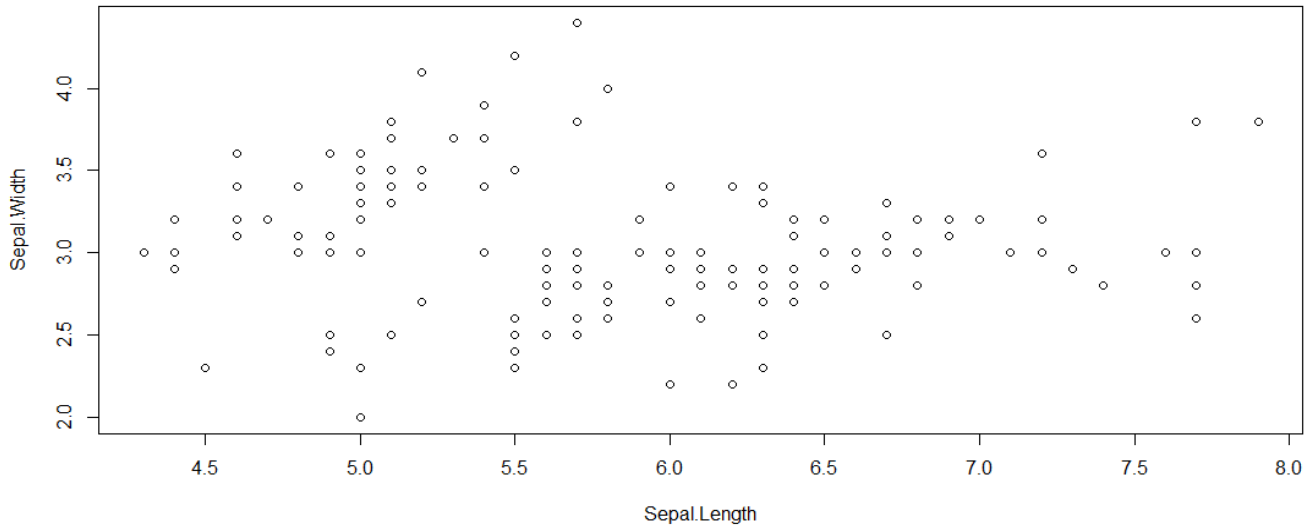
- **Cluster identification and confusion matrix:**

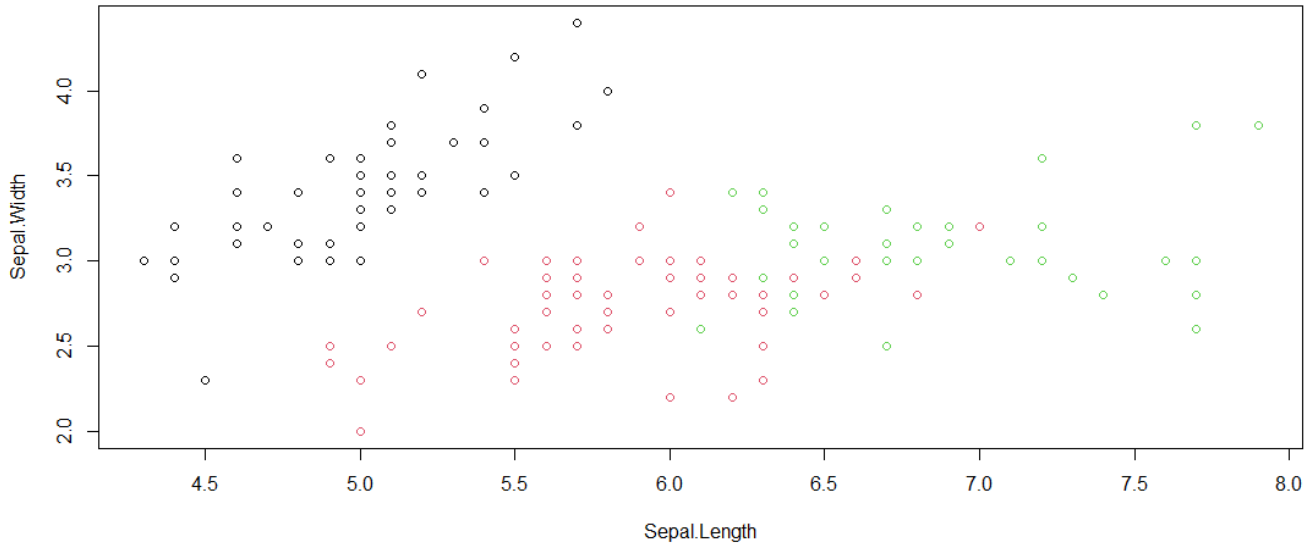
```
> # Cluster identification for
> # each observation
> kmeans.re$cluster
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [44] 1 1 1 1 1 1 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2
 [87] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3 3 2 2 3 3 3 2 3 2 3 2 3 3 2 2 3
[130] 3 3 3 3 2 3 3 3 3 2 3 3 3 2 3 3 3 2 3 3 2
> # Confusion Matrix
> cm <- table(iris$Species, kmeans.re$cluster)
> cm

              1  2  3
  setosa     50  0  0
  versicolor  0 48  2
  virginica   0 14 36
```

50 Setosa are so appropriately categorized as Setosa. 48 of the 62 Versicolor are appropriately categorized as Versicolor, and 14 are categorized as virginica. 19 of the 36 virginica have been reliably identified as virginica, while two have been identified as Versicolor.

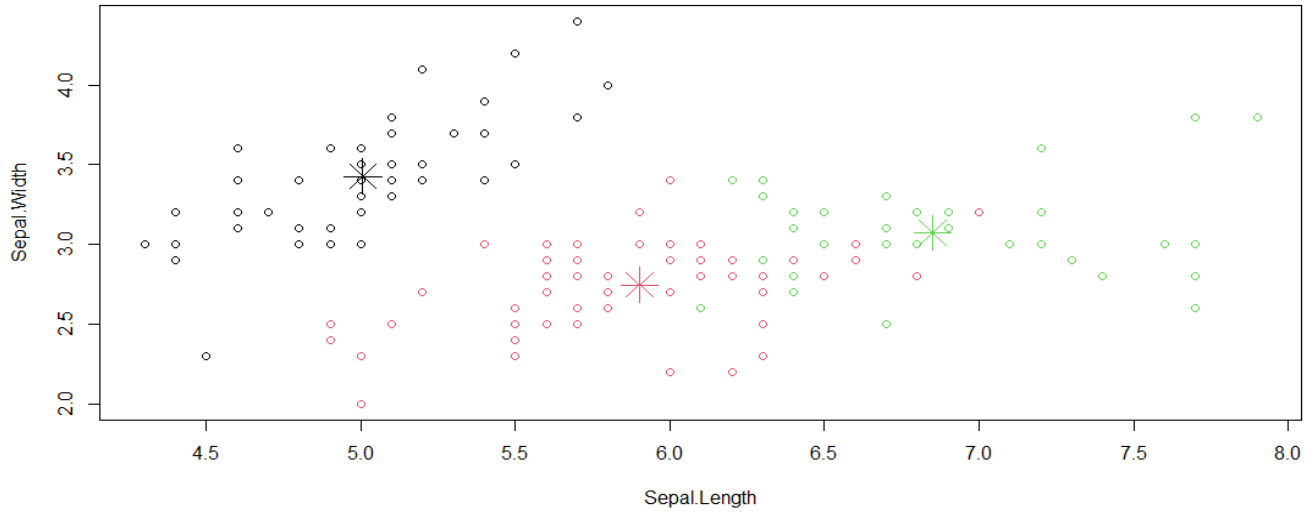
- **K-means with 3 clusters plot:**

```
# Model Evaluation and visualization
plot(iris_1[c("Sepal.Length", "Sepal.Width")])
plot(iris_1[c("Sepal.Length", "Sepal.Width")],
     col = kmeans.re$cluster)
plot(iris_1[c("Sepal.Length", "Sepal.Width")],
     col = kmeans.re$cluster,
     main = "K-means with 3 clusters")

## Plotting cluster centers
kmeans.re$centers
kmeans.re$centers[, c("Sepal.Length", "Sepal.Width")]
```

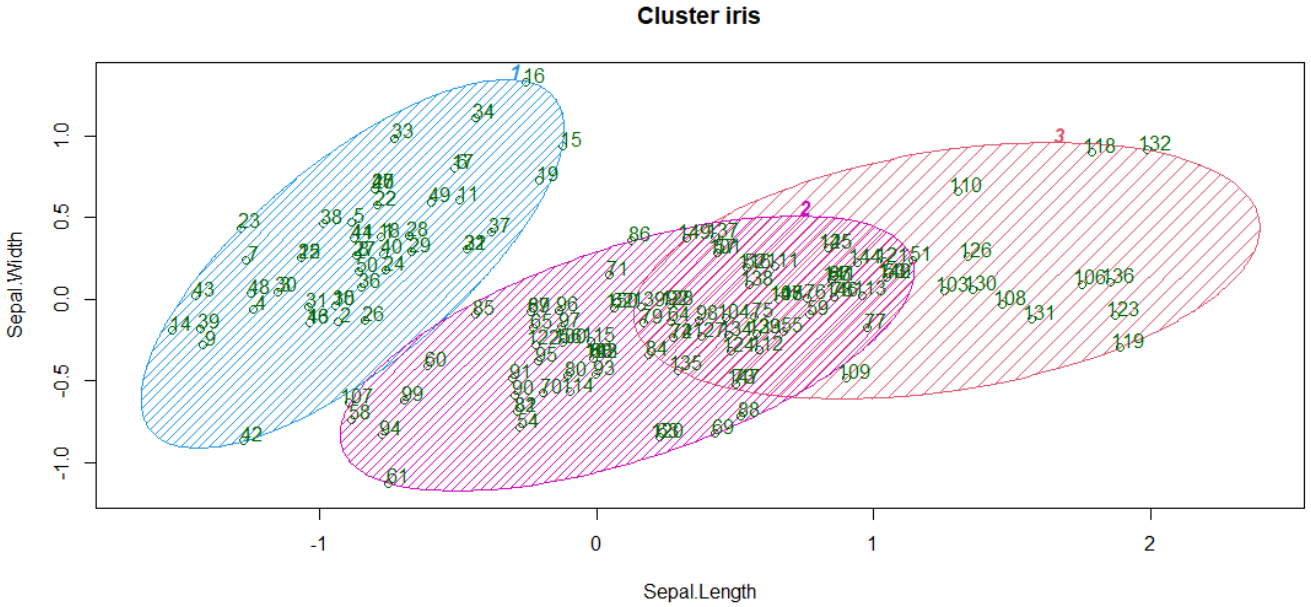




K-means with 3 clusters



- Plot of clusters:

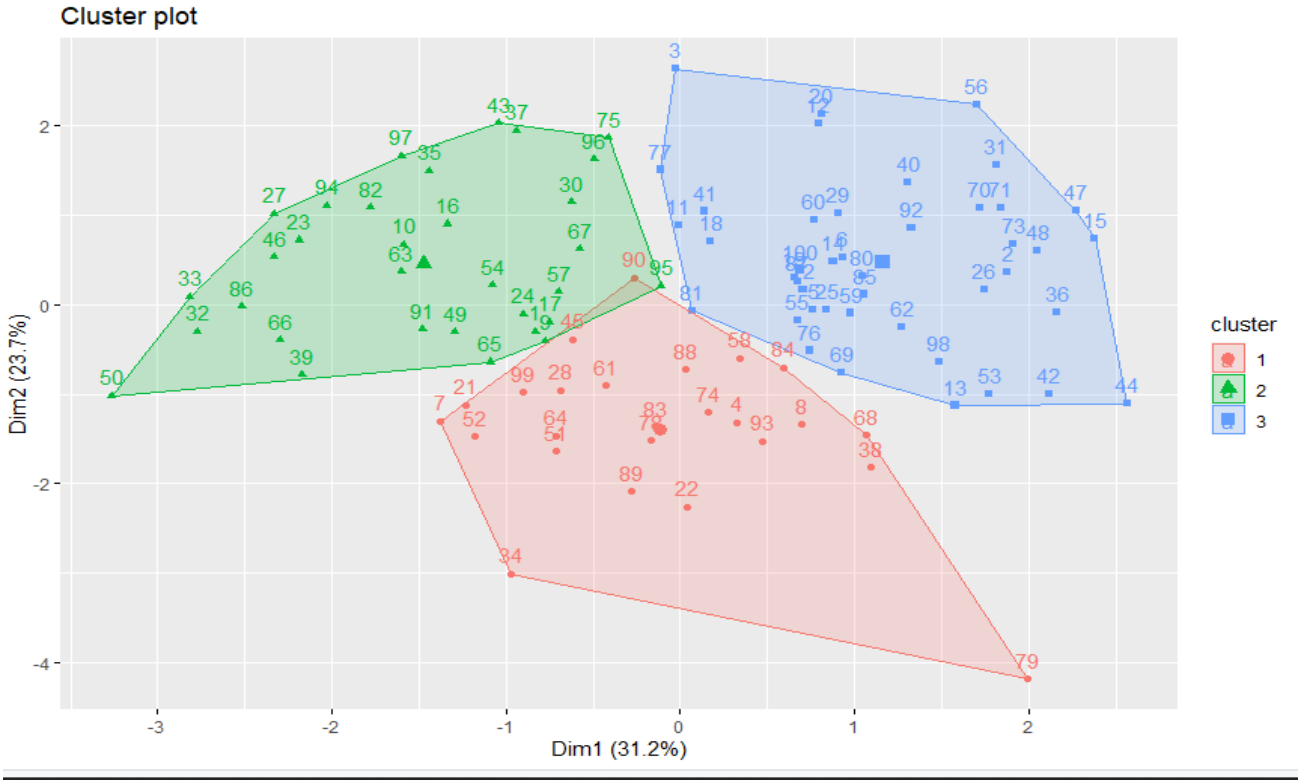


For Wine dataset:

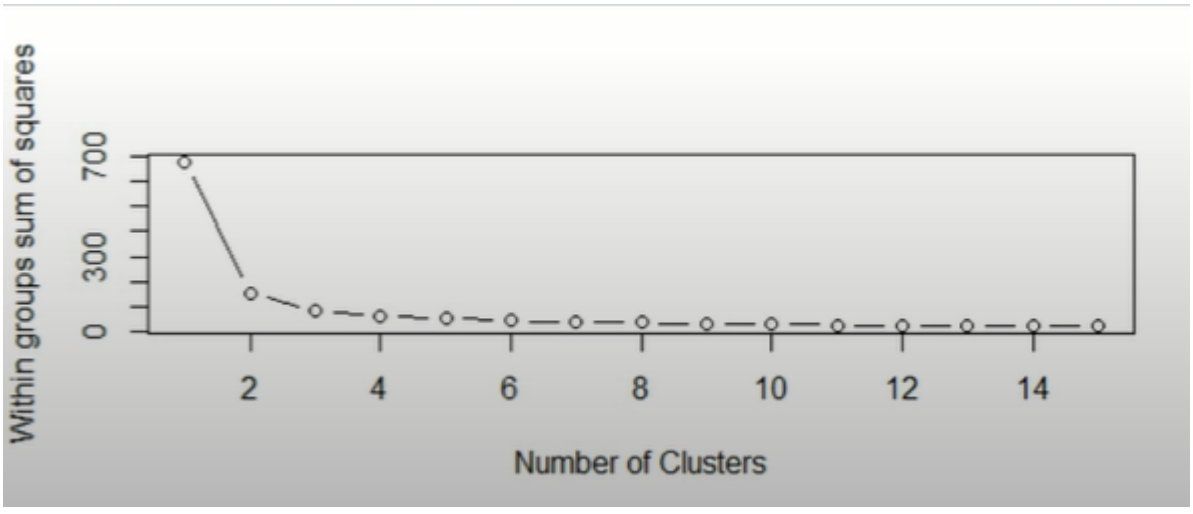
- Model kmeans_re:

```
> # standardize continuous variables
> data <- Wine %>% select(-1, -14) %>% scale()
> # create clusters with k-means
> kmeans(data, centers = 3, iter.max = 100, nstart = 100)
K-means clustering with 3 clusters of sizes 62, 64, 51

Cluster means:
       14.23       1.71       2.43       15.6        127        2.8       3.06          .28       2.29       5.64
1  0.7987121 -0.3638891  0.3546289 -0.5644599  0.51157963  0.9610990  1.01856614 -0.589199261  0.628154853  0.1719609
2 -0.9121255 -0.3361819 -0.4927039  0.1349607 -0.44381878 -0.1572472 -0.02551182 -0.002287425  0.006431426 -0.9139094
3  0.1736447  0.8642504  0.1871775  0.5168437 -0.06497127 -0.9710650 -1.20624204  0.719151948 -0.771710042  0.9378162
         1.04       3.92
1  0.5150541  0.7837745
2  0.4227232  0.2664770
3 -1.1566204 -1.2872265

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [59] 2 2 3 2 1 2 2 1 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[117] 2 3 2 2 1 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[175] 3 3 3

Within cluster sum of squares by cluster:
[1] 372.2401 512.5285 319.0040
 (between_SS / total_SS =  43.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
```

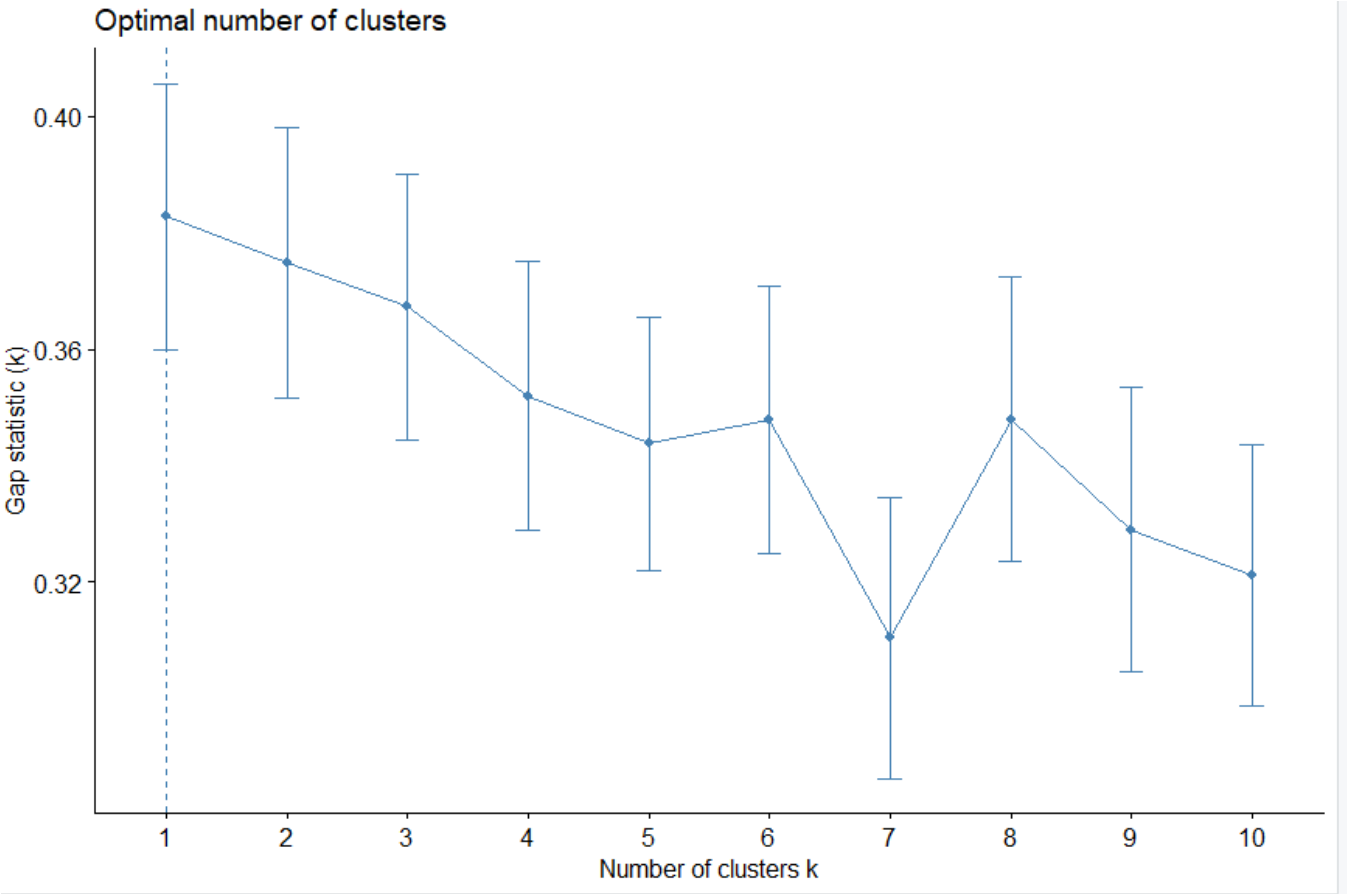
- **Plot of clusters:**



2. Explain the centroid initialization of the k-means used in the first question.



We base on the elbow method to choose k = 3 for the Iris Dataset



And silhouette method to choose k = 3 for Wine Dataset

3. Analyze and compare the results with different values of k.

For Iris dataset: k = 2

```
> # Fitting K-Means clustering Model to training dataset
> set.seed(240) # Setting seed
> kmeans.re <- kmeans(iris_1, centers = 2, nstart = 20)
> kmeans.re
K-means clustering with 2 clusters of sizes 53, 97

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.005660    3.369811     1.560377    0.290566
2     6.301031    2.886598     4.958763    1.695876

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1 2 2
 [61] 2 2 2 2 2 2 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 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[121] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1]  28.55208 123.79588
 (between_SS / total_SS =  77.6 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
> |
```

For Wine dataset: k = 4

```
> # create clusters with k-means
> kmeans(data, centers = 4, iter.max = 100, nstart = 177)
K-means clustering with 4 clusters of sizes 41, 57, 50, 29

Cluster means:
       14.23       1.71       2.43       15.6        127        2.8       3.06        .28       2.29       5.64
1 -0.7723064 -0.4174020 -1.0048046 -0.3903739 -0.47336765  0.2393353  0.2678889 -0.6055783  0.2822938 -0.7787523
2  0.8919484 -0.3157252  0.4629576 -0.5293868  0.61826550  0.9426657  1.0193255 -0.5562158  0.6554689  0.2075293
3  0.1858631  0.8990441  0.2160888  0.5434689 -0.06685264 -0.9817944 -1.2180815  0.7066309 -0.7533079  0.9505909
4 -0.9817122 -0.3393926  0.1380678  0.6554114 -0.43070439 -0.4984474 -0.2821009  0.7310850 -0.3886337 -0.9458576
        1.04        3.92
1  0.3963681  0.50032327
2  0.5039521  0.78214923
3 -1.1817618 -1.28420092
4  0.4866113 -0.03054188

Clustering vector:
  [1] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 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
 [59] 1 4 4 1 1 4 1 1 1 4 1 4 2 4 2 1 1 1 4 1 4 1 1 4 3 1 1 4 4 4 4 4 4 4 1 1 2 4 1 1 1 1 1 4 1 1 4 1 4 1 1 1 1 4 4 4 4 1
[117] 4 3 1 1 2 4 1 1 1 1 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[175] 3 3 3

Within cluster sum of squares by cluster:
[1] 290.3908 326.5426 307.6225 184.4747
 (between_SS / total_SS =  47.5 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
```

4. Calculate the clustering quality (any criteria in slides).

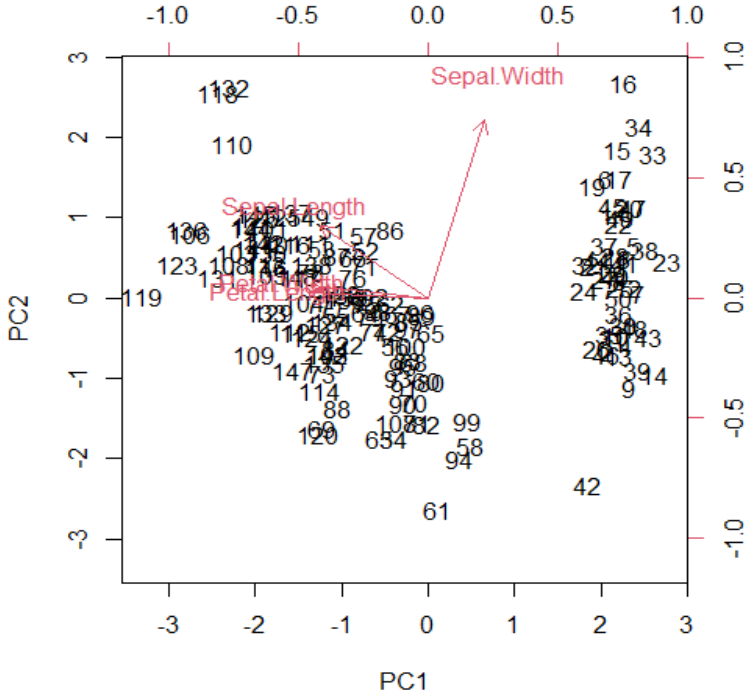
```
> confusionMatrix(cm)
Confusion Matrix and Statistics


            setosa versicolor virginica
  setosa         20          0         0
  versicolor      0         20         0
  virginica       0          0        20

Overall Statistics

               Accuracy : 1
                 95% CI : (0.9404, 1)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            1.0000           1.0000
Specificity                 1.0000            1.0000           1.0000
Pos Pred Value              1.0000            1.0000           1.0000
Neg Pred Value              1.0000            1.0000           1.0000
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3333           0.3333
Detection Prevalence        0.3333            0.3333           0.3333
Balanced Accuracy           1.0000            1.0000           1.0000
```

The model's accuracy was 100% and its p-value was less than 1.

This suggests that the model is good.

5. Use PCA or SVD to visualize the data distribution in 2D/3D.

- PCA for Iris dataset :



- PCA for Wine dataset: