GRIP Task 2 - Unsupervised Algorithm

Sanju Hyacinth C

14/11/2020

PROJECT AIM:

To find the optimum number of clusters from the dataset and visualise them

DATA:

Iris - Open source dataset

ALGORITHM:

K-Means - Unsupervised Algorithm

INSTALLING REQUIRED LIBRARIES:

```
# install.packages("patchwork")
# install.packages("tidyverse")
# install.packages("gridExtra")
# install.packages("ggExtra")
# install.packages("gtable")
# install.packages("ggpubr")
```

EXPLORATORY DATA ANALYSIS:

```
# loading data
dsba2 = read.csv("dsba_2.csv",header = TRUE)

# top 10 rows
head(dsba2, 10)
```

```
##    Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm     Species
## 1   1           5.1          3.5           1.4          0.2 Iris-setosa
## 2   2           4.9          3.0           1.4          0.2 Iris-setosa
## 3   3           4.7          3.2           1.3          0.2 Iris-setosa
## 4   4           4.6          3.1           1.5          0.2 Iris-setosa
## 5   5           5.0          3.6           1.4          0.2 Iris-setosa
## 6   6           5.4          3.9           1.7          0.4 Iris-setosa
## 7   7           4.6          3.4           1.4          0.3 Iris-setosa
## 8   8           5.0          3.4           1.5          0.2 Iris-setosa
## 9   9           4.4          2.9           1.4          0.2 Iris-setosa
## 10 10           4.9          3.1           1.5          0.1 Iris-setosa
```

```
# shape of the data
dim(dsba2)
```

```
## [1] 150   6
```

```r
# structure of the data
str(dsba2)
```

```
## 'data.frame':    150 obs. of  6 variables:
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ SepalLengthCm: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ SepalWidthCm : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ PetalLengthCm: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ PetalWidthCm : 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 "Iris-setosa",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
# summary of the data:
summary(dsba2)
```

```
##        Id         SepalLengthCm    SepalWidthCm   PetalLengthCm
##  Min.   :  1.00   Min.   :4.300   Min.   :2.000   Min.   :1.000
##  1st Qu.: 38.25   1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600
##  Median : 75.50   Median :5.800   Median :3.000   Median :4.350
##  Mean   : 75.50   Mean   :5.843   Mean   :3.054   Mean   :3.759
##  3rd Qu.:112.75   3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100
##  Max.   :150.00   Max.   :7.900   Max.   :4.400   Max.   :6.900
##   PetalWidthCm              Species
##  Min.   :0.100   Iris-setosa    :50
##  1st Qu.:0.300   Iris-versicolor:50
##  Median :1.300   Iris-virginica :50
##  Mean   :1.199
##  3rd Qu.:1.800
##  Max.   :2.500
```

Inference:

- No NA values are present in the data.
- We have 50 numbers each of the species - Setosa, Versicolor & Virginica
- The mean and median are not far apart indicating less number of outliers.

DATA VISUALIZATION:

Boxplots - to examine outliers:

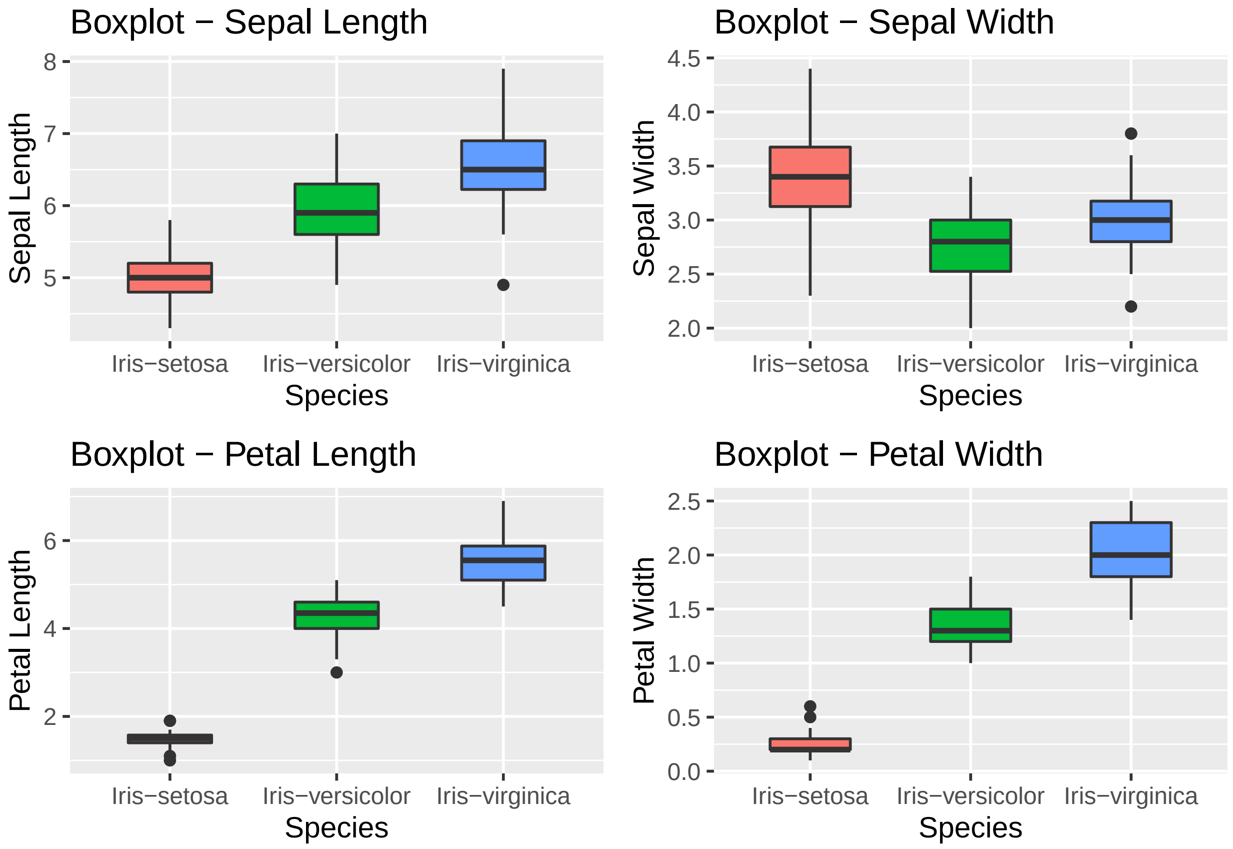
```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.6.3
```

```r
library(ggplot2)
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 3.6.3
```

```r
# SEPAL LENGTH:

plot1 = ggplot(data = dsba2)+ geom_boxplot(aes(x = dsba2$Species,
                                               y = dsba2$SepalLengthCm,
                                               fill = dsba2$Species), width=0.5) +
  labs(title = "Boxplot - Sepal Length",
       x = "Species", y = "Sepal Length") + theme(legend.position = "none")

# SEPAL WIDTH:

plot2 = ggplot(data = dsba2)+ geom_boxplot(aes(x = dsba2$Species,
                                               y = dsba2$SepalWidthCm,
                                               fill = dsba2$Species), width=0.5) +
  labs(title = "Boxplot - Sepal Width", x = "Species",
       y = "Sepal Width") + theme(legend.position = "none")

# PETAL LENGTH:

plot3 = ggplot(data = dsba2)+ geom_boxplot(aes(x = dsba2$Species,
                                               y = dsba2$PetalLengthCm,
                                               fill = dsba2$Species), width=0.5) +
  labs(title = "Boxplot - Petal Length", x = "Species",
       y = "Petal Length") + theme(legend.position = "none")

# PETAL WIDTH:

plot4 = ggplot(data = dsba2)+ geom_boxplot(aes(x = dsba2$Species,
                                               y = dsba2$PetalWidthCm,
                                               fill = dsba2$Species), width=0.5) +
  labs(title = "Boxplot - Petal Width", x = "Species",
       y = "Petal Width") + theme(legend.position = "none")


grid.arrange(plot1, plot2, plot3, plot4, ncol=2)
```



Inference:

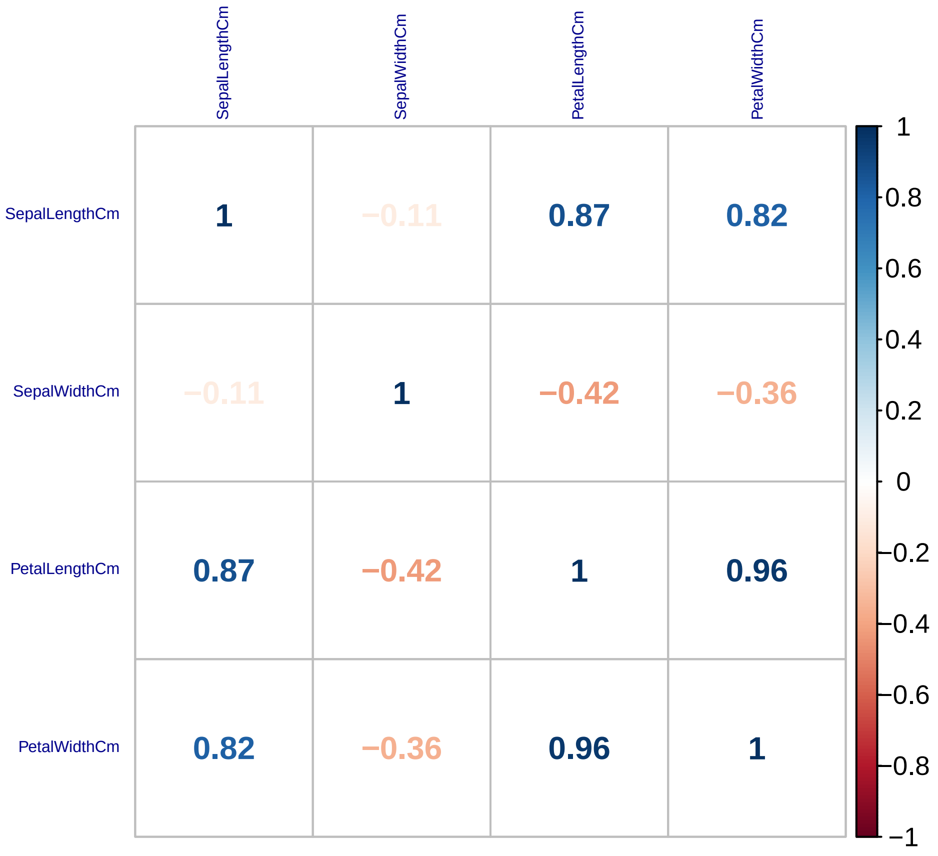
- We have some outliers in all the columns.
- Virginica and Setosa are respectively the largest and the smallest of the flowers.
- Sepal Width is different from the other attributes.

```
# Correlation:

library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.6.2
```

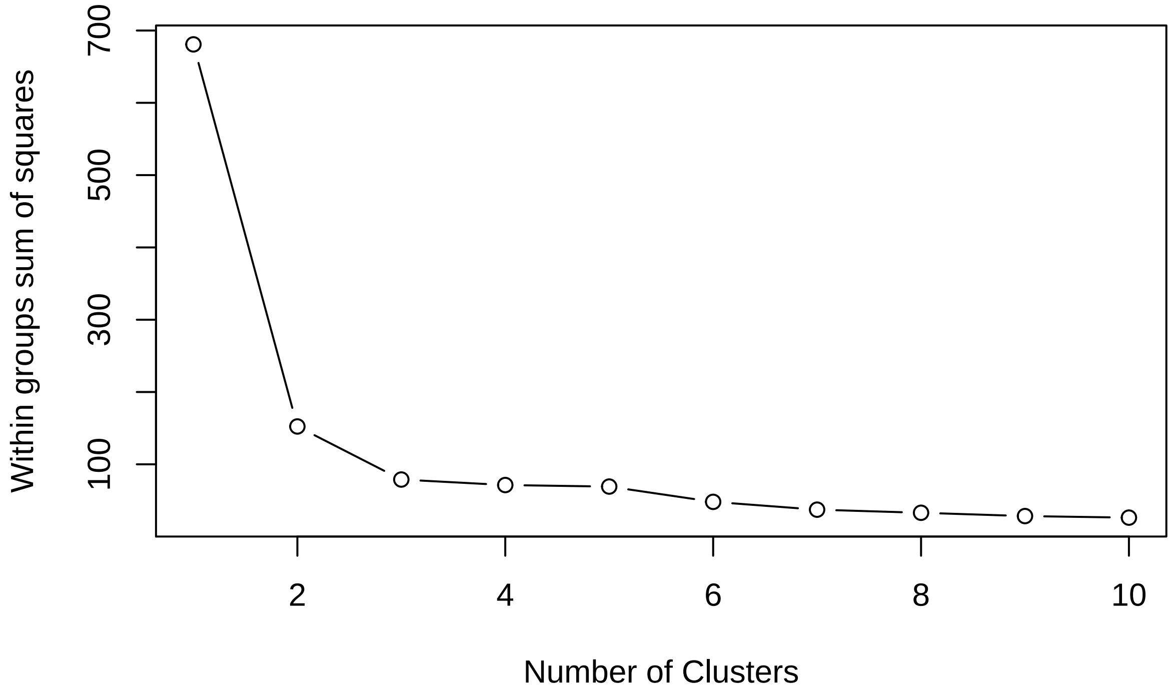
```
## corrplot 0.84 loaded
```

```
dsba2_plot = corrplot(cor(dsba2[, 2:5]),
                      method = "number",
                      tl.cex = 0.5,
                      tl.col = "dark blue")
```

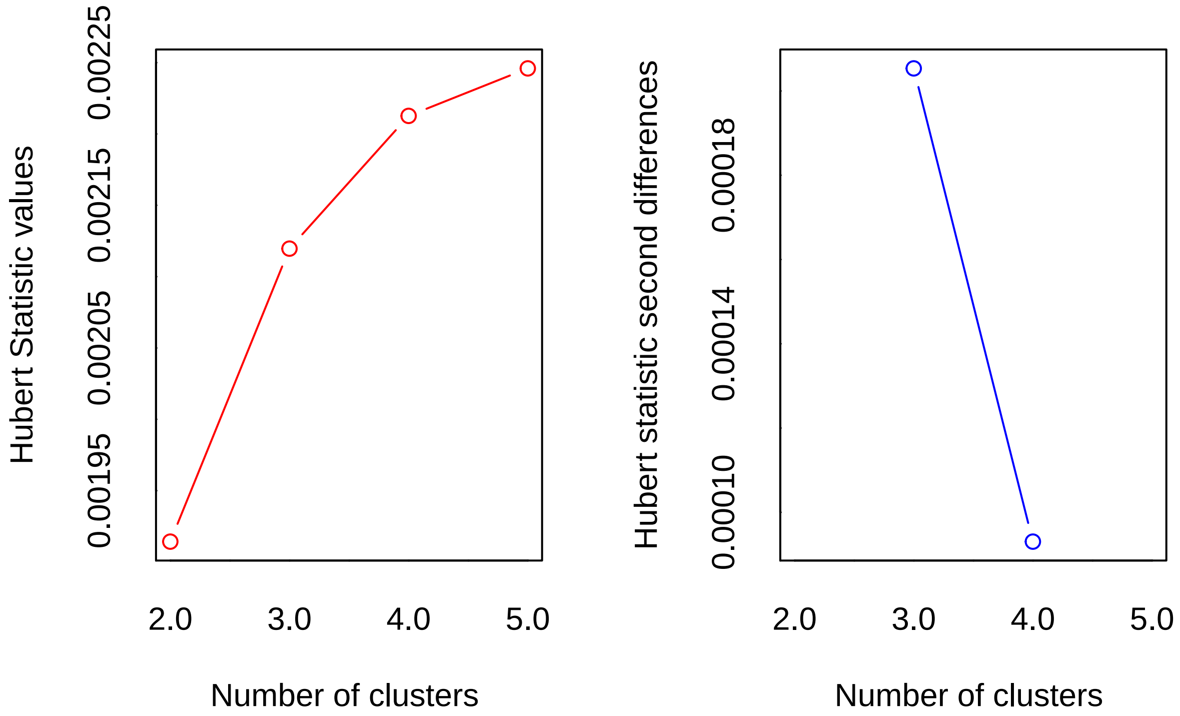


#Finding the optimal number of clusters:

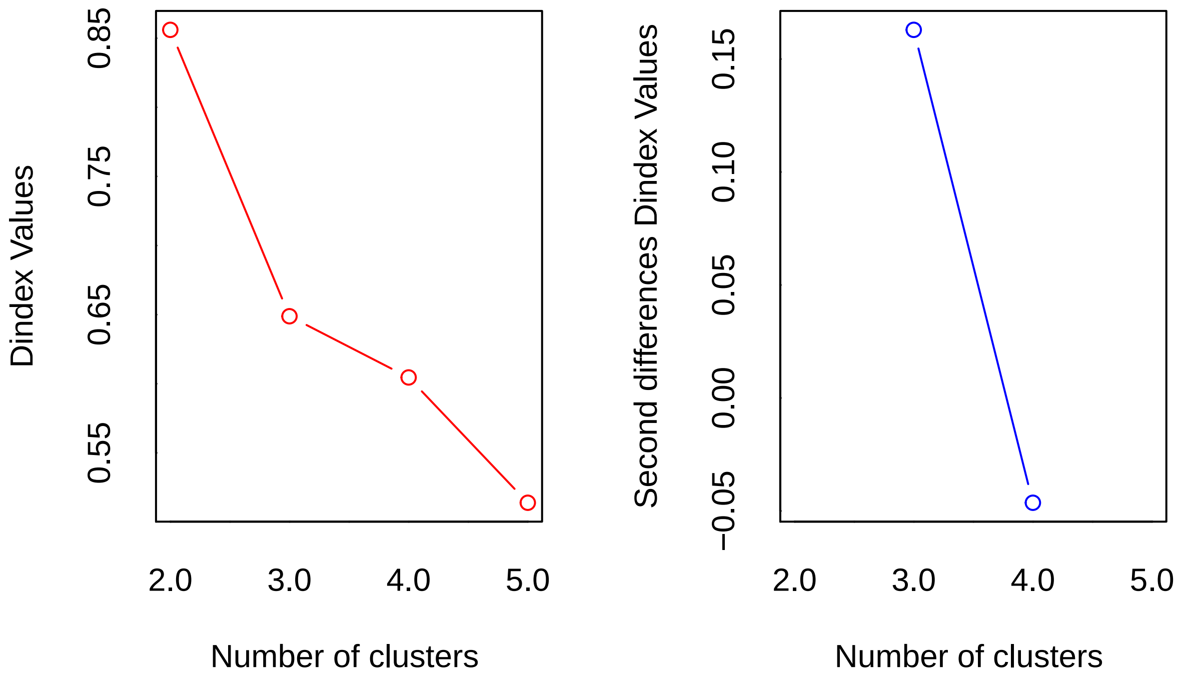
```
## Finding optimal number of clusters from WSS

wssplot = function(data, nc=15, seed=123){
  wss = (nrow(data)-1)*sum(apply(data,2,var))
  for (i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
  plot(1:nc, wss, type="b", xlab="Number of Clusters",
       ylab="Within groups sum of squares")}

wssplot(dsba2[,2:5], nc=10)
```



```
## Identifying the optimal number of clusters using NbClust:

library(NbClust)

set.seed(123)
Nclus <- NbClust(dsba2[,2:5], min.nc=2, max.nc=5, method="kmeans")
```



```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```



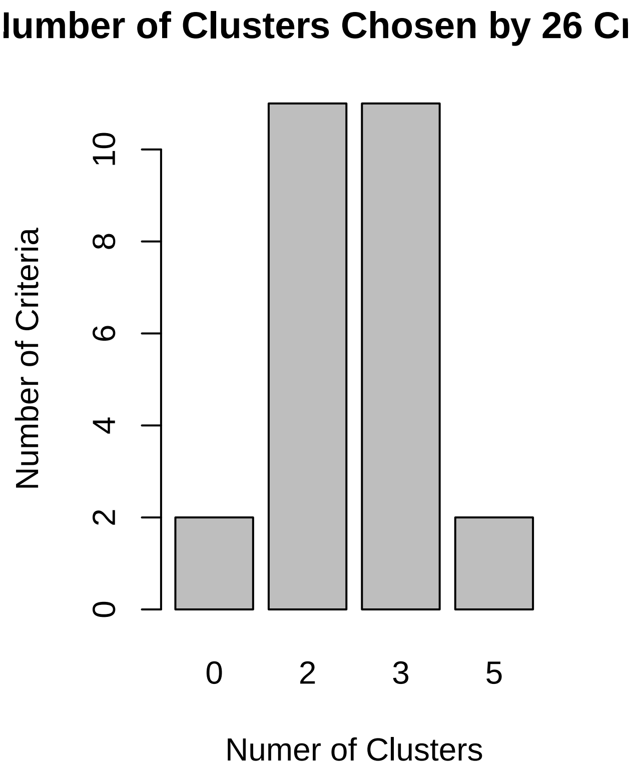
```
## *** : The D index is a graphical method of determining the number of clusters.
##                  In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                  second differences plot) that corresponds to a significant increase of the value of
##                  the measure.
##
## *******************************************************************
## * Among all indices:
## * 11 proposed 2 as the best number of clusters
## * 11 proposed 3 as the best number of clusters
## * 2 proposed 5 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

```
table(Nclus$Best.n[1,])
```

```
##
##  0  2  3  5
##  2 11 11  2
```

```
barplot(table(Nclus$Best.n[1,]),
          xlab="Numer of Clusters", ylab="Number of Criteria",
          main="Number of Clusters Chosen by 26 Criteria")
```

```
# According to the majority rule, the best number of clusters is  2
```



Forming & Plotting the clusters:

```
kmeans_clust = kmeans(x=dsba2[,2:5], centers = 2, nstart = 5)
kmeans_clust
```

```
## K-means clustering with 2 clusters of sizes 97, 53
##
## Cluster means:
##   SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm
## 1      6.301031     2.886598      4.958763    1.6958763
## 2      5.005660     3.360377      1.562264    0.2886792
##
## Clustering vector:
##   [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 2 2 2 2 2 2 2 2 2 2 2
##  [36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
##  [71] 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 2 1 1 1 1 1 1
## [106] 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
## [141] 1 1 1 1 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 123.79588  28.57283
##  (between_SS / total_SS =  77.6 %)
```

```
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

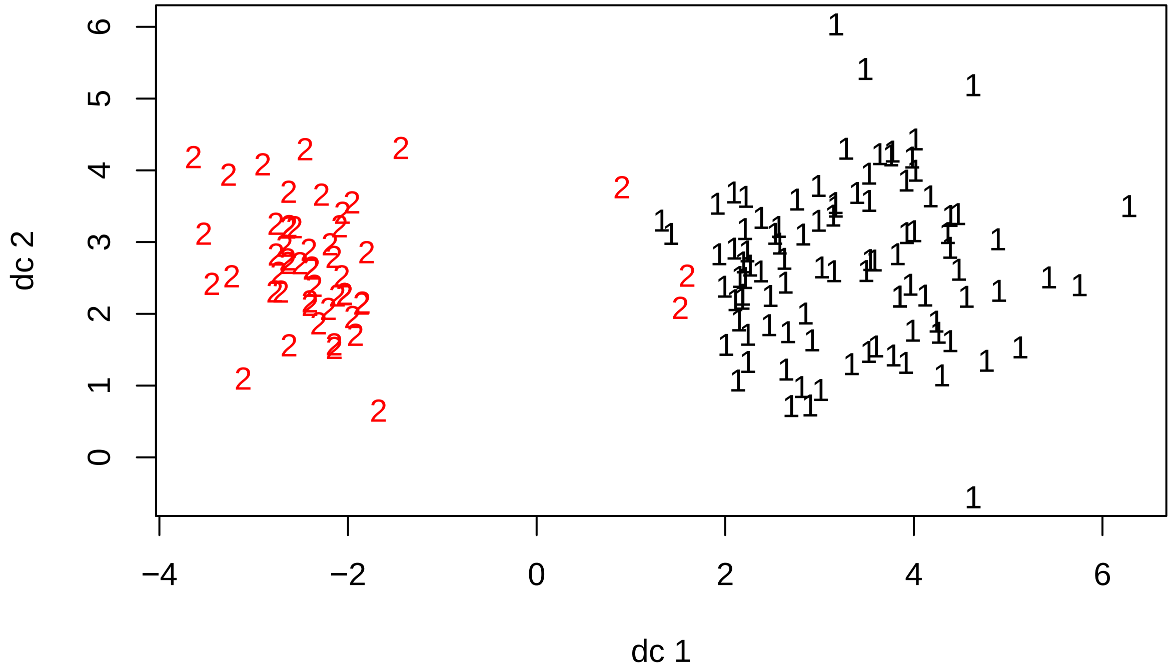
```r
# K-means clustering with 2 clusters of sizes 97, 53
# the percentage similarity between data in the same cluster is 77.6


library(fpc)
```

```
## Warning: package 'fpc' was built under R version 3.6.3
```

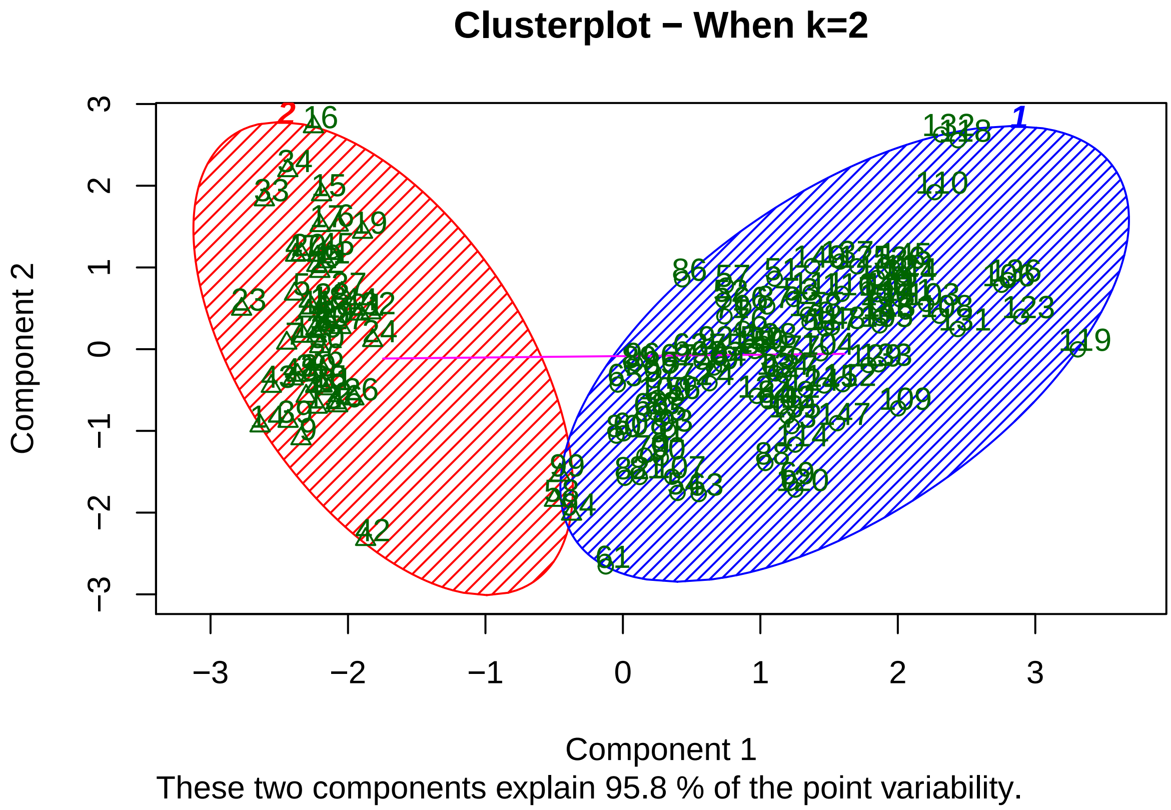
```r
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 3.6.1
```

```r
## plotting the clusters

plotcluster(dsba2[,2:5], kmeans_clust$cluster)
```



```r
# More complex

clusplot(dsba2[,2:5], main = "Clusterplot - When k=2",
         kmeans_clust$cluster,
         color=TRUE, shade=TRUE, labels=2, lines=1)
```



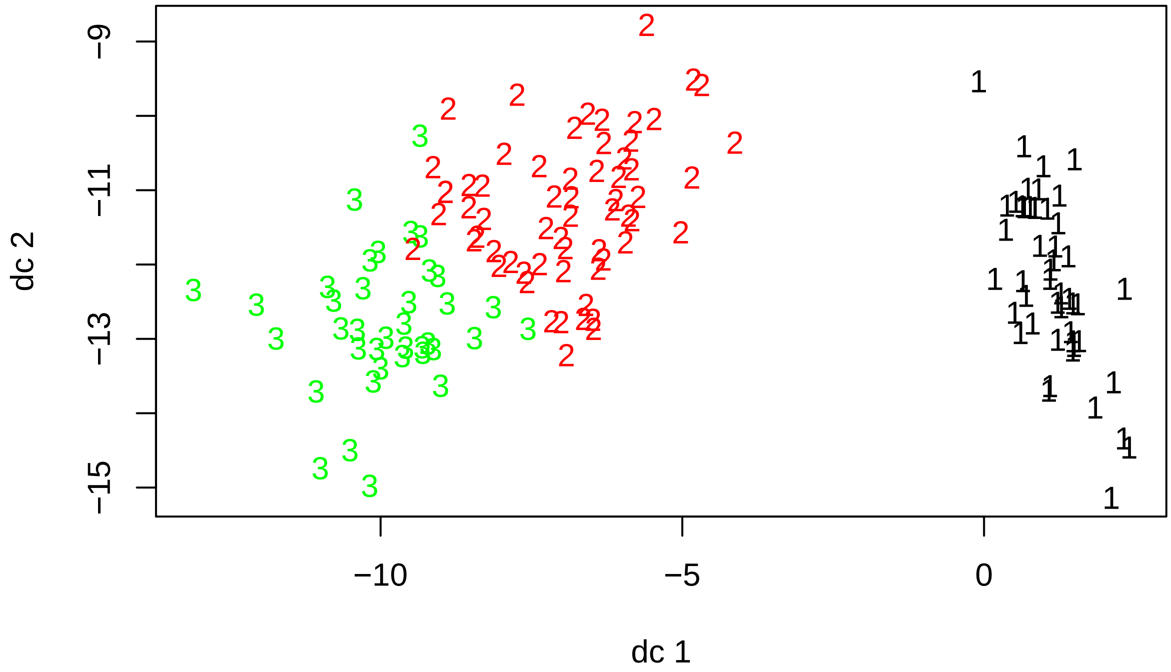
Let us look for the similarity percentage with 3 clusters:

```
# Forming & Plotting the clusters:

kmeans_clust2 = kmeans(x=dsba2[,2:5], centers = 3, nstart = 5)
kmeans_clust2
```

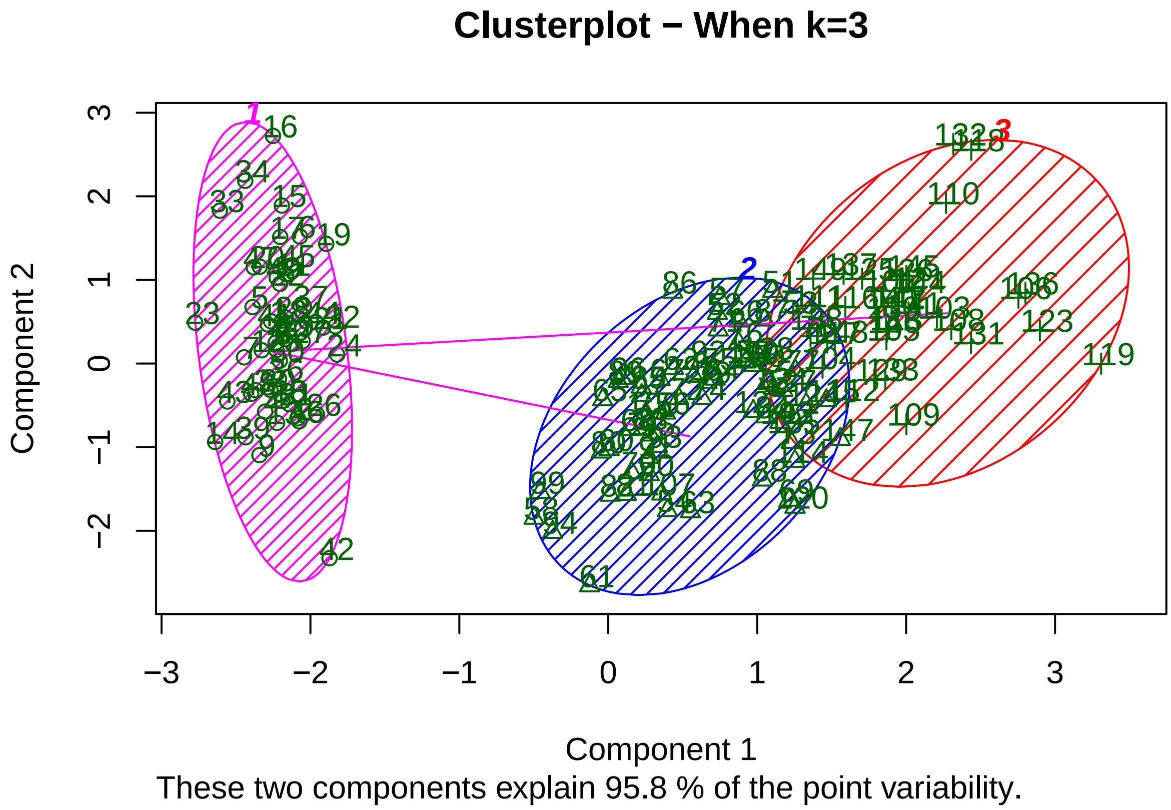
```
## K-means clustering with 3 clusters of sizes 50, 62, 38
##
## Cluster means:
##   SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm
## 1      5.006000     3.418000      1.464000     0.244000
## 2      5.901613     2.748387      4.393548     1.433871
## 3      6.850000     3.073684      5.742105     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
##  [36] 1 1 1 1 1 1 1 1 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
##  [71] 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 2 2 2 2 3 2 3 3 3
## [106] 3 2 3 3 3 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
## [141] 3 3 2 3 3 3 2 3 3 2
##
## Within cluster sum of squares by cluster:
## [1] 15.24040 39.82097 23.87947
##  (between_SS / total_SS =  88.4 %)
```

```
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
# K-means clustering with 3 clusters of sizes 38, 50, 62
# the percentage similarity between data in the same cluster is 88.4 %
# The increase of one cluster can increase similarity about 11% which is great.

## plotting the clusters

plotcluster(dsba2[,2:5], kmeans_clust2$cluster)
```



```r
# More complex

clusplot(dsba2[,2:5], main = "Clusterplot - When k=3",
         kmeans_clust2$cluster,
         color=TRUE, shade=TRUE, labels=2, lines=1)
```



Clustering Validation:

We may use the silhouette coefficient (silhouette width) to evaluate the goodness of our clustering. We will see the clusters silhouette plot and measure the average silhouette width to confirm which is better. The more the value is closer to 1, the better is the clustering of data points. Hence, the model (k value) that has the best silhouette coefficient will be confirmed.

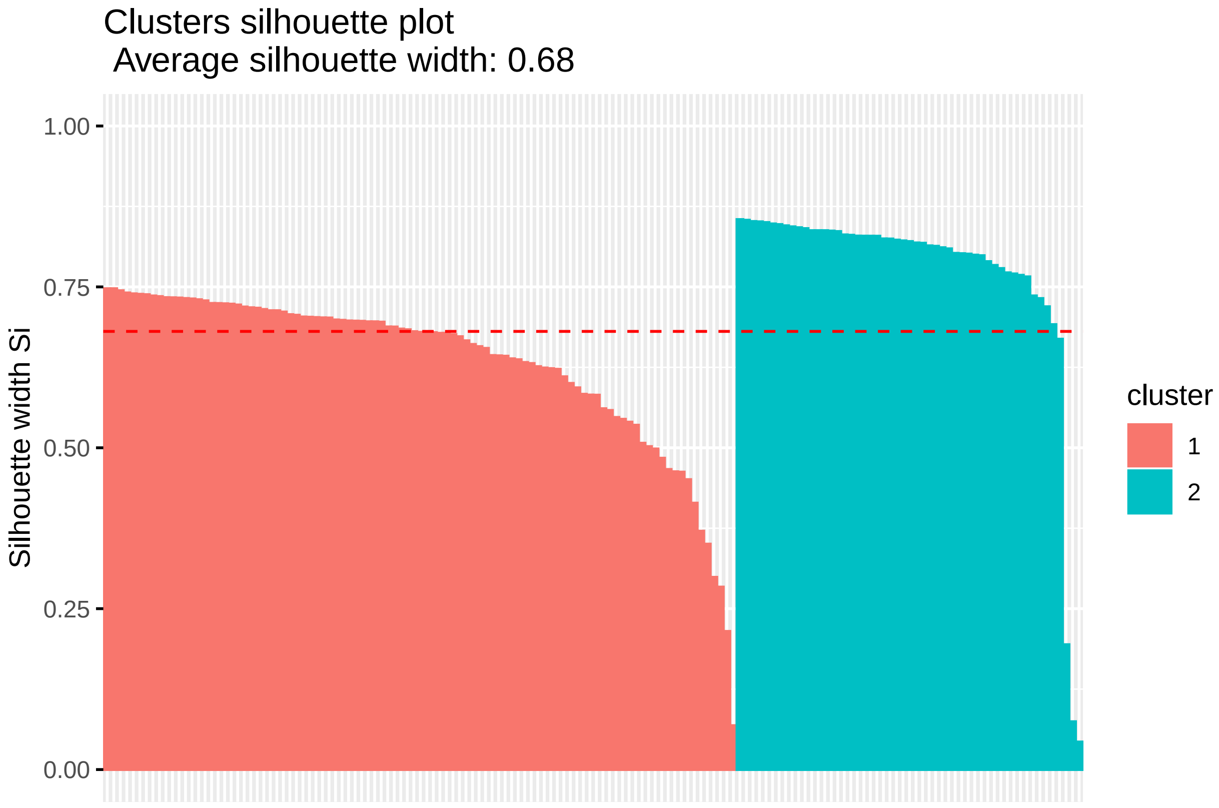
```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 3.6.1
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

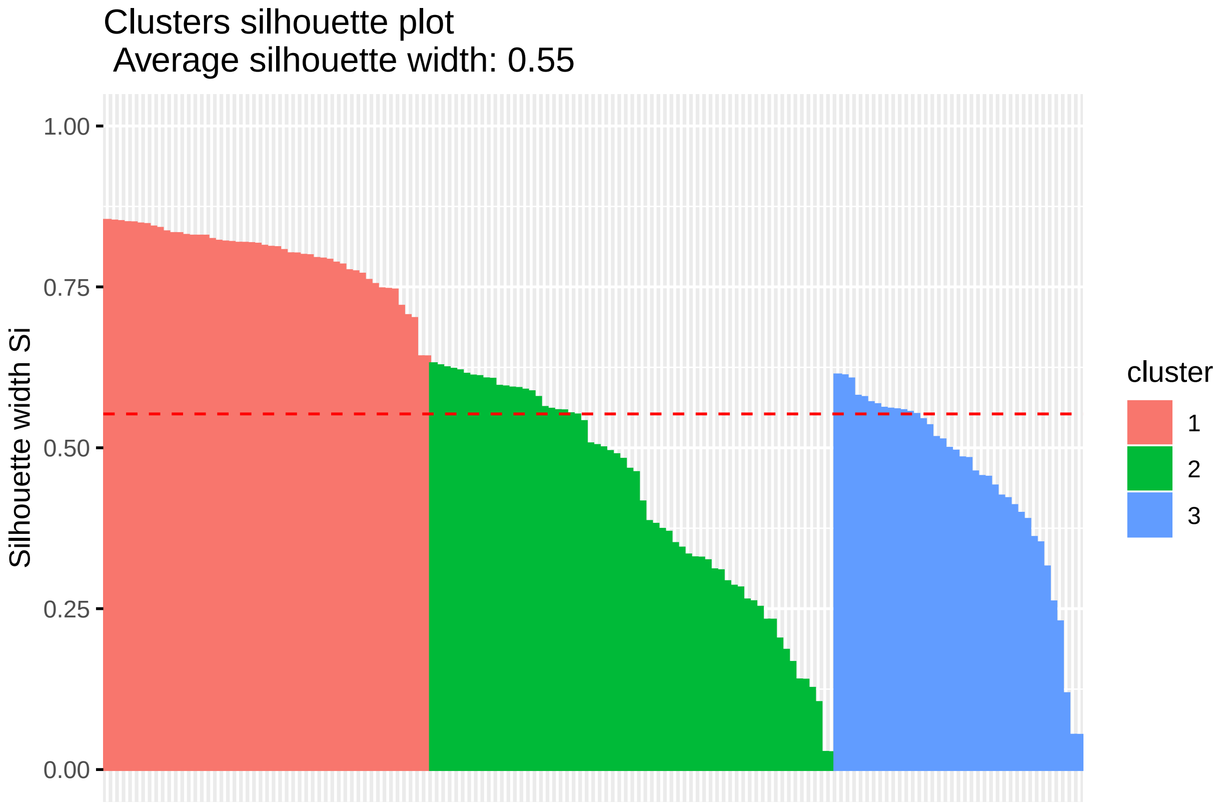
```
# When k=2
sil_k2 <- silhouette(kmeans_clust$cluster, dist(dsba2[,2:5]))
fviz_silhouette(sil_k2)
```

```
##   cluster size ave.sil.width
## 1       1   97          0.63
## 2       2   53          0.77
```



```
# Cluster 1 (97 data points) has avg.sil.width of 0.63
# Cluster 2 (53 data points) has avg.sil.width of 0.77
# Average silhouette width : 0.68

# When k=3
sil_k3 <- silhouette(kmeans_clust2$cluster, dist(dsba2[,2:5]))
fviz_silhouette(sil_k3)
```

```
##   cluster size ave.sil.width
## 1       1   50          0.80
## 2       2   62          0.42
## 3       3   38          0.45
```



```
# Cluster 1 (38 data points) has avg.sil.width of 0.45
# Cluster 2 (50 data points) has avg.sil.width of 0.80
# Cluster 3 (62 data points) has avg.sil.width of 0.42
# Average silhouette width : 0.55
```

Inference:

Upon observing the clusters, we find that the average silhouette width when k=2 is more than when k=3. And since the individual silhouette width of both the clusters in model 1 is more than 0.5 (closer to 1) than the 3 clusters from the next model, we choose k=2 as the optimum number of clusters.