TSF-GRIP (Feb`23)

Name : **Bhanuprakash**

Task : **Prediction using Un Supervised ML (Task#2)**

To Do : **predict the optimum number of clusters and represent it visually.**

Tool : **R**

13/02/2023

Loading required libraries

```
library(factoextra)
```

```
## Loading required package: ggplot2
```

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

Lets prepare the dataset

```
#Loading iris data
df1=iris
head(df1)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
tail(df1)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 145          6.7         3.3          5.7         2.5 virginica
## 146          6.7         3.0          5.2         2.3 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 148          6.5         3.0          5.2         2.0 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

Dimension of the data

```
dim(df1)
```

```
## [1] 150   5
```

Summary

```
summary(df1)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

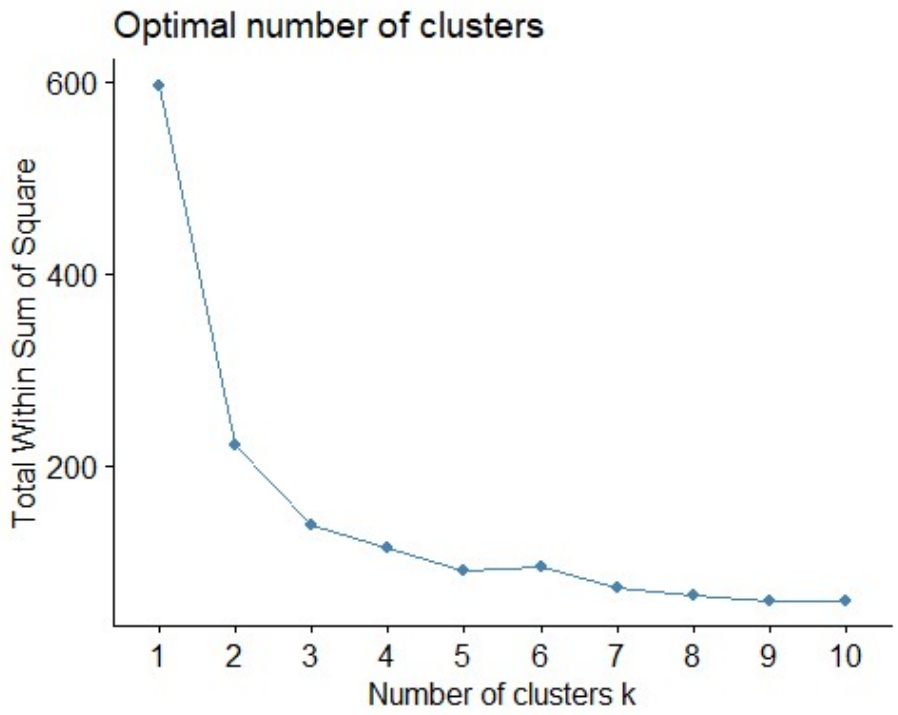
Scaling variables for cluster analysis

```
scale.df1=scale(df1[1:4])
head(scale.df1)
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## [1,]   -0.8976739  1.01560199    -1.335752   -1.311052
## [2,]   -1.1392005 -0.13153881    -1.335752   -1.311052
## [3,]   -1.3807271  0.32731751    -1.392399   -1.311052
## [4,]   -1.5014904  0.09788935    -1.279104   -1.311052
## [5,]   -1.0184372  1.24503015    -1.335752   -1.311052
## [6,]   -0.5353840  1.93331463    -1.165809   -1.048667
```

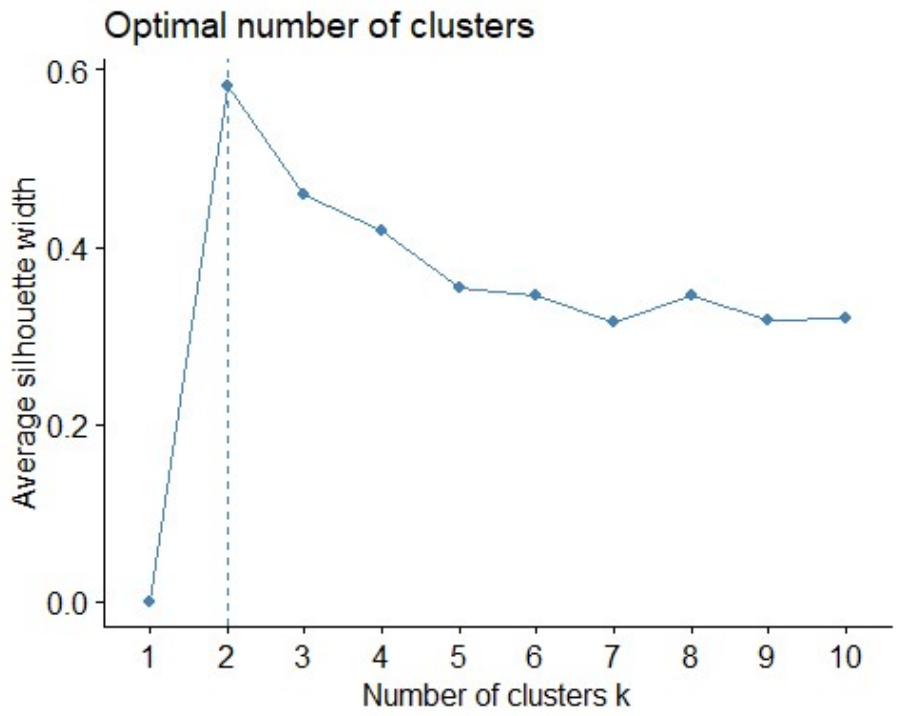
Finding Optimal number of clusters for kmeans using plots.

```
# Using method of total within sum of squares
#an Elbow plot
fviz_nbclust(scale.df1, kmeans, method = "wss")
```



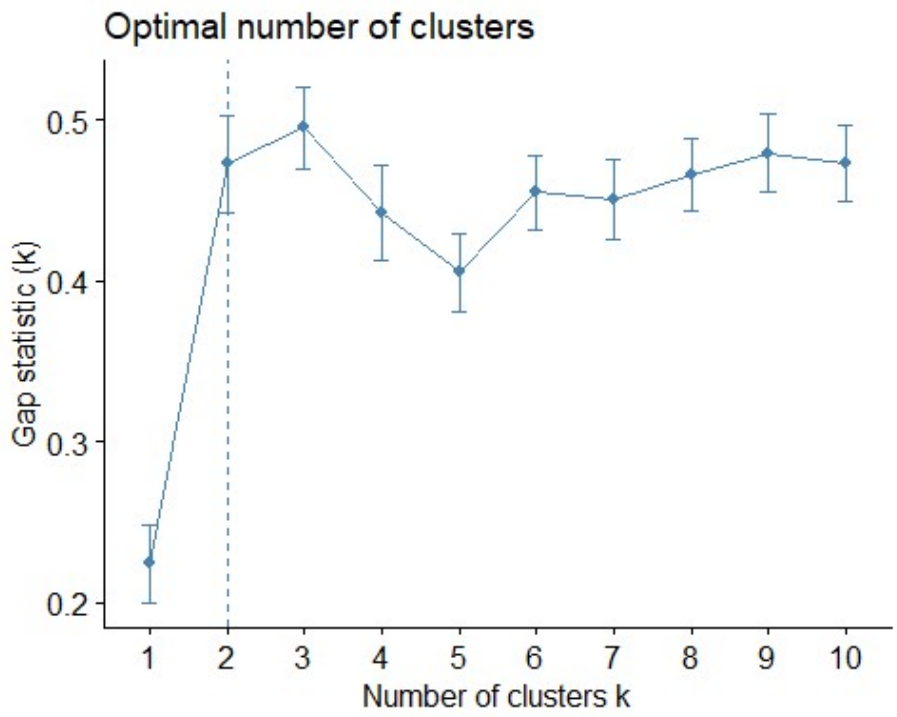
In this method,the cluster point at 3 had some bent position declining continously. therefore optimal no.of clusters is k=3 a/c to this plot.Lets see another method..

```
# Using method of silhouette
fviz_nbclust(scale.df1, kmeans, method = "silhouette")
```



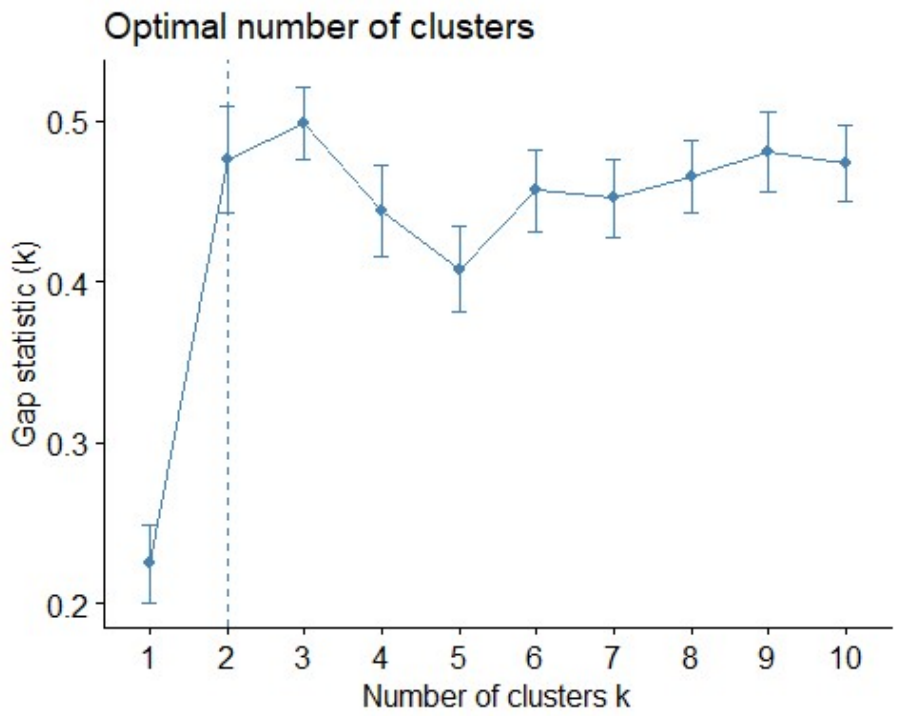
In this method also cluster point 3 had some bent position. therefore optimal no.of clusters is k=3 .Lets see another method..

```
# Using method of gap_stat
fviz_nbclust(scale.df1, kmeans, method = "gap_stat")
```



we can see that gap statistic is highest at k = 3 clusters, which matches the wss method.i.e..Elbow method. Lets see what will happen for 50 obs`..

```
# Using method of gap_stat for 50 obs`
fviz_nbclust(scale.df1, kmeans, method = "gap_stat", nboot = 50)
```



It is same as above...

From all these plots we can interprest that,the optimal number of clusters occurs at the k = 3.Hence ,our data has 3 cluster formations.

Computing K-means clustring for k=3 clusters

```
set.seed(246)

k.means = kmeans(scale.df1, centers = 3, nstart = 30)
k.means

## K-means clustering with 3 clusters of sizes 53, 50, 47
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1  -0.05005221 -0.88042696    0.3465767   0.2805873
## 2  -1.01119138  0.85041372   -1.3006301  -1.2507035
## 3   1.13217737  0.08812645    0.9928284   1.0141287
##
## 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 2
##  [38] 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 1 1 3 1 1 1 1 1 1 3 1 1 1 3 1 1 1
##  [75] 1 3 3 3 1 1 1 1 1 1 3 3 1 1 1 1 1 1 1 1 1 1 1 3 1 3 3 3 3 1 3 3 3 3
## [112] 3 3 1 1 3 3 3 3 1 3 1 3 1 3 3 1 3 3 3 3 3 3 1 1 3 3 3 1 3 3 3 1 3 3 3 1 3
## [149] 3 1
##
## Within cluster sum of squares by cluster:
## [1] 44.08754 47.35062 47.45019
##  (between_SS / total_SS =  76.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

It can be identified that 53,50,47 percent of species are assigned to first,second,third clusters respectively.

Lets find the means or centroids of each cluster

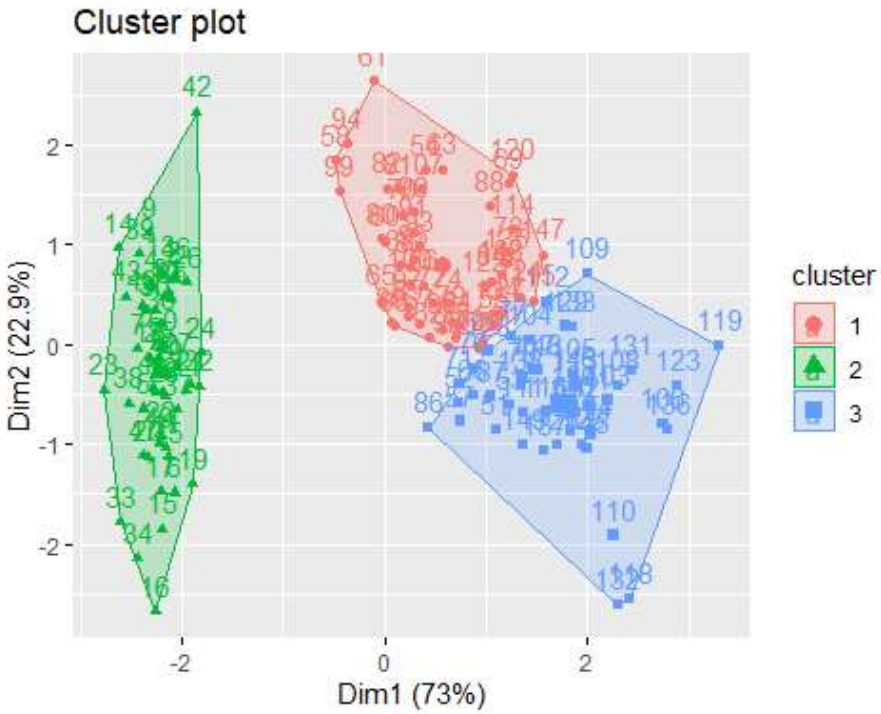
```
aggregate(scale.df1, by=list(cluster=k.means$cluster), mean)

##   cluster Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1       1  -0.05005221 -0.88042696    0.3465767   0.2805873
## 2       2  -1.01119138  0.85041372   -1.3006301  -1.2507035
## 3       3   1.13217737  0.08812645    0.9928284   1.0141287
```

These are all the centroids of all species dimensions for each cluster.

k-means model visualization

```
fviz_cluster(k.means, data = scale.df1)
```



Finally, With respect to the problem statement ,it is concluded that there are 3 clusters (k=3) for this iris dataset and visualized it graphically.

—— THANK YOU ——