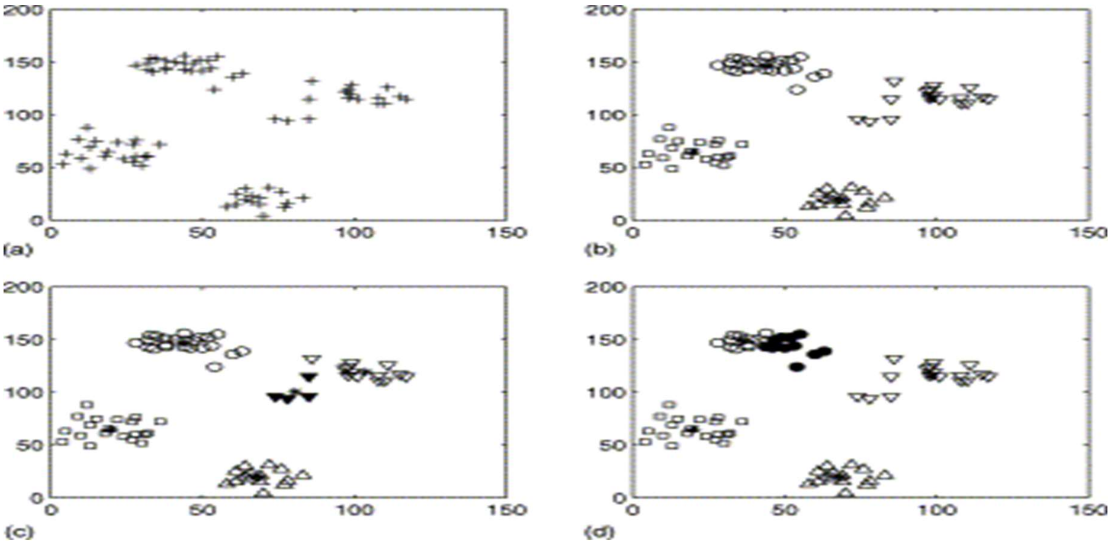
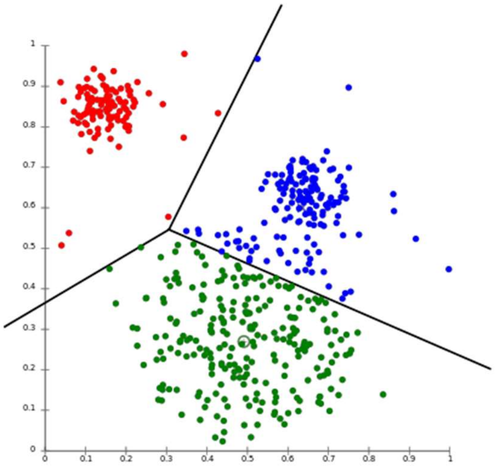
K-MEANS CLUSTERING ANALYSIS

An introduction to K-Means clustering algorithm, how it works and its analysis on “iris” dataset in R Studio.

K-means is a partitioning method (non-hierarchical method) that divides observations in data into k mutually exclusive clusters (Lletí et al., 2004). In this way, it treats each observation in data as an object having a location in space and finds a partition in which objects within each cluster are as close to each other as possible, and as far from objects in other clusters as possible. Each cluster in the partition is defined by its member objects and by its centroid, or Centre. The centroid for each cluster is the point to which the sum of distances from all objects in that cluster is minimized. It is important to highlight that k-means is an iterative algorithm that minimizes the sum of distances from each object to its cluster centroid, over all clusters by moving objects between clusters until the sum cannot be decreased further. The result is thus a set of clusters that are as compact and well separated as possible (Lletí et al., 2004).

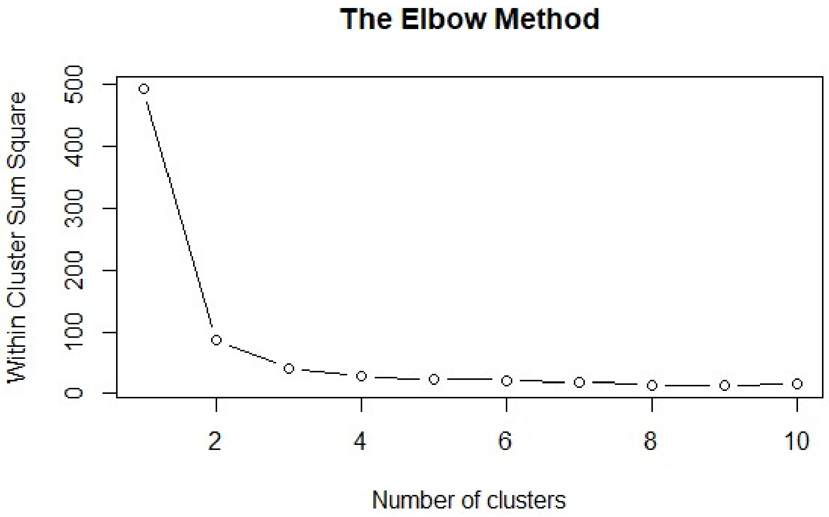


(Image source: Lletí et al., 2004)



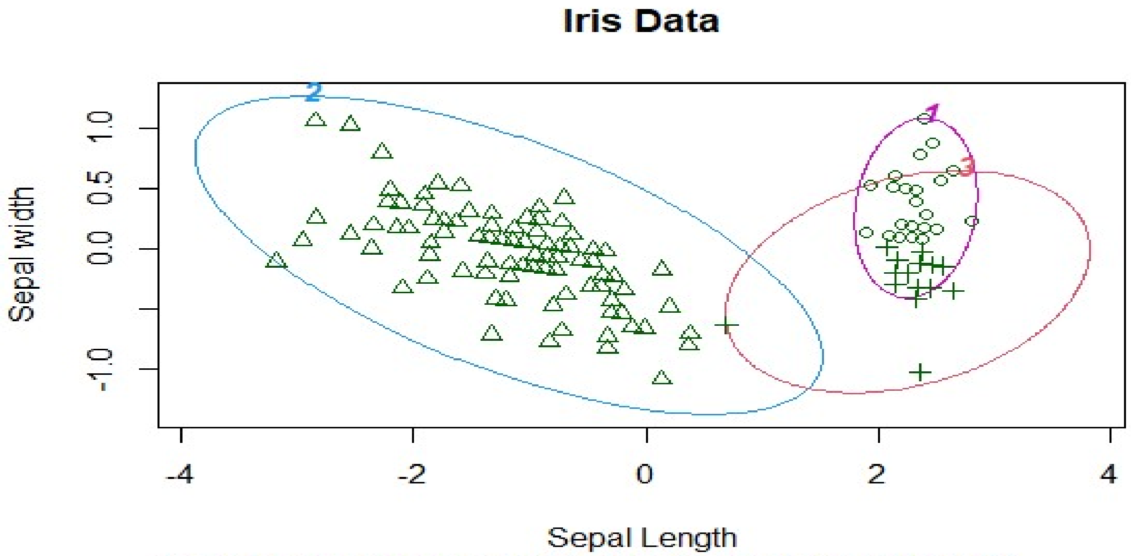
(Image source: Google)

The oldest method for determining the true number of clusters in a data set is inelegantly called the elbow method (Kodinariya & Makwana, 2013). It is a visual method. The idea is that Start with K=2, and keep increasing it in each step by 1, calculating your clusters and the cost that comes with the training. At some value for K the cost drops dramatically, and after that it reaches a plateau when you increase it further. This is the K value you want. The calculation of cluster code, with the iris data using the elbow method (A) produces the following output

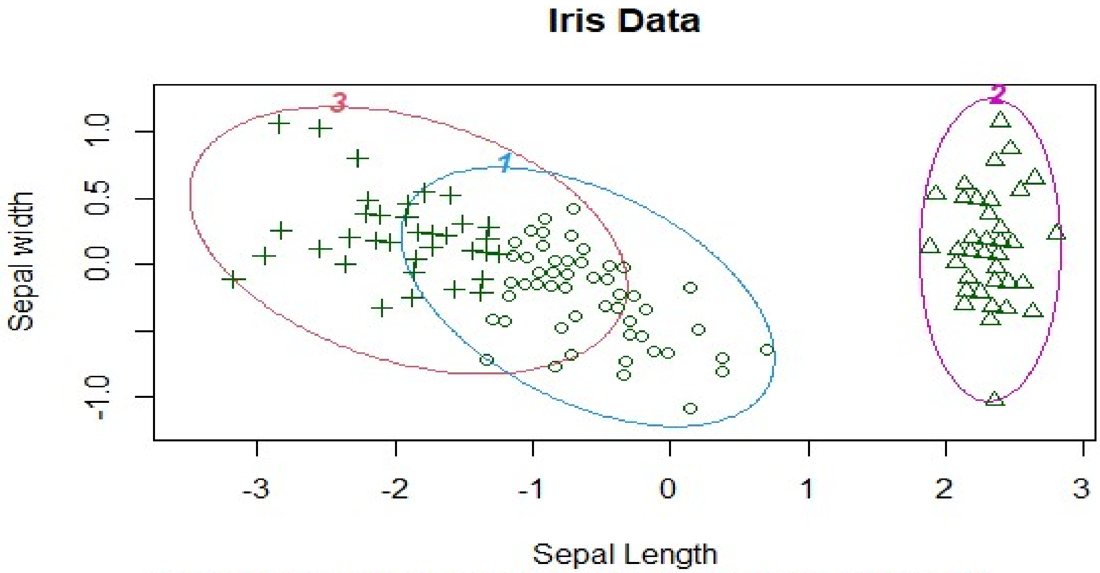


in R:-

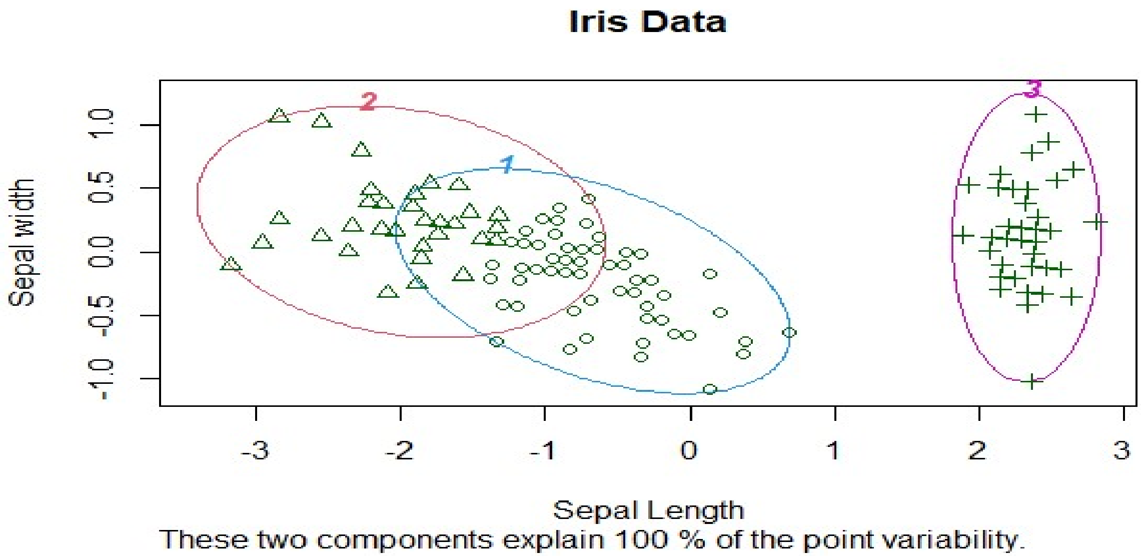
The difference in dip between values 3 and 4 are significantly dropped, as compared between 1-2 and 2-3, which in the process make an elbow shape structure. As a result, the value from which the deviation is reduced is picked, in this case, 3. After working with how to calculate the number of clusters, we move on ahead with plotting the clusters by executing the built in k-means function in R (B) and plotting the results. Some plots can be shown below:-



These two components explain 100 % of the point variability.



These two components explain 100 % of the point variability.



For a general idea, the formula with which it calculates this is:-

$$\text{objective function} \leftarrow J = \sum_{j=1}^{\overset{\text{number of clusters}}{k}} \sum_{i=1}^{\overset{\text{number of cases}}{n}} \underbrace{\left\| \overset{\text{case } i}{x_i^{(j)}} - \overset{\text{centroid for cluster } j}{c_j} \right\|^2}_{\text{Distance function}}$$

As it is seen, all the outputs contain different data points in different clusters. Also, some of the points which were in cluster 1 were in a different cluster the next code execution. Each time the code (B) is executed it will show a different output, resulting in different points in different centroids, or changing the position of clusters altogether. This is happening due to the K mean's random positioning of its centroid in the first step.

K-Means arguably is the most popular clustering method. Due to its properties its interest is to increase numbers of practitioners in marketing research, bioinformatics, customer management, engineering and other application areas, other than data mining and machine learning communities (Kodinariya & Makwana, 2013).

APPENDIX – A

Code to generate the number of clusters in a dataset using the elbow method.

```
iris
newiris <- iris
newiris$Species <- NULL
newiris
newiris <- newiris[2:3]
newiris

# Using the elbow method to find the optimal number of clusters
#set.seed(6)*
clstrvalue = vector()
for (i in 1:10) clstrvalue[i] = sum(kmeans(newiris, i)$withinss)

table(clstrvalue)
plot(1:10, clstrvalue,
    type = 'b',
    main = paste('The Elbow Method'),
    xlab = 'Number of clusters',
    ylab = 'Within Cluster Sum Square')
```

*set.seed() can be used to get one randomly generated value. It has been commented so that it will generate multiple values and cluster value can be decided from that.

APPENDIX – B

Code to generate the K- Means clusters and plotting the same.

```
iris
newiris <- iris
newiris$Species <- NULL
newiris
newiris <- newiris[2:3]
newiris

# Fitting K-Means to the dataset
kc = kmeans(newiris, 3)
kc

# Visualising the clusters
clusplot(newiris, kc$cluster,
         lines = 0, shade = FALSE, color = TRUE, labels = 4,
         plotchar = TRUE, span = FALSE,
         main = 'Iris Data', xlab = 'Sepal Length', ylab = 'Sepal width')
```

(clusplot() and logic to calculate the cluster using the elbow method was referenced from one of the courses in udemy, which was then used for number of cluster calculation and plotting the K-Means clusters.)

Reference

- Wikipedia contributors, "K-means clustering," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=K-means_clustering&oldid=1010544533
- Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. *International Journal of Advance Research in Computer Science and Management Studies*, *1*(6). www.ijarcsms.com
- Lletí, R., Ortiz, M. C., Sarabia, L. A., & Sánchez, M. S. (2004). Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes. *Analytica Chimica Acta*, *515*(1), 87–100. https://doi.org/10.1016/j.aca.2003.12.020