Manojkumar Ravichandran

Kmeans Investigation

K-means is a prototype-based, simple partitional clustering algorithm that attempts to find K non-overlapping clusters. These clusters are represented by their centroids (a cluster centroid is typically the mean of the points in that cluster). The clustering process of K-means is as follows. First, K initial centroids are selected, where K is specified by the user and indicates the desired number of clusters. Every point in the data is then assigned to the closest centroid, and each collection of points assigned to a centroid form a cluster. The centroid of each cluster is then updated based on the points assigned to that cluster. This process is repeated until no point changes clusters. (Wu, 2012)

As per the given code, every time I run it, I get a slightly different output. Sometimes the centroid also changes. The difference can be seen between Fig 1 and Fig 2.

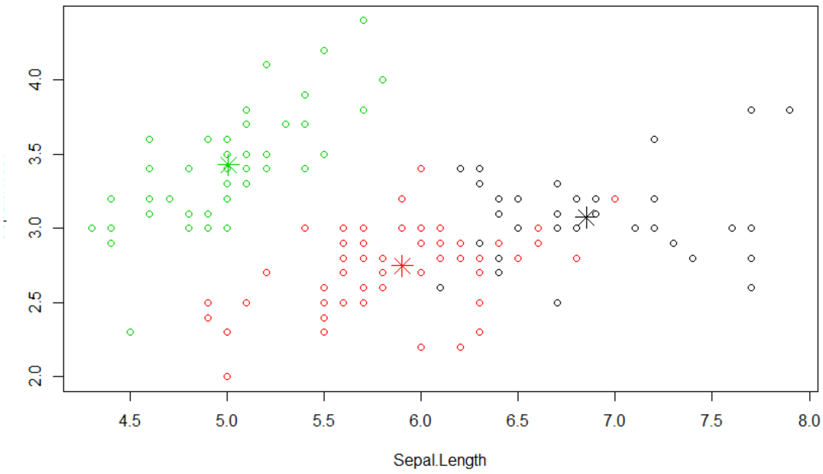

Fig 1

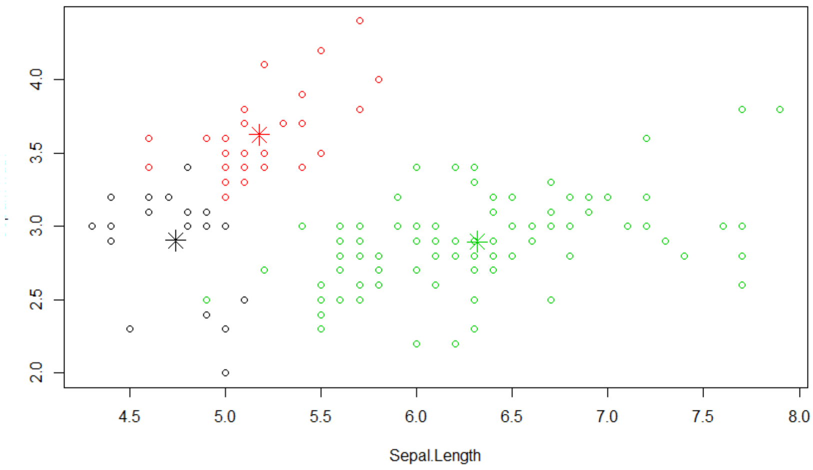

Fig 2

The phenomenon behind K-means is that it picks three points in the 2d plot and it uses the Euclidean distance. In 2 dimensions, the Euclidian distance is the same thing as the Pythagorean theorem. Then we assign the point to the nearest cluster. We then calculate the center of each cluster and create a new position for the cluster.

$$a^2 + b^2 = c^2$$

With the given code, I tried 3D plotting with three different libraries such as Scatter Plot 3D, Plotly and RGL. So that I can have a better understanding of the clusters and its overlaps. To make it a 3D plot graph, I must add another column called Petal Length. These 3D Visualizations can be seen in Fig 3, Fig 4 and Fig 5.

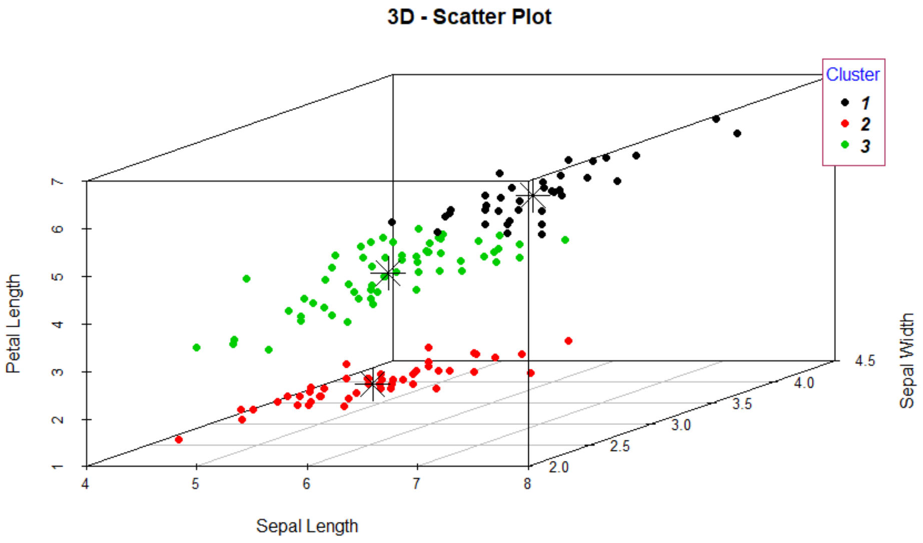

Fig 3 (3D Scatter Plot)

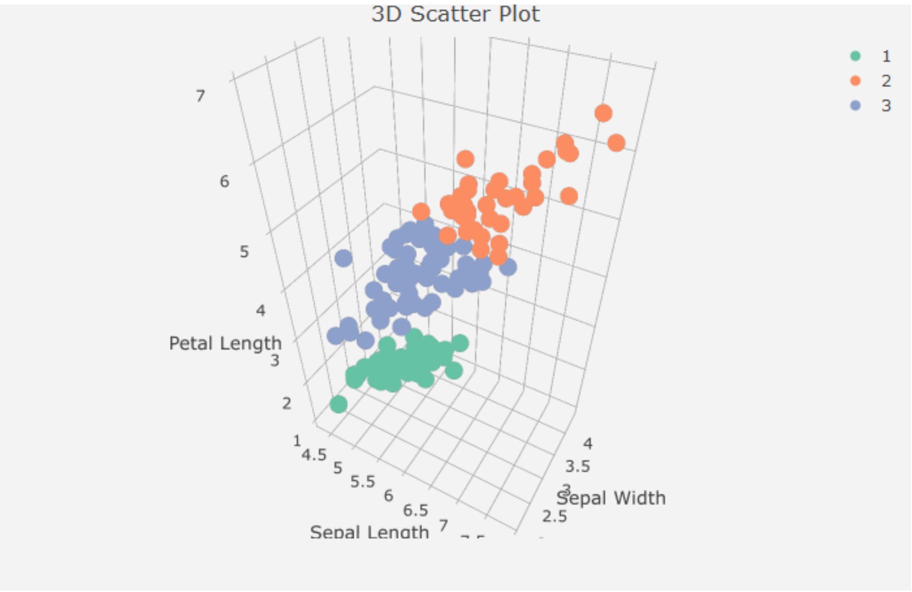

Fig 4.1 (Plotly View 01)

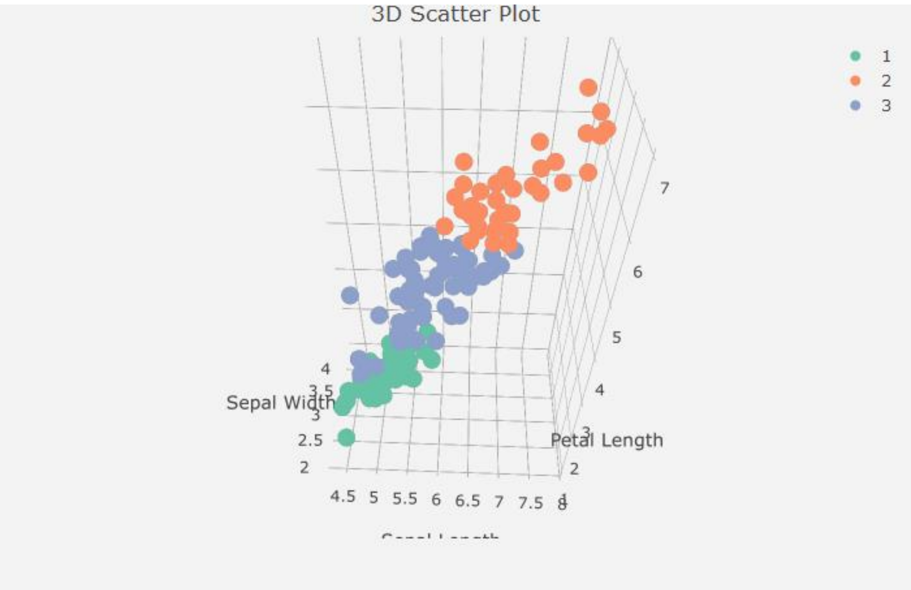

Fig 4.2 (Plotly View 02)

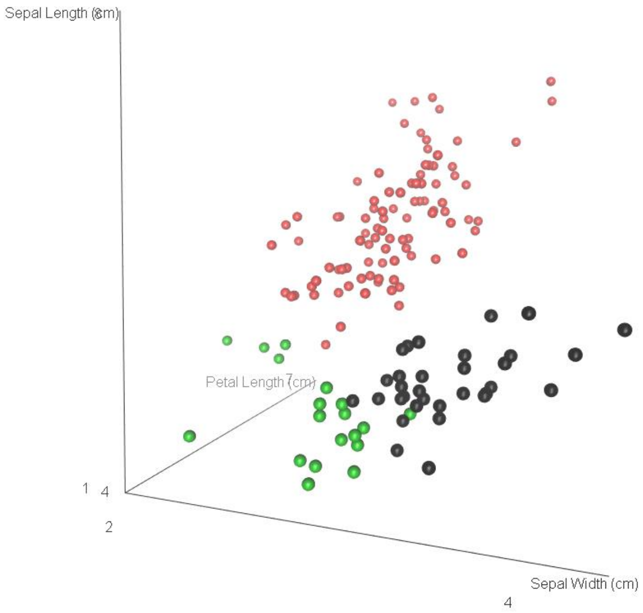

Fig 5.1 (RGL View 01)

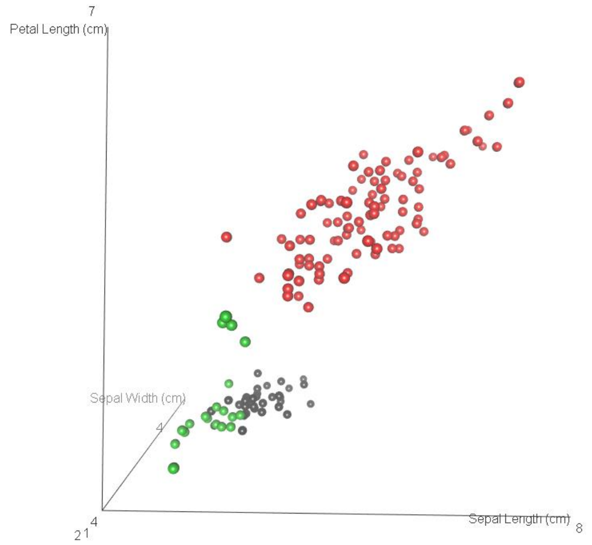

Fig 5.2 (RGL View 02)

The above figures show that there is absence of overlapping. Thus, using these visualizations, we can understand the behavior of the k-means.

APPENDIX

```
install.packages("plotly")
install.packages("scatterplot3d")
library(plotly)
library(scatterplot3d)
iris
newiris <- iris
newiris$Species <- NULL
newiris
kc <- kmeans(newiris, 3)
kc$centers
#3D Plot
plt <- scatterplot3d(x = newiris$Sepal.Length, y = newiris$Sepal.Width, z= newiris$Petal.Length,
                     xlab ='Sepal Length', ylab = 'Sepal Width', zlab = 'Petal Length', main = '3D - Scatter Plot',
                     color = kc$cluster, pch = 16)
plt$points3d(x = kc$centers[,'Sepal.Length'],y = kc$centers[,'Sepal.Width'], z = kc$centers[, 'Petal.Length'],
             pch = 8, cex = 3 )
legend("topright", legend = levels(as.factor(kc$cluster)),
       col = c(1,2,3), pch = 16, box.col = 'maroon', text.font = 4, title = 'Cluster', title.col = 'blue')
#3D Plot 2
plot_ly(newiris, x= ~Sepal.Length, y=~Sepal.Width, z=~Petal.Length,
         color = as.factor(kc$cluster))%>%add_markers()%>%
  layout(title = '3D Scatter Plot', scene = list(xaxis = list(title = 'Sepal Length'),
                                                 yaxis = list(title = 'Sepal Width'),
                                                 zaxis = list(title = 'Petal Length')),
          paper_bgcolor = 'rgb(243, 243, 243)', plot_bgcolor = 'rgb(243, 243, 243)')
#3D Plot 3
install.packages('rgl')
library(rgl)
library(car)
scatter3d(x = newiris$Sepal.Length, y = newiris$Sepal.Width, z= newiris$Petal.Length,
           xlab = 'Sepal Length (cm)', ylab = 'Sepal Width (cm)', zlab = 'Petal Length (cm)',
           groups = as.factor(kc$cluster), type = 's', grid = FALSE, surface = FALSE,
           surface.col = c(1,2,3), axis.col = c('black', 'black', 'black'), add = TRUE)
```

Bibliography

Wu, J. (2012). Cluster Analysis and K-means Clustering: An Introduction. In J. Wu, *Advances in K-means Clustering* (pp. 1-16). Springer, Berlin, Heidelberg.