# Assignment 02

Hajar Faiyad 119200096

2023-06-04

## Chosen dataset description and source

The Iris Dataset, is popular for clustering and classification analysis. It is classically used in the machine learning field where clustering and classification tasks are considered.

For this project, I used the built-in "iris" dataset. This dataset contains measurements of four variables/features for three different iris flower species: Setosa, Virginica, and Versicolor. In more detail, the four features are sepal length and width and petal length and width, in centimeters. Moreover, the dataset contains 150 samples in total, 50 for each species. My purpose is to classify the iris flower into their respective species using clustering based on the previously mentioned four features.

```
# Loading the iris data and printing the summary
data(iris)
summary(iris)
```

```
##   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  
##                 
##                 
##                 
```

As for the source, as stated in the "help(iris)", the data were collected by Anderson, Edgar (1935). The irises of the Gaspe Peninsula, Bulletin of the American Iris Society, 59, 2–5. Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7, Part II, 179–188. .

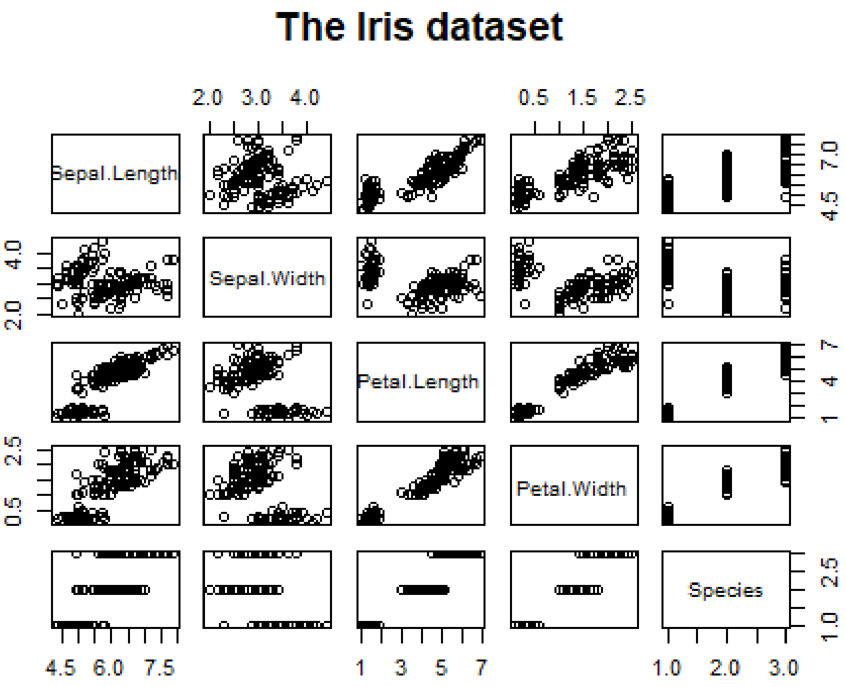
Displaying a sample of the data. As could be understood from the below sample, each row represents an observation (aka sample) and each column represents a feature. This dataset also dialysis the target variable, which is the flower's species.

```
head(iris)
```

```
##   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
```

Plotting the data.

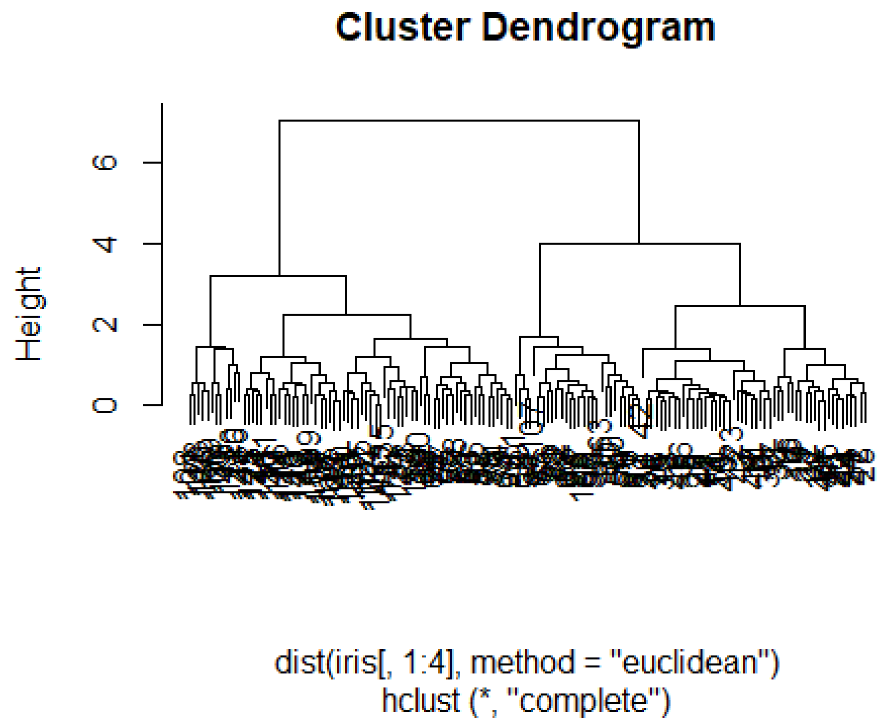
```
plot(iris, main="The Iris dataset")
```



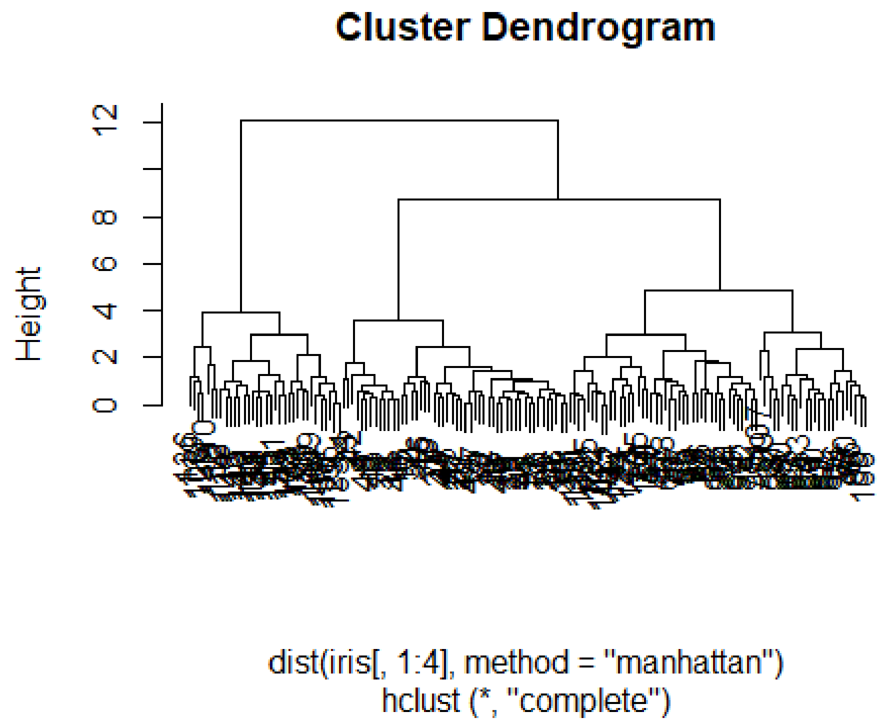
## Analysisng the data.

Displaying the hierarchical clustering represented by the Cluster Dendrogram. Although it is not really visible in our case, I chose to display the labels; I believe they are of importance.

```
plot(hclust(dist(iris[ , 1:4], method = "euclidean")))
```



```
plot(hclust(dist(iris[ , 1:4], method = "manhattan")))
```



After comparing these two hierarchical clustering techniques, I chose to cluster the data using the Manhattan method as my reference; because it displays a more certain or clear dissimilarity between the clusters.

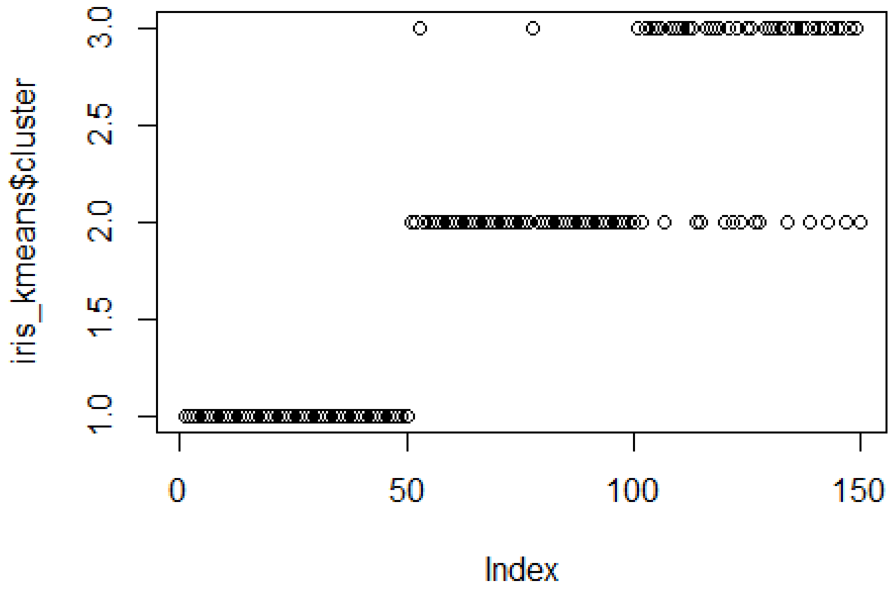
From the Manhattan hierarchical clustering, I noticed that 3 clusters should be used to cluster the data in a more efficient manner. The cluster number benchmark is around 4 for the Manhattan method. For more comparison, the benchmark is around 2 for the euclidean method.

To analyse the data, I used the K-means algorithm to form the clusters.

```
set.seed(123)

# I perform k-means clustering on the first 4 columns, which are the feature
columns.
iris_kmeans <- kmeans(iris[ , 1:4], centers = 3)

# Printing the assignments of clusters
print(iris_kmeans$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
##  [38] 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 2
2 2 2
##  [75] 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 3 2 3
3 3 3
## [112] 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
## [149] 3 2
```

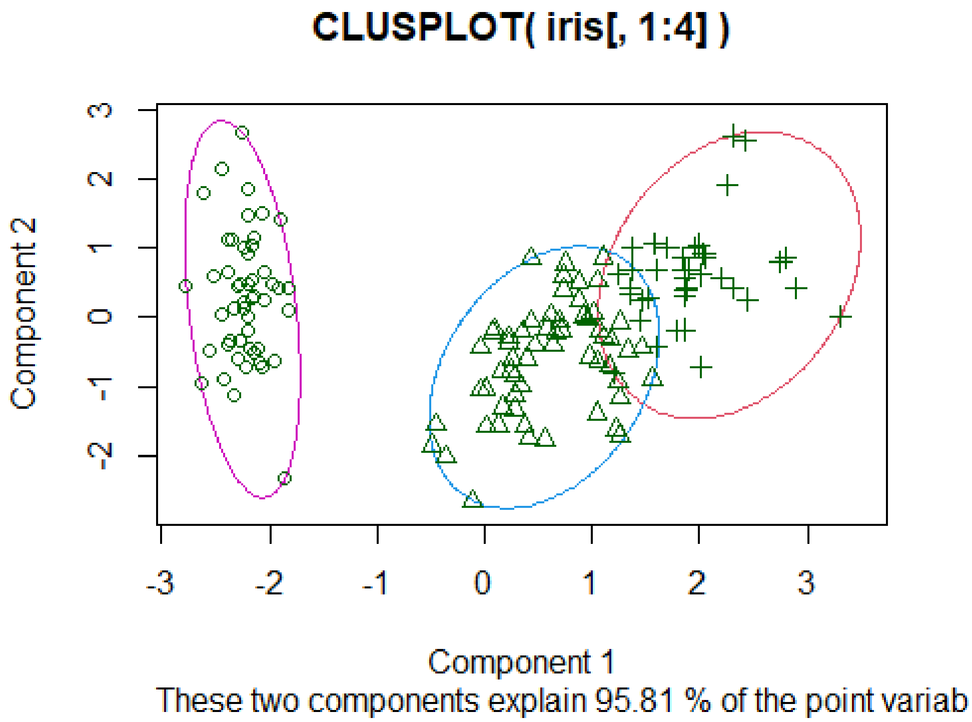
Plotting the clusters for visualization purposes.

```
plot(iris_kmeans$cluster)
```



Visualizing the clusters in a clearer way.

```
library(cluster)
clusplot(iris[ , 1:4], iris_kmeans$cluster, color = TRUE, shade =
FALSE,labels = 1, lines = 0)
```

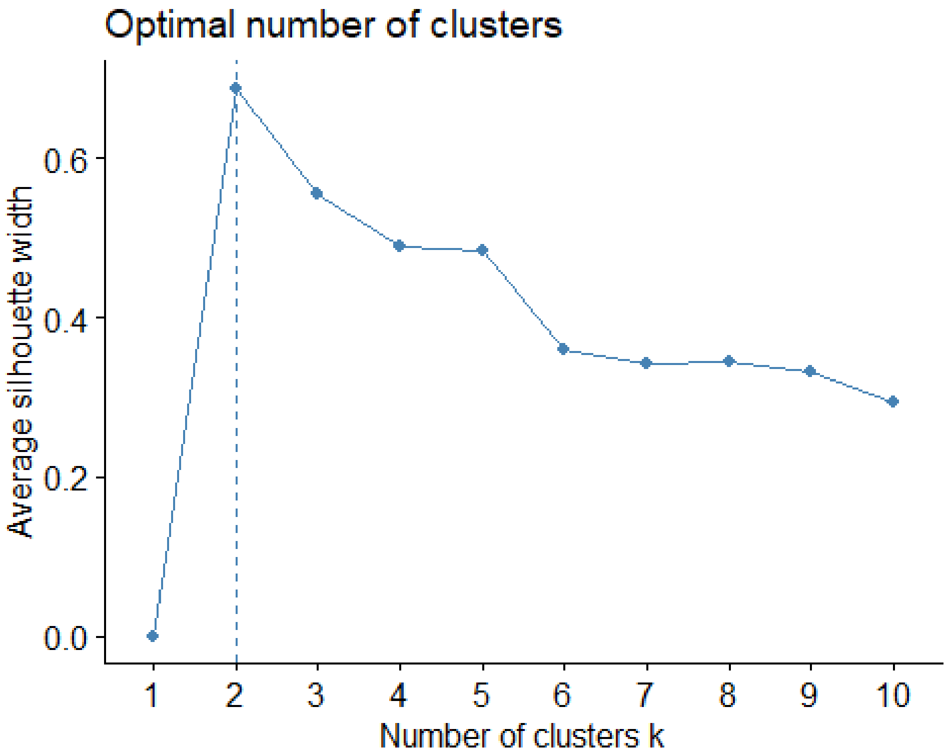


## Quality measures (elbow graph)

To find the best number of clusters, I used the elbow graph.

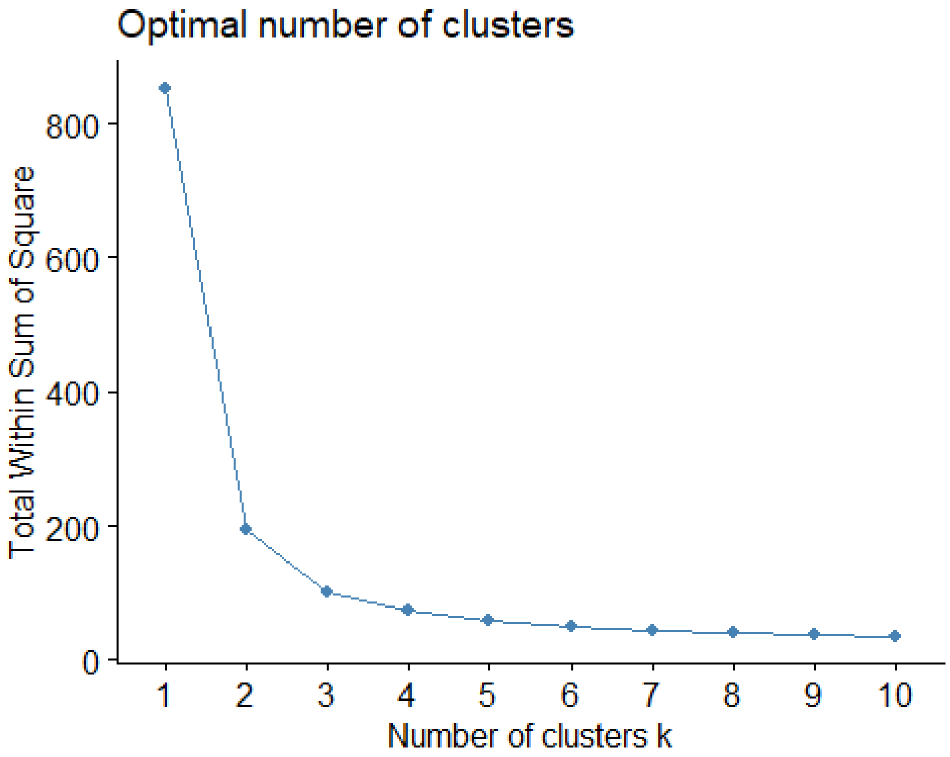
```
library('factoextra')

## Loading required package: ggplot2

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

# if not already done, install the factoextra package.
require("factoextra")
fviz_nbclust(iris, hcut, method = "silhouette", print.summary = T)

## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
```

```
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion
```



```
fviz_nbclust(iris, hcut, method = "wss", print.summary = T)

## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion
```

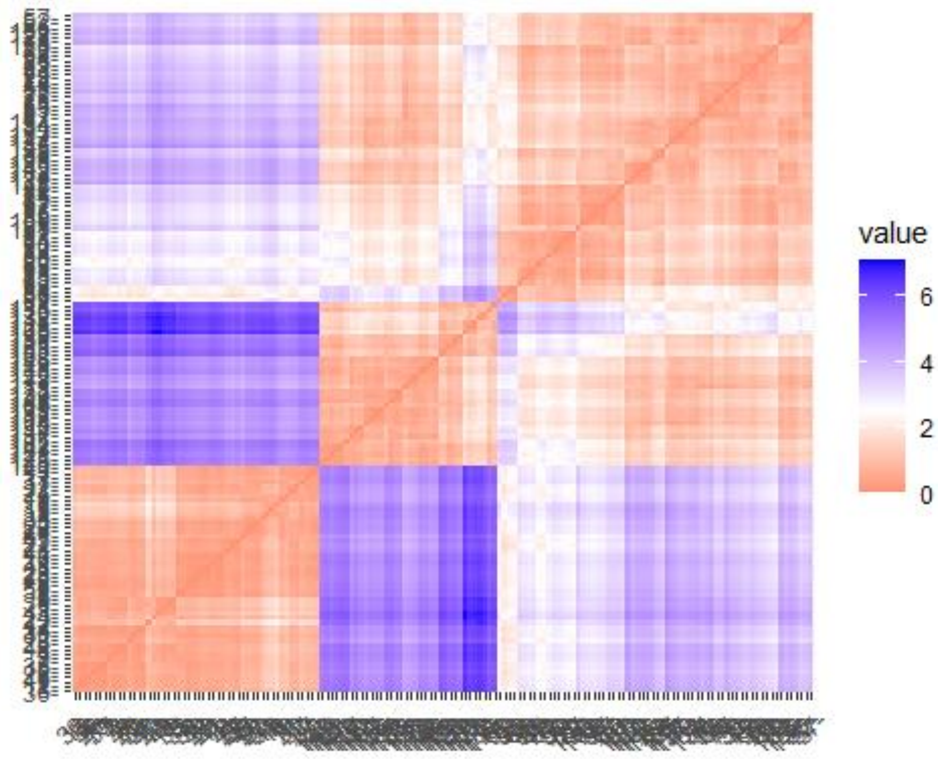
```
## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by
coercion
```



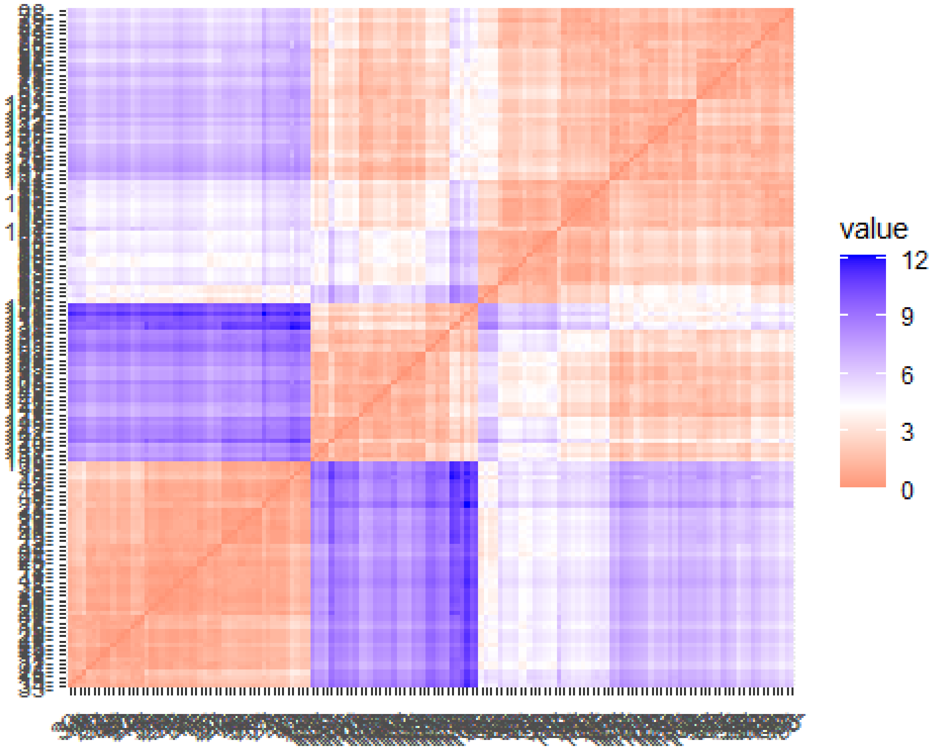
As analysed from the above graphs, the best cluster number is either 2 or 3.

## Interpretting the results

```
fviz_dist(dist(iris[, 1:4]))
```



```
fviz_dist(dist(iris[, 1:4], method = "manhattan"))
```



As seen when compared both the manhattan method and the euclidean method, the manhattan seems to have more tight and clear difference between the clusters than the euclidean method where the 3D visualization is more smooth and unclear. Furthermore, the 2nd and the third clusters seem to be somewhat close to each other with a low(small) distance between them.

In conclusion, I used Solid/Hard clustering approach to analyse the iris dataset. After analyzing the Dendrograms, I decided to use 3 clusters and the k-means method for clustering the data.