4 Hands On: Descriptive Modelling

Load the package `tidyverse`. Install and load the packages `cluster`, `dbscan` and `factoextra`.

4.1 Descriptive Analytics

1. Load the `iris` dataset to answer the following questions.

(a) Create a new tibble `iris1` by removing the `Species` attribute and select the first 5 examples. Use the function `dist()` to obtain the distance matrix. Which is the default distance function?

(b) Obtain the distance matrix for different distance functions, namely *manhattan*, *minkowski* with $p = 2$ and $p = 4$, *supremum*.

(c) Use the function `daisy()` from the package `cluster` to obtain the distance matrix for different distance functions, but now including the `Species` attribute (check the help of the function to see how it handles different types of attributes.).

4.2 Clustering

2. Load the `iris` data set to answer the following questions.

(a) Create a new tibble `iris2` by removing the `Species` attribute and scaling the remaining attributes through the functions `mutate_all()` and `scale()`. Why do you think this might be important?

(b) Apply the k-means algorithm with 3 clusters, through the function `kmeans()`. Inspect the returned object.

```
k3 <- kmeans(iris2,centers=3)
k3
```

(c) Plot the obtained clusters, using the function `fviz_cluster` from the package `factoextra`.

```
fviz_cluster(k3,iris2)
```

(d) Obtain the silhouette coefficient for each observation, using the function `silhouette()` from package `cluster`, and plot the silhouette information for the 3 clusters, using the function `fviz_silhouette()` from package `factoextra`.

```
si_coefs_k3 <- silhouette(k3$cluster,dist(iris2))
fviz_silhouette(si_coefs_k3) + coord_flip()
```

(e) Identify the optimal number of clusters by the average silhouette, using the function `fviz_nbclust()`

```
fviz_nbclust(iris2, kmeans, method = "silhouette")
```

(f) Identify again the optimal number of clusters but now by the "elbow method" using the total within sum of squares.

(g) Using 3 clusters, repeat the process above for the the algorithms PAM and CLARA.

(h) Use the function `dbscan()` from the package `dbscan` with `eps=0.9` to perform density-based clustering. Visualize the obtained clusters.

(i) Use the function `hclust()` to perform agglomarative hierarchical clustering with single, complete and average link. Visualize the obtained clusters, using the function `fviz_dend()`.

```
dm <- dist(iris2)
hclust.sing <- hclust(dm,"single")
fviz_dend(hclust.sing,k=3)

c <- cutree(hclust.sing,k=3)
si_coefs_hclust_sing_3 <- silhouette(c,dm)

# to complete ...
```