**Assignment – 5**

**-Sai Sree Mithra Sripathi**

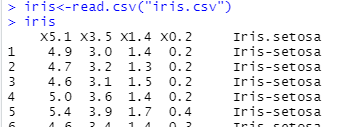
**Load “iris.csv” dataset and store it into a local R variable named “iris” using “read.csv” R command.**

**Display the content of the “iris” variable by typing the variable name in RStudio. The first line of the dataset has the feature names in the form of something like:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **X5.1** | **X3.5** | **X1.4** | **X0.2** | **Iris-setosa** |
| **1** | **4.9** | **3.0** | **1.4** | **0.2** | **Iris-setosa** |  |
| **2** | **...** | **...** | **...** | **...** | **...** |  |

*iris<-read.csv("iris.csv")*

*iris*



*str(iris)*

*names(iris)*

*names(iris)[names(iris)=="X5.1"]<-"sepal\_length"*

*names(iris)<-gsub("X3.5","sepal\_width", names(iris))*

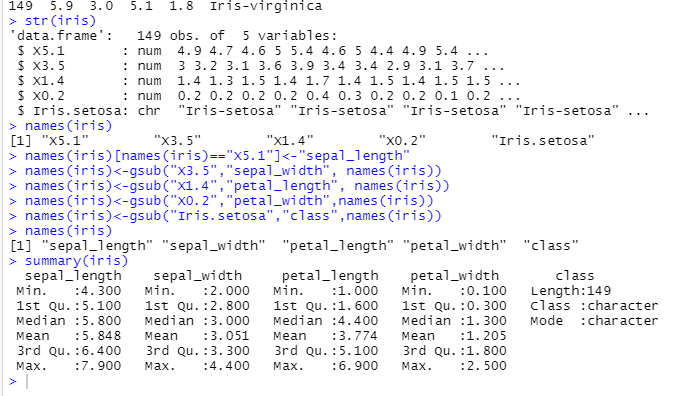
*names(iris)<-gsub("X1.4","petal\_length", names(iris))*

*names(iris)<-gsub("X0.2","petal\_width",names(iris))*

*names(iris)<-gsub("Iris.setosa","class",names(iris))*

*names(iris)*

*summary(iris)*



**According to the dataset description on the UCI Machine Learning Repository, these features correspond to sepal\_length, sepal\_width, petal\_length, petal\_width, and class. To clean this up, use the R command colnames to rename the column names according to the dataset description in the UCI Machine Learning Repository.**

**After doing so, if you display the content of the “iris” variable you should see something like:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **sepal\_length** | **sepal\_width** | **petal\_length** | **petal\_width** | **class** |
| **1** | **4.9** | **3.0** | **1.4** | **0.2** | **Iris-setosa** |
| **2** | **...** | **...** | **...** | **...** | **...** |

**Now that your data is ready, select the features you will use for clustering using the k-means algorithm. In one run, use the sepal\_length and sepal\_width features. You can do so by storing the two features in a local variable called   
  
  
  
kmeans\_sepal: kmeans\_sepal <- data.frame(iris$sepal\_length, iris$sepal\_width)**

**In another run, use petal\_length and petal\_width features and save them in a local variable called kmeans\_petal.**

**Start training a clustering model of your data using the k-means algorithm.**

**For the kmeans\_sepal variable, run the k-means algorithm for 3 clusters.**

K-MEANS Clustering:

K-means clustering is one of the most commonly used unsupervised machine learning algorithm for partitioning a given data set into a set of k groups (i.e. k clusters), where k represents the number of groups pre-specified by the analyst.

Kmeans command : Perform k-means clustering on a data matrix.

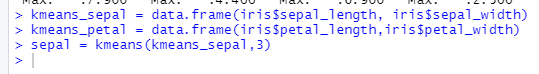
Kmeans() function returns a list of components, including:

* cluster: A vector of integers (from 1:k) indicating the cluster to which each point is allocated
* centers: A matrix of cluster centers (cluster means)
* totss: The total sum of squares (TSS), i.e ∑(xi−x¯)2∑(xi−x¯)2. TSS measures the total variance in the data.
* withinss: Vector of within-cluster sum of squares, one component per cluster
* tot.withinss: Total within-cluster sum of squares, i.e. sum(withinss)sum(withinss)
* betweenss: The between-cluster sum of squares, i.e. totss−tot.withinsstotss−tot.withinss
* size: The number of observations in each cluster

*kmeans\_sepal = data.frame(iris$sepal\_length, iris$sepal\_width)*

*kmeans\_petal = data.frame(iris$petal\_length,iris$petal\_width)*

*sepal = kmeans(kmeans\_sepal,3)*



**Plot the outcome of your clustering model using clusplot R command, e.g., clusplot(iris, sepal$cluster, color=TRUE, shade=TRUE, labels=5, lines=0).**

**For the kmeans\_petal variables, run the k-means algorithm for 4 clusters and plot the outcome.**

**Note that to use the clusplot R command, you need to load cluster library.**

*library("cluster")*

*clusplot(iris, sepal$cluster, color = TRUE, shade = TRUE, labels = 5, lines = 0)*

Diagram

Description automatically generated

*petal = kmeans(kmeans\_petal,4)*

*clusplot(iris, petal$cluster, color = TRUE, shade = TRUE, labels = 5, lines = 0)*

Chart, diagram

Description automatically generated

When we use 4 in sepal length –

*sepal = kmeans(kmeans\_sepal,4)*

*clusplot(iris, sepal$cluster, color = TRUE, shade = TRUE, labels = 5, lines = 0)*

Then the plot looks like:

Chart, diagram

Description automatically generated

The count defines the number of clusters to be categorized into. First when a count of 3 was given the data is divided as 3 clusters. When we give a count of 4, the entire data is categorized as 4 clusters.

**References:**

https://www.datanovia.com/en/lessons/k-means-clustering-in-r-algorith-and-practical-examples/