**Practical no 8**

1. **Means Clustering**

** Apply the K-Means algorithm to group similar data points into clusters.**

# Loading data

data(iris)

# Structure

str(iris)

# Installing Packages

install.packages("ClusterR")

install.packages("cluster")

# Loading package

library(ClusterR)

library(cluster)

iris\_1 <- iris[, -5]

# Fitting K-Means clustering Model

# to training dataset

set.seed(240) # Setting seed

kmeans.re <- kmeans(iris\_1, centers = 3, nstart = 20)

kmeans.re

# each observation

kmeans.re$cluster

# Confusion Matrix

cm <- table(iris$Species, kmeans.re$cluster)

cm

plot(iris\_1[c("Sepal.Length", "Sepal.Width")])

plot(iris\_1[c("Sepal.Length", "Sepal.Width")],

col = kmeans.re$cluster)

plot(iris\_1[c("Sepal.Length", "Sepal.Width")],

col = kmeans.re$cluster,

main = "K-means with 3 clusters")

## Plotiing cluster centers

kmeans.re$centers

kmeans.re$centers[, c("Sepal.Length", "Sepal.Width")]

# cex is font size, pch is symbol

points(kmeans.re$centers[, c("Sepal.Length", "Sepal.Width")],

col = 1:3, pch = 8, cex = 3)

## Visualizing clusters

y\_kmeans <- kmeans.re$cluster

clusplot(iris\_1[, c("Sepal.Length", "Sepal.Width")],

y\_kmeans,

lines = 0,

shade = TRUE,

color = TRUE,

labels = 2,

plotchar = FALSE,

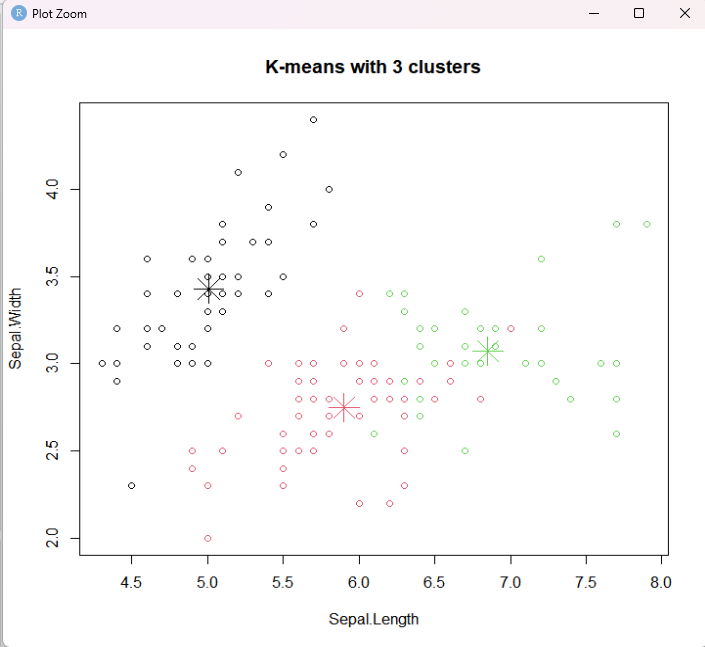
span = TRUE,

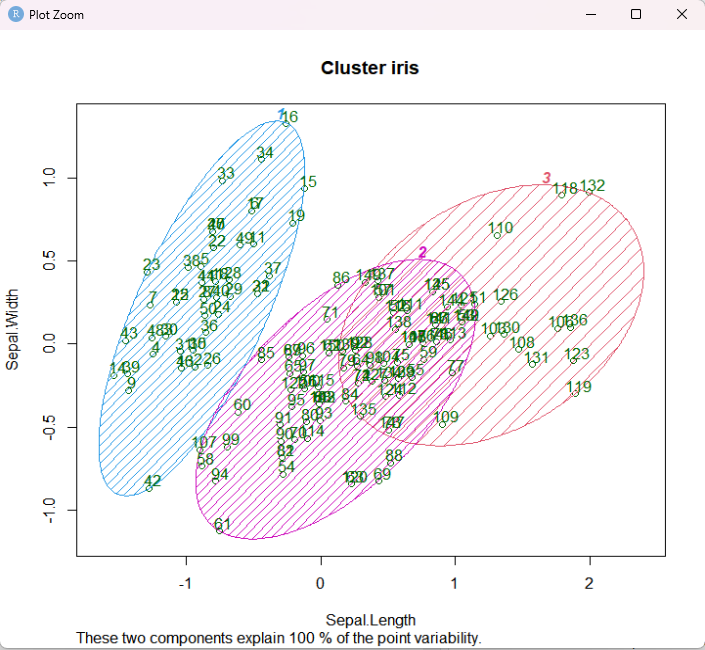
main = paste("Cluster iris"),

xlab = 'Sepal.Length',

ylab = 'Sepal.Width')

Output





** Determine the optimal number of clusters using elbow method or silhouette analysis**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import matplotlib.cm as cm

from sklearn.datasets import load\_iris

from sklearn.cluster import KMeans

#Load the Dataset

X, y = load\_iris(return\_X\_y=True)

#Find optimum number of cluster using Elbow method

sse = [] #SUM OF SQUARED ERROR

for k in range(1,11):

km = KMeans(n\_clusters=k, random\_state=2)

km.fit(X)

sse.append(km.inertia\_)

#Plot the Elbow graph to find the optimum number of cluster

sns.set\_style("whitegrid")

g=sns.lineplot(x=range(1,11), y=sse)

g.set(xlabel ="Number of cluster (k)",

ylabel = "Sum Squared Error",

title ='Elbow Method')

plt.show()

#Build the Kmeans clustering model

kmeans = KMeans(n\_clusters = 3, random\_state = 2)

kmeans.fit(X)

#Find the cluster center

print(kmeans.cluster\_centers\_)

#Predict the cluster group:

pred = kmeans.fit\_predict(X)

print(pred)

#Plot the cluster center with data points

plt.figure(figsize=(12,5))

plt.subplot(1,2,1)

plt.scatter(X[:,0],X[:,1],c = pred, cmap=cm.Accent)

plt.grid(True)

for center in kmeans.cluster\_centers\_:

center = center[:2]

plt.scatter(center[0],center[1],marker = '^',c = 'red')

plt.xlabel("petal length (cm)")

plt.ylabel("petal width (cm)")

plt.subplot(1,2,2)

plt.scatter(X[:,2],X[:,3],c = pred, cmap=cm.Accent)

plt.grid(True)

for center in kmeans.cluster\_centers\_:

center = center[2:4]

plt.scatter(center[0],center[1],marker = '^',c = 'red')

plt.xlabel("sepal length (cm)")

plt.ylabel("sepal width (cm)")

plt.show()

print("End....")

**Output**

