#Assignment22\_Session22

#Problem

#1. Use the below given data set

# DataSet

#2. Perform the below given activities:

#a. apply K-means clustering to identify similar recipies

#b. apply K-means clustering to identify similar attributes

#c. how many unique recipies that people order often

#d. what are their typical profiles

#Answers

#reading the dataset

#using wine dataset

library(pvclust)

library(tidyverse) # data manipulation

library(cluster) # clustering algorithms

library(factoextra) # clustering algorithms & visualization

library(ggplot2)

#reading the dataset

data <- read.csv("D:/acadgild/mail things/24 feb/wine.csv")

View(data)

#structure of it

str(data)

# Normalization/scaling

df <- scale(data[,-c(1,2)])

#finding optimal clusters for our data by looking into plot

wssplot <- function(data, nc=15, seed=1234){

wss <- (nrow(data)-1)\*sum(apply(data,2,var))

for (i in 2:nc){

set.seed(seed)

wss[i] <- sum(kmeans(data, centers=i)$withinss)}

plot(1:nc, wss, type="b", xlab="Number of Clusters",

ylab="Within groups sum of squares")}

#The data parameter is the numeric dataset to be analyzed,

#nc is the maximum number of clusters to consider,

#and seed is a random number seed.

#of our main dataset df

wssplot(df)

library(NbClust)

set.seed(1234)

nc <- NbClust(df, min.nc=2, max.nc=15, method="kmeans")

nc

table(nc$Best.n[1,])

barplot(table(nc$Best.n[1,]),

xlab="Numer of Clusters", ylab="Number of Criteria",

main="Number of Clusters Chosen by 26 Criteria")

#The plot above represents the variance within the clusters.

#It decreases as k increases, but it can be seen a bend (or “elbow”) at k = 3.

#This bend indicates that additional clusters beyond the third have little value.

#In figure 2, 14 of 24 criteria provided by the NbClust package suggest a 3-cluster solution.

#Note that not all 30 criteria can be calculated for every dataset.

#k means

set.seed(1234)

fit.km <- kmeans(df, 3, nstart=25)

print(fit.km)

#As the final result of k-means clustering result is sensitive to the random starting assignments, we specify nstart = 25. This means that R will try 25 different random starting assignments and then select

#the best results corresponding to the one with the lowest within cluster variation

#The printed output displays:

#the cluster means or centers: a matrix, which rows are cluster number (1 to 4) and columns are variables

#the clustering vector: A vector of integers (from 1:k) indicating the cluster to which each point is allocated

fit.km$size

fit.km$centers

#we can extract the clusters and add to our initial data to do

#some descriptive statistics at the cluster level:

#to compute the mean of each variables by clusters using the original data:

aggregate(data[,-c(1,2)], by=list(cluster=fit.km$cluster), mean)

#same thing but new way of doing it previous aggregate things

data %>%

mutate(Cluster = fit.km$cluster) %>%

group\_by(Cluster) %>%

summarise\_all("mean")

#A final cluster solution is obtained with kmeans() function and the cluster centroids are printed (#3).

#Since the centroids provided by the function are based on standardized data,

#the aggregate() function is used along with the cluster memberships

#to determine variable means for each cluster in the original metric.

# to add the point classifications to the original data, use this:

dd <- cbind(data[,-c(1,2)], cluster = fit.km$cluster)

head(dd)

#visualize

fviz\_cluster(fit.km, data = df)

# Plotting Different cluster formed graphically

clusplot(data[,-c(1,2)],fit.km$cluster,color = TRUE)

#A cross-tabulation of Type (wine varietal) and cluster membership is given by

ct.km <- table(data$Type, fit.km$cluster)

ct.km

#We can execute the same process for 3, 4, and 5 clusters, and the results are shown in the figure:

k2 <- kmeans(df, centers = 2, nstart = 25)

k3 <- kmeans(df, centers = 3, nstart = 25)

k4 <- kmeans(df, centers = 4, nstart = 25)

k5 <- kmeans(df, centers = 5, nstart = 25)

# plots to compare

p1 <- fviz\_cluster(k2, geom = "point", data = df) + ggtitle("k = 2")

p2 <- fviz\_cluster(k3, geom = "point", data = df) + ggtitle("k = 3")

p3 <- fviz\_cluster(k4, geom = "point", data = df) + ggtitle("k = 4")

p4 <- fviz\_cluster(k5, geom = "point", data = df) + ggtitle("k = 5")

library(gridExtra)

grid.arrange(p1, p2, p3, p4, nrow = 2)