Name: Shobhit Agrawal
Reg. No.: 20BDS0162
Slot: L31+L32

Subject: CSE3050 – Data Visualization and Presentation
Date: 13th February, 2023
Faculty: Dr. Prakash M.

Exercise Number – 1

K-Means Clustering

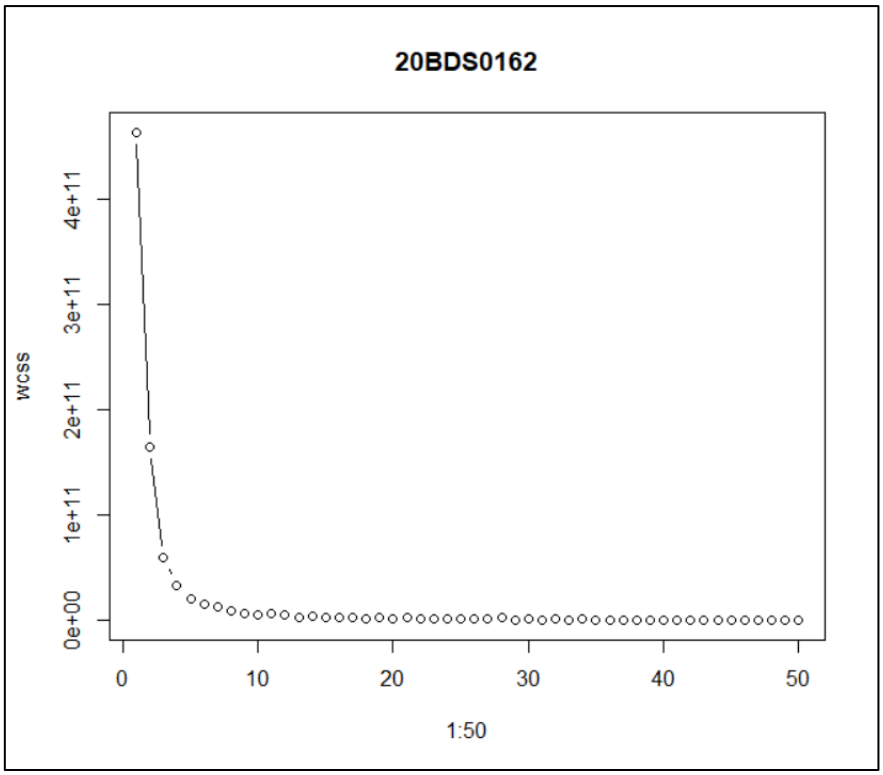
Q1. K-mean Clustering using Social_Network_Ads dataset.

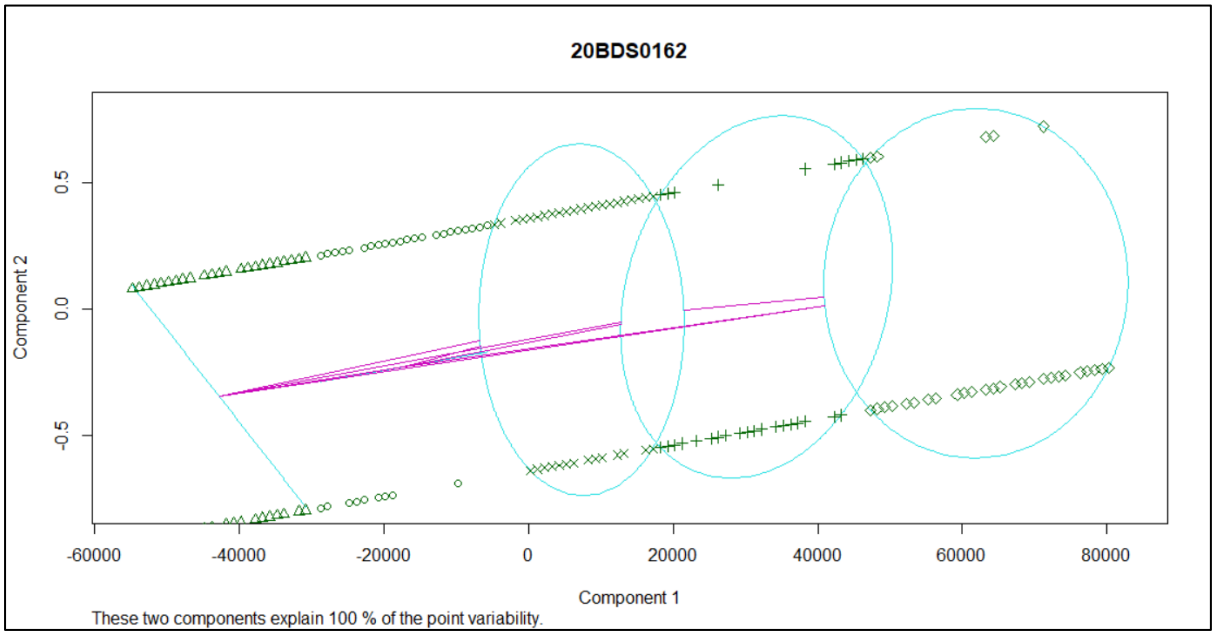
Aim: To use the Social_Network_Ads dataset with the K-Means Clustering Algorithm and visualise the clusters

Code:

```
library(arules)

library(dplyr)


#k means clustering

df2 = read.csv("D:\\Sem6\\DVP\\ELA\\Assessment2\\Social_Network_Ads.csv")

df2 = df2[4:5]

df2


library(cluster)


set.seed(5000)

wcss = vector()

wcss


for(i in 1:50)

  wcss[i] = sum(kmeans(df2, i)$withinss)


plot(1:50, wcss, type = 'b') #5 cluster reqd


kmeans = kmeans(x = df2, centers = 5)


y_kmeans = kmeans$cluster

z=clusplot(df2, y_kmeans,main = "20BDS0162")

z
```

Output:





Result:

K-Means Clustering for the dataset Social Network Ads was completed successfully. Based on the Estimated Salary column, I had first put the cluster centroid at 5000. (4th). I used the algorithm to produce five clusters. Following that, all of the clusters are shown, and it is also indicated which data points belong to particular clusters and how far away they are from each other.

Exercise Number - 2

Market Basket Analysis

Q1. Market-Basket Data analysis Visualization

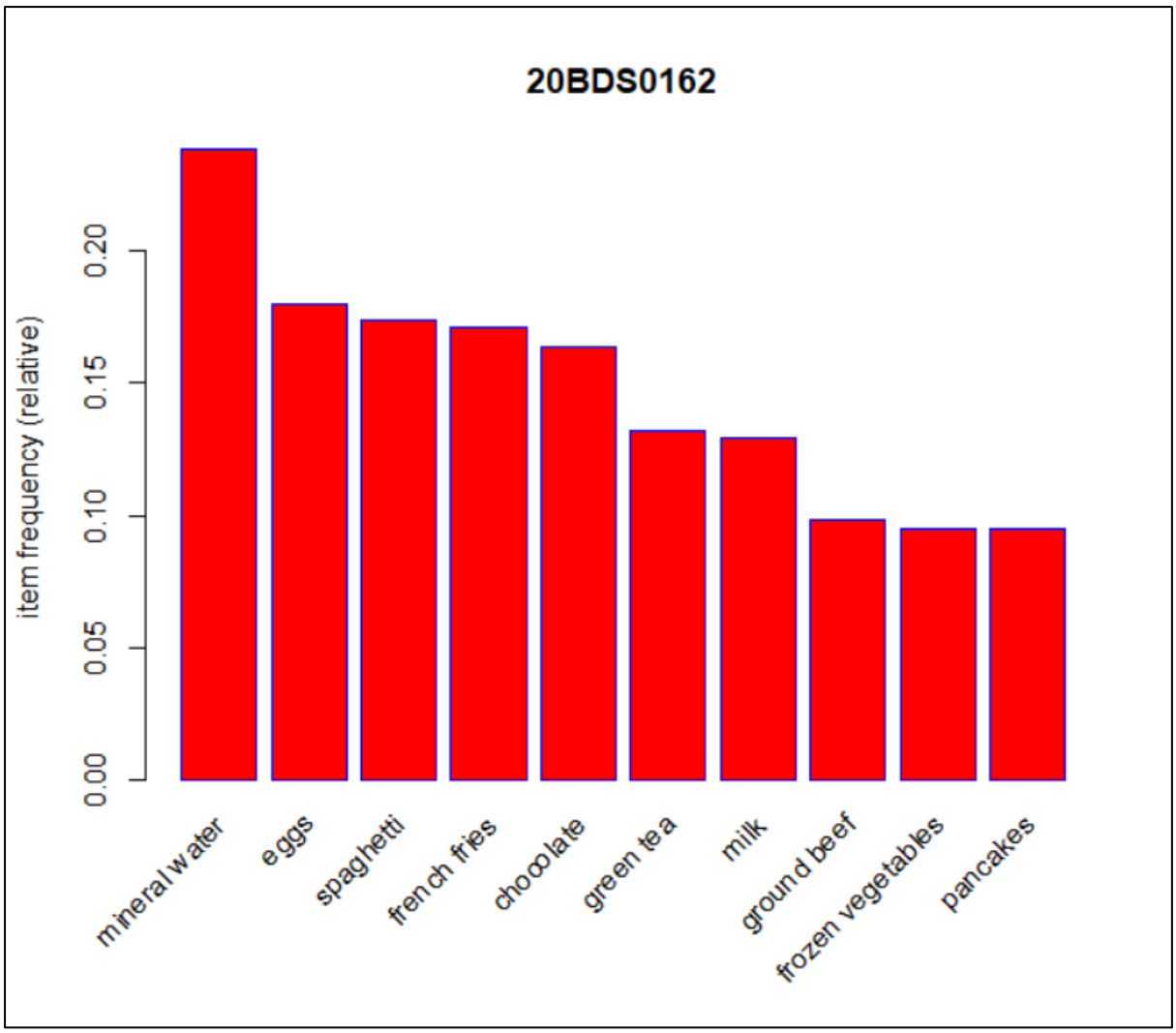
Aim: To perform Market-Basket-Data-Analysis Visualization using Apriori Algorithm and visualize the results

Code:

```
install.packages("dplyr")

install.packages("arules")


library(arules)

library(dplyr)


df1 = read.csv("D:\\Sem6\\DVP\\ELA\\Assessment2\\Market_Basket_Optimisation.csv", header = FALSE)


summary(df1)

dim(df1)

str(df1)


#sparse matrix

df1 = read.transactions(file = "D:\\Sem6\\DVP\\ELA\\Assessment2\\Market_Basket_Optimisation.csv",

                        sep = ",",

                        rm.duplicates = T)


summary(df1)


itemFrequencyPlot(x = df1, topN = 10,main = "20BDS0162",col = "red", border = "blue")


#apriori algo

rules = apriori(data = df1,

                parameter = list(support = 0.004,
```

```
                    confidence = 0.2))
#visualizing
inspect(sort(rules, by = 'lift')[1:10])
```

Output:



Result:

Calculated the necessary support and confidence that properly suit the provided dataset in order to successfully use the Apriori algorithm and visualize the results. According to the findings, one can predict what a person would purchase in total while visiting the market based on the things they initially choose.

Exercise Number - 3

Dashboard in R

Q1. Create a Simple dashboard using Shiny.

Aim: To create a simple dashboard in R using the Shiny library

Code:

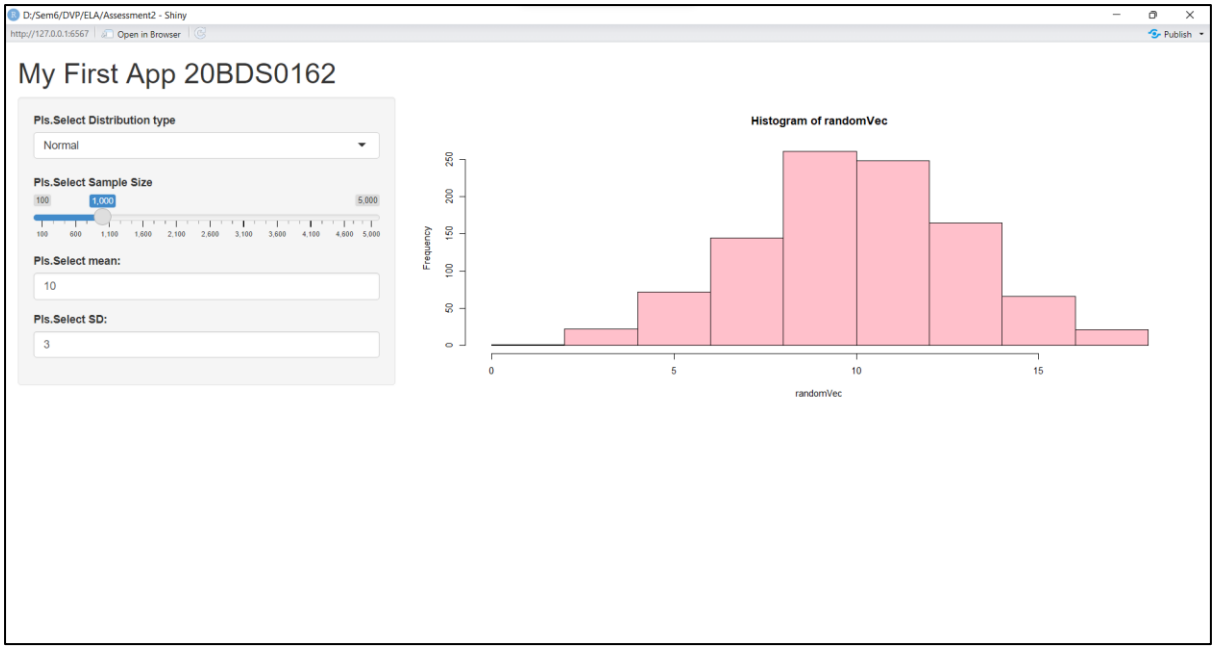
ui.R

```
install.packages("shiny")
install.packages("shinydashboard")

library(shiny)
library(shinydashboard)

shinyServer(
  pageWithSidebar(
    headerPanel("My First App 20BDS0162"),
    sidebarPanel(
      selectInput("Distribution",'Pls.Select Distribution type',
                  choices =c('Normal','Exponential')),
      sliderInput("sampleSize",
                  'Pls.Select Sample Size',
                  min =100,max=5000,
                  value=1000,step=100),
      conditionalPanel(condition="input.Distribution=='Normal'",
                       textInput("mean","Pls.Select mean:",10),
                       textInput("sd","Pls.Select SD:",3)),
      conditionalPanel(condition="input.Distribution=='Exponential'",
                       textInput("lambda","Pls.Select Exp lamda:",1))
    ),
    mainPanel(plotOutput('myPlot'))
  )
)
```

server.R

```
shinyServer(
  function(input,output,session){
    output$myPlot<-renderPlot({
      distType = input$Distribution
      size <-input$sampleSize
      if(distType=="Normal"){
        randomVec <- rnorm(size,mean=as.numeric(input$mean),
                           sd=as.numeric(input$sd))
      }else{
        randomVec <- rexp(size,rate=1/as.numeric(input$lambda))
      }
      hist(randomVec,col ="pink")
    })
  }
)
```

Output:



Result:

Successfully created a straightforward dashboard in R using the Shiny package. The dashboard shows a randomVec histogram, and the user may select the normal or exponential distribution type, as well as the sample size, mean, and standard deviation for each distribution.