Grocery Analysis Project

TEAM 102

23011224	أشرف خميس أحمد عيسى
23010120	أحمد محمد أحمد فؤاد توفيق محمد
23012039	أحمد عماد عبد التواب أبو زامل
23011198	أحمد أيمن خيري محمد
23010098	أحمد محمد عبد الوهاب ابراهيم

1- What will the program do?

- The program will take some data purchased from a Grocery store that needs to be cleaned and display the information that the system will process from the data into visual graphs which makes it easy for the Grocery managers to understand what the customers like to do whenever he is in their store and what mostly like he will do the next time he visits their store.

2- What the input to the program will be?

The user will enter:

1- The data file that he wants to analyze.
2- The number of clusters.
3- The minimum confidence.
4- The minimum support.

3- What is the output from the program will be?

The user will see:

1- Pie chart that compares between Credit and Cash.
2- Histogram that compares Age and Total Spending.
3- Bar plot that compares Cities and Total Spending.
4- Box plot that describes the Distribution of Total Spending.
5- Kmeans.
6- Apriori rules.

Full Description of Dataset:

The dataset consists of 8 different columns every column describes important data about each purchase:

1- **items ~~>** The items which the customer purchased.

2- **count ~~>** The number of items that are purchased.

3- **total ~~>** the total of money spend by the customer in that purchase.

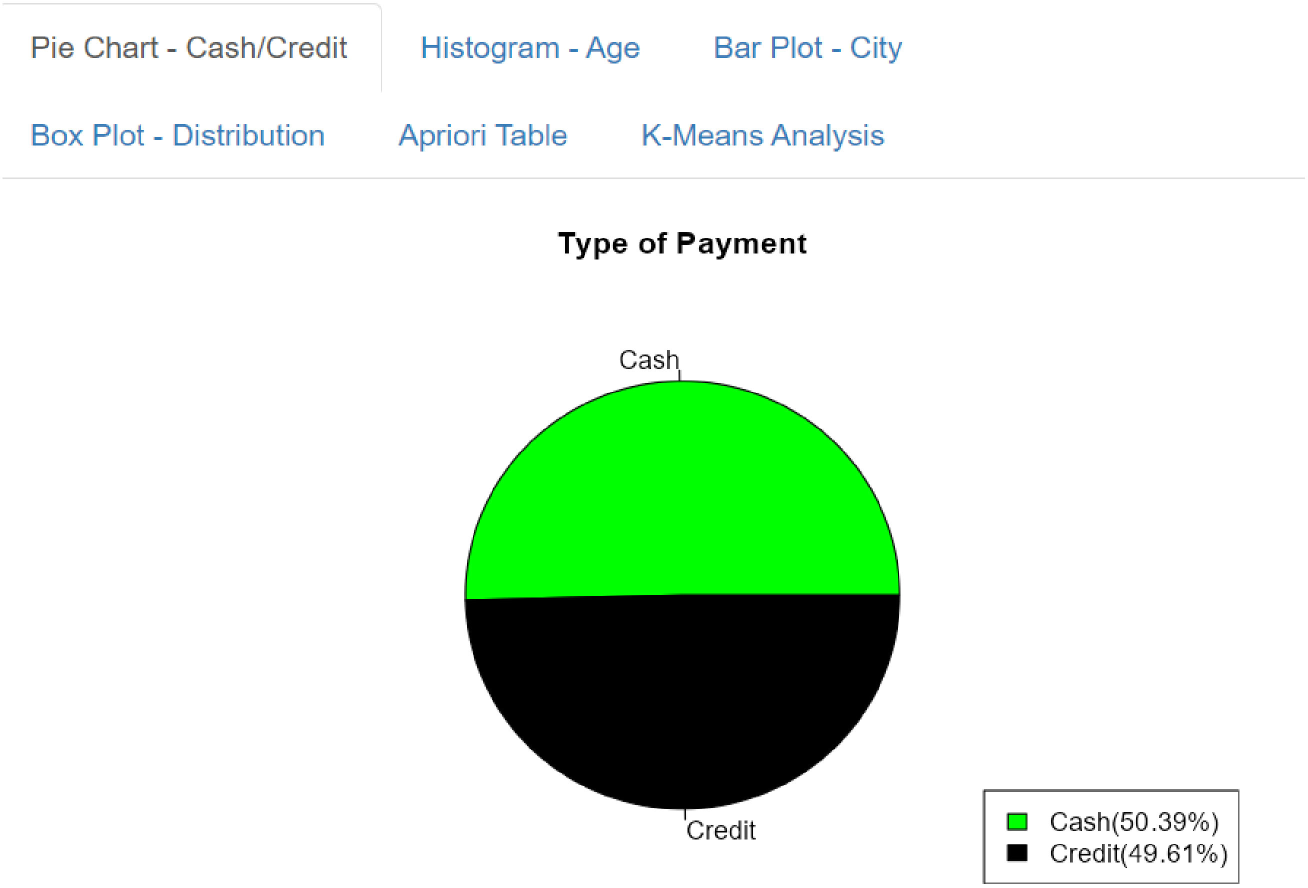
4- **rnd ~~>** the customer id.

5- **customer ~~>** the customer name.

6- **age ~~>** the customer age.

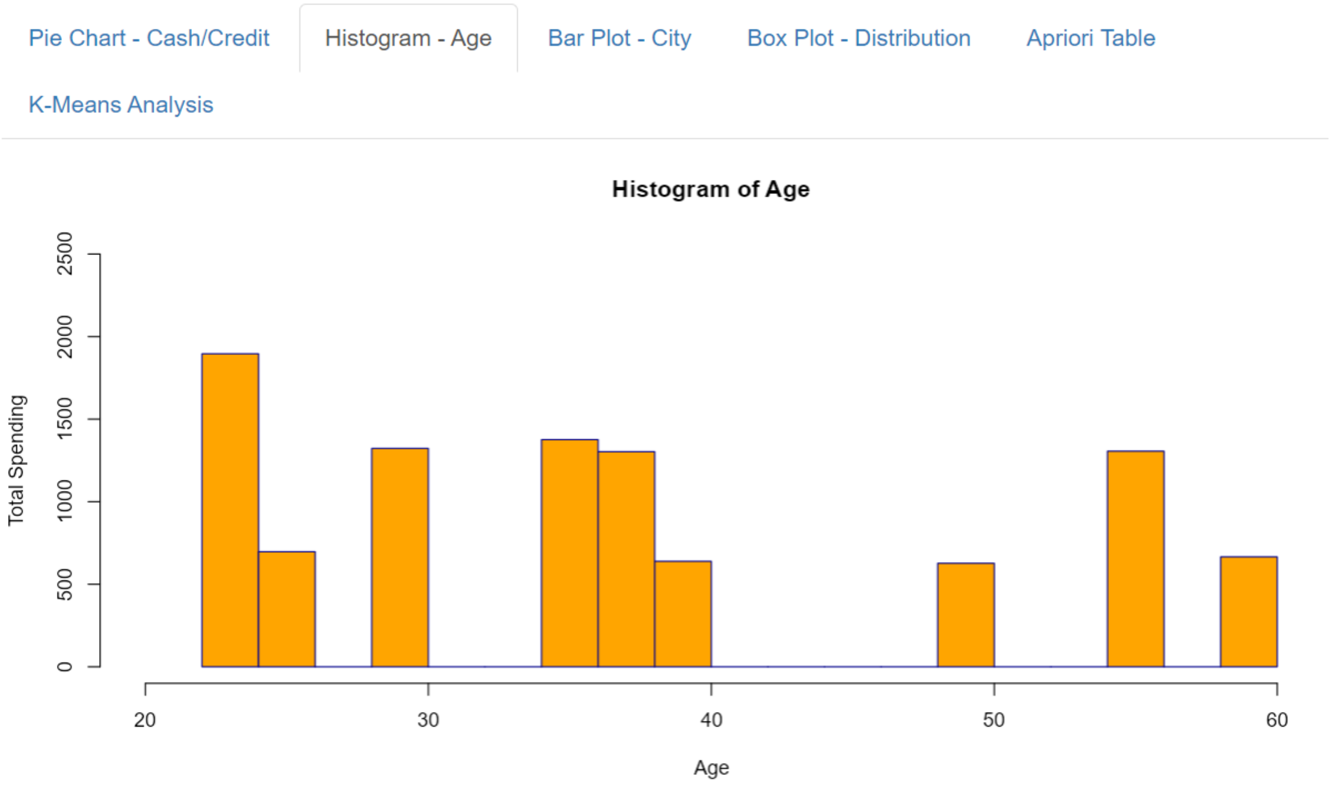
7- **city ~~>** the city where the customer purchased the items.

8- **paymentType ~~>** the Payment Type whether Cash or Credit



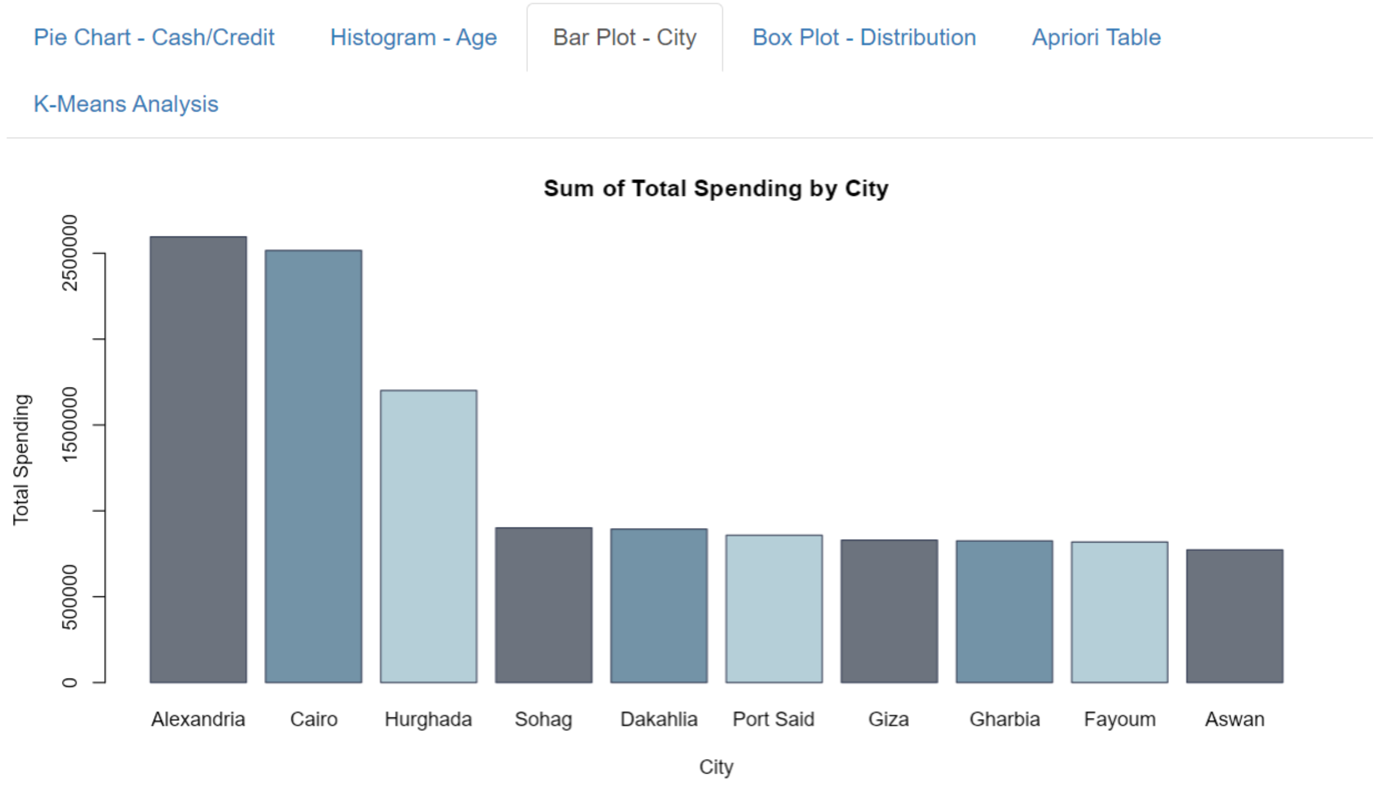
In the following graph, we made a comparison between the purchases which are made by the two different payment types (cash–credit). There is a legend on the side that presents the slight difference between them both where the cash is more than the credit about 1 percent of the users.

We used the Pie plot for this comparison to illustrate the small difference between these 2.



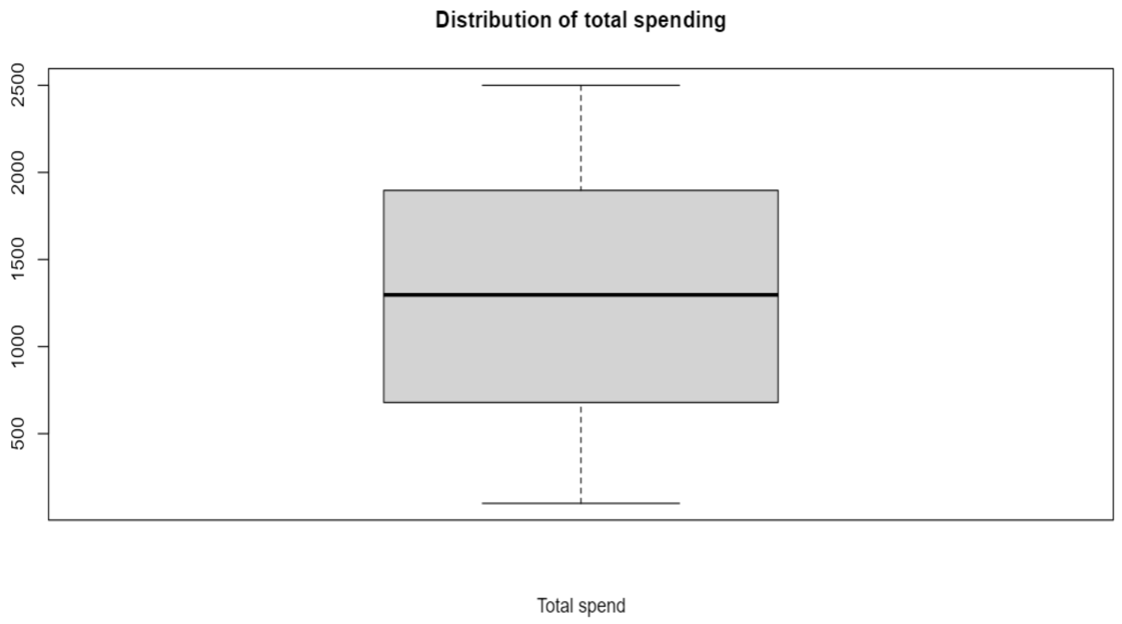
The following Histogram compares the age difference and their total spending, showing that people in their early 20s spend the most people spending money and people in the age of 39 spend the least in the grocery.

We chose the Histogram because the age is ongoing time and the only comparison is by Total spending where we use only one axis in the comparison.



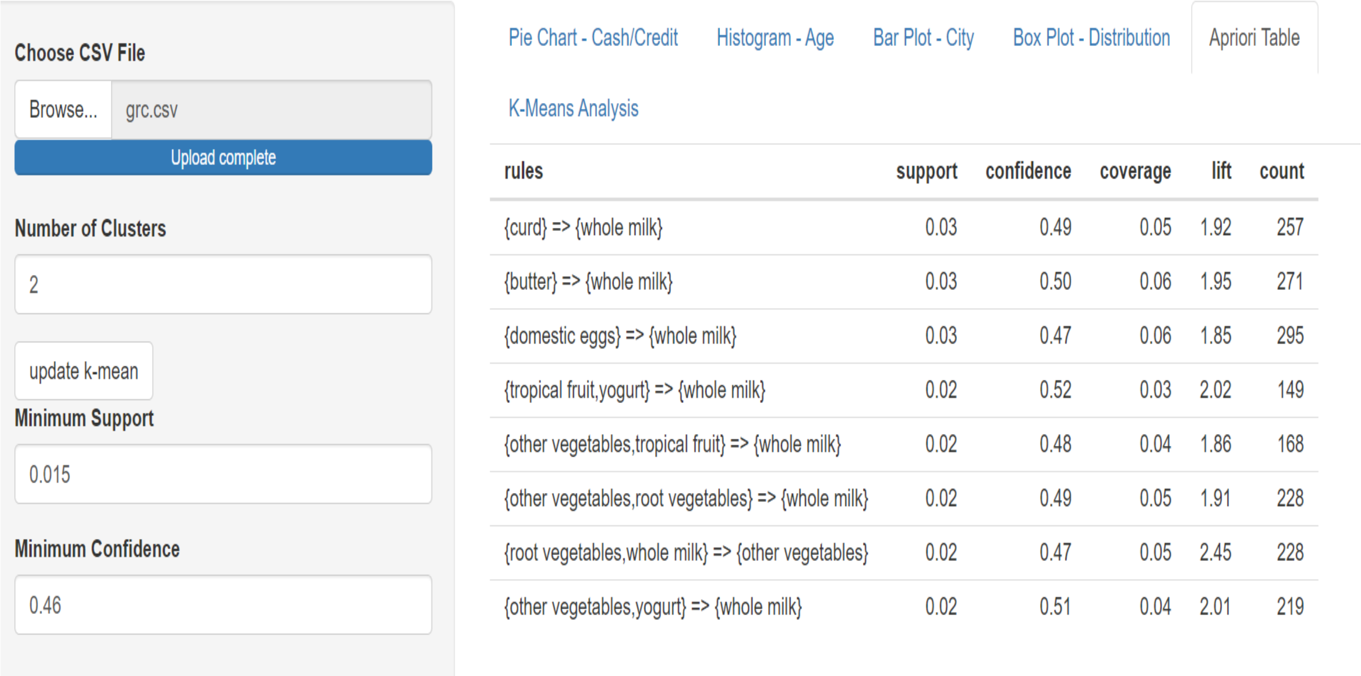
The following Bar plot compares the city's Total Spending with each other. As we can see Alexandria has the highest in money total spending. While Aswan is the least in money total spending.

We used the Bar plot because there are many variables that are used on the X-axis (Cities) While on the Y-axis there is total spending. Also, we used the Bar plot to see which of them is bigger than the other and we can arrange it descendingly.



The following graph represents the Distribution of the total spending where you can see that every purchase procedure is nearly to each other. As it is concentrated in about 1293$.

We used the Box plot because it helps you know where the main of the total spending is. and where the lowest and the highest and also helps you know if there is outlier data.



Choose CSV File: grc.csv

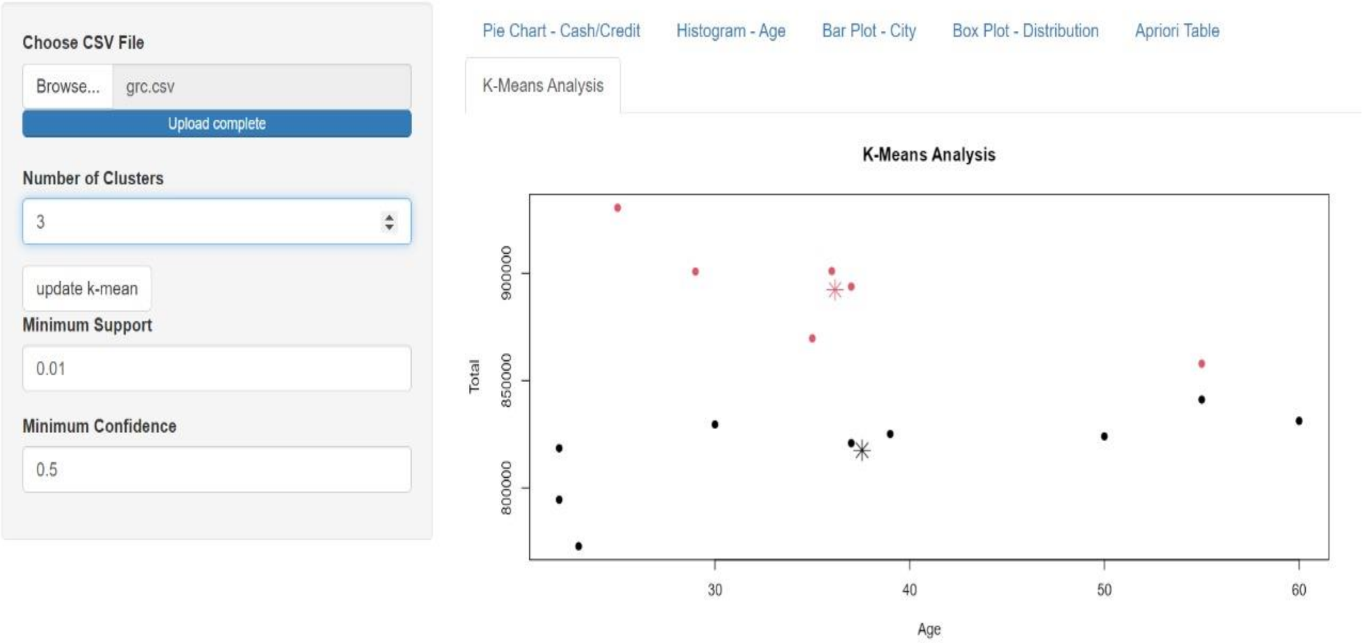
Number of Clusters: 2

Minimum Support: 0.015

Minimum Confidence: 0.46

rules	support	confidence	coverage	lift	count
{curd} => {whole milk}	0.03	0.49	0.05	1.92	257
{butter} => {whole milk}	0.03	0.50	0.06	1.95	271
{domestic eggs} => {whole milk}	0.03	0.47	0.06	1.85	295
{tropical fruit,yogurt} => {whole milk}	0.02	0.52	0.03	2.02	149
{other vegetables,tropical fruit} => {whole milk}	0.02	0.48	0.04	1.86	168
{other vegetables,root vegetables} => {whole milk}	0.02	0.49	0.05	1.91	228
{root vegetables,whole milk} => {other vegetables}	0.02	0.47	0.05	2.45	228
{other vegetables,yogurt} => {whole milk}	0.02	0.51	0.04	2.01	219

In the following table, you can see all the association rules that help us identify patterns and relationships between items in the data set. The Apriori algorithm gives us the results we see in the table with the minimum support and minimum confidence.



Choose CSV File

grc.csv

Upload complete

Number of Clusters

3

Minimum Support

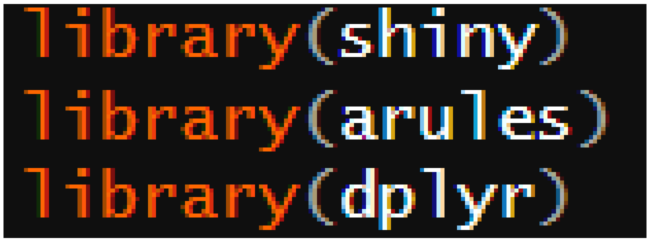
0.01

Minimum Confidence

0.5

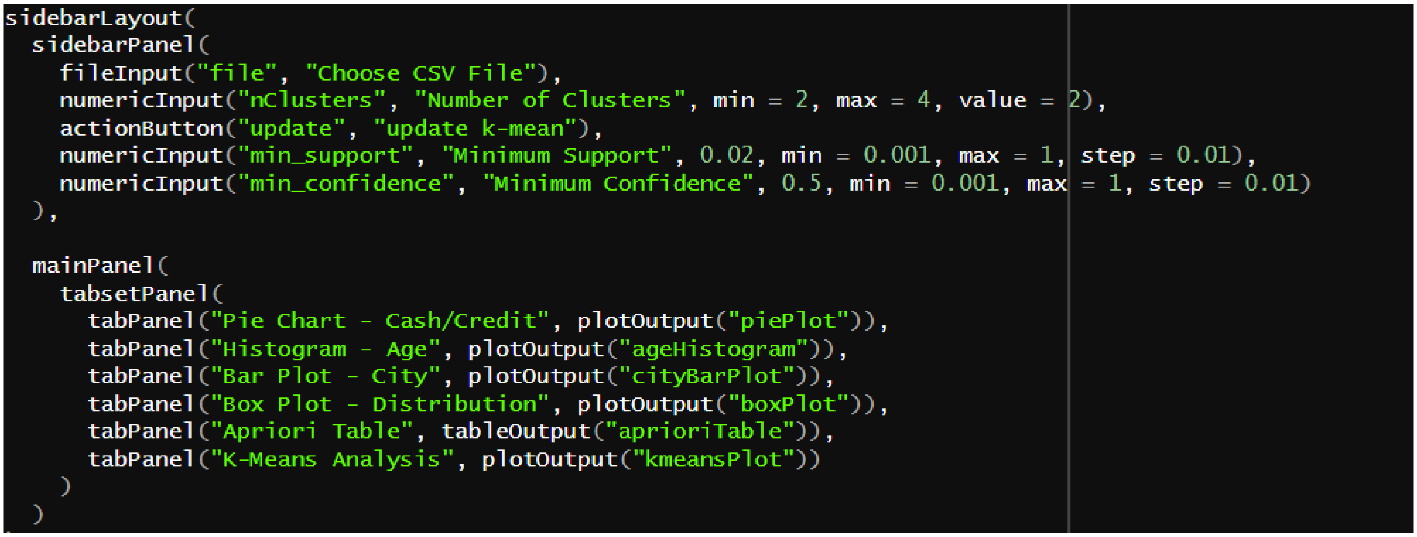
The following graph represents each of the 15 customers which we grouped by the id total spending and age then we made 3 clusters **(the number of clusters that the user wrote in the dashboard)** and classified each one into only one of the clusters.

Libraries Used:



```
library(shiny)
library(arules)
library(dplyr)
```

DASHBOARD/UI: (أحمد محمد أحمد فؤاد توفيق)



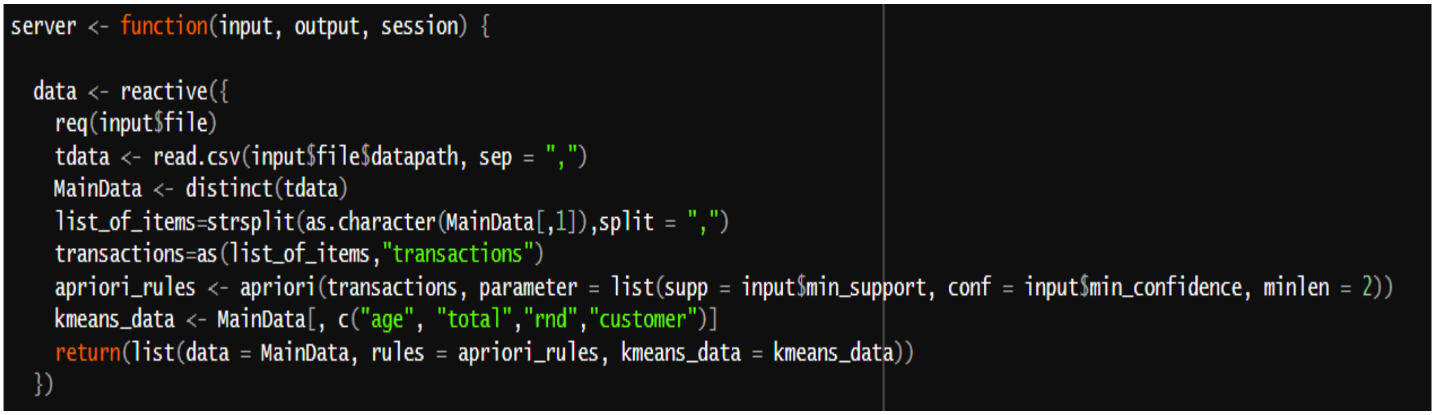
```
sidebarLayout(
  sidebarPanel(
    fileInput("file", "Choose CSV File"),
    numericInput("nClusters", "Number of Clusters", min = 2, max = 4, value = 2),
    actionButton("update", "update k-mean"),
    numericInput("min_support", "Minimum Support", 0.02, min = 0.001, max = 1, step = 0.01),
    numericInput("min_confidence", "Minimum Confidence", 0.5, min = 0.001, max = 1, step = 0.01)
  ),

  mainPanel(
    tabsetPanel(
      tabPanel("Pie Chart - Cash/Credit", plotOutput("piePlot")),
      tabPanel("Histogram - Age", plotOutput("ageHistogram")),
      tabPanel("Bar Plot - City", plotOutput("cityBarPlot")),
      tabPanel("Box Plot - Distribution", plotOutput("boxPlot")),
      tabPanel("Apriori Table", tableOutput("aprioriTable")),
      tabPanel("K-Means Analysis", plotOutput("kmeansPlot"))
    )
  )
```

The following code is the user interface code where we made the sidebar Panel into a menu to put all the input the program would use :

1. file input ~> to take the data file.
2. number of clusters ~> the number of clusters which can be updated from range 2 to 4.
3. Action Button ~> to update the k-means visualization whenever the number of clusters is changed.
4. Min Supp ~> to take the min sup from the range 0.001 to 1 with an initial value of 0.02.
5. Min Conf ~> to take the min conf from the range 0.001 to 1 with an initial value of 0.05.

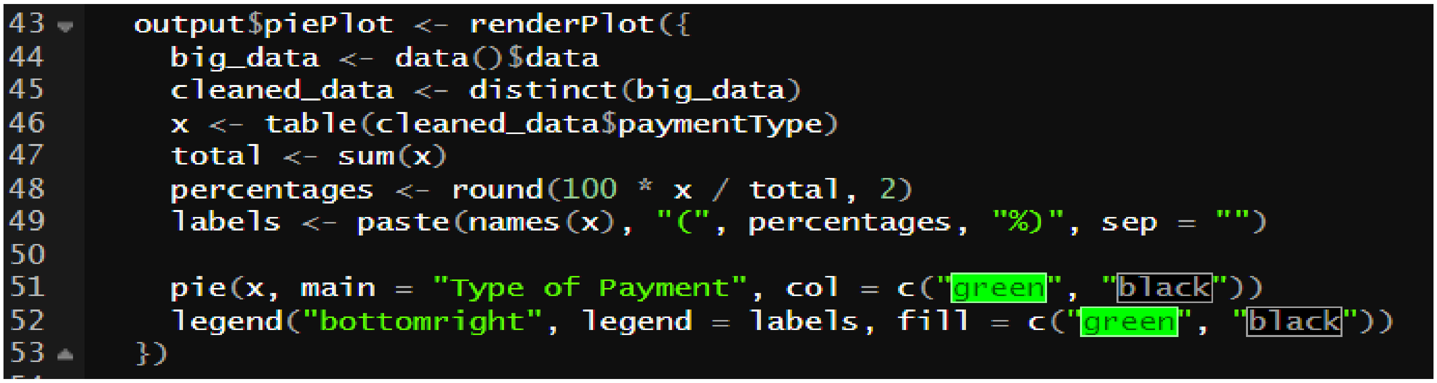
Then we used the main panel to have several tabs where each tab has a different graph of different variables to compare and as a user you can choose easily by clicking on the tab name.

Data input/flow throw the program :(أحمد عماد عبد التواب)



```
server <- function(input, output, session) {

  data <- reactive({
    req(input$file)
    tdata <- read.csv(input$file$datapath, sep = ",")
    MainData <- distinct(tdata)
    list_of_items=strsplit(as.character(MainData[,1]),split = ",")
    transactions=as(list_of_items,"transactions")
    apriori_rules <- apriori(transactions, parameter = list(supp = input$min_support, conf = input$min_confidence, minlen = 2))
    kmeans_data <- MainData[, c("age", "total","rnd","customer")]
    return(list(data = MainData, rules = apriori_rules, kmeans_data = kmeans_data))
  })
```

The data is taken when you choose the file so we get the file path and read it as CSV to make it a data frame that the program can analyze for visual outputs, then we take the first data column which is the items, and change it to transactions so we can use it to make apriori rules on the data items.

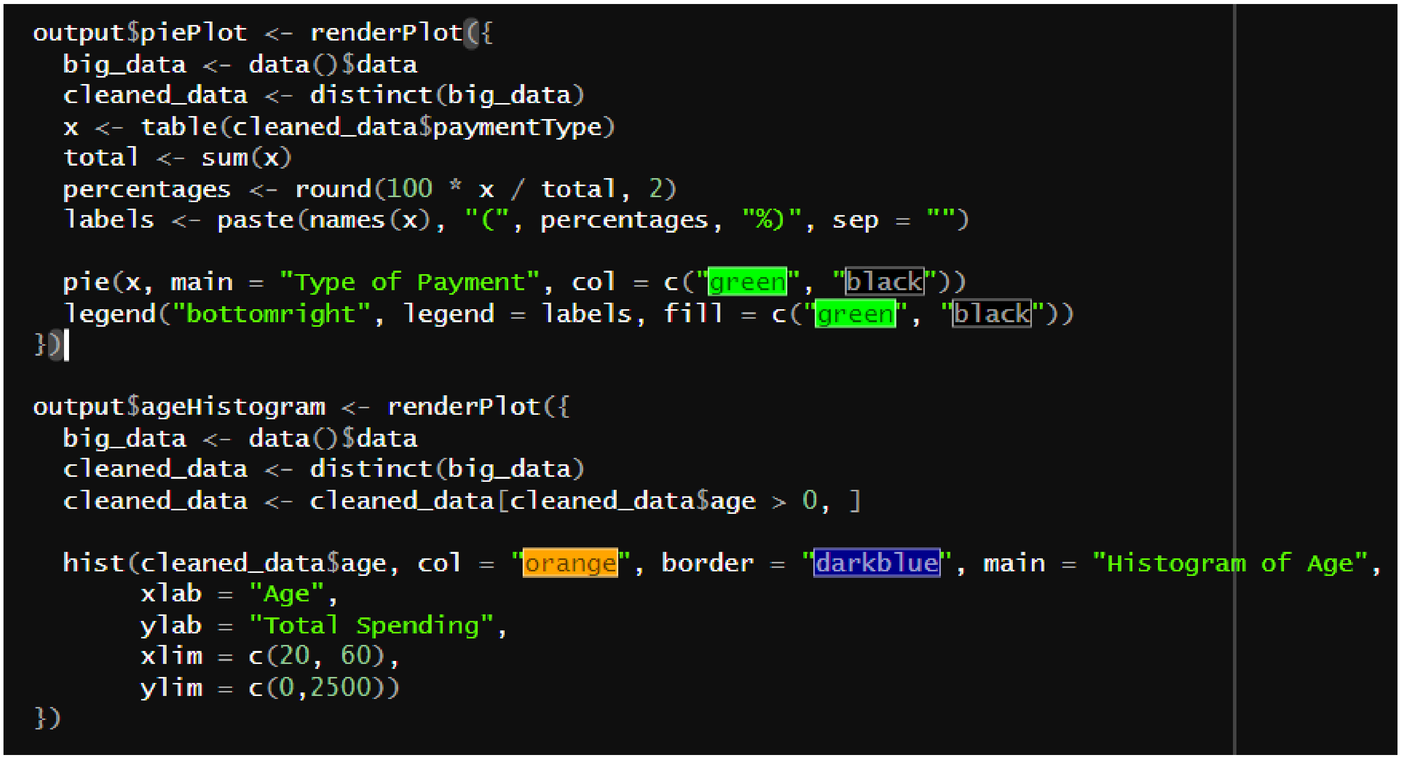


```
43    output$piePlot <- renderPlot({
44      big_data <- data()$data
45      cleaned_data <- distinct(big_data)
46      x <- table(cleaned_data$paymentType)
47      total <- sum(x)
48      percentages <- round(100 * x / total, 2)
49      labels <- paste(names(x), "(", percentages, "%)", sep = "")
50
51      pie(x, main = "Type of Payment", col = c("green", "black"))
52      legend("bottomright", legend = labels, fill = c("green", "black"))
53    })
```

So like in line 44 in the code we called the data, in line 45 we cleaned the data so the program can use it to process the data to visual outputs.

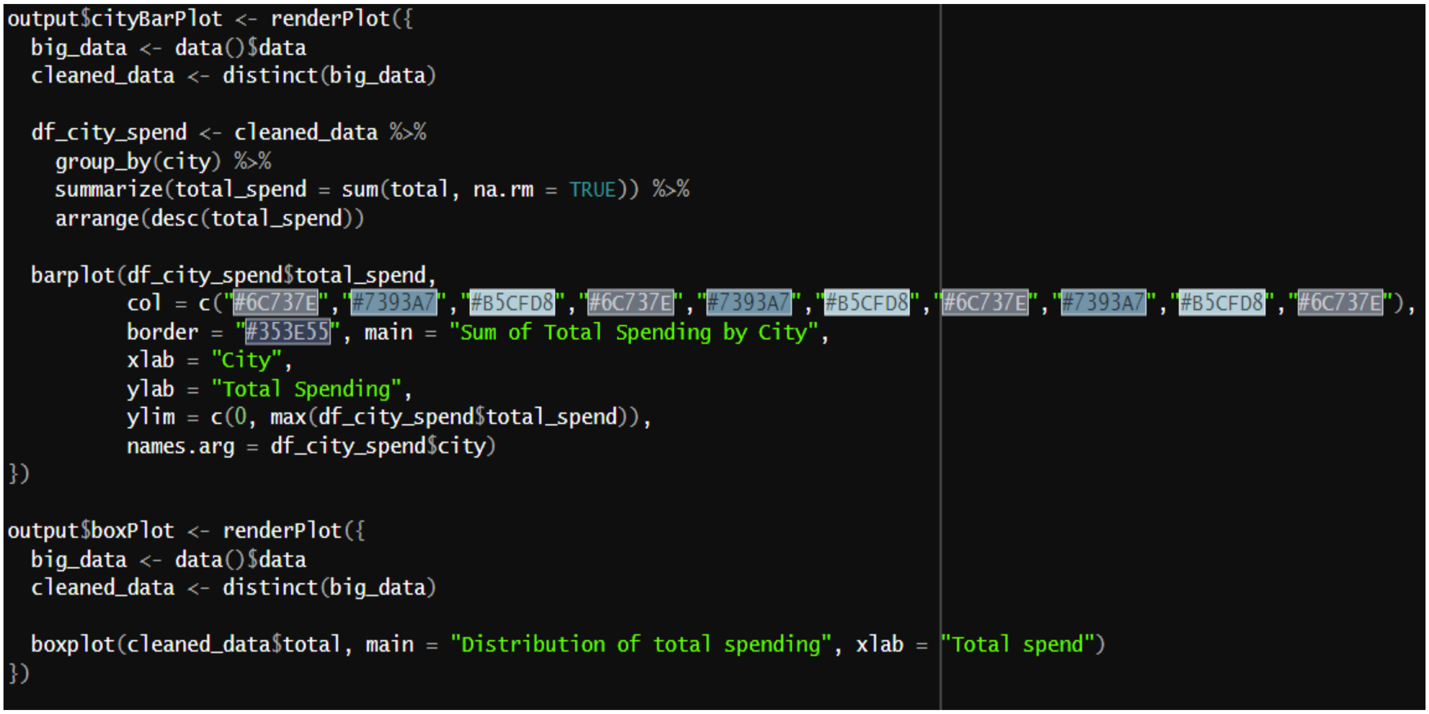
Pie plot (Cash – Credit) / Histogram (Age):

(أحمد أيمن خيري)



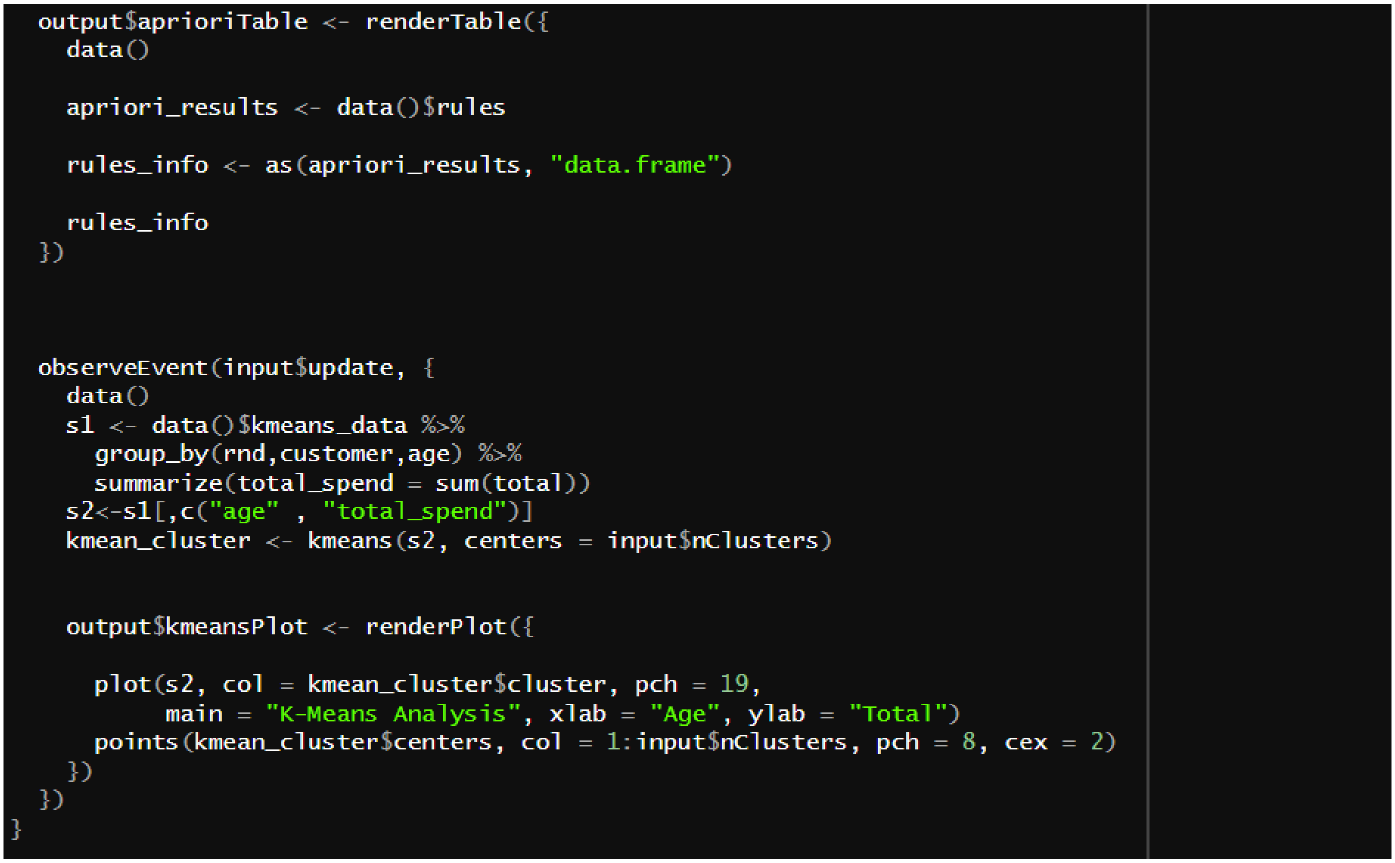
```
output$piePlot <- renderPlot({
  big_data <- data()$data
  cleaned_data <- distinct(big_data)
  x <- table(cleaned_data$paymentType)
  total <- sum(x)
  percentages <- round(100 * x / total, 2)
  labels <- paste(names(x), "(", percentages, "%)", sep = "")

  pie(x, main = "Type of Payment", col = c("green", "black"))
  legend("bottomright", legend = labels, fill = c("green", "black"))
})

output$ageHistogram <- renderPlot({
  big_data <- data()$data
  cleaned_data <- distinct(big_data)
  cleaned_data <- cleaned_data[cleaned_data$age > 0, ]

  hist(cleaned_data$age, col = "orange", border = "darkblue", main = "Histogram of Age",
       xlab = "Age",
       ylab = "Total Spending",
       xlim = c(20, 60),
       ylim = c(0,2500))
})
```

Bar plot (cities) / Box plot (Distribution):(أشرف عيسى)



```
output$cityBarPlot <- renderPlot({
  big_data <- data()$data
  cleaned_data <- distinct(big_data)

  df_city_spend <- cleaned_data %>%
    group_by(city) %>%
    summarize(total_spend = sum(total, na.rm = TRUE)) %>%
    arrange(desc(total_spend))

  barplot(df_city_spend$total_spend,
          col = c("#6C737E","#7393A7","#B5CFD8","#6C737E","#7393A7","#B5CFD8","#6C737E","#7393A7","#B5CFD8","#6C737E"),
          border = "#353E55", main = "Sum of Total Spending by City",
          xlab = "City",
          ylab = "Total Spending",
          ylim = c(0, max(df_city_spend$total_spend)),
          names.arg = df_city_spend$city)
})

output$boxPlot <- renderPlot({
  big_data <- data()$data
  cleaned_data <- distinct(big_data)

  boxplot(cleaned_data$total, main = "Distribution of total spending", xlab = "Total spend")
})
```

A priori / K-means: (أحمد محمد عبد الوهاب ابراهيم)



```
  output$aprioriTable <- renderTable({
    data()

    apriori_results <- data()$rules

    rules_info <- as(apriori_results, "data.frame")

    rules_info
  })



  observeEvent(input$update, {
    data()
    s1 <- data()$kmeans_data %>%
      group_by(rnd,customer,age) %>%
      summarize(total_spend = sum(total))
    s2<-s1[,c("age" , "total_spend")]
    kmean_cluster <- kmeans(s2, centers = input$nClusters)


    output$kmeansPlot <- renderPlot({

      plot(s2, col = kmean_cluster$cluster, pch = 19,
           main = "K-Means Analysis", xlab = "Age", ylab = "Total")
      points(kmean_cluster$centers, col = 1:input$nClusters, pch = 8, cex = 2)
    })
  })
}
```

In the apriori, it takes the data which is already been to transactions and then gets called at the start of its code so then take the rules and change it to a data frame so it can be visualized as a table in the program output.

In the Kmeans code, we grouped the data by the rnd which was a unique 15-user ID, and then we visualized the output after taking the number of clusters into a plot that shows the clusters by stars and the 15 users as 15 different points and clustered them in easy to understand graph.