Grocery Sales Analysis Dashboard

Team ID: 104

Mustafa Ashraf (GUI Server Backend) 23012069:

- implements the server-side logic for the Shiny dashboard.

Mustafa Abdelaziz (Data Visualization) 23011542:

- Creates and designs visualizations for the Shiny dashboard.

Mustafa Mohamed (GUI Frontend) 23011546:

- Responsible for creating the graphical user interface (GUI) using Shiny.

Alibar Atef (Data Cleaning-Data Exploration) 23011225:

- Responsible for data cleaning and exploration.

Problem Description:

What will the program do?

The program is a Shiny Application designed to perform comprehensive analysis on a grocery sales dataset. It offers a user-friendly interface with functionalities tailored towards exploring, visualizing, and deriving insights from the data. Key features of the application include data visualization, customer segmentation through clustering, and association rule mining.

What will the input to the program be?

1. Dataset: The program expects a CSV file containing grocery sales data.
2. Parameters: Number of Clusters, Minimum Support, Minimum Confidence

What will the output from the program be?

1. Dashboard Tab:

Visualization of various aspects of the dataset, including payment type distribution, total spending by age, total spending by city and distribution of total spending.

2. Customer Clustering and Table Tab:

A table displaying the original dataset with an additional column indicating the cluster assignment for each customer. The clustering is performed using the K-means algorithm based on the specified number of clusters.

3. Association Rules Tab:

Output of association rule mining based on the Apriori algorithm. This includes discovered association rules along with support, confidence, and other relevant metrics. These rules provide insights into item co-occurrence patterns in transactions, aiding in market basket analysis and decision-making processes.

Description of Dataset:

The dataset comprises records representing individual grocery sales transactions, encompassing various attributes providing insights into customer purchasing behavior, transaction details, and demographic information. Below is a comprehensive description of the dataset columns:

Items: This column contains a list of items purchased in each transaction. Each entry may include the name, or category of the item, separated by commas.

Count: The count column represents the quantity of each item purchased in the transaction. It indicates the number of units or packages bought for each item listed.

Total: This column denotes the total monetary value of the transaction, representing the sum of prices for all items purchased. It reflects the overall expenditure incurred by the customer during the transaction.

rnd :The "rnd" column functions similarly to a Customer ID but with broader applications. It contains randomly generated values or identifiers for transactions, serving purposes like data augmentation, anonymization, or simulation.

Customer: This column includes the names of the customers involved in the transactions. While it does not contain unique identifiers or codes, it serves to identify individual customers and track their purchasing behavior, preferences, and transaction history based on their names.

Age: The age column specifies the age of the customer involved in the transaction. It provides demographic information that enables segmentation and analysis based on age demographics.

City: This column indicates the city or location where the transaction took place or where the customer resides. It helps in understanding geographic patterns in purchasing behavior and targeting localized marketing strategies.

PaymentType: The paymentType column categorizes the payment method used by the customer to complete the transaction. It includes options such as cash and credit, indicating whether the payment was made in cash or via credit/debit card.

```r
#-----------Data Exploration-------------------------#
df <- read.csv("grc.csv",TRUE,",")
head(df)

# how many duplicated rows are in your data
sum(duplicated(df))

# To remove duplicate rows in a data frame.
df_without= unique(df)
head(df_without)
sum(duplicated(df_without))

sum(is.na(df))

outlier <- boxplot(df[, 3])$out
outlier
```

Initially, we load our dataset 'grc.csv' into a dataframe 'df' and identify any duplicated rows for data integrity. Duplicates are removed to update the dataframe as 'df_without'. We then check for and handle missing values to ensure data completeness. Finally, outlier detection is performed using a boxplot, providing insights into data distribution. This thorough data exploration process lays the groundwork for subsequent analysis, enhancing the dataset's reliability.

```r
#-----------GUI---------#
# Load required libraries
library(shiny)
library(ggplot2)
library(dplyr)
library(arules)
library(DT)

# UI definition
ui <- fluidPage(
  titlePanel("Grocery Data Dashboard"),

  sidebarLayout(
    sidebarPanel(
      fileInput("dataset_file", "Upload Dataset", accept = ".csv"),
      numericInput("num_clusters", "Number of Clusters", value = 2, min = 2, max = 4),
      numericInput("min_support", "Minimum Support", value = 0.1, min = 0.001, max = 1, step = 0.001),
      numericInput("min_confidence", "Minimum Confidence", value = 0.1, min = 0.001, max = 1, step = 0.001)
    ),

    mainPanel(
      tabsetPanel(
        tabPanel("Dashboard",
                 plotOutput("dashboard_plot")
        ),

        tabPanel("Customer Clustering and Table",
                 DT::DTOutput("customer_table")),

        tabPanel("Association Rules",
                 verbatimTextOutput("association_rules_output"))
      )
    )
  )
)
```

This code defines a Shiny application for a Grocery Data Dashboard. It loads necessary libraries and sets up the user interface (UI) layout. Users can upload a CSV dataset file and input parameters such as the number of clusters, minimum support, and minimum confidence.

shiny: Shiny is an R package that enables the creation of interactive web applications directly from R.

ggplot2: ggplot2 is a popular R package for creating elegant and customizable data visualizations.
dplyr: dplyr is a versatile R package for data manipulation and transformation. It offers a set of intuitive functions for filtering, selecting, summarizing, and arranging data, making it easier to perform data wrangling tasks.

arules: arules is an R package for mining association rules and frequent itemsets from transaction data. It provides functions for discovering patterns in transactional datasets, such as market basket analysis, which can uncover relationships between items purchased together.

DT: DT is an R package for creating interactive data tables in Shiny applications. It enables users to display large datasets in a tabular format with features such as sorting, searching, and pagination, enhancing the usability and interactivity of Shiny dashboards

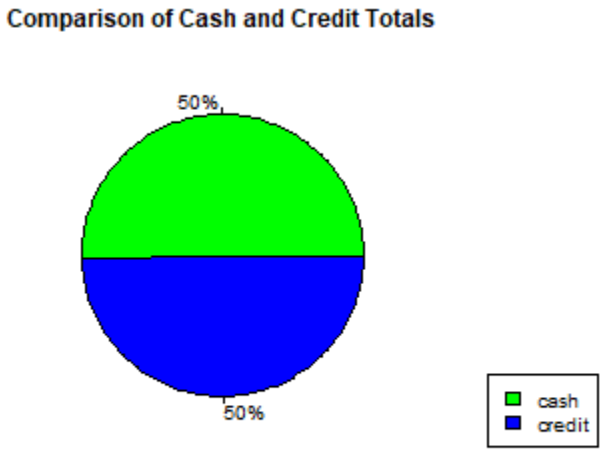
```r
  # Perform clustering and output table
  output$customer_table <- renderDT({
    req(input$dataset_file) # Require dataset to be uploaded
    df <- dataset()
    set.seed(123)
    kmeans_result <- kmeans(df[, c("age", "total")], centers = input$num_clusters)
    df$cluster <- as.factor(kmeans_result$cluster)
    df
  })

  # Generate association rules
  output$association_rules_output <- renderPrint({
    req(input$dataset_file) # Require dataset to be uploaded
    df <- dataset()
    item <- strsplit(df$items, ",")
    items <- as(item, "transactions")

    min_support <- input$min_support
    min_confidence <- input$min_confidence

    apriori <- apriori(items, parameter = list(supp = min_support, conf = min_confidence))
    inspect(apriori)
  })
}

# Run the application
shinyApp(ui = ui, server = server)
```

server(input, output)

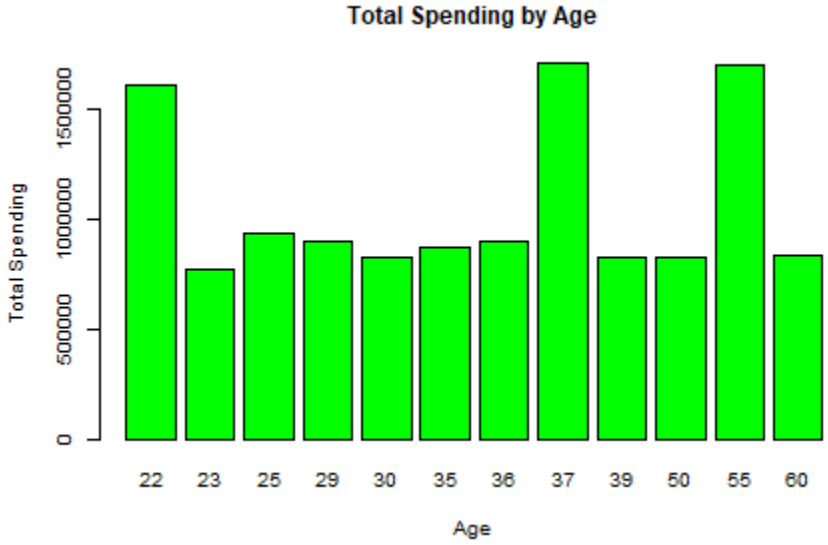
The code presents a Shiny application for interactive data analysis. Users can upload CSV datasets for analysis. Dashboard plots compare cash vs credit transactions, age vs total spending, city total spending, and total spending distribution. Clustering analysis uses the K-means algorithm on age and total spending attributes. Association rules are generated using the Apriori algorithm on transactional data. Users can set parameters like the number of clusters and minimum support/confidence thresholds. The application utilizes R packages like shiny, dplyr, arules, and arulesViz. Improvements could include error handling for file uploads and optimization for large datasets.

Plotted Graphs Insights

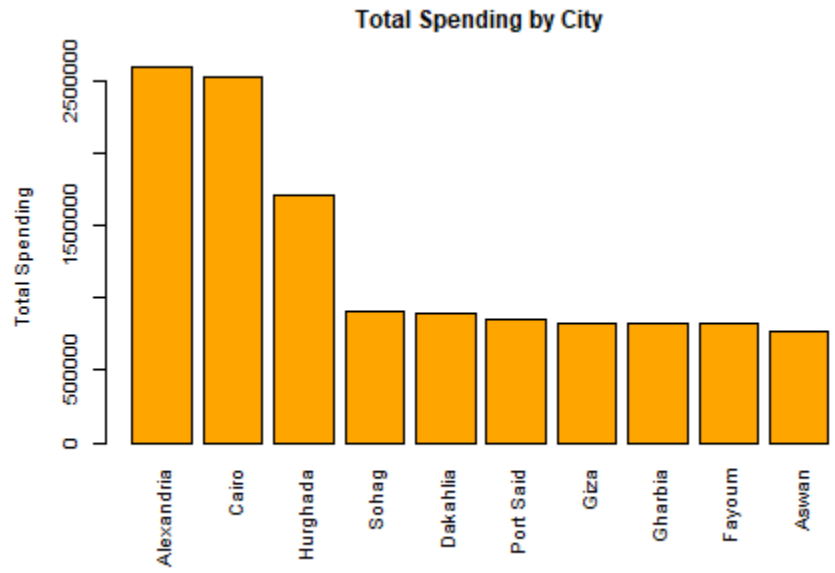
Dashboard Section:



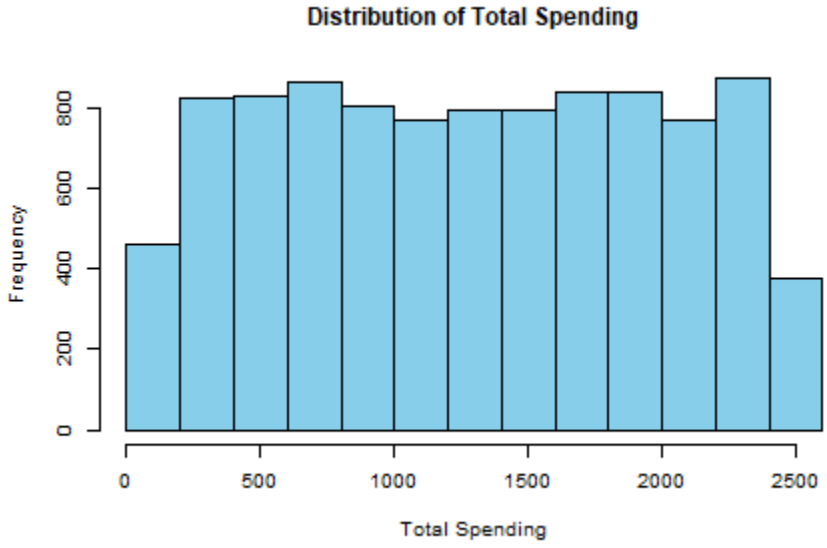
1. **Visualization of Cash vs. Credit Transactions:** The dashboard includes a pie chart comparing the total number of transactions made using cash and credit payment methods. It suggests that about the same number of customers use cash as those who use credit. So, the chart looks even between cash and credit payments.



2. **Total Spending by Age:** A bar plot illustrates the total spending aggregated by different age groups. This visualization helps identify spending patterns across different age demographics. It suggests that those who are 22, 37, and 55 spend the most.



3. **Total Spending by City:** Another bar plot displays the total spending in different cities or locations and arranges it descending order, highlighting that the most crowded cities such as Alexandria and Cairo have the highest spending totals in comparison with the other cities.



4. **Distribution of Total Spending:** A histogram showcases the distribution of total spending across all transactions. This visualization provides insights into the spread and concentration of spending amounts, aiding in understanding the overall spending behavior of customers.

Customer Clustering and Association Rules Sections:

Customer Clustering and Table: The dashboard allows users to perform customer segmentation using the k-means clustering algorithm based on age and total spending. The resulting clusters are displayed in a table format, providing insights into distinct customer groups and their characteristics.

Association Rules Mining: Users can explore association rules mined from the dataset using the Apriori algorithm. These rules reveal patterns of item co-occurrence in transactions, indicating potential product affinities and market basket insights.