Data Science Final Project

Mohamed Walied Yakout
23011501

Ahmed Abdelmoaty *(Team Leader)*
23012214

Seif Eldin Amgad
23011085

Omar Mahmoud Nabil
23012090

Mohamed Mohamed Elbassat
23010047

Abstract

This report details a customer behavior analysis project utilizing the "Groceries Market Basket Dataset." The project employs a data-driven approach to uncover customer purchasing patterns and relationships within the grocery market.

Methodology:

- **Exploratory Data Analysis (EDA):** A comprehensive initial analysis of the dataset was conducted to understand its characteristics, identify potential issues, and gain insights into customer behavior.
- **Data Cleaning:** Data cleaning procedures addressed missing values, inconsistencies, and outliers to ensure data quality for downstream analysis.
- **Association Rule Learning:** Association rule learning was employed to identify frequently co-purchased items, providing valuable insights into product placement strategies and promotional campaigns.
- **K-Means Clustering:** K-means clustering was utilized to segment customers based on their purchasing habits, revealing distinct customer groups with unique basket compositions.
- **GUI Application Development:** A user-friendly Graphical User Interface (GUI) application was developed to facilitate interactive exploration of the data, visualization of insights, and application of the derived knowledge.

Outcomes:

This project successfully leveraged data analysis techniques to extract valuable customer insights from the "Groceries Market Basket Dataset." The k-means clustering uncovered distinct customer segments, and association rule learning revealed product relationships. The developed GUI application empowers users to interactively explore these findings and gain a deeper understanding of customer behavior within the grocery market.

Benefits:

The project's findings provide valuable information for businesses in the grocery sector, enabling them to:

- **Target customer segments:** Develop targeted marketing campaigns and promotions for specific customer groups based on their purchasing habits.
- **Optimize product placement:** Strategically arrange products on shelves to encourage co-purchases based on association rules.

Exploratory Data Analysis (EDA) and Data Cleaning

Initial Data Exploration

Before applying the cleaning steps within the function, the code performs some initial data exploration to understand the state of the data. Here's a breakdown of these exploration steps:

Duplicate Check: The code uses *sum(duplicated(data))* to count the number of duplicate rows in the data *(num_duplicates).* It then prints num_duplicates to reveal the number of duplicates. Finally, it uses *data[duplicated(data),]* to identify and print the actual duplicate rows.

Data Type Verification: The code uses unique(data) to get a representative sample of the data (dataclean1). It then checks the data types of specific columns using *is.integer()* and *is.character()* for integer and character data, respectively. This helps identify any inconsistencies in data types.

Missing Value Check: The code uses *sum(is.na(dataclean1))* to calculate the total number of missing values (NA) across all columns in dataclean1.

Outlier Analysis: The code utilizes boxplot(dataclean1[,c(2,3,4,6)]) to create boxplots for specific columns with numerical data to visually identify potential outliers.

It then extracts the outliers from the boxplot results using *boxplot(dataclean1$count)$out.* This helps determine if there are extreme values that might require further investigation. However, after investigating the outliers, particularly in the "count" column, it was determined that these outliers could represent legitimate purchases with a large number of items. Therefore, the decision was made to not remove these outliers from the data

```
num_duplicates <- sum(duplicated(data))
print(num_duplicates)
print(data[duplicated(data),])
dataclean1=unique(data)
print(dataclean1)
is.integer(dataclean1$age)
is.integer(dataclean1$count)
is.integer(dataclean1$total)
is.integer(dataclean1$rnd)
is.character(dataclean1$items)
is.character(dataclean1$customer)
is.character(dataclean1$city)
is.character(dataclean1$paymentType)
dataclean1$count=as.integer(dataclean1$count)
dataclean1$age=as.integer(dataclean1$age)
dataclean1$rnd=as.integer(dataclean1$rnd)
dataclean1$total=as.integer(dataclean1$total)
print(n=9836,dataclean1)
sum(is.na(dataclean1))

boxplot(dataclean1[,c(2,3,4,6)])
outlier=boxplot(dataclean1$count)$out
```

Data Cleaning Steps in the main code after exploration. Based on the initial exploration, the data_clean function performs the following cleaning steps:

- Removes duplicates: It uses *unique(data())* to eliminate duplicate rows.
- Converts data types: It explicitly converts the data types of specific columns *("count", "age", "rnd", "total")* to integers using *as.integer()* to ensure consistency.
- Removes missing values: It removes rows with missing values (NA) using *na.omit()* to avoid potential issues during analysis.

```
data_clean <- function(){
  data_unique <- unique(data())


  # Convert columns to appropriate data types
  data_unique$count <- as.integer(data_unique$count)
  data_unique$age <- as.integer(data_unique$age)
  data_unique$rnd <- as.integer(data_unique$rnd)
  data_unique$total <- as.integer(data_unique$total)
  data_cleaned <- na.omit(data_unique)
  # Print cleaned data
  print(data_unique)

  return(data_unique)
}
```

Data Visualization

In this phase of the project, various visualization techniques were employed to gain insights and effectively communicate patterns and trends present in the dataset. The following visualization methods were utilized:

1. Cash and Credit Totals: To depict the distribution of total spending across cash and credit transactions, a visually engaging pie chart was created. This interactive chart allows for intuitive exploration of the proportion of spending attributed to each payment type. *(Figure 1)*

2. Age and Total Spending: A scatter plot was employed to visualize the relationship between age and total spending. This plot facilitates the identification of any potential correlations or patterns between these two variables, aiding in the understanding of customer spending behavior across different age groups. *(Figure 2)*

3. Total Spending by City (Descending): To showcase the disparity in total spending across various cities, a descending bar plot was generated. This visually striking plot provides a clear depiction of the cities with the highest total spending, enabling quick identification of key contributors to overall expenditure. *(Figure 3)*

4. Distribution of Total Spending: A box plot was utilized to illustrate the distribution of total spending within the dataset. This plot offers insights into the spread and central tendency of spending data, highlighting any outliers or concentration of values. *(Figure 4)*

The utilization of *ggplot*, *plotly*, and *ggplotly* enabled the creation of dynamic and interactive visualizations, enhancing the accessibility and interpretability of the findings. These visualizations serve as powerful tools for data exploration and storytelling, facilitating informed decision-making and actionable insights.

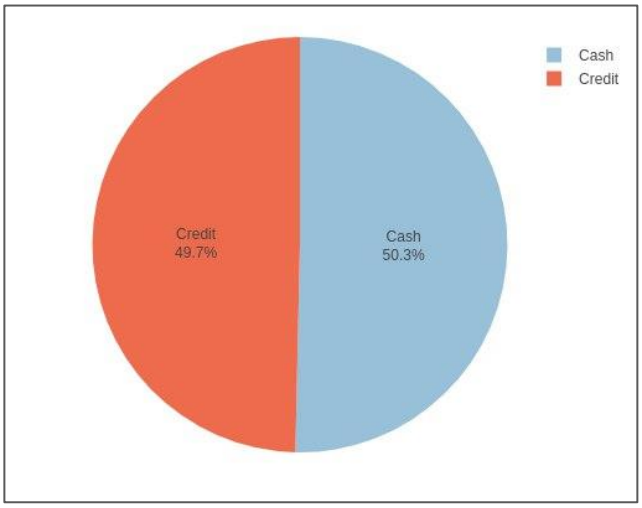


Figure 1

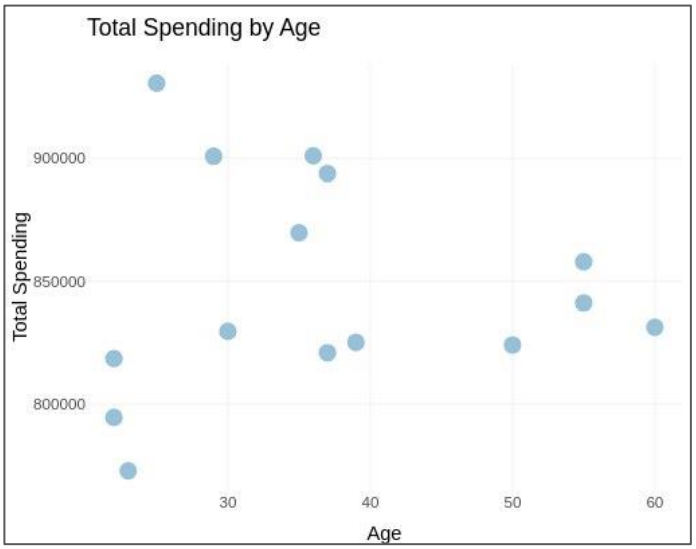


Figure 2

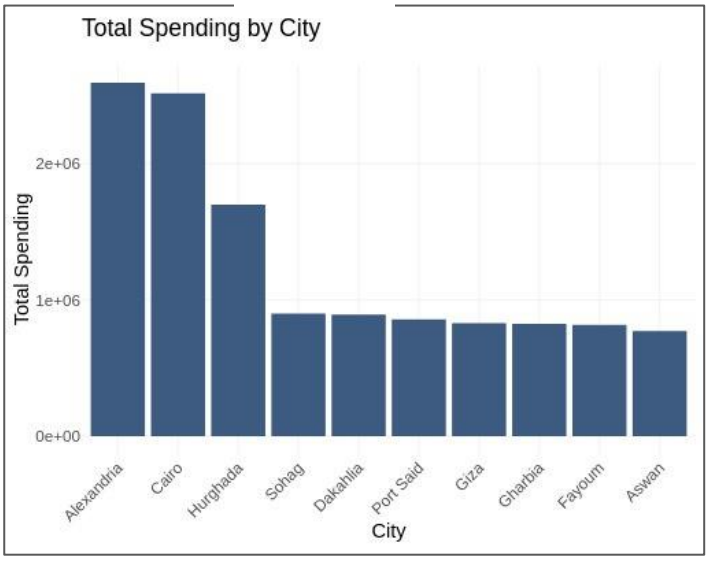


Figure 3

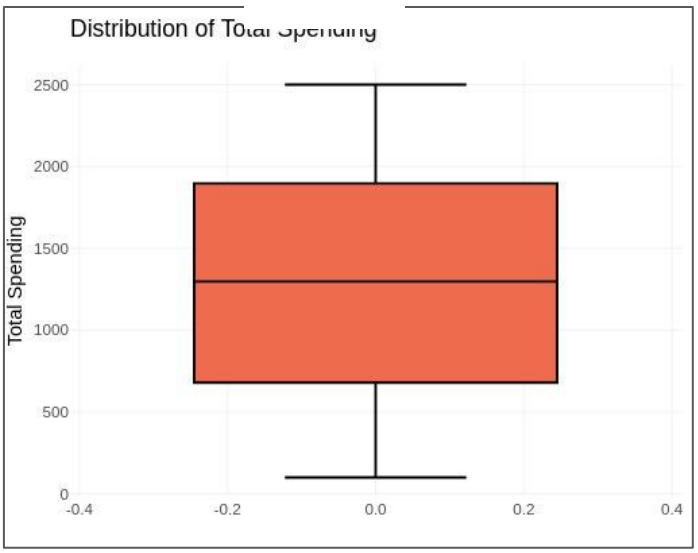
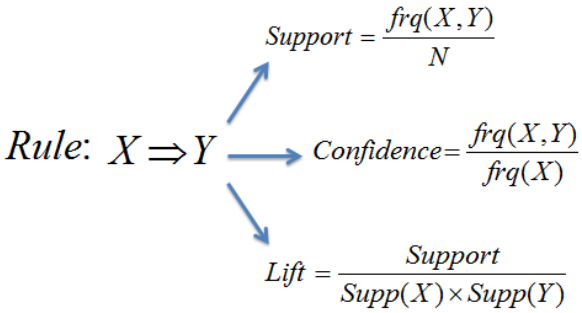


Figure 4

Association rules for Basket analysis

is a technique used to discover frequent item sets and relationships within transactional data. In your case, it helps identify products that customers frequently purchase together in the "Groceries Market Basket Dataset."



Here's some information to consider for your report's association rule section:

- **Metrics:** The metrics used to evaluate association rules, such as support (frequency of itemset occurrence) and confidence (likelihood of Y appearing given X).
- **Top Rules:** Present the most interesting association rules discovered. These could highlight frequently co-purchased products or unexpected relationships.
- **Business Implications:** Explain how these association rules can be used to improve business strategies, such as product placement or targeted promotions.

Digging into the Rules:

- We dive into the dataset, extract those items, and organize them into transactions.
- Then, we let the **Apriori algorithm** do its thing *(apriori function)*, generating association rules based on the specified support and confidence thresholds.
- Putting it in Plain Sight: But we don't stop there! We convert those freshly minted association rules into a good data frame *(rules_df)* for better visibility and analysis.

```
# Generate association rules
items <- data_clean()$items
items_list <- strsplit(items, ",")
transactions <- as(items_list, "transactions")
rules <- apriori(transactions, parameter = list(support = input$support, confidence = input$confidence))
```

Association Rules

rules	support	confidence	coverage	lift	count
{} => {whole milk}	0.26	0.26	1.00	1.00	2513
{} => {other vegetables}	0.19	0.19	1.00	1.00	1903
{} => {rolls/buns}	0.18	0.18	1.00	1.00	1809
{} => {soda}	0.17	0.17	1.00	1.00	1714
{} => {yogurt}	0.14	0.14	1.00	1.00	1372
{} => {bottled water}	0.11	0.11	1.00	1.00	1087
{} => {root vegetables}	0.11	0.11	1.00	1.00	1072
{} => {tropical fruit}	0.10	0.10	1.00	1.00	1032
{} => {shopping bags}	0.10	0.10	1.00	1.00	969
{} => {sausage}	0.09	0.09	1.00	1.00	924
{other vegetables} => {whole milk}	0.07	0.39	0.19	1.51	736
{whole milk} => {other vegetables}	0.07	0.29	0.26	1.51	736
{rolls/buns} => {whole milk}	0.06	0.31	0.18	1.20	557
{whole milk} => {rolls/buns}	0.06	0.22	0.26	1.20	557
{yogurt} => {whole milk}	0.06	0.40	0.14	1.57	551

Figure 5 - The association rules

K-Means Clustering for Customer Segmentation

K-means clustering is an unsupervised machine learning technique commonly used for customer segmentation. It groups customers into distinct clusters based on similarities in their purchasing behavior within the "Groceries Market Basket Dataset." This allows us to identify distinct customer segments with unique characteristics.

Process:

1. **Feature Selection:** We selected relevant features from the dataset that best represent customer buying habits. Examples might include:
    - Frequently purchased categories (e.g., dairy, produce)
    - Total purchase amount
    - Number of unique items per basket
2. **K Determination:** We determined the optimal number of clusters (k) through techniques like the elbow method or silhouette analysis. This ensures we capture meaningful segments without over- or under-segmentation.
3. **Clustering:** The k-means algorithm iteratively groups customers into k clusters by:
    - Initializing random centroids (cluster centers)
    - Assigning each customer to the nearest centroid based on a distance metric (e.g., Euclidean distance)
    - Recomputing the centroid of each cluster based on the assigned customers.
    - Repeating these steps until convergence (centroids stabilize).

Outcomes:

K-means clustering revealed distinct customer segments with unique purchasing patterns. Analyzing these segments allows us to:

- **Understand customer profiles:** Identify characteristics and preferences that define each segment (e.g., budget-conscious shoppers, health-conscious buyers, families with young children).
- **Targeted marketing:** Develop targeted marketing campaigns and promotions for specific segments based on their needs and preferences.
- **Product recommendations:** Recommend relevant products to customers based on their segment affiliation.

By segmenting customers using k-means clustering, we gain valuable insights into customer behavior within the grocery market. This information can be leveraged to improve customer targeting, product recommendations, and overall marketing strategies.

- **Feature Selection:**

Selecting only the *"total_amount"* and *"avg_age"* features from the grouped_data for clustering. It's essential to choose numerical features for K-means clustering.

- **K-Means Clustering:**

Performing K-means clustering on the *numerical_data*. The centers argument specifies the number of clusters to create (set to 2 in this case). K-means assigns each data point (customer in this case) to the nearest cluster centroid based on Euclidean distance.

- **Adding Cluster Labels:**

Combining the original *grouped_data* with the cluster labels assigned by K-means. The *cbind* function creates a new data frame by binding columns together. The cluster column now contains the cluster ID for each customer.

```r
library(dplyr)
grouped_data <- cleaned_data %>%
  group_by(customer) %>%
  summarise(
    total_transactions = n(),
    total_amount = sum(total),
    avg_age = mean(age),
    count = sum(count),
    city = unique(city)
  )

#K-means Clustering
numerical_data <- grouped_data[, c("total_amount","avg_age")]
K_means_Clusterung = kmeans(numerical_data,centers = 2)

data_final <- cbind(grouped_data,cluster=K_means_Clusterung$cluster)
```

- **Data Grouping:** Group the by the "customer" identifier. This creates groups of transactions for each unique customer.

- **Feature Engineering:**

Within each customer group, the function calculates various summary statistics:

  - Counts the number of transactions for each customer.
  - Calculates the total amount spent by each customer by summing the "total" variable (assuming it represents transaction amount).
  - Computes the average age of customers within each group (assuming "age" is a numeric variable representing customer age).
  - Calculates the total count of a variable named "count". However, without context about the meaning of "count" in your data, it's unclear what this value represents. It's recommended to rename the variable or the summary function (`sum`) if it doesn't represent a meaningful count.
  - Extracts the unique city for each customer group, assuming a customer belongs to a single city. If customers can have multiple cities, consider alternative approaches like keeping all cities or using a dominant city based on criteria relevant to your analysis.

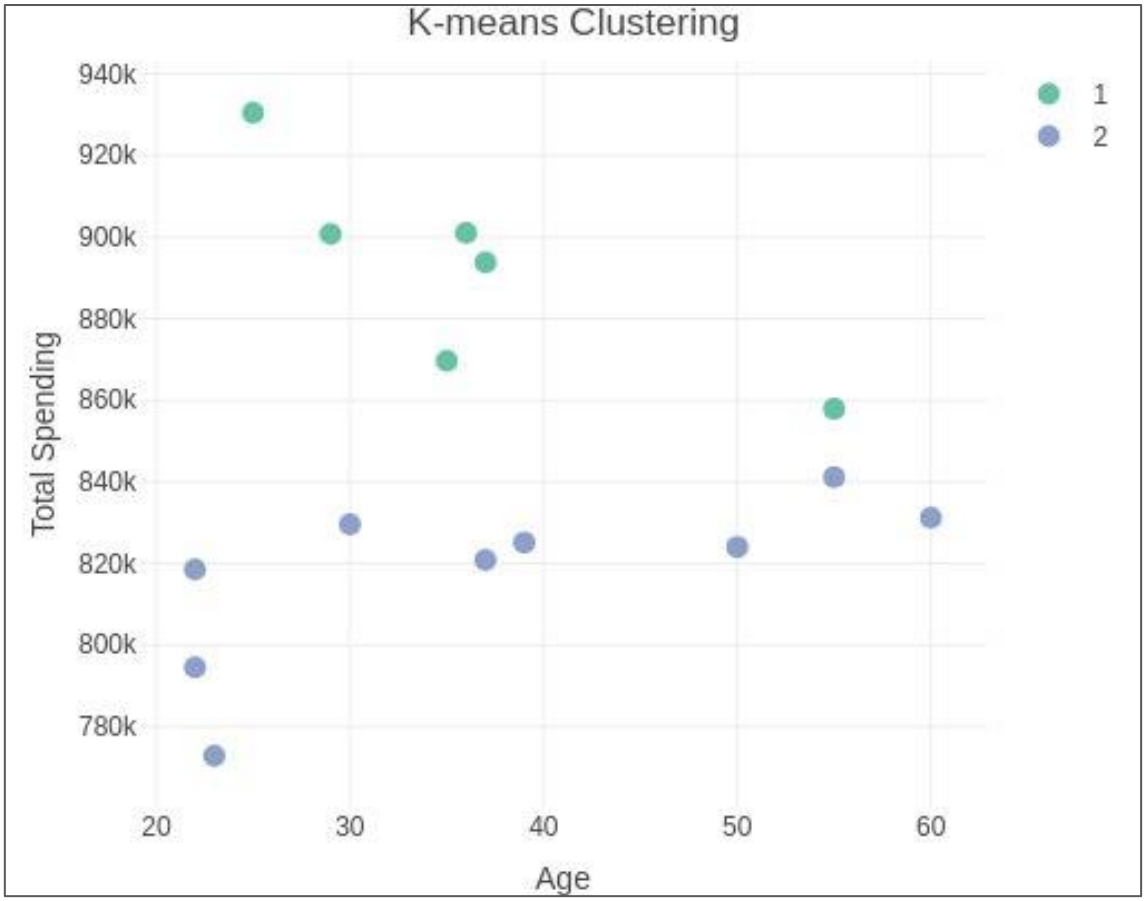


Figure 6 - 2 Clusters

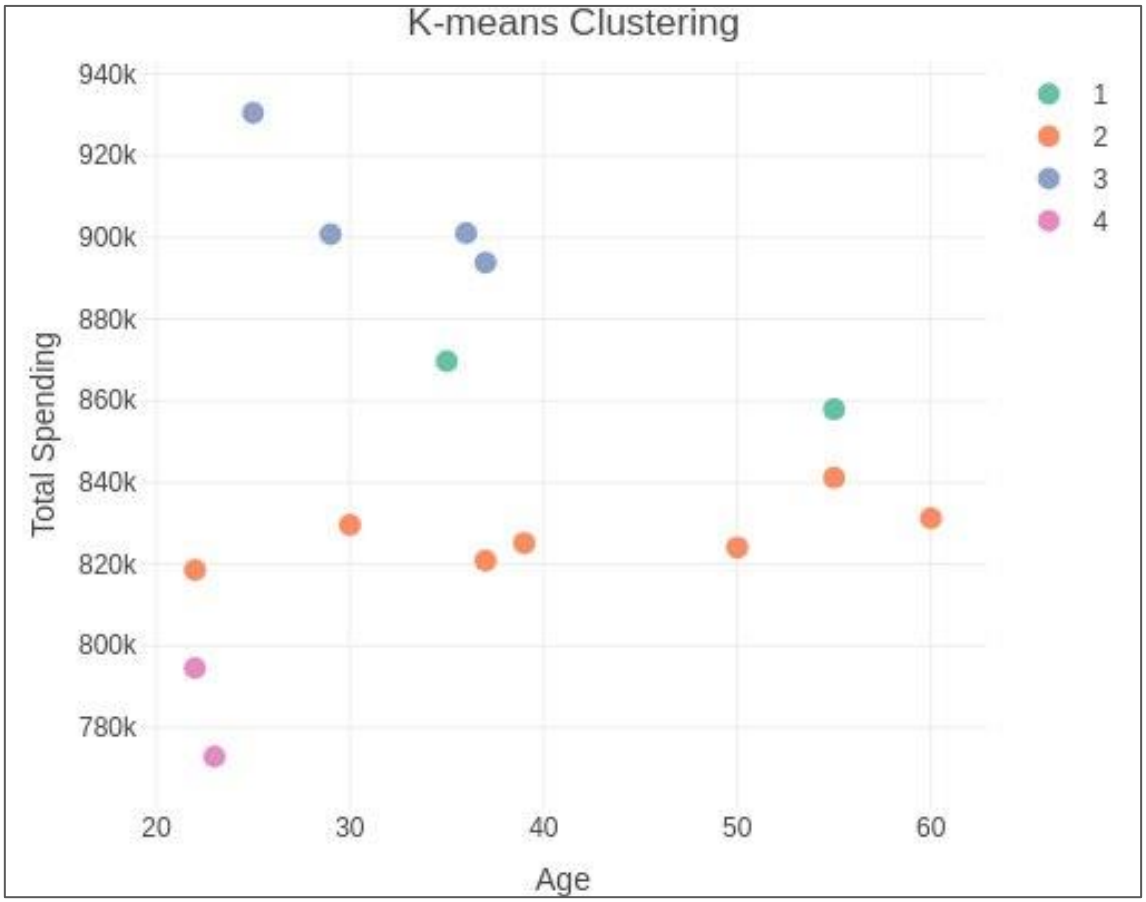


Figure 7 - 4 Clusters

GUI Application Development

Implementation with Shiny Dashboard

The Shiny dashboard framework was leveraged to develop a user-friendly graphical user interface (GUI) for the project. The implementation consisted of the following key components:

1. User Interface (UI: Within the UI function, fluid rows were utilized to organize the layout. Each row contained boxes housing select input, slider input, and file input elements to gather the necessary parameters and data for analysis.

2. Server Function: In the server function, the various components were integrated to ensure seamless data processing and visualization. Reactive expressions were employed to dynamically apply changes based on user inputs.

3. Reactive Functions: Several reactive functions were defined to streamline data handling and processing. The *data()* function was created to retrieve the uploaded dataset via file input. Another function, *data_clean(),* was developed to perform data cleaning operations, returning the cleaned dataset. Additionally, the *data_grouped()* function facilitated data grouping by customer name, enhancing data organization and analysis.

4. Output Process: The reactive functions were strategically utilized in the output process throughout the GUI. This ensured that the cleaned and grouped data were seamlessly integrated into the visualizations and analyses presented to the user.

The utilization of Shiny dashboard not only facilitated the development of an intuitive and interactive user interface but also enhanced the efficiency and effectiveness of data processing and analysis. By encapsulating complex data operations within a user-friendly interface, Shiny dashboard enabled streamlined data exploration and visualization for users of varying technical backgrounds.

In conclusion, this data science project successfully showcased the power of K-means clustering and association rule mining in analyzing grocery data. The findings offer practical insights that can be employed to enhance customer targeting, product placement, and ultimately, drive sales growth.