WELCOME
TO OUR PROJECT

- **Ahmed Mohamed Ahmed Fouad**
- **Project Part:** Report Writing

- **Ahmed Al-Sayed Mohamed Mors**
- **Project Part:** Data Analysis

- **Hossam Samir Abd-Elati Kamel**
- **Project Part:** Data Cleaning

- **Omar Adel Awad**
- **Project Part:** Graphical User Interface (GUI)

- **Adham Ahmed Mohamed Abdel-Razak**
- **Project Part:** Clustering-----TeamLeader

Project Goal : TO apply and practice what has been learned in real-world task.

Rules:

1. Cheating is strictly prohibited; both parties will face strict measures if found.
2. Groups should consist of 2 to 6 students.
    1. Groups will be assigned a number announced on Teams.
    2. Discussion of the projects will be announced soon.
3. Each group must prepare a PDF report named “Project Report + Group No.” including:
    1. Students’ names, IDs, and group number.
    2. Problem description answering:
        1. What will the program do?
        2. What will the input be?
        3. What will the output be?
    3. Full dataset description.
    4. Project steps screenshots.
    5. Explanation of results and insights via plotted graphs.
    6. Discussion of the code (libraries used, attributes) with screenshots and descriptions.

Project Details:

- **Tasks to be implemented in R:**
  - User interface allowing customers to enter data and see results.
  - Data assessment and cleaning.
  - Data Visualization:
    - Compare cash and credit totals.
    - Compare age groups with total spending.
    - Show total spending by city in descending order.
    - Display distribution of total spending.
  - Consolidate all plots in one dashboard.
  - Split customers into groups based on:
    - Total spending., Age.
    - Display a table with customer name, age, total spending, and cluster number
    - Generate association rules between items with user-defined minimum support and confidince

Problem Description

The project involves development of an R program to perform analytics and visualization of a grocery dataset. Major tasks to be performed involve cleaning the data, analyzing it, and then presenting the results in a clear and intuitive form. The following tasks can be identified:

Inputs: grocery dataset, user-specified parameters (e.g. the number of clusters, minimum support and confidence thresholds).

Outputs: visual representations of the data, results of the customer segmentation, association rule metrics, and a dashboard.

.Solution Description

The solution is structured as follows:

1. Interface Development: A GUI is developed using R packages such as shiny for data entry and specification of parameters.

2. Data Cleaning: The dataset is prepared by cleaning the inconsistencies it contains.

3. Data Visualization: Charts are created to examine payment methods, payment for goods by age and city, and distribution of payment.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

Untitled1 | project.R | project.R | Untitled2*

Source on Save | Run | Source

```
1  read.csv("C:/Users/MOHAMED REDA/Downloads/grc.csv")
2  data=read.csv("C:/Users/MOHAMED REDA/Downloads/grc.csv")
3  #Function for data frequent
4  duplicated('data')
5  #F for N of data frequent
6  sum(duplicated('data'))
7  #F for save data new
8  data_c1=unique(data)
9  #F for Number of NA values
10 sum(is.na(data_c1))
11 #F for save data new
12 data_c2=na.omit(data_c1)
13 #F for any outlier
14 boxplot(data_c2[, 2:4])
15 boxplot(data_c2[, 6])
16 #F for correct data types
```

2:57 (Top Level) R Script

Console | Terminal | Background Jobs

R 4.4.1 · C:/Users/MOHAMED REDA/OneDrive/Desktop/Material/intro to data science/projects/New folder/

```
> read.csv("C:/Users/MOHAMED REDA/Downloads/grc.csv")

items
1
citrus fruit,tropical fruit,whole milk,butter,curd,yogurt,flour,bottled water,dishes
2
beef
3
yogurt
4
other vegetables
5
pastry,bottled water
6
packaged fruit/vegetables,brown bread,canned beer
7
other vegetables,whole milk,frozen vegetables,canned fish,salty snack,seasonal products,de
tergent
```

Environment | History | Connections | Tutorial

Import Dataset | 148 MiB | List

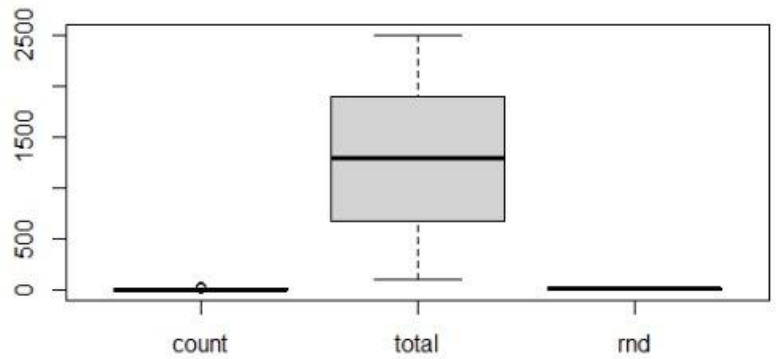
R | Global Environment

Data

data	9835 obs. of 8 variables
data_c1	9834 obs. of 8 variables
data_c2	9833 obs. of 8 variables

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

Untitled1 | project.R | project.R | Untitled2*

Source on Save | Run | Source

```
12  data_c2=na.omit(data_c1)
13  #F for any outlier
14  boxplot(data_c2[, 2:4])
15  boxplot(data_c2[, 6])
16  #F for correct data types
17  data_c2$count=as.integer(data_c2$count)
18  data_c2$total=as.integer(data_c2$total)
19  data_c2$rnd=as.integer(data_c2$rnd)
20  data_c2$age=as.integer(data_c2$age)
21  data_c2$items=as.character(data_c2$items)
22  data_c2$customer=as.character(data_c2$customer)
23  data_c2$city=as.character(data_c2$city)
24  data_c2$paymentType=as.character(data_c2$paymentType)
25  #F for save clean data
26  write.csv(data_c2,"cleaned6_grc.csv")
27
```

2:57 (Top Level) R Script

Console | Terminal | Background Jobs

R 4.4.1 · C:/Users/MOHAMED REDA/OneDrive/Desktop/Material/intro to data science/projects/New folder/

```
> #F for save data new
> data_c2=na.omit(data_c1)
> #F for any outlier
> boxplot(data_c2[, 2:4])
> boxplot(data_c2[, 6])
> #F for correct data types
> data_c2$count=as.integer(data_c2$count)
> data_c2$total=as.integer(data_c2$total)
> data_c2$rnd=as.integer(data_c2$rnd)
> data_c2$age=as.integer(data_c2$age)
> data_c2$items=as.character(data_c2$items)
> data_c2$customer=as.character(data_c2$customer)
> data_c2$city=as.character(data_c2$city)
> data_c2$paymentType=as.character(data_c2$paymentType)
> #F for save clean data
> write.csv(data_c2,"cleaned6_grc.csv")
>
```

Environment | History | Connections | Tutorial

Import Dataset | 149 MiB | List

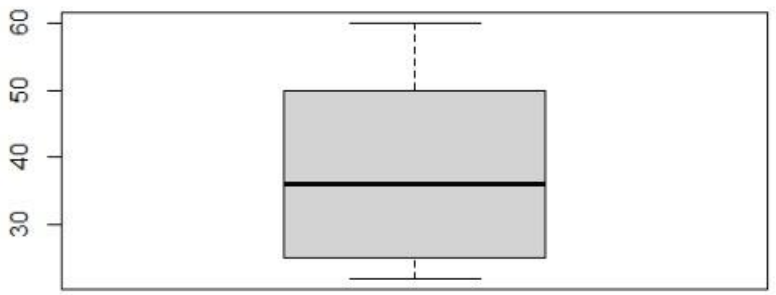
R | Global Environment

Data

data	9835 obs. of 8 variables
data_c1	9834 obs. of 8 variables
data_c2	9833 obs. of 8 variables

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish



Console | Terminal | Background Jobs

R 4.4.1 · C:/Users/MOHAMED REDA/OneDrive/Desktop/Material/intro to data science/projects/New folder/

```
> #F for save data new
> data_c2=na.omit(data_c1)
> #F for any outlier
> boxplot(data_c2[, 2:4])
> boxplot(data_c2[, 6])
> #F for correct data types
> data_c2$count=as.integer(data_c2$count)
> data_c2$total=as.integer(data_c2$total)
> data_c2$rnd=as.integer(data_c2$rnd)
> data_c2$age=as.integer(data_c2$age)
> data_c2$items=as.character(data_c2$items)
> data_c2$customer=as.character(data_c2$customer)
> data_c2$city=as.character(data_c2$city)
> data_c2$paymentType=as.character(data_c2$paymentType)
> #F for save clean data
> write.csv(data_c2,"cleaned6_grc.csv")
>
```

	items	count	total	rnd	customer	age	city	paymentTy
1	citrus fruit	9	1965	1	Farida	22	Cairo	Credit
2	beef	1	784	11	Hanan	22	Fayoum	Cash
3	yogurt	1	599	1	Farida	22	Cairo	Cash
4	other vege	1	1906	1	Farida	22	Cairo	Credit
5	pastry,bott	2	730	1	Farida	22	Cairo	Credit
6	packaged f	3	2236	11	Hanan	22	Fayoum	Cash
7	other vege	7	2146	11	Hanan	22	Fayoum	Credit
8	whole milk	1	453	1	Farida	22	Cairo	Credit
9	flour,salt,b	5	2079	1	Farida	22	Cairo	Credit
10	fruit/veget	3	1133	1	Farida	22	Cairo	Credit
11	sausage,sli	9	2213	1	Farida	22	Cairo	Credit
12	sausage,w	10	1695	1	Farida	22	Cairo	Credit
13	curd chees	1	717	11	Hanan	22	Fayoum	Credit
14	newspaper	1	1854	11	Hanan	22	Fayoum	Cash
15	canned be	1	111	1	Farida	22	Cairo	Credit
16	frankfurter	10	1379	11	Hanan	22	Fayoum	Credit
17	grapes,oth	6	311	1	Farida	22	Cairo	Cash
18	ham,tropic	8	1201	11	Hanan	22	Fayoum	Cash
19	bottled be	1	280	11	Hanan	22	Fayoum	Credit
20	other vege	2	2103	11	Hanan	22	Fayoum	Cash
21	soda,bottl	4	1922	1	Farida	22	Cairo	Credit
22	curd,crean	7	853	11	Hanan	22	Fayoum	Credit
23	curd,salt,li	5	1544	11	Hanan	22	Fayoum	Cash
24	sausage,bo	3	2461	11	Hanan	22	Fayoum	Credit
25	curd,rolls/l	2	1076	11	Hanan	22	Fayoum	Cash
26	canned be	1	1961	11	Hanan	22	Fayoum	Credit
27	pork,root v	5	1702	11	Hanan	22	Fayoum	Cash
28	pot plants	1	876	11	Hanan	22	Fayoum	Cash
29	sausage,so	4	1961	1	Farida	22	Cairo	Credit
30	meat,othe	4	2413	11	Hanan	22	Fayoum	Cash
31	ice cream	1	2186	11	Hanan	22	Fayoum	Credit
32	frankfurter	5	906	11	Hanan	22	Fayoum	Cash
33	chicken,pip	11	2371	11	Hanan	22	Fayoum	Credit
34	sausage	1	2035	11	Hanan	22	Fayoum	Credit
35	sausage,ch	9	140	1	Farida	22	Cairo	Credit
36	chicken,wh	5	416	11	Hanan	22	Fayoum	Cash
37	pork,citrus	6	1112	1	Farida	22	Cairo	Credit
38	meat,hard	3	143	11	Hanan	22	Fayoum	Cash
39	other vege	4	591	11	Hanan	22	Fayoum	Credit
40	grapes,roo	2	1798	1	Farida	22	Cairo	Credit
41	hamburger	3	923	11	Hanan	22	Fayoum	Cash
42	yogurt	1	1761	11	Hanan	22	Fayoum	Cash
43	bottled wa	1	1098	1	Farida	22	Cairo	Credit
44	rolls/buns,	2	107	1	Farida	22	Cairo	Credit
45	citrus fruit	4	1396	1	Farida	22	Cairo	Credit
46	berries,cof	2	1274	11	Hanan	22	Fayoum	Credit
47	ham,citrus	8	1893	11	Hanan	22	Fayoum	Cash
48	onions,but	11	400	11	Hanan	22	Fayoum	Credit
49	hamburger	7	1985	11	Hanan	22	Fayoum	Credit
50	sausage,liv	7	674	1	Farida	22	Cairo	Credit
51	hard chees	3	1819	1	Farida	22	Cairo	Credit
52	hamburger	1	1404	11	Hanan	22	Fayoum	Credit
53	meat,othe	4	1385	11	Hanan	22	Fayoum	Cash
54	bottled wa	1	2445	11	Hanan	22	Fayoum	Cash
55	domestic e	4	1562	11	Hanan	22	Fayoum	Cash
56	tropical fru	3	2455	1	Farida	22	Cairo	Credit
57	soda	1	1126	1	Farida	22	Cairo	Cash
58	tropical fru	8	1050	1	Farida	22	Cairo	Credit
59	pork,onion	8	1076	11	Hanan	22	Fayoum	Credit
60	hamburger	5	2487	11	Hanan	22	Fayoum	Credit
61	white wine	2	1318	11	Hanan	22	Fayoum	Credit
62	soda,misc.	2	373	1	Farida	22	Cairo	Cash
63	domestic e	1	1143	1	Farida	22	Cairo	Cash
64	bottled be	1	430	1	Farida	22	Cairo	Credit

1. **data_c2 <- na.omit(data_c1)**:
    - This line removes any rows with missing values from data_c1 and assigns the cleaned data to data_c2.
2. **boxplot(data_c2[, 2:4])**:
    - This line generates a boxplot for the columns 2, 3, and 4 in data_c2. Boxplots help visualize the distribution of data and identify outliers.
3. **boxplot(data_c2[, 6])**:
    - This line generates a boxplot for the 6th column in data_c2.
4. **Converting data types for correct analysis:**
    - **data_c2$count <- as.integer(data_c2$count)**: Converts the count column to integer.
    - **data_c2$total <- as.integer(data_c2$total)**: Converts the total column to integer.
    - **data_c2$rnd <- as.integer(data_c2$rnd)**: Converts the rnd column to integer.
    - **data_c2$age <- as.integer(data_c2$age)**: Converts the age column to integer.
    - **data_c2$items <- as.character(data_c2$items)**: Converts the items column to character.
    - **data_c2$customer <- as.character(data_c2$customer)**: Converts the customer column to character.
    - **data_c2$city <- as.character(data_c2$city)**: Converts the city column to character.
    - **data_c2$paymentType <- as.character(data_c2$paymentType)**: Converts the paymentType column to character.
5. **write.csv(data_c2, "cleaned6_grc.csv")**:
    - This line saves the cleaned and processed data in data_c2 to a CSV file named "cleaned6_grc.csv".

Overall, the script cleans the data by removing rows with missing values, converts the data types of specific columns for accurate analysis, generates boxplots to identify outliers, and finally saves the cleaned data to a new CSV file.

..Description of the R Code:

1. **Removing Missing Values:**

R

```
data_c2 <- na.omit(data_c1)
```

- **This line creates a new dataset data_c2 by removing all rows with missing values from the dataset data_c1.**

2. **Boxplot for Outliers:**

R

```
boxplot(data_c2[, 2:4]) boxplot(data_c2[, 6])
```

- **These lines generate boxplots for the specified columns in data_c2.**
- **data_c2[, 2:4] creates a boxplot for columns 2, 3, and 4.**
- **data_c2[, 6] creates a boxplot for column 6.**
- **Boxplots are useful for identifying outliers in the data.**

3. **Correcting Data Types:**

R

```
data_c2$count <- as.integer(data_c2$count) data_c2$total <- as.integer(data_c2$total) data_c2$rnd <- as.integer(data_c2$rnd) data_c2$age <- as.integer(data_c2$age) data_c2$items <- as.character(data_c2$items) data_c2$customer <- as.character(data_c2$customer) data_c2$city <- as.character(data_c2$city) data_c2$paymentType <- as.character(data_c2$paymentType)
```

- **These lines convert the data types of specific columns to the appropriate types for analysis:**
  - **Converts count, total, rnd, and age columns to integer.**
  - **Converts items, customer, city, and paymentType columns to character.**

4. **Saving the Clean Data:**

R

```
write.csv(data_c2, "cleaned6_grc.csv")
```

- **This line saves the cleaned dataset data_c2 to a CSV file named "cleaned6_grc.csv".**

Summary:

The script performs data cleaning and type conversion on a dataset. It first removes any rows with missing values, then generates boxplots for specific columns to identify outliers. After that, it ensures that each column has the correct data type. Finally, the cleaned data is saved to a new CSV file.

..R Script (project.R):

1.Reading CSV Files:

R

```
data <- read.csv("C:/Users/MOHAMED REDA/Downloads/grc.csv") data <- read.csv("C:/Users/MOHAMED REDA/Downloads/grc.CSV")
```

- These lines read the CSV file "grc.csv" from the specified directory into the data frame data.

2.Functions and Data Cleaning:

R

```
sum(duplicated('data')) data_c1 <- unique(data) sum(is.na(data_c1)) data_c2 <- na.omit(data_c1)
```

- **sum(duplicated('data'))**: Calculates the number of duplicate rows in data.
- **data_c1 <- unique(data)**: Removes duplicate rows from data and stores the result in data_c1.
- **sum(is.na(data_c1))**: Calculates the number of missing values in data_c1.
- **data_c2 <- na.omit(data_c1)**: Removes rows with missing values from data_c1 and stores the result in data_c2.

3.Boxplot for Outliers:

R

```
boxplot(data_c2[, 2:4]) boxplot(data_c2[, 6])
```

- These lines generate boxplots for specific columns in data_c2 to visualize the distribution and identify any outliers.
- **boxplot(data_c2[, 2:4])**: Creates boxplots for columns 2 to 4.
- **boxplot(data_c2[, 6])**: Creates a boxplot for column 6.

4.Correcting Data Types (not visible in full but inferred from previous images):

- The code that follows (not fully visible here) likely converts columns to appropriate data types for analysis.

..Console Output:

- **Reading CSV and displaying content:**

R

> read.csv("C:/Users/MOHAMED REDA/Downloads/grc.csv")

  - This line in the console shows the content of the CSV file, which includes various food items listed by their categories.

Environment Panel:

- Displays three data frames:
  - **data**: Contains the raw data from the CSV file with 9835 observations of 8 variables.
  - **data_c1**: The data after removing duplicate rows, with 9834 observations of 8 variables.
  - **data_c2**: The data after removing rows with missing values, with 9833 observations of 8 variables.

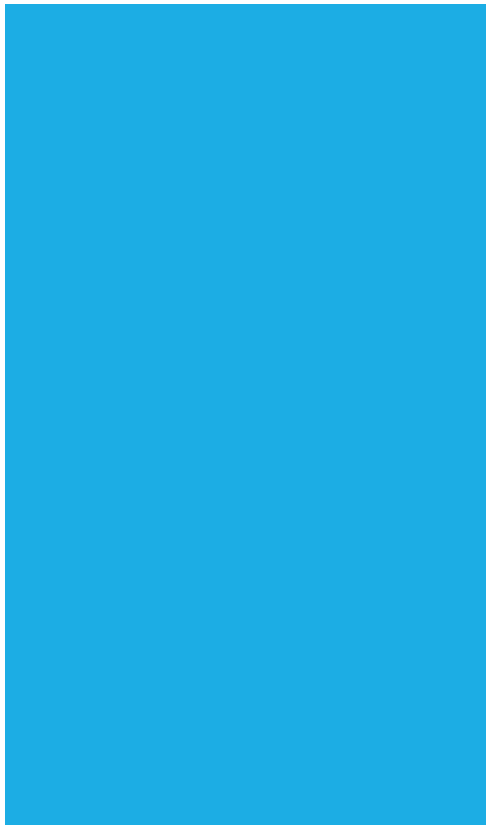
Plot Panel:

- Shows a boxplot for columns 2 to 4 of data_c2, with labels count, total, and rnd. This boxplot helps identify outliers and understand the distribution of these columns.

Summary:

- The script reads a CSV file, removes duplicates and missing values, and generates boxplots to visualize the data. It also prepares the data for further analysis by correcting data types (inferred from previous context). The environment panel shows the progression of data cleaning, and the plot panel displays the visual representation of data distributions.

```
# Load the necessary packages
library(arules)
library(readxl)

# Load the data from an Excel file
data <- read_excel("path_to_your_file.xlsx")

# Split the items in the 'items' column into lists
item_list <- strsplit(as.character(data$items), ",")

# Convert the lists into a 'transactions' object
transactions <- as(item_list, "transactions")

# Apply the apriori algorithm
rules <- apriori(transactions, parameter = list(supp = 0.1, conf = 0.1))

# Display the discovered rules
inspect(rules)

# Plot the frequency of the most common items
itemFrequencyPlot(transactions, topN = 5, type = "relative")
```



Explanation:

1. **Load the necessary packages:**

r

library(arules) library(readxl)

These lines load the arules package for association rule mining and the readxl package for reading Excel files.

2. **Load the data from an Excel file:**

r

data <- read_excel("path_to_your_file.xlsx")

This line reads the data from an Excel file. You need to replace "path_to_your_file.xlsx" with the actual path to your Excel file.

3. **Split the items in the 'items' column into lists:**

r

item_list <- strsplit(as.character(data$items), ",")

This line splits the strings in the 'items' column (where items are assumed to be separated by commas) into lists.

4. **Convert the lists into a 'transactions' object:**

r

transactions <- as(item_list, "transactions")

This line converts the list of items into a format suitable for transaction analysis.

5. **Apply the apriori algorithm:**

r

rules <- apriori(transactions, parameter = list(supp = 0.1, conf = 0.1))

This line runs the Apriori algorithm to find association rules. The supp parameter sets the minimum support, and the conf parameter sets the minimum confidence. Here, both are set to 0.1, meaning 10%.

6. **Display the discovered rules:**

r

inspect(rules)

This line displays the rules discovered by the Apriori algorithm.

7. **Plot the frequency of the most common items:**

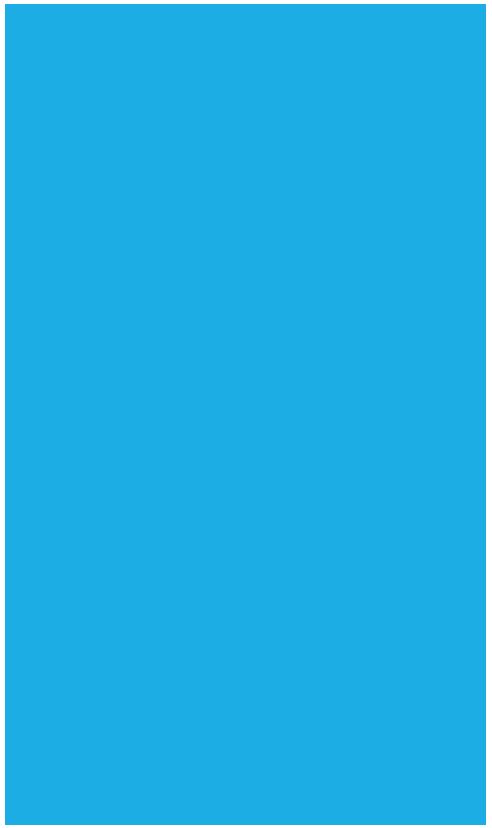
r

itemFrequencyPlot(transactions, topN = 5, type = "relative")

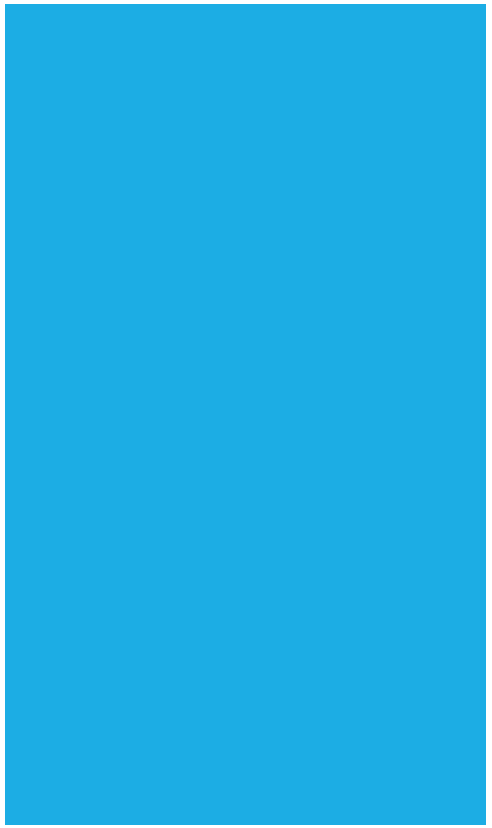
This line plots the relative frequency of the top 5 most common items in the transactions.

This code performs market basket analysis on the items listed in an Excel file, discovering association rules and visualizing the frequency of the most common items.

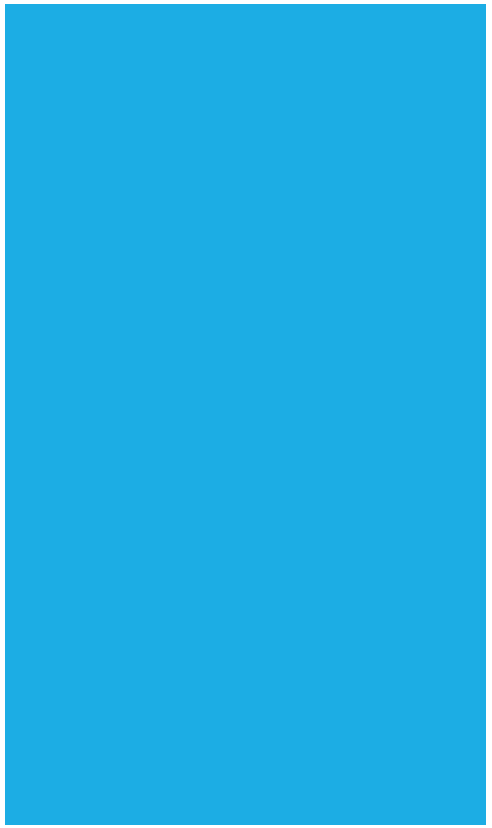
```
# Importing necessary packages

library(stats)
library(readxl)
library(ggplot2)
library(tidyr)
library(dplyr)

# Load the data from the Excel file
data <- read.csv("C:\\programming\\R language\\DS project\\cleaned6_grc.csv")

# Check column names and the number of missing values in each column
print("Column names in the data:")
print(colnames(data))

print("Missing values before cleaning:")
print(sapply(data, function(x) sum(is.na(x))))

# Remove rows with any missing values (NA) in 'age' or 'total'
data_clean <- data %>%
  select(age, total) %>%
  filter(!is.na(age) & !is.na(total))
```



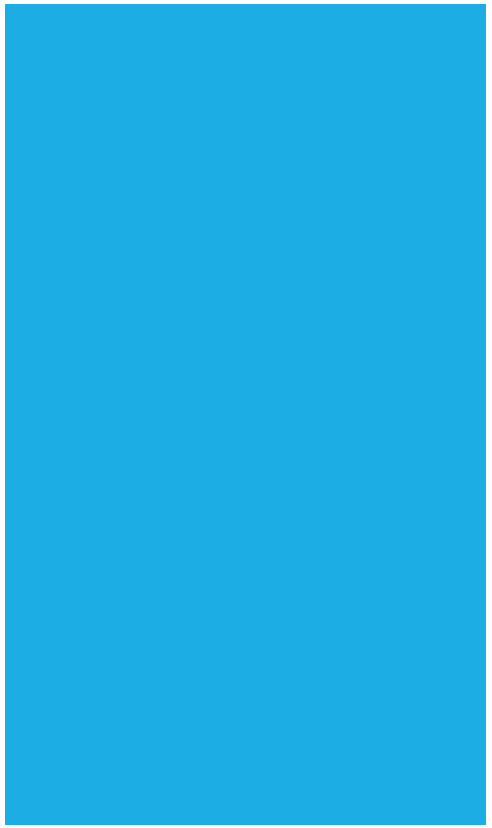
```
# Remove duplicate rows from the cleaned data to prevent any many-to-many relationships
data_clean <- data_clean %>% distinct()

# Verify the cleaned data (show the first few rows)
print("Data after cleaning:")
print(head(data_clean))

# Check for any missing values in the cleaned data
print("Missing values after cleaning:")
print(sapply(data_clean, function(x) sum(is.na(x))))

# Set the number of clusters and a random seed for reproducibility
set.seed(200)
n_clusters <- 3
```



```
# Apply K-means clustering to 'age' and 'total' columns
kmeans_result <- kmeans(data_clean, centers = n_clusters)

# Add the cluster assignment to the cleaned data
data_clean$Cluster <- kmeans_result$cluster

# Merge the cluster assignments back to the original data
data_with_cluster <- left_join(data, data_clean, by = c("age", "total"))

# Check if there are any rows in the original data that were not matched with the cleaned data
unmatched_rows <- data %>% anti_join(data_clean, by = c("age", "total"))
print("Rows in data not matched with data_clean:")
print(unmatched_rows)

# Display the final data with the cluster assignments
print("Final Data with Clusters:")
print(head(data_with_cluster))
```



```
# Check for missing values in the final data after the merge
print("Missing values in the final data:")
print(sapply(data_with_cluster, function(x) sum(is.na(x))))


# Plot the data using ggplot to visualize customer segmentation by cluster
ggplot(data_with_cluster, aes(x = age, y = total, color = factor(Cluster))) +
  geom_point(size = 3) +  # Plotting data points
  labs(
    title = "Customer Segmentation",  # Title of the plot
    x = "Age",  # Label for the x-axis
    y = "Total Spending",  # Label for the y-axis
    color = "Cluster"  # Legend label for cluster colors
  ) +
  theme_minimal()  # Using a minimal theme for the plot

#create a table with customer names, age, total spending, and their assigned cluster number.
customer_table <- data_with_cluster %>%
  select(customer, age, total, Cluster)

# Print the table displaying customer information
print("Customer Table with Name, Age, Total Spending, and Cluster:")
print(customer_table)
```



Explanation:

1. **Importing necessary packages:**

r

library(stats) library(readxl) library(ggplot2) library(tidyr) library(dplyr)

These lines load various packages for statistical analysis, reading Excel files, data visualization, and data manipulation.

2. **Load the data from the CSV file:**

r

data <- read.csv("C:\\programming\\R language\\DS project\\cleaned6_grc.csv")

This line reads the data from a CSV file located at the specified path.

3. **Check column names and the number of missing values:**

r

print("Column names in the data:") print(colnames(data)) print("Missing values before cleaning:") print(sapply(data, function(x) sum(is.na(x))))

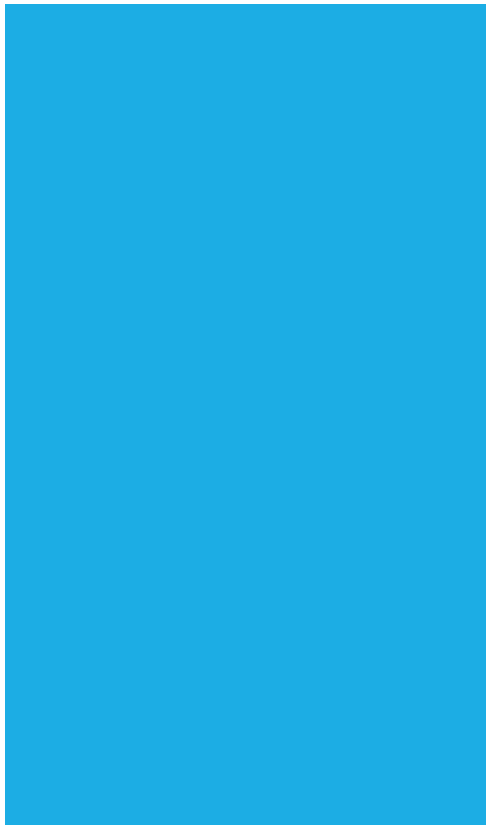
These lines print the column names and count the number of missing values (NA) in each column.

4. **Remove rows with any missing values (NA) in 'age' or 'total':**

r

data_clean <- data %>% select(age, total) %>% filter(!is.na(age) & !is.na(total))

This block selects the 'age' and 'total' columns and removes rows with missing values in these columns.



5. **Remove duplicate rows:**

r

data_clean <- data_clean %>% distinct()

This line removes duplicate rows from the cleaned data.

6. **Verify the cleaned data:**

r

print("Data after cleaning:") print(head(data_clean)) print("Missing values after cleaning:") print(sapply(data_clean, function(x) sum(is.na(x))))

These lines print the first few rows of the cleaned data and check for any remaining missing values.

7. **Set the number of clusters and a random seed for reproducibility:**

r

set.seed(200) n_clusters <- 3

This block sets the random seed for reproducibility and specifies the number of clusters for K-means clustering.

8. **Apply K-means clustering:**

r

kmeans_result <- kmeans(data_clean, centers = n_clusters)

10-Merge the cluster assignments back to the original data:

data_with_cluster <- left_join(data, data_clean, by = c("age", "total"))

This line merges the cluster assignments back to the original data based on the 'age' and 'total' columns.

11-Check for unmatched rows:

r

unmatched_rows <- data %>% anti_join(data_clean, by = c("age", "total")) print("Rows in data not matched with data_clean:") print(unmatched_rows)

These lines check if there are any rows in the original data that were not matched with the cleaned data.

12-Display the final data with the cluster assignments:

r

print("Final Data with Clusters:") print(head(data_with_cluster)) print("Missing values in the final data:") print(sapply(data_with_cluster, function(x) sum(is.na(x))))

These lines print the first few rows of the final data and check for missing values after the merge.

13-Plot the data using ggplot:

r

ggplot(data_with_cluster, aes(x = age, y = total, color = factor(Cluster))) + geom_point(size = 3) + labs(title = "Customer Segmentation", x = "Age", y = "Total Spending", color = "Cluster") + theme_minimal()

This block uses ggplot2 to create a scatter plot of 'age' vs. 'total' spending, colored by cluster. The plot helps visualize customer segmentation.

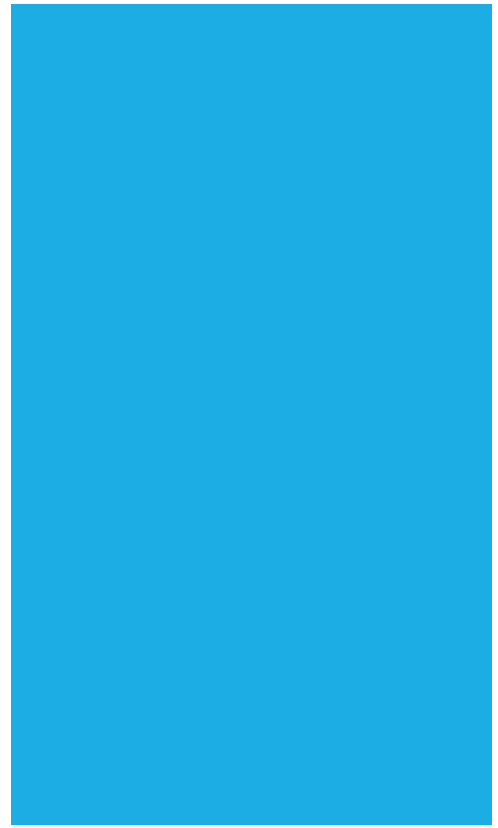
14-Create a table with customer information and cluster assignments:

r

customer_table <- data_with_cluster %>% select(customer, age, total, Cluster) print("Customer Table with Name, Age, Total Spending, and Cluster:") print(customer_table)

This block creates a table with customer names, ages, total spending, and their assigned cluster. It then prints the table.

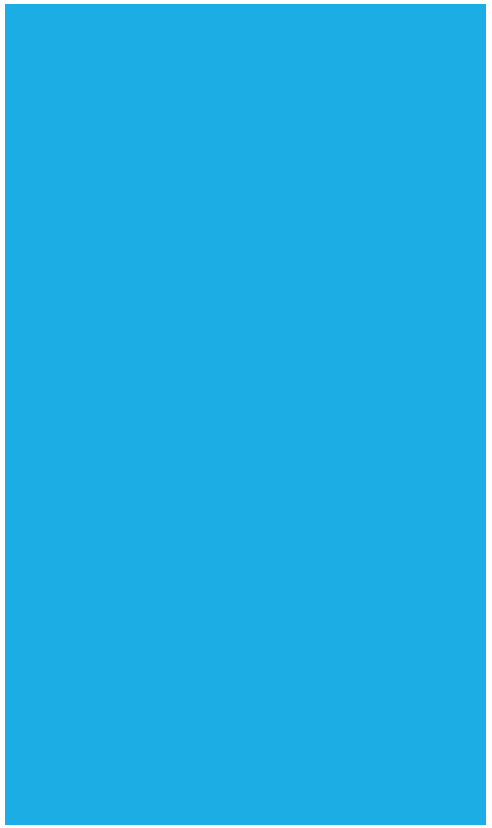
This script reads customer data, cleans it, performs K-means clustering to segment customers, and visualizes the segmentation. It also generates a table of customer information with cluster assignments.



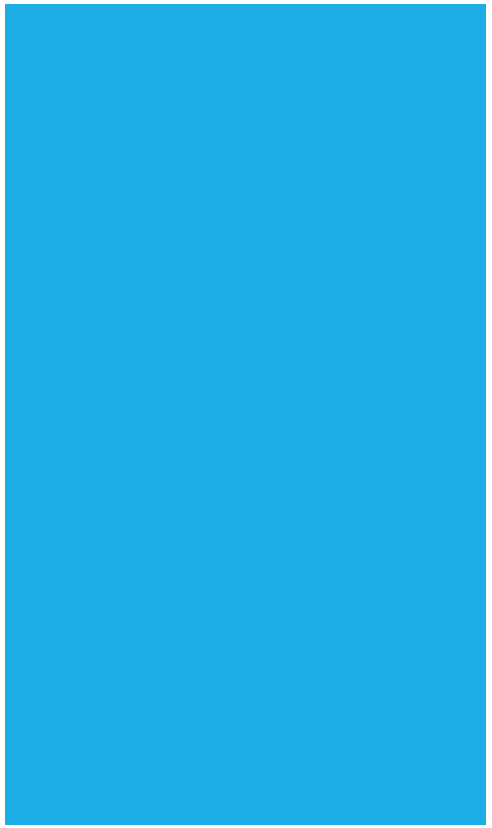
items	count	total	rnd	customer	age	city	paymentTy
citrus fruit,	4	1612	9	Maged	60	Hurghada	Cash
tropical fru	3	509	12	Eman	23	Aswan	Cash
whole milk	1	2084	8	Rania	37	Dakahlia	Cash
pip fruit,yo	4	788	8	Rania	37	Dakahlia	Cash
other vege	4	1182	14	Magdy	36	Sohag	Cash
whole milk	5	1771	3	Ahmed	30	Giza	Credit
rolls/buns	1	2196	7	Huda	39	Gharbia	Cash
other vege	5	1657	6	Walaa	29	Cairo	Cash
pot plants	1	2373	2	Mohamed	25	Alexandria	Credit
whole milk	2	343	5	Shimaa	55	Port Said	Cash
tropical fru	5	1381	2	Mohamed	25	Alexandria	Cash
citrus fruit,	9	1965	1	Farida	22	Cairo	Credit
beef	1	784	11	Hanan	22	Fayoum	Cash
frankfurter	3	1001	7	Huda	39	Gharbia	Credit
chicken,tro	2	1579	13	Sayed	37	Cairo	Credit
butter,suga	4	585	8	Rania	37	Dakahlia	Credit
fruit/veget	1	184	5	Shimaa	55	Port Said	Cash
packaged f	1	1737	12	Eman	23	Aswan	Cash
chocolate	1	184	6	Walaa	29	Cairo	Cash
specialty b	1	469	7	Huda	39	Gharbia	Cash
other vege	1	408	5	Shimaa	55	Port Said	Cash
butter milk	2	2252	7	Huda	39	Gharbia	Credit
whole milk	1	1538	3	Ahmed	30	Giza	Cash
tropical fru	5	1215	9	Maged	60	Hurghada	Credit
tropical fru	11	1762	4	Adel	50	Alexandria	Credit
bottled wa	2	2384	15	Sameh	35	Hurghada	Cash
yogurt	1	599	1	Farida	22	Cairo	Cash
sausage,ro	4	2360	10	Samy	55	Alexandria	Credit
other vege	1	1906	1	Farida	22	Cairo	Credit
brown bre	6	391	13	Sayed	37	Cairo	Cash
yogurt,bev	4	556	3	Ahmed	30	Giza	Cash
hamburger	7	1821	5	Shimaa	55	Port Said	Cash
root veget	5	676	3	Ahmed	30	Giza	Credit
pork,berrie	8	988	8	Rania	37	Dakahlia	Credit
beef,grape	3	148	13	Sayed	37	Cairo	Credit
pastry,sod	2	1009	8	Rania	37	Dakahlia	Credit
fruit/veget	1	1760	14	Magdy	36	Sohag	Cash
canned be	1	1465	6	Walaa	29	Cairo	Credit
root veget	4	2374	6	Walaa	29	Cairo	Credit
citrus fruit,	3	416	4	Adel	50	Alexandria	Cash
sausage,ro	6	2117	10	Samy	55	Alexandria	Credit
tropical fru	13	1287	13	Sayed	37	Cairo	Cash
berries,yog	2	443	15	Sameh	35	Hurghada	Cash
canned be	1	1225	15	Sameh	35	Hurghada	Credit
butter milk	8	1413	8	Rania	37	Dakahlia	Credit
coffee	1	1983	2	Mohamed	25	Alexandria	Cash
pastry,bott	2	730	1	Farida	22	Cairo	Credit
rolls/buns	1	2430	10	Samy	55	Alexandria	Credit
misc. beve	1	194	12	Eman	23	Aswan	Credit
root veget	10	2232	5	Shimaa	55	Port Said	Credit
sausage,ro	4	2276	9	Maged	60	Hurghada	Cash
canned be	1	569	15	Sameh	35	Hurghada	Credit
ham,grape	4	1597	10	Samy	55	Alexandria	Cash
turkey,trop	10	433	5	Shimaa	55	Port Said	Cash
whole milk	5	114	9	Maged	60	Hurghada	Cash
whole milk	4	1026	2	Mohamed	25	Alexandria	Credit
packaged f	3	2236	11	Hanan	22	Fayoum	Cash
rolls/buns,	7	1372	5	Shimaa	55	Port Said	Credit
ham,beef,v	6	384	4	Adel	50	Alexandria	Cash
rolls/buns,	3	1743	8	Rania	37	Dakahlia	Cash
other vege	7	2146	11	Hanan	22	Fayoum	Credit
sausage,pa	2	912	12	Eman	23	Aswan	Cash
sausage,be	3	222	4	Adel	50	Alexandria	Credit
frankfurter	5	688	14	Magdy	36	Sohag	Cash

```
data <- read.csv("C:\\programming\\R language\\DS project\\grc.csv" , header = TRUE)
# i. Compare cash and credit totals.

# extract the frequency of type method => the number of credit , the number of cash
paymentsmethods <- table(data$paymentType)
# the lable of the two paymnets methods
lable = c("cash" , "credit")

# the ratio of the payments method the method type  divided by the (sum of methods table count) *100
ratio <-  round(    paymentsmethods/sum(paymentsmethods)*100 , 1)

# creating the pie chart
pie(paymentsmethods,labels = paste(names(paymentsmethods) , ratio , "%") , main = "cash vs credit" , col = c("navy", "orange"))
# ii. Compare each age and sum of total spending.

# the sum of the total and ages
ageByTotall <- tapply(data$total , data$age , sum)

sort(table(data$customer), decreasing = TRUE)
# sorting them
sortetTotal <- sort(ageByTotall ,decreasing = TRUE)

# bar chart
```

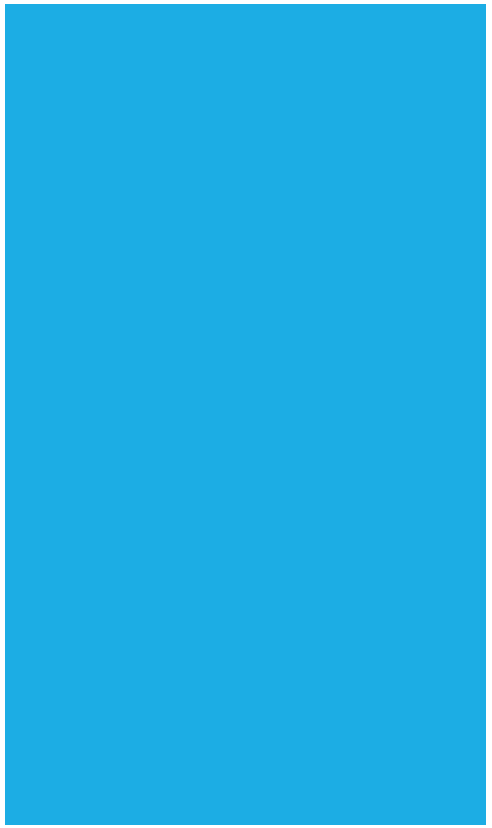
```
barplot(
  sortetTotal , xlab  = "age" , ylab = "total spending" , col = c("navy" , "orange")
)


# iii. Show each city total spending and arrange it by total descending.


# extracting the cities and how much is the total payment for each city
# alex => ????
# cairo => ???
# awsan => ???

totalCitySpending <- tapply(data$total, data$city, sum)

sortedCities <- sort(totalCitySpending, decreasing = TRUE)

barplot(sortedCities,
        xlab = "Cities",
        ylab = "Total Spending",
        main = "Total Spending per City",
        las = 2,
        col = rainbow(length(sortedCities)))

hist(data$total,
     main = "Distribution of Total Spending",
     xlab = "Total Spending",
```



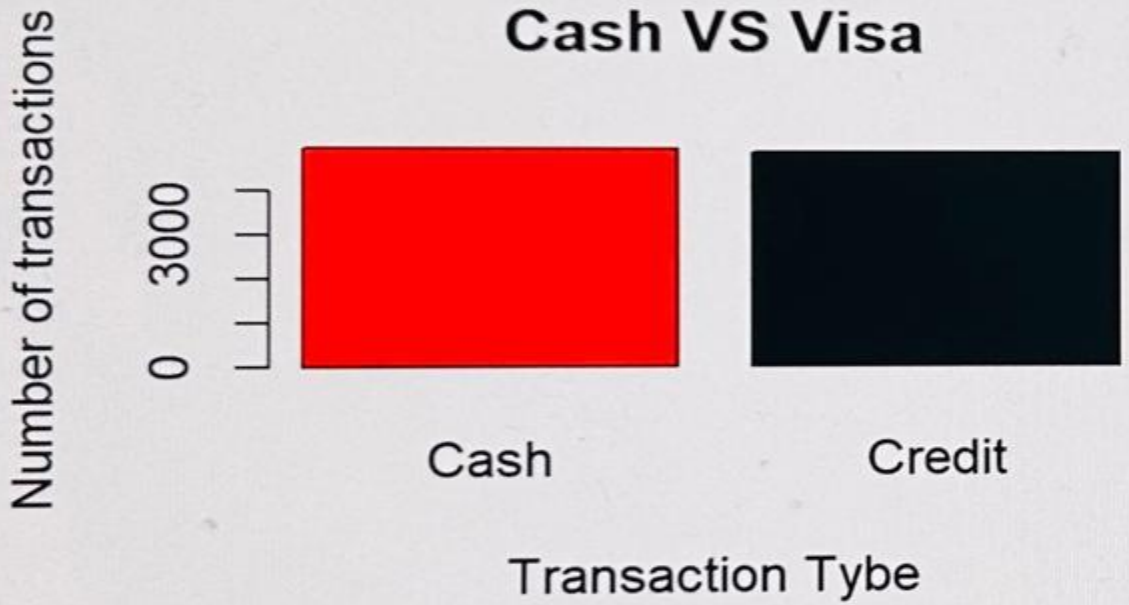
```
col = "orange",
     border = "black")



# another way
plot(
  sortetTotal,
  xlab = "Age",
  ylab = "Total Spending",
  pch = 12,  # pattern shape
 col = "navy",
  main = "Total Spending by Age"
)



ageByTotal2 <- tapply(data$total , data$age , sum)

# Total ranking from highest to lowest
sortedTotal <- sort(ageByTotal2 , decreasing = FALSE)
```

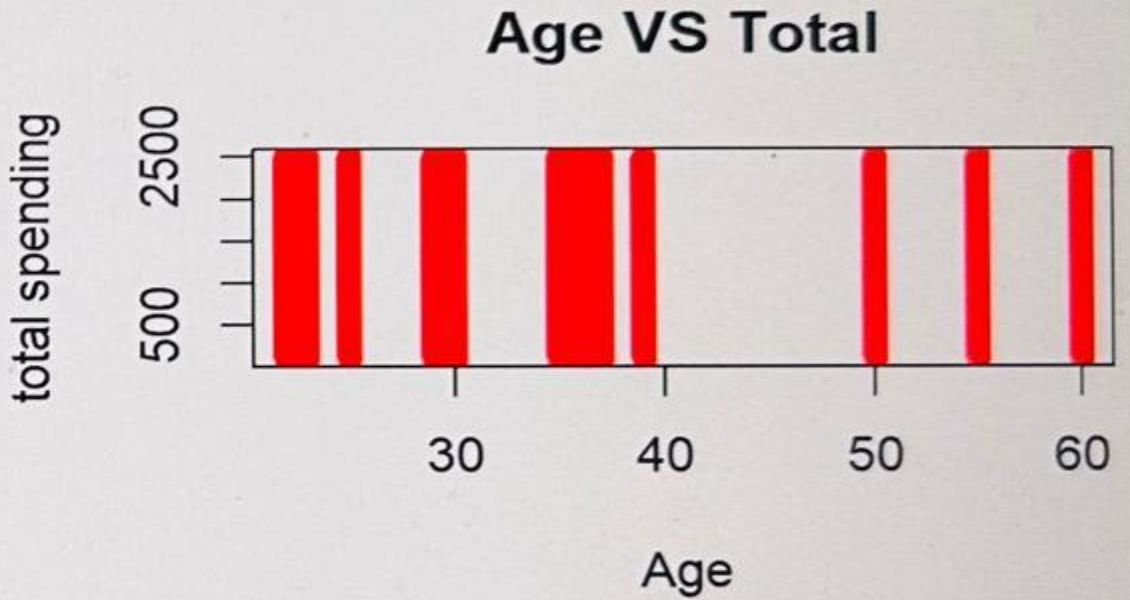


```
#graph
dotchart(sortetTotal, main = "Age vs Total Spending",
         xlab = "Total Spending", ylab = "Age",
         col = "navy")
boxplot(data$total,
        main = "Distribution of Total Spending",

        xlab = "Total Spending",

        col = "navy")
customers = table(data$customer)
summ  = sum(customers)

print(summ)

library(arules)

trans = data$items
x <- apriori(trans , parameter = )
```

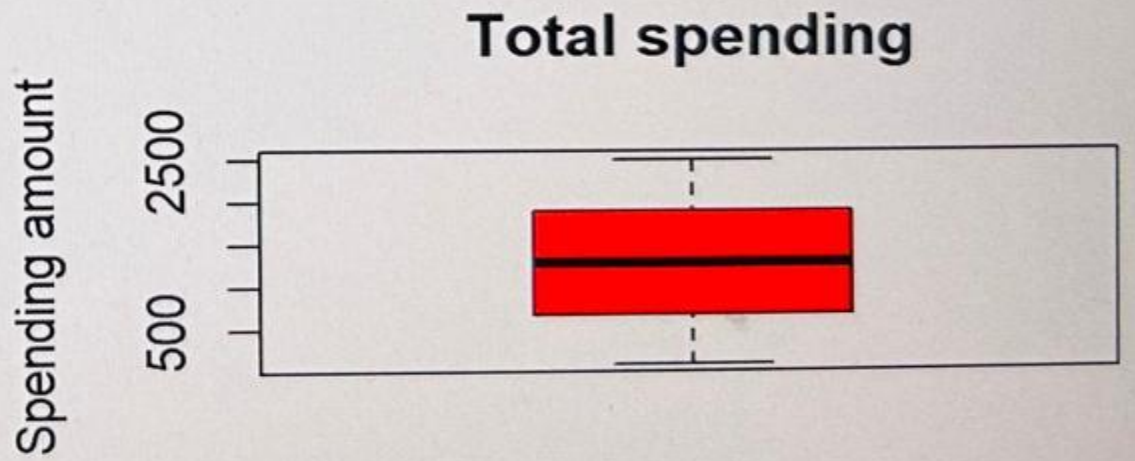


```
# Load the data
project_data <- read.csv("C:/Users/Mega Store/Desktop/project1.csv")
# Install the ggplot2 package if not already installed
install.packages("ggplot2") library(ggplot2)
# Set the layout for the plots
 par(mfrow=c(2,2)) # Compare cash and credit sales transaction <-
table(project_data$paymentType) barplot(transaction, main = "Cash VS Visa", col = c("red",
"black"), xlab = "Transaction Type", ylab = "Number of Transactions")
# Compare customer ages with total spending
plot(project_data$age, project_data$total, main = "Age VS Total", col = "red", xlab = "Age",
ylab = "Total Spending")
 # Calculate average age and spending
mean_age <- mean(project_data$age) mean_spending <- mean(project_data$total)
# Rank cities by total spending
 total_spending <- aggregate(total ~ city, data = project_data, sum) sorted_data <-
total_spending[order(-total_spending$total),] barplot(sorted_data$total, names.arg =
sorted_data$city, main = "Total Spending by City", col = "red", xlab = "City", ylab = "Total
Spending")
 # Display the distribution of total spending
boxplot(project_data$total, main = "Total Spending", ylab = "Spending Amount", col = "red")
```


Cash VS Visa
Number of transactions
0
3000
Cash
Credit
Transaction Tybe


Age VS Total
total spending
500
2500
30
40
50
60
Age


Total spending by city
Total spending
0
2000000
Alexandria
Sohag
Giza
Aswan
City


Total spending
Spending amount
500
2500