RETAIL MARKETING ANALYTICS

R.D & S.H National College & S.W.A Science College

Bandra, Mumbai – 400050

DEPARTMENT OF COMPUTER SCIENCE

M.Sc. Data Science (Part I) – **Semester I**

Course Code: **PSDS506b**

Course Name: **Retail Marketing Analytics**

Practical Journal

2023-2024

Seat No: ____________________

R.D. & S. H. National College & S. W.A. Science College
Bandra, Mumbai – 400050.

Department of Computer Science

CERTIFICATE

This is to certify that ~~Mr.~~/Ms. TANISHQA CHANDRASHEKHAR TAMBE of M.SC. DATA SCIENCE class (FIRST Semester) has satisfactorily completed. 8 Practicals, in the subject of RETAIL MARKETING ANALYTICS as a part of M.Sc. in Computer Science Program during the academic year 2023 – 2024.

Date of Certification:

Subject Incharge,

Co-ordinator,

Department Computer Science

Signature of Examiner

INDEX

Practical 1: Tabulate And Summarize Marketing Data

Learn how to tabulate and summarize marketing data using R.

- Clean and preprocess the marketing data.
- Generate a simple histogram plot to visualize data distribution.
- Use tabulation and summary functions to gain insights from the data.
- Interpret the findings and discuss the implications for marketing analysis.

Code:

```
#Step 1: Loading and Exploring the Data# Load necessary libraries library(tidyverse)
# Load the dataset
superstore_data <- read.csv("D:/MSc DS/Semester 1/Retail Market Analysis/Practical/superstore_data.csv")

#Step 2: Cleaning and Preprocessing
# Check for missing values
missing_values <- colSums(is.na(superstore_data))
# Replace missing values with the mean of each columnfor (col in names(superstore_data)) {
if (sum(is.na(superstore_data[[col]])) > 0 & is.numeric(superstore_data[[col]])) {
superstore_data[[col]][is.na(superstore_data[[col]])] <- mean(superstore_data[[col]], na.rm = TRUE)
}}
# Remove duplicates or irrelevant columns if necessary.
selected_data <- superstore_data[, c("Year_Birth", "Marital_Status", "Education", "Dt_Customer", "Recency",
"NumStorePurchases", "NumWebPurchases", "NumWebVisitsMonth")]
selected_data <- unique(selected_data)
#Convert data types if needed (e.g., converting a column to the correct date format). selected_data$Dt_Customer <-
as.Date(selected_data$Dt_Customer, format = "%Y-%m-%d")selected_data$Year_Birth <-
as.numeric(as.character(selected_data$Year_Birth))

#Step 3: Visualization with Histogram Plot
# Simple histogram plot for a numerical variable (e.g., 'age')
selected_data$Age <- as.integer(format(Sys.Date(), "%Y")) - selected_data$Year_Birthhist(selected_data$Age,
main = "Histogram of Age",xlab = "Age",
col = "salmon")

#Step 4: Tabulation and Summary# Summary statistics summary(superstore_data)
# Tabulation for categorical variables table(selected_data$NumWebPurchases)
table(selected_data$NumStorePurchases)
```

Output:

```
>    #Step 1: Loading and Exploring the Data
>    # Load necessary libraries
>    library(tidyverse)
>    # Load the dataset
>    superstore_data <- read.csv("D:/MSc DS/Semester 1/Retail Market Analysis/Practical/superstore_data.csv")
>
>
>    #Step 2: Cleaning and Preprocessing
>    # Check for missing values
>    missing_values <- colSums(is.na(superstore_data))
>
>    # Replace missing values with the mean of each column
>    for (col in names(superstore_data)) {
+   if (sum(is.na(superstore_data[[col]])) > 0 & is.numeric(superstore_data[[col]])) {
+       superstore_data[[col]][is.na(superstore_data[[col]])] <- mean(superstore_data[[col
], na.rm = TRUE)
+   }}
>
>    # Remove duplicates or irrelevant columns if necessary.
>    selected_data <- superstore_data[, c("Year_Birth", "Marital_Status", "Education", "Dt_Customer", "Recency",
"NumStorePurchases", "NumWebPurchases", "NumWebVisitsMonth")]
>    selected_data <- unique(selected_data)
>
>    #Convert data types if needed (e.g., converting a column to the correct date format).
>    selected_data$Dt_Customer <- as.Date(selected_data$Dt_Customer, format = "%Y-%m-%d")
>    selected_data$Year_Birth <- as.numeric(as.character(selected_data$Year_Birth))
>
>    #Step 3: Visualization with Histogram Plot
>    # Simple histogram plot for a numerical variable (e.g., 'age')
>    hist(selected_data$Age,
+         main = "Histogram of Age",
+         xlab = "Age",
+         col = "salmon")
Error in hist.default(selected_data$Age, main = "Histogram of Age", xlab = "Age", :'x' must be numeric
>    #Step 1: Loading and Exploring the Data
>    # Load necessary libraries
>    library(tidyverse)
>    # Load the dataset
>    superstore_data <- read.csv("D:/MSc DS/Semester 1/Retail Market Analysis/Practical/superstore_data.csv")
>
>
>    #Step 2: Cleaning and Preprocessing
>    # Check for missing values
>    missing_values <- colSums(is.na(superstore_data))
>
>    # Replace missing values with the mean of each column
>    for (col in names(superstore_data)) {
+   if (sum(is.na(superstore_data[[col]])) > 0 & is.numeric(superstore_data[[col]])) {
+       superstore_data[[col]][is.na(superstore_data[[col]])] <- mean(superstore_data[[col
], na.rm = TRUE)
+   }}
>
>    # Remove duplicates or irrelevant columns if necessary.
>    selected_data <- superstore_data[, c("Year_Birth", "Marital_Status", "Education", "Dt_Customer", "Recency",
"NumStorePurchases", "NumWebPurchases", "NumWebVisitsMonth")]
>    selected_data <- unique(selected_data)
>
>    #Convert data types if needed (e.g., converting a column to the correct date format).
>    selected_data$Dt_Customer <- as.Date(selected_data$Dt_Customer, format = "%Y-%m-%d")
>    selected_data$Year_Birth <- as.numeric(as.character(selected_data$Year_Birth))
>
>    #Step 3: Visualization with Histogram Plot
>    # Simple histogram plot for a numerical variable (e.g., 'age')
>    selected_data$Age <- current_year - selected_data$Year_BirthError: object 'current_year' not found
>    #Step 1: Loading and Exploring the Data
>    # Load necessary libraries
>    library(tidyverse)
>    # Load the dataset
>    superstore_data <- read.csv("D:/MSc DS/Semester 1/Retail Market Analysis/Practical/superstore_data.csv")
>    #Step 2: Cleaning and Preprocessing
>    # Check for missing values
>    missing_values <- colSums(is.na(superstore_data))
>
>    # Replace missing values with the mean of each column
>    for (col in names(superstore_data)) {
+   if (sum(is.na(superstore_data[[col]])) > 0 & is.numeric(superstore_data[[col]])) {
+        superstore_data[[col]][is.na(superstore_data[[col]])] <- mean(superstore_data[[col
], na.rm = TRUE)
+    }}
>
>    # Remove duplicates or irrelevant columns if necessary.
>    selected_data <- superstore_data[, c("Year_Birth", "Marital_Status", "Education", "Dt_Customer", "Recency",
"NumStorePurchases", "NumWebPurchases", "NumWebVisitsMonth")]
>    selected_data <- unique(selected_data)
>
>    #Convert data types if needed (e.g., converting a column to the correct date format).
>    selected_data$Dt_Customer <- as.Date(selected_data$Dt_Customer, format = "%Y-%m-%d")
>    selected_data$Year_Birth <- as.numeric(as.character(selected_data$Year_Birth))
>
>    #Step 3: Visualization with Histogram Plot
>    # Simple histogram plot for a numerical variable (e.g., 'age')
>    selected_data$Age <- as.integer(format(Sys.Date(), "%Y")) - selected_data$Year_Birth
>    hist(selected_data$Age,
+       main = "Histogram of Age",
+       xlab = "Age",
+       col = "salmon")
>
> #Step 4: Tabulation and Summary

> # Summary statistics
> summary(superstore_data)
                                               Education          Marital_Status         Income


       Id          Year_Birth
 Min.   :    0   Min.   :1893   Length:2240        Length:2240        Min.   :  1730
 1st Qu.: 2828   1st Qu.:1959   Class :character   Class :character   1st Qu.: 35539
 Median : 5458   Median :1970   Mode  :character   Mode  :character   Median : 51742
 Mean   : 5592   Mean   :1969                                         Mean   : 52247
 3rd Qu.: 8428   3rd Qu.:1977                                         3rd Qu.: 68290
 Max.   :11191   Max.   :1996                                         Max.   :666666
    Kidhome          Teenhome         Dt_Customer           Recency         MntWines
 Min.   :0.0000   Min.   :0.0000   Length:2240        Min.   : 0.00   Min.   :   0.00
 1st Qu.:0.0000   1st Qu.:0.0000   Class :character   1st Qu.:24.00   1st Qu.:  23.75
 Median :0.0000   Median :0.0000   Mode  :character   Median :49.00   Median : 173.50
 Mean   :0.4442   Mean   :0.5062                      Mean   :49.11   Mean   : 303.94
 3rd Qu.:1.0000   3rd Qu.:1.0000                      3rd Qu.:74.00   3rd Qu.: 504.25
 Max.   :2.0000   Max.   :2.0000                      Max.   :99.00   Max.   :1493.00
   MntFruits      MntMeatProducts   MntFishProducts  MntSweetProducts  MntGoldProds
 Min.   :  0.0   Min.   :   0.0   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
 1st Qu.:  1.0   1st Qu.:  16.0   1st Qu.:  3.00   1st Qu.:  1.00   1st Qu.:  9.00
 Median :  8.0   Median :  67.0   Median : 12.00   Median :  8.00   Median : 24.00
 Mean   : 26.3   Mean   : 166.9   Mean   : 37.53   Mean   : 27.06   Mean   : 44.02
 3rd Qu.: 33.0   3rd Qu.: 232.0   3rd Qu.: 50.00   3rd Qu.: 33.00   3rd Qu.: 56.00
```

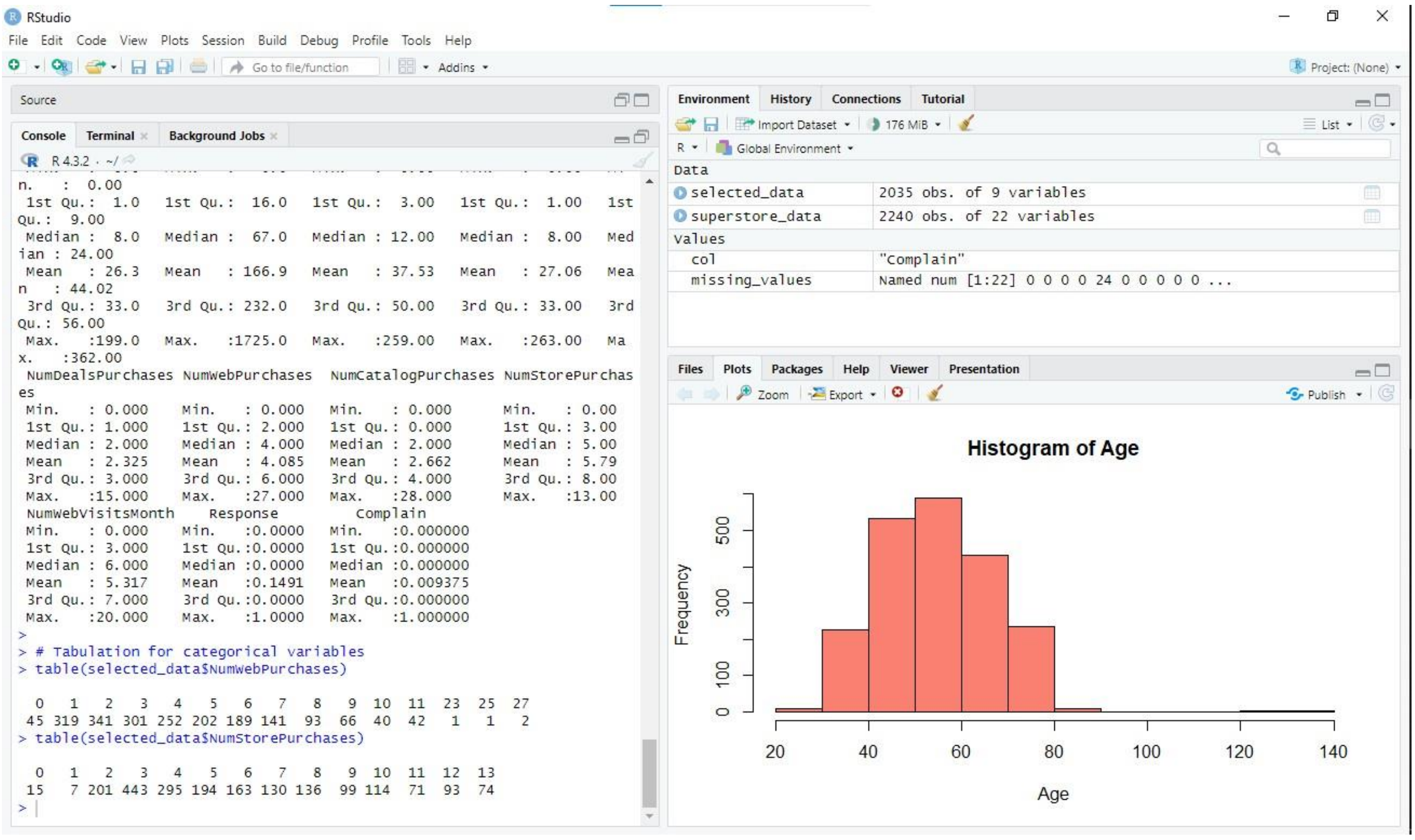
```
1st Qu.: 1.000        1st Qu.: 2.000       1st Qu.: 0.000         1st Qu.: 3.00
Median : 2.000          Median :           Median : 2.000         Median : 5.00
                         4.000
Mean  : 2.325         Mean : 4.085         Mean    : 2.662          Mean : 5.79
3rd Qu.: 3.000        3rd Qu.: 6.000       3rd Qu.: 4.000         3rd Qu.: 8.00
Max.  :15.000         Max. :27.000         Max.    :28.000          Max. :13.00
NumWebVisitsM           Response           Complain
onth
Min.   : 0.000        Min.  :0.0000        Min.     :0.000000
1st Qu.: 3.000        1st Qu.:0.0000       1st Qu.:0.000000
Median : 6.000          Median             Median :0.000000
                        :0.0000
Mean  : 5.317         Mean :0.1491         Mean    :0.009375
3rd Qu.: 7.000            3rd              3rd Qu.:0.000000
                        Qu.:0.0000
Max.  :20.000         Max. :1.0000         Max.    :1.000000
```

```
>   # Tabulation for categorical variables
>   table(selected_data$NumWebPurchases)

 0 1    2    3    4    5    6    7        8          9  10  11          2    2
                                                                 23          5    7
     45 319 341 301 252 202 189 141         9      66  40  42    1       1    2
                                            3
>   table(selected_data$NumStorePurchases)



   0          2    3    4    5    6    7    8    9    1    1    1    1
                                                        0    1    2    3
   1          2    4    2    1    1    1    1    9    1    7    9    7
   5          0    4    9    9    6    3    3    9    1    1    3    4
              1    3    5    4    3    0    6         4
```



```
n.    :  0.00
 1st Qu.:  1.0   1st Qu.:  16.0   1st Qu.:  3.00   1st Qu.:  1.00   1st
Qu.:  9.00
 Median :  8.0   Median :  67.0   Median : 12.00   Median :  8.00   Med
ian : 24.00
 Mean   : 26.3   Mean   : 166.9   Mean   : 37.53   Mean   : 27.06   Mea
n   : 44.02
 3rd Qu.: 33.0   3rd Qu.: 232.0   3rd Qu.: 50.00   3rd Qu.: 33.00   3rd
Qu.: 56.00
 Max.   :199.0   Max.   :1725.0   Max.   :259.00   Max.   :263.00   Ma
x.   :362.00
 NumDealsPurchases NumWebPurchases  NumCatalogPurchases NumStorePurchas
es
 Min.   : 0.000    Min.   : 0.000   Min.   : 0.000      Min.   : 0.00
 1st Qu.: 1.000    1st Qu.: 2.000   1st Qu.: 0.000      1st Qu.: 3.00
 Median : 2.000    Median : 4.000   Median : 2.000      Median : 5.00
 Mean   : 2.325    Mean   : 4.085   Mean   : 2.662      Mean   : 5.79
 3rd Qu.: 3.000    3rd Qu.: 6.000   3rd Qu.: 4.000      3rd Qu.: 8.00
 Max.   :15.000    Max.   :27.000   Max.   :28.000      Max.   :13.00
 NumWebVisitsMonth    Response          Complain
 Min.   : 0.000    Min.   :0.0000   Min.   :0.000000
 1st Qu.: 3.000    1st Qu.:0.0000   1st Qu.:0.000000
 Median : 6.000    Median :0.0000   Median :0.000000
 Mean   : 5.317    Mean   :0.1491   Mean   :0.009375
 3rd Qu.: 7.000    3rd Qu.:0.0000   3rd Qu.:0.000000
 Max.   :20.000    Max.   :1.0000   Max.   :1.000000
>
> # Tabulation for categorical variables
> table(selected_data$NumWebPurchases)

  0   1   2   3   4   5   6   7   8   9  10  11  23  25  27
 45 319 341 301 252 202 189 141  93  66  40  42   1   1   2
> table(selected_data$NumStorePurchases)

  0   1   2   3   4   5   6   7   8   9  10  11  12  13
 15   7 201 443 295 194 163 130 136  99 114  71  93  74
> 
```

```
Data
selected_data        2035 obs. of 9 variables
superstore_data      2240 obs. of 22 variables
Values
  col                "Complain"
  missing_values     Named num [1:22] 0 0 0 0 24 0 0 0 0 0 ...
```

Interpretation and Implications:

Histogram Visualization: Analysing the distribution of age provides insight into the target audience's age range, aiding intargeted marketing campaigns. The majority of the target audience falls between 40-60.

Tabulation and Summary: Understanding the distribution of categorical variables like Web purchases/Store Purchases and summary statistics of numerical features such as income, age, etc. Helps identify customer demographics. Insights into instore and web purchase levels provide indications of purchasing power and potential market segments is higher for instore purchase than web purchases

Practical 2: Gain Proficiency in Visualizing Marketing Data

Gain proficiency in visualizing marketing data using R.

- Understand the key elements of data visualization.
- Create various visualizations such as histograms, scatter plots, line plots, and bar charts using the ggplot() function in R.
- Apply appropriate visualization techniques to effectively communicate marketing insights.

Theory:

Key Elements of Data Visualization

Histogram: Use a histogram to visualize the distribution of a single continuous variable. Helpful for understanding the shape, central tendency, and spread of data. Suitable for identifying patterns, skewness, and outliers in data.

Boxplot (Box-and-Whisker plot): Ideal for displaying the distribution of a numerical variable across different categoriesor group. Useful for comparing distributions and identifying outliers or variability between group. Shows key statistics like median, quartiles, and outliers.

Scatter plot: Use a scatter plot to visualize the relationship between two continuous variables Helpful for identifying correlations, trends, clusters, or patterns between variables. Suitable for assessing the strength and direction of relationships between variables.

Code:

#Step 1: Loading and Preparing the Data

```
# Load necessary librarieslibrary (ggplot2)
# Load the dataset
superstore_data <- read.csv("D:/MSc DS/Semester 1/Retail Market Analysis/Practical/superstore_data.csv")
selected_data <- superstore_data[, c("Id","Year_Birth", "Marital_Status", "Education", "Dt_Customer", "Recency", "NumStorePurchases", "NumWebPurchases", "NumWebVisitsMonth")]
selected_data <- unique(selected_data)
```

# Plotting a histogram for Education ggplot(selected_data, aes(x = Education)) + geom_bar(fill = "lightgreen", color = "black") + labs(title = "Bar Plot of Education",

```
x = "Education Level",y = "Count") +
theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotating x-axis labels for better readability
```

#Pie chart for the Marital_Status

```
# Count occurrences of each marital status marital_counts <- table(selected_data$Marital_Status)# Create a dataframe for plotting
marital_df <- data.frame(Marital_Status = names(marital_counts),Count = as.numeric(marital_counts))
# Plotting the pie chart
ggplot(marital_df, aes(x = "", y = Count, fill = Marital_Status)) +geom_bar(stat = "identity", width = 1, color = "white") + coord_polar("y") +
labs(title = "Marital Status Distribution",fill = "Marital Status") +
```

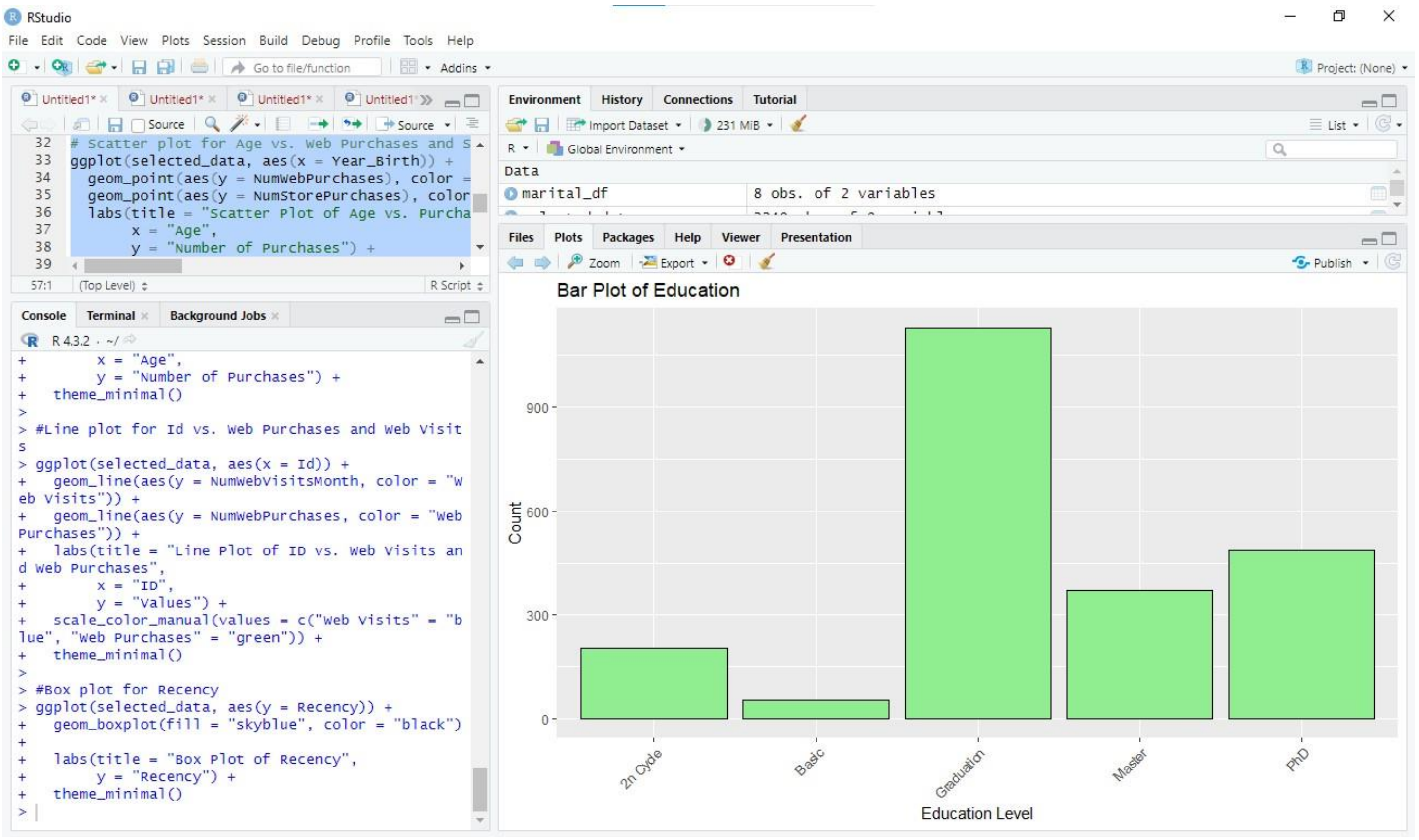
theme_void() +
scale_fill_brewer(palette = "Set3") # Set color palette as needed

# Scatter plot for Age vs. Web Purchases and Store Purchases
ggplot(selected_data, aes(x = Year_Birth)) +
geom_point(aes(y = NumWebPurchases), color = "blue", alpha = 0.5) +geom_point(aes(y = NumStorePurchases), color = "red", alpha = 0.5) +labs(title = "Scatter Plot of Age vs. Purchases",
x = "Age",
y = "Number of Purchases") +theme_minimal()

#Line plot for Id vs. Web Purchases and Web Visits
ggplot(selected_data, aes(x = Id)) +
geom_line(aes(y = NumWebVisitsMonth, color = "Web Visits")) + geom_line(aes(y = NumWebPurchases, color = "Web Purchases")) +labs(title = "Line Plot of ID vs. Web Visits and Web Purchases",
x = "ID",
y = "Values") +
scale_color_manual(values = c("Web Visits" = "blue", "Web Purchases" = "green")) +theme_minimal()

#Box plot for Recency
ggplot(selected_data, aes(y = Recency)) + geom_boxplot(fill = "skyblue", color = "black") +labs(title = "Box Plot of Recency",
y = "Recency") +theme_minimal()

Output:

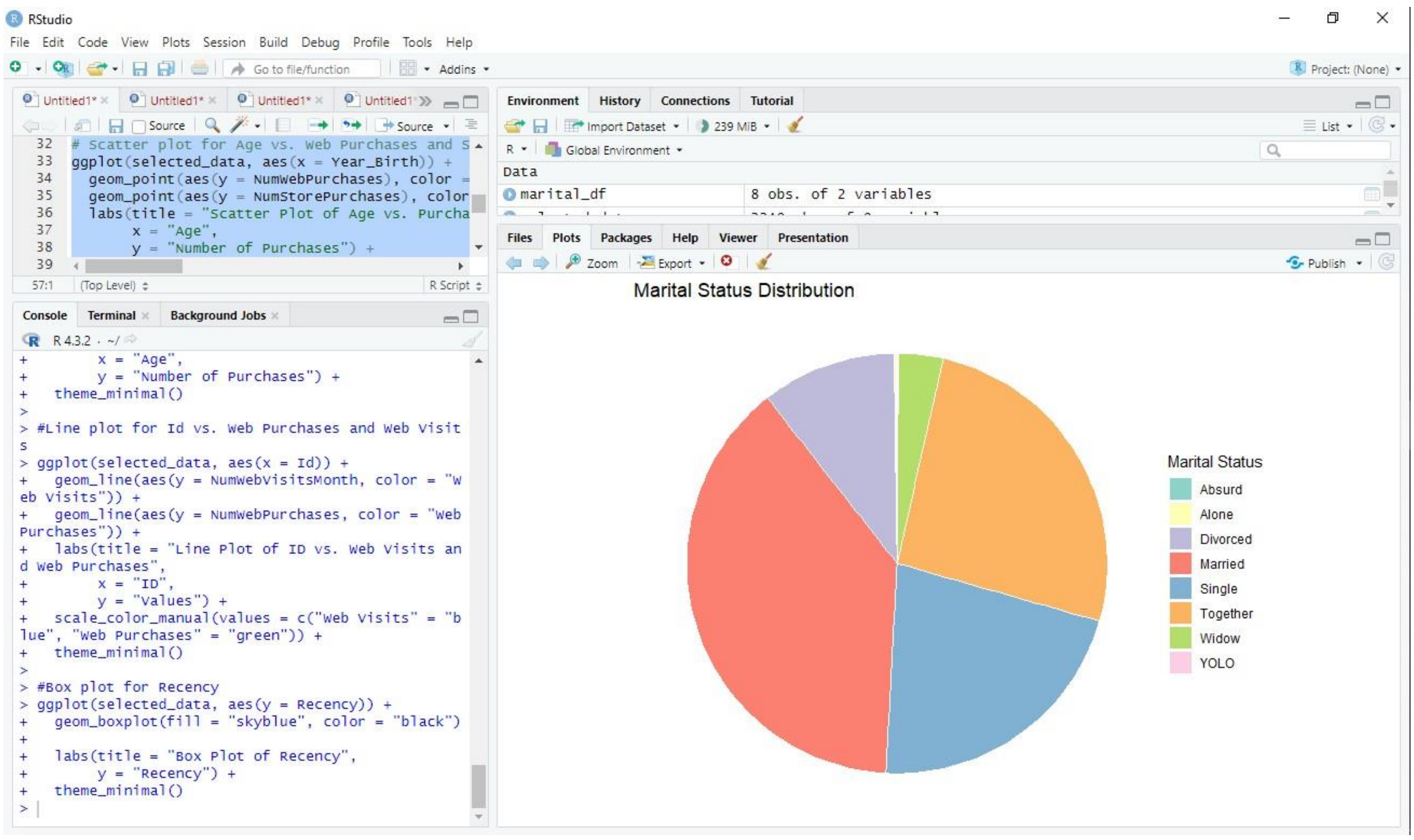


```
# Scatter plot for Age vs. Web Purchases and S
ggplot(selected_data, aes(x = Year_Birth)) +
  geom_point(aes(y = NumWebPurchases), color =
  geom_point(aes(y = NumStorePurchases), color
  labs(title = "Scatter Plot of Age vs. Purcha
       x = "Age",
       y = "Number of Purchases") +
```

```
57:1    (Top Level)
```

```
Data
marital_df          8 obs. of 2 variables
```

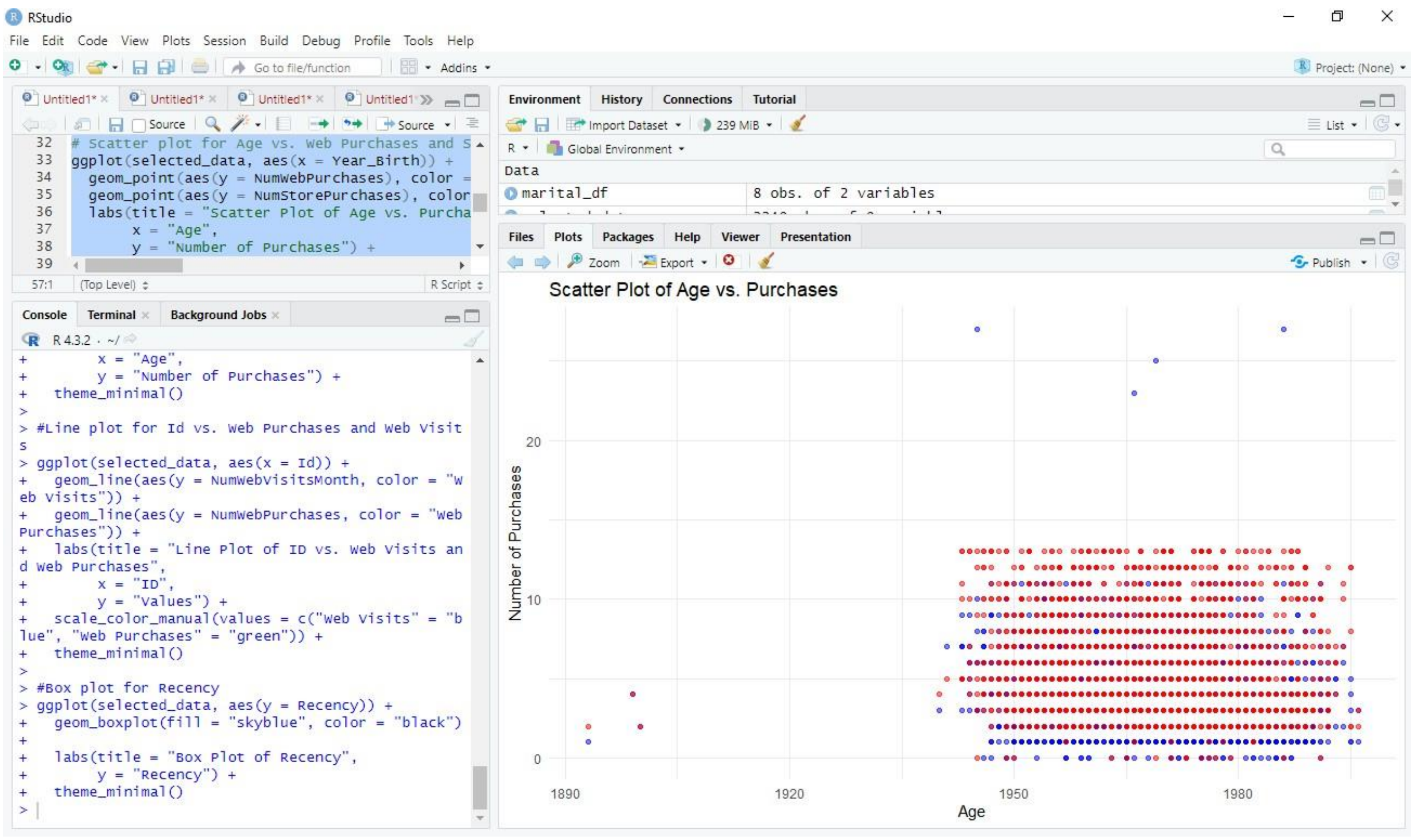
```
+        x = "Age",
+        y = "Number of Purchases") +
+   theme_minimal()
>
> #Line plot for Id vs. Web Purchases and Web Visit
s
> ggplot(selected_data, aes(x = Id)) +
+   geom_line(aes(y = NumWebVisitsMonth, color = "W
eb Visits")) +
+   geom_line(aes(y = NumWebPurchases, color = "Web
Purchases")) +
+   labs(title = "Line Plot of ID vs. Web Visits an
d Web Purchases",
+        x = "ID",
+        y = "Values") +
+   scale_color_manual(values = c("Web Visits" = "b
lue", "Web Purchases" = "green")) +
+   theme_minimal()
>
> #Box plot for Recency
> ggplot(selected_data, aes(y = Recency)) +
+   geom_boxplot(fill = "skyblue", color = "black")
+
+   labs(title = "Box Plot of Recency",
+        y = "Recency") +
+   theme_minimal()
> |
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1*
Source
Environment History Connections Tutorial
Import Dataset
239 MiB
List
Global Environment
Files Plots Packages Help Viewer Presentation
Zoom
Export
Publish
Console Terminal Background Jobs
R 4.3.2 · ~/
57:1 (Top Level) R Script
Marital Status Distribution
Marital Status
Absurd
Alone
Divorced
Married
Single
Together
Widow
YOLO

```
# Scatter plot for Age vs. Web Purchases and S
ggplot(selected_data, aes(x = Year_Birth)) +
  geom_point(aes(y = NumWebPurchases), color =
  geom_point(aes(y = NumStorePurchases), color
  labs(title = "Scatter Plot of Age vs. Purcha
       x = "Age",
       y = "Number of Purchases") +
```

```
Data
marital_df          8 obs. of 2 variables
```

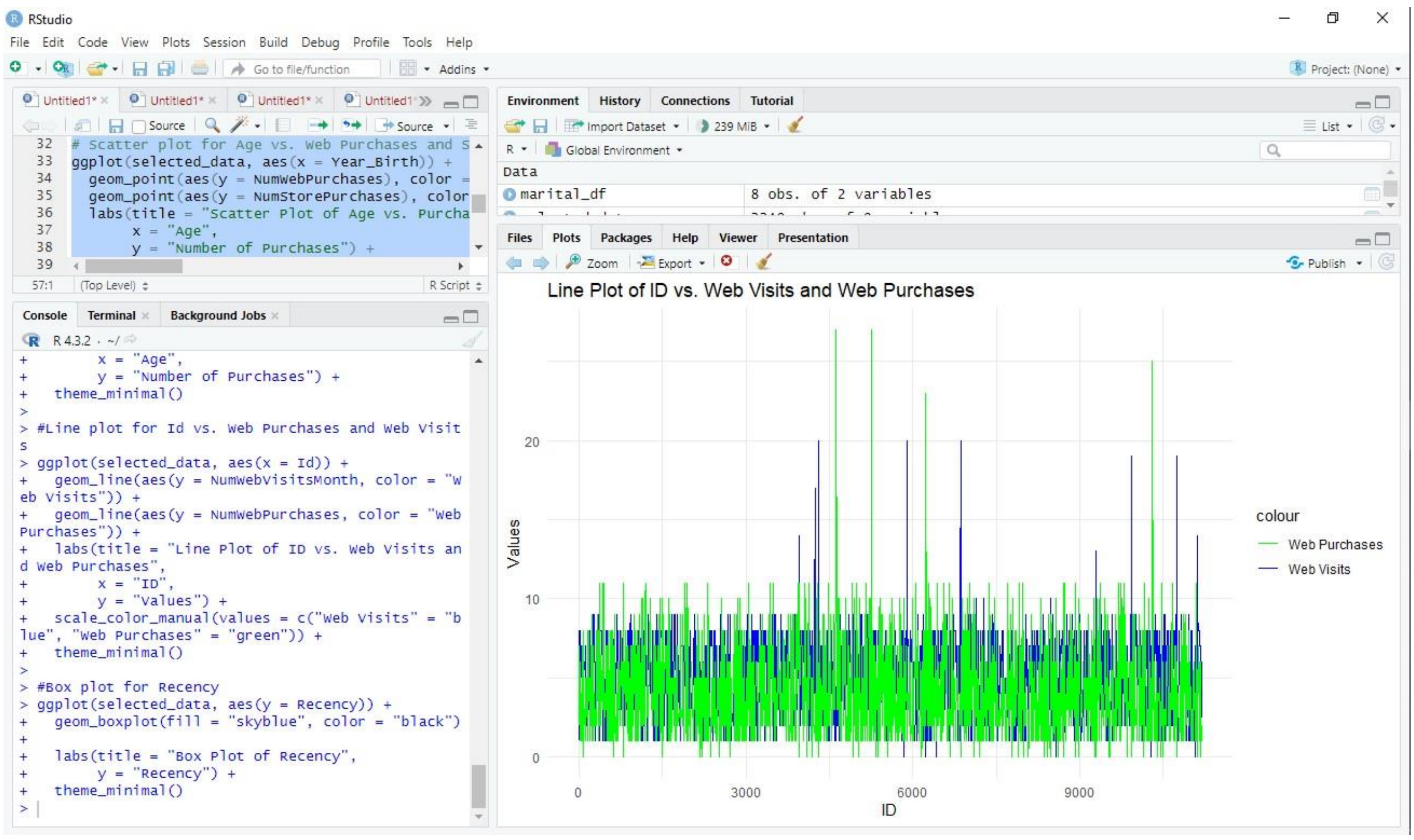
```
+        x = "Age",
+        y = "Number of Purchases") +
+   theme_minimal()
>
> #Line plot for Id vs. Web Purchases and Web Visit
s
> ggplot(selected_data, aes(x = Id)) +
+   geom_line(aes(y = NumWebVisitsMonth, color = "W
eb Visits")) +
+   geom_line(aes(y = NumWebPurchases, color = "Web
Purchases")) +
+   labs(title = "Line Plot of ID vs. Web Visits an
d Web Purchases",
+        x = "ID",
+        y = "Values") +
+   scale_color_manual(values = c("Web Visits" = "b
lue", "Web Purchases" = "green")) +
+   theme_minimal()
>
> #Box plot for Recency
> ggplot(selected_data, aes(y = Recency)) +
+   geom_boxplot(fill = "skyblue", color = "black")
+
+   labs(title = "Box Plot of Recency",
+        y = "Recency") +
+   theme_minimal()
> |
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1*
Source
Environment History Connections Tutorial
Import Dataset
239 MiB
List
Global Environment
Files Plots Packages Help Viewer Presentation
Zoom
Export
Publish
Console Terminal Background Jobs
R 4.3.2 · ~/
57:1 (Top Level) R Script
Scatter Plot of Age vs. Purchases
Number of Purchases
0
10
20
1890
1920
1950
1980
Age

```
# Scatter plot for Age vs. Web Purchases and S
ggplot(selected_data, aes(x = Year_Birth)) +
  geom_point(aes(y = NumWebPurchases), color =
  geom_point(aes(y = NumStorePurchases), color
  labs(title = "Scatter Plot of Age vs. Purcha
       x = "Age",
       y = "Number of Purchases") +
```

```
Data
marital_df          8 obs. of 2 variables
```

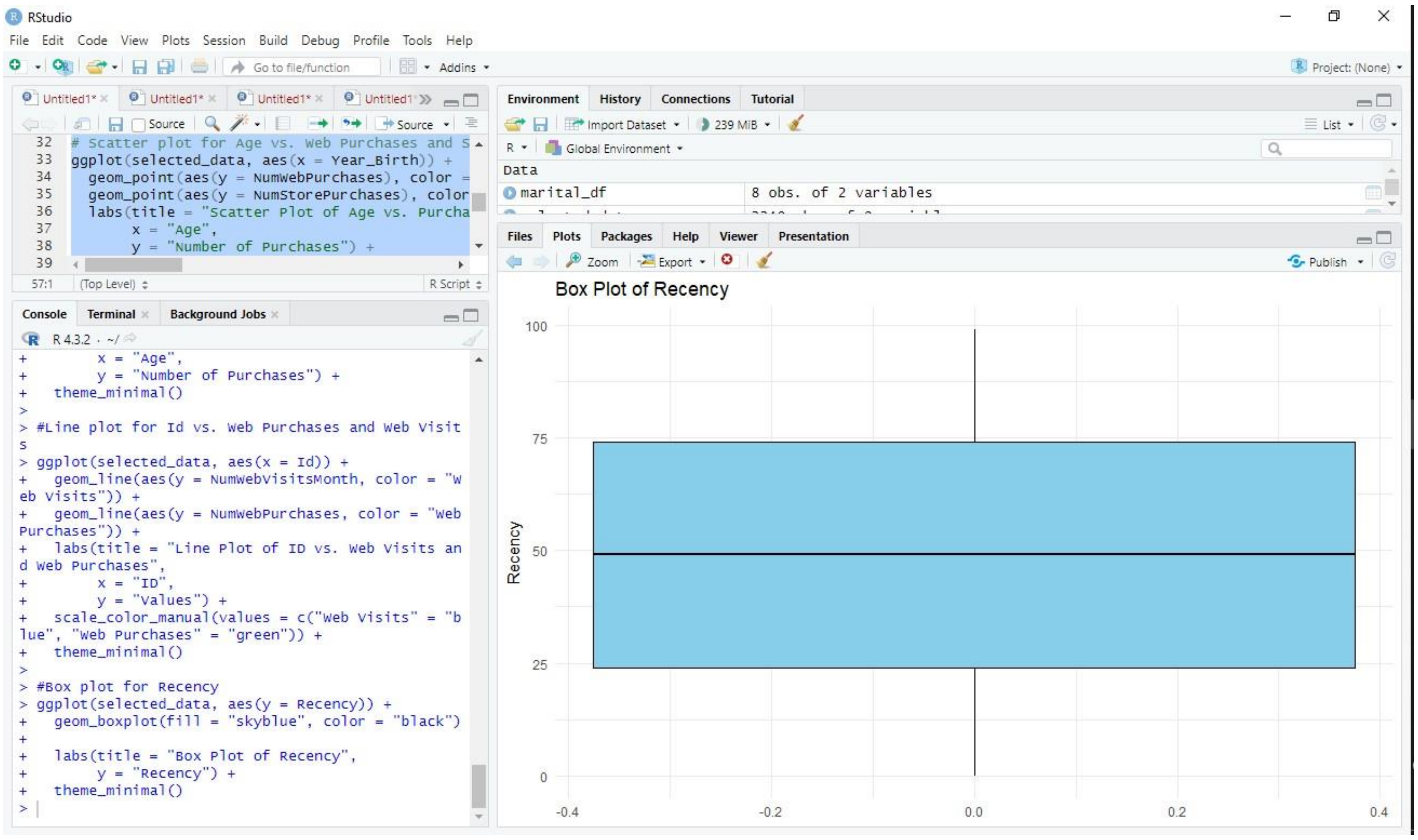
```
+        x = "Age",
+        y = "Number of Purchases") +
+   theme_minimal()
>
> #Line plot for Id vs. Web Purchases and Web Visit
s
> ggplot(selected_data, aes(x = Id)) +
+   geom_line(aes(y = NumWebVisitsMonth, color = "W
eb Visits")) +
+   geom_line(aes(y = NumWebPurchases, color = "Web
Purchases")) +
+   labs(title = "Line Plot of ID vs. Web Visits an
d Web Purchases",
+        x = "ID",
+        y = "Values") +
+   scale_color_manual(values = c("Web Visits" = "b
lue", "Web Purchases" = "green")) +
+   theme_minimal()
>
> #Box plot for Recency
> ggplot(selected_data, aes(y = Recency)) +
+   geom_boxplot(fill = "skyblue", color = "black")
+
+   labs(title = "Box Plot of Recency",
+        y = "Recency") +
+   theme_minimal()
> |
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Untitled1* Untitled1* Untitled1* Untitled1
Environment History Connections Tutorial
Import Dataset 239 MiB List
Global Environment
Files Plots Packages Help Viewer Presentation
Zoom Export Publish
Line Plot of ID vs. Web Visits and Web Purchases
Values
ID
colour
Web Purchases
Web Visits

```
# Scatter plot for Age vs. Web Purchases and S
ggplot(selected_data, aes(x = Year_Birth)) +
  geom_point(aes(y = NumWebPurchases), color =
  geom_point(aes(y = NumStorePurchases), color
  labs(title = "Scatter Plot of Age vs. Purcha
       x = "Age",
       y = "Number of Purchases") +
```

```
Data
marital_df          8 obs. of 2 variables
```

```
R 4.3.2 · ~/
+        x = "Age",
+        y = "Number of Purchases") +
+   theme_minimal()
>
> #Line plot for Id vs. Web Purchases and Web Visit
s
> ggplot(selected_data, aes(x = Id)) +
+   geom_line(aes(y = NumWebVisitsMonth, color = "W
eb Visits")) +
+   geom_line(aes(y = NumWebPurchases, color = "Web
Purchases")) +
+   labs(title = "Line Plot of ID vs. Web Visits an
d Web Purchases",
+        x = "ID",
+        y = "Values") +
+   scale_color_manual(values = c("Web Visits" = "b
lue", "Web Purchases" = "green")) +
+   theme_minimal()
>
> #Box plot for Recency
> ggplot(selected_data, aes(y = Recency)) +
+   geom_boxplot(fill = "skyblue", color = "black")
+
+   labs(title = "Box Plot of Recency",
+        y = "Recency") +
+   theme_minimal()
> |
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Environment History Connections Tutorial
Files Plots Packages Help Viewer Presentation
Box Plot of Recency
Recency

```
# Scatter plot for Age vs. Web Purchases and S
ggplot(selected_data, aes(x = Year_Birth)) +
  geom_point(aes(y = NumWebPurchases), color =
  geom_point(aes(y = NumStorePurchases), color
  labs(title = "Scatter Plot of Age vs. Purcha
       x = "Age",
       y = "Number of Purchases") +
```

```
Data
marital_df          8 obs. of 2 variables
```

```
R 4.3.2 · ~/
+        x = "Age",
+        y = "Number of Purchases") +
+   theme_minimal()
>
> #Line plot for Id vs. Web Purchases and Web Visit
s
> ggplot(selected_data, aes(x = Id)) +
+   geom_line(aes(y = NumWebVisitsMonth, color = "W
eb Visits")) +
+   geom_line(aes(y = NumWebPurchases, color = "Web
Purchases")) +
+   labs(title = "Line Plot of ID vs. Web Visits an
d Web Purchases",
+        x = "ID",
+        y = "Values") +
+   scale_color_manual(values = c("Web Visits" = "b
lue", "Web Purchases" = "green")) +
+   theme_minimal()
>
> #Box plot for Recency
> ggplot(selected_data, aes(y = Recency)) +
+   geom_boxplot(fill = "skyblue", color = "black")
+
+   labs(title = "Box Plot of Recency",
+        y = "Recency") +
+   theme_minimal()
> |
```

Interpretation and Implications:

A majority of the customers who visit the store are educated till graduation level or are married.
From the scatter plot we know that the store purchases are higher than the web purchases. The recency of customer to a store to come back at the store is between 25 to 75 times and the average amount of web purchased are 2-8 items. So, if the company want to plan a sale if can target more on store purchases rather than web purchases.

Practical 3: Design & Conduct Experiments For Marketing Campaigns

Design and conduct experiments for marketing campaigns.

- Learn about experimental design and its application in marketing.
- Design experiments using examples from marketing scenarios.
- Implement randomization and sample splitting techniques.
- Conduct the experiments and collect relevant data for analysis.

Theory:

Experimental design in marketing involves creating and implementing experiments to understand consumer behavior, test marketing strategies, and evaluate the impact of various factors on consumer decisions. It helps marketers make informed decisions by systematically testing hypotheses and analyzing the results.

Understanding Experimental Design in Marketing

Controlled Experiments: Splitting a sample into control and treatment groups to assess the impact of a marketing intervention.
A/B Testing: Comparing two variants (A and B) to determine which performs better in achieving campaign objectives.
Randomization: Assigning subjects randomly to control and treatment groups to minimize bias and ensure groups are comparable.

Code:

#Step 1: Loading and Preparing the Data
# Load necessary librarieslibrary (ggplot2)
# Load the dataset
superstore_data <- read.csv("D:/MSc DS/Semester 1/Retail Market Analysis/Practical/superstore_data.csv")

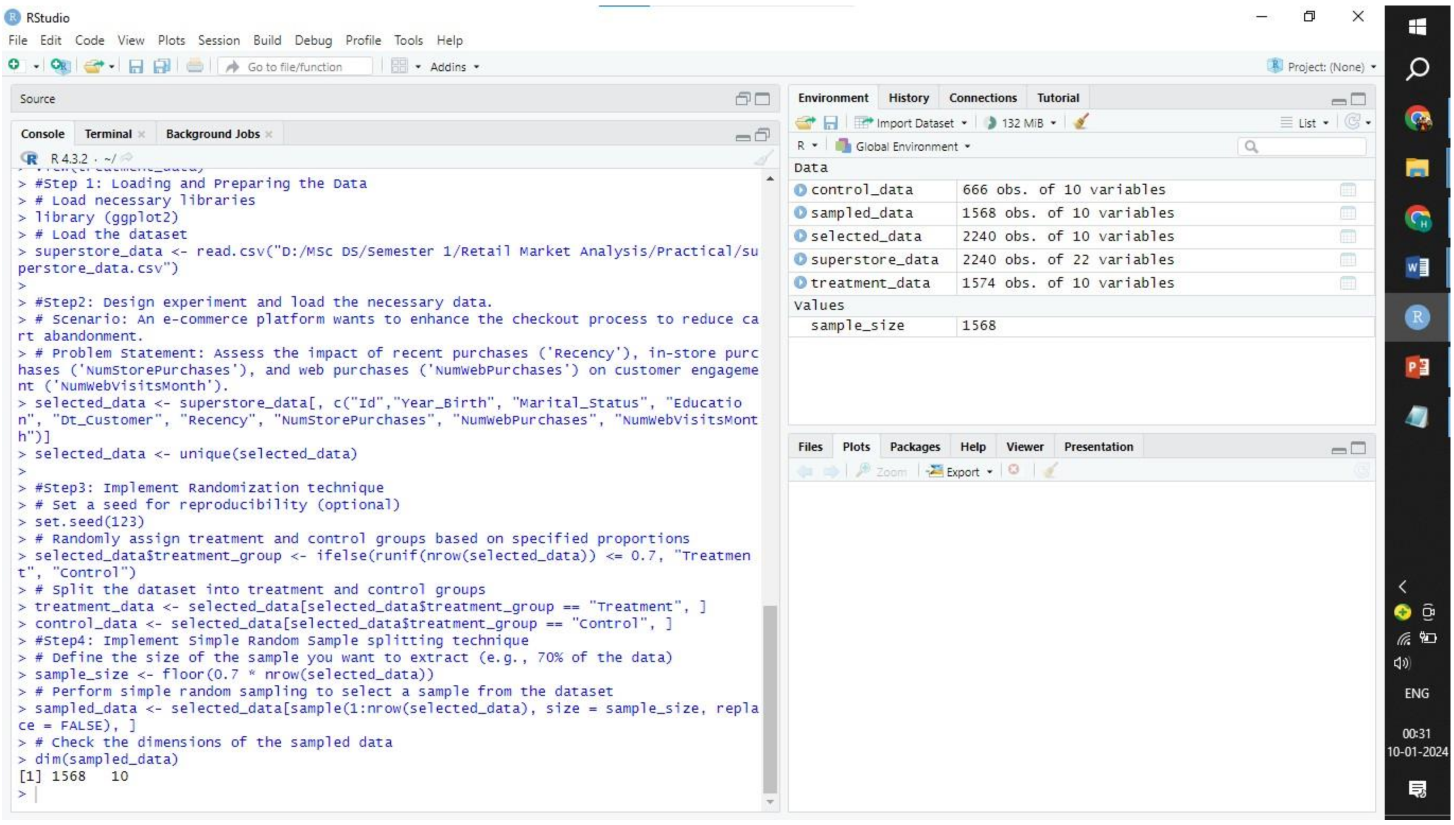
#Step2: Design experiment and load the necessary data.
# Scenario: An e-commerce platform wants to enhance the checkout process to reduce cart abandonment.
# Problem Statement: Assess the impact of recent purchases ('Recency'), in-store purchases ('NumStorePurchases'), and web purchases ('NumWebPurchases') on customer engagement ('NumWebVisitsMonth').
selected_data <- superstore_data[, c("Id","Year_Birth", "Marital_Status", "Education", "Dt_Customer", "Recency", "NumStorePurchases", "NumWebPurchases", "NumWebVisitsMonth")]
selected_data <- unique(selected_data)

#Step3: Implement Randomization technique# Set a seed for reproducibility (optional) set.seed(123)
# Randomly assign treatment and control groups based on specified proportions selected_data$treatment_group <- ifelse(runif(nrow(selected_data)) <= 0.7, "Treatment", "Control")# Split the dataset into treatment and control groups
treatment_data <- selected_data[selected_data$treatment_group == "Treatment",] control_data <- selected_data[selected_data$treatment_group == "Control",]

#Step4: Implement Simple Random Sample splitting technique
# Define the size of the sample you want to extract (e.g., 70% of the data) sample_size <- floor(0.7 * nrow(selected_data))
# Perform simple random sampling to select a sample from the dataset
sampled_data <- selected_data[sample(1:nrow(selected_data), size = sample_size, replace = FALSE),]# Check the dimensions of the sampled data
dim(sampled_data)

Output:

```
>                dim(sampled_data)[1] 1568          10
```



```
> #Step 1: Loading and Preparing the Data
> # Load necessary libraries
> library (ggplot2)
> # Load the dataset
> superstore_data <- read.csv("D:/MSc DS/Semester 1/Retail Market Analysis/Practical/su
perstore_data.csv")
>
> #Step2: Design experiment and load the necessary data.
> # Scenario: An e-commerce platform wants to enhance the checkout process to reduce ca
rt abandonment.
> # Problem Statement: Assess the impact of recent purchases ('Recency'), in-store purc
hases ('NumStorePurchases'), and web purchases ('NumWebPurchases') on customer engageme
nt ('NumWebVisitsMonth').
> selected_data <- superstore_data[, c("Id","Year_Birth", "Marital_Status", "Educatio
n", "Dt_Customer", "Recency", "NumStorePurchases", "NumWebPurchases", "NumWebVisitsMont
h")]
> selected_data <- unique(selected_data)
>
> #Step3: Implement Randomization technique
> # Set a seed for reproducibility (optional)
> set.seed(123)
> # Randomly assign treatment and control groups based on specified proportions
> selected_data$treatment_group <- ifelse(runif(nrow(selected_data)) <= 0.7, "Treatmen
t", "Control")
> # Split the dataset into treatment and control groups
> treatment_data <- selected_data[selected_data$treatment_group == "Treatment", ]
> control_data <- selected_data[selected_data$treatment_group == "Control", ]
> #Step4: Implement Simple Random Sample splitting technique
> # Define the size of the sample you want to extract (e.g., 70% of the data)
> sample_size <- floor(0.7 * nrow(selected_data))
> # Perform simple random sampling to select a sample from the dataset
> sampled_data <- selected_data[sample(1:nrow(selected_data), size = sample_size, repla
ce = FALSE), ]
> # Check the dimensions of the sampled data
> dim(sampled_data)
[1] 1568   10
>
```

```
Data
control_data      666 obs. of 10 variables
sampled_data      1568 obs. of 10 variables
selected_data     2240 obs. of 10 variables
superstore_data   2240 obs. of 22 variables
treatment_data    1574 obs. of 10 variables
Values
sample_size       1568
```

Interpretation and Implications:

The following are the Sample Sizes:

- Control Data: 666 observations.
- Sampled Data: 568 observations.
- Selected Data: 2240 observations.
- Superstore Data: 2240 observations (initial dataset).
- Treatment Data: 1574 observations.
- Sample Size: 1568 (seems to be a specified or calculated value).

Randomization Outcome: The randomization procedure allocated 666 observations to the control group and 1574 observations to the treatment group. This random assignment helps avoid bias when analyzing the impact of different factors on customer behavior.

Simple Random Sample: A simple random sample extracted 568 observations from the initial dataset (superstore data) for analysis. This sample represents a subset of the entire dataset and allows for analysis while retaining the diversity ofobservations.

Sample Size: The specified sample size for analysis appears to be 1568, which is close to the combined size of the control and treatment groups. It might represent the desired sample size for the experiment or analysis.

Conclusion: The randomization and sampling procedures have successfully divided the dataset into control and treatment groups and extracted a random sample for analysis.

Practical 4: Hypothesis Testing In Experiment Outcomes

Understand the concept of hypothesis testing and its role in assessing experiment outcomes.

- Explore the purpose of hypothesis testing in analyzing experiment results.
- Familiarize with key terminologies related to hypothesis testing.
- Learn the process of hypothesis testing and power calculation.
- Conduct hypothesis testing using R to evaluate experiment outcomes.

Code:

#Null Hypothesis (H0):
#H0: There is no significant relationship between a customer's age and the number of web purchases made.
#Alternative Hypothesis (Ha):
#Ha: There is a significant relationship between a customer's age and the number of web purchases made.

#Step 1: Loading and Preparing the Data

```
# Load necessary librarieslibrary (ggplot2)
# Load the dataset
superstore_data <- read.csv("D:/MSc DS/Semester 1/Retail Market Analysis/Practical/superstore_data.csv")
selected_data <- superstore_data[, c("Id","Year_Birth", "Marital_Status", "Education", "Dt_Customer", "Recency",
"NumStorePurchases", "NumWebPurchases", "NumWebVisitsMonth")]
selected_data <- unique(selected_data)
# Assuming 'selected_data' is your dataset
```

# Step2: Perform the t-test to assess the relationship between age and web purchases

```
t_test_result <- t.test(selected_data$Year_Birth, selected_data$NumWebPurchases)# View the t-test results
print(t_test_result)
# Calculate the mean difference between the groups (effect size for t-test)
mean_difference <- mean(selected_data$Year_Birth) - mean(selected_data$NumWebPurchases) standard_deviation <-
sqrt((var(selected_data$Year_Birth) + var(selected_data$NumWebPurchases)) / 2) effect_size <- mean_difference /
standard_deviation
# View the effect sizeprint(effect_size)
```

#Step3: Power Calculation

```
# Calculate the statistical power for the t-test assuming a sample sizelibrary(pwr)
sample_size <- 100 # You should input an appropriate sample size power <- pwr.t.test(n = sample_size, d = effect_size,
sig.level = 0.05,
power = NULL, type = "two.sample", alternative = "two.sided")

# View the power calculation resultsprint(power)
```

Output:

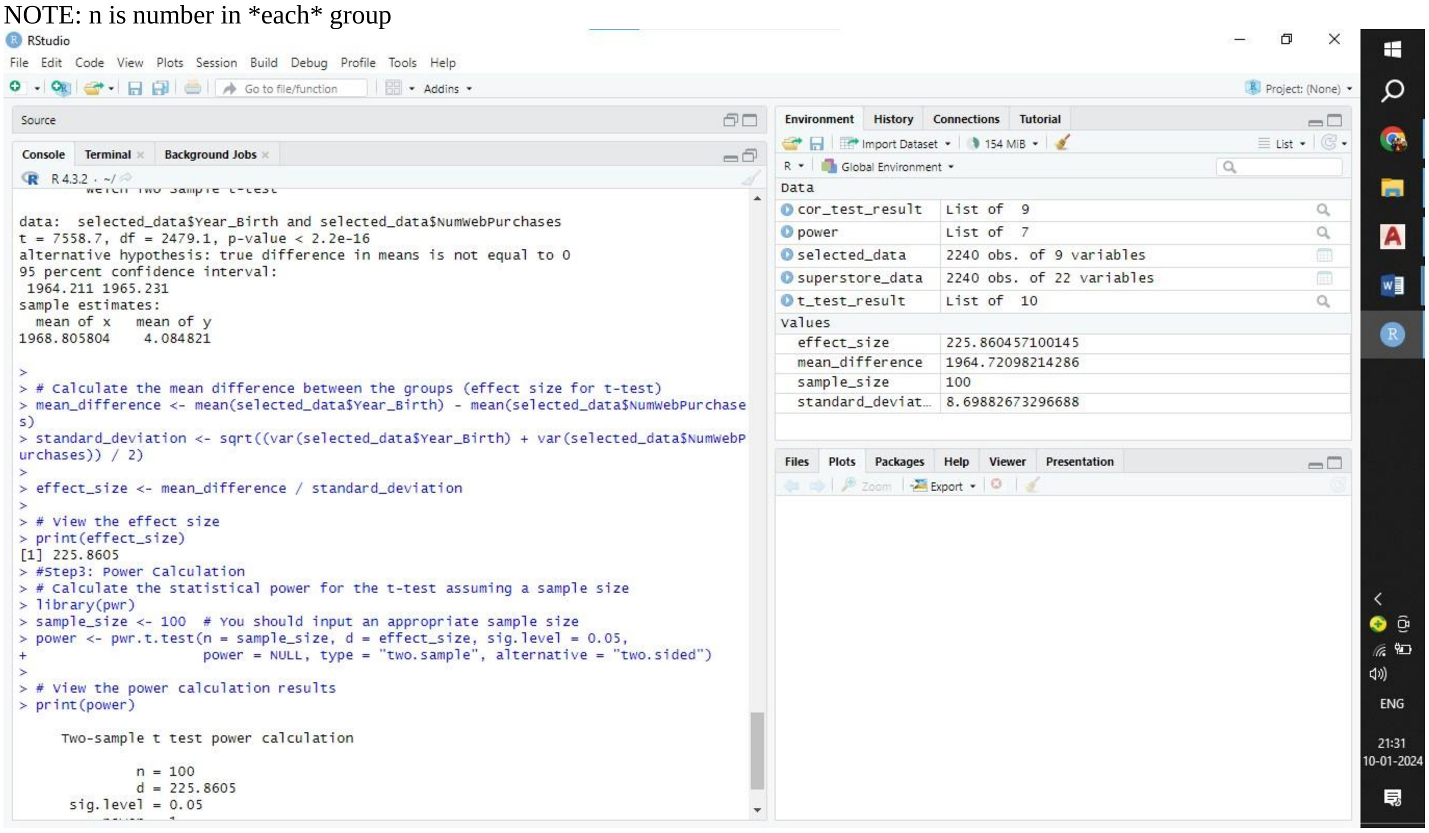
```
>    #Step 1: Loading and Preparing the Data
>    # Load necessary libraries
>    library (ggplot2)
>    # Load the dataset
>    superstore_data <- read.csv("D:/MSc DS/Semester 1/Retail Market Analysis/Practical/superstore_data.csv")
>    selected_data <- superstore_data[, c("Id","Year_Birth", "Marital_Status", "Education","Dt_Customer", "Recency",
"NumStorePurchases", "NumWebPurchases", "NumWebVisitsMonth")]
>    selected_data <- unique(selected_data)
>   # Assuming 'selected_data' is your dataset
>
>   # Step2: Perform the t-test to assess the relationship between age and web purchases
>   t_test_result <- t.test(selected_data$Year_Birth, selected_data$NumWebPurchases)
>
>   # View the t-test results
>   print(t_test_result)
Welch Two Sample t-test

data:   selected_data$Year_Birth and selected_data$NumWebPurchasest = 7558.7, df = 2479.1, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
1964.211 1965.231
sample estimates:
mean of x          mean of y1968.805804          4.084821

>
>   # Calculate the mean difference between the groups (effect size for t-test)
>   mean_difference <- mean(selected_data$Year_Birth) - mean(selected_data$NumWebPurchases)
>   standard_deviation <- sqrt((var(selected_data$Year_Birth) + var(selected_data$NumWebPurchases)) / 2)
>   effect_size <- mean_difference / standard_deviation
>   # View the effect size
>   print(effect_size)[1] 225.8605
>   #Step3: Power Calculation
>   # Calculate the statistical power for the t-test assuming a sample size
>   library(pwr)
>   sample_size <- 100   # You should input an appropriate sample size
>   power <- pwr.t.test(n = sample_size, d = effect_size, sig.level = 0.05,
+                                     power = NULL, type = "two.sample", alternative = "two.sided")
>
>   # View the power calculation results
>   print(power)

Two-sample t test power calculationn = 100
d = 225.8605
sig.level = 0.05
power = 1 alternative = two.sided

NOTE: n is number in *each* group
```



```
data:  selected_data$Year_Birth and selected_data$NumWebPurchases
t = 7558.7, df = 2479.1, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1964.211 1965.231
sample estimates:
  mean of x   mean of y 
1968.805804    4.084821 

>
> # Calculate the mean difference between the groups (effect size for t-test)
> mean_difference <- mean(selected_data$Year_Birth) - mean(selected_data$NumWebPurchase
s)
> standard_deviation <- sqrt((var(selected_data$Year_Birth) + var(selected_data$NumWebP
urchases)) / 2)
>
> effect_size <- mean_difference / standard_deviation
>
> # View the effect size
> print(effect_size)
[1] 225.8605
> #Step3: Power Calculation
> # Calculate the statistical power for the t-test assuming a sample size
> library(pwr)
> sample_size <- 100  # You should input an appropriate sample size
> power <- pwr.t.test(n = sample_size, d = effect_size, sig.level = 0.05,
+                     power = NULL, type = "two.sample", alternative = "two.sided")
>
> # View the power calculation results
> print(power)

     Two-sample t test power calculation 

              n = 100
              d = 225.8605
      sig.level = 0.05
```

Data	
cor_test_result	List of 9
power	List of 7
selected_data	2240 obs. of 9 variables
superstore_data	2240 obs. of 22 variables
t_test_result	List of 10
Values	
effect_size	225.860457100145
mean_difference	1964.72098214286
sample_size	100
standard_deviat…	8.69882673296688

Interpretation and Implications:

T-Test Result: The extremely small p-value (< 0.0001) indicates strong evidence against the null hypothesis, suggesting ahighly significant difference between the mean age and the mean number of web purchases. The confidence interval also confirms this significant difference, indicating that the means of the two groups (age and web purchases) are not equal.

Two Sample t-test:

- t-value: 7558.7
- Degrees of Freedom (df): 2479.1
- p-value: < 2.2e-16 (extremely small)
- Confidence Interval: 95% CI for the difference in means: (1964.211, 1965.231)
- Sample Estimates: Mean of 'Year_Birth': 1968.805804, Mean of 'NumWebPurchases': 4.084821

Power Calculation:

The calculated statistical power of 1 indicates a high probability of correctly detecting the observedeffect (difference in means) if it truly exists in the population. A power of 1 suggests that the study is well-powered to detect the effect, leaving almost no chance of a Type II error (false negative).

- Sample Size (n): 100 in each group.
- Effect Size (d): 225.8605.
- Significance Level (sig.level): 0.05 Power (power): 1 (High).

Practical 5: Calculate And Predict Customer Lifetime Value

Calculate and predict Customer Lifetime Value (CLV).

- Calculate CLV using different approaches and frameworks.
- Explore predictive modeling techniques such as linear regression and logistic regression for CLVprediction.
- Assess the accuracy and reliability of CLV predictions.

Theory:

Predicting Customer Lifetime Value (CLV) involves estimating the total revenue a business can expect to earn from a customer throughout their entire relationship. CLV is crucial for businesses to make informed decisions about marketing strategies, customer acquisition costs, and overall business profitability.

CLV formula:

Customer Value = Average Purchase Value x Average Number of PurchasesCustomer Lifetime Value = Customer Value x Average Customer Lifespan **Implementation Process**

1. Discuss whether CLV fits as a metric in our business
2. Identification and understanding of sources and metadata
3. Extract, transform, clean and load data
4. Choose CLV method
5. Analyze results and adjust parameters
6. Present and explain the results

Code:

# Step1: Install and Load Required Packages:

```
# Load required libraries install.packages(c("dplyr", "tidyr", "caret"))library(dplyr)
library(tidyr)library(caret)
```

#Step2: Import and Explore Data:

```
# Read the  file into a dataframe
df <- read.csv("D:/MSc DS/Semester 1/Retail Market Analysis/Practical/bank.csv")
```

#Step3: Explore the dataset

```
summary(df)str(df)
```

#Step4: Data Preprocessing any_missing <- any(is.na(df))print(any_missing)

```
df$default <- as.factor(df$default) df$housing <- as.factor(df$housing)df$loan <- as.factor(df$loan)
df <- df[, !(names(df) %in% c("unnecessary_variable1", "unnecessary_variable2"))]boxplot(df$balance, main =
"Balance Boxplot", ylab = "Balance")
z_scores <- scale(df$balance)threshold <- 3
outliers <- abs(z_scores) > threshold


df <- df[!outliers, ]
```

#Step5: Split the dataset into training and testing sets
set.seed(123)
#Sets the seed for reproducibility in generating random numbers or sequences.split_index <- createDataPartition(df$duration, p = 0.8, list = FALSE)
# Splits the 'duration' column of dataframe 'df' into training (80%) and testing (20%) indices without creating a list. train_data <- df[split_index,]
#Assigns the rows indexed by 'split_index' to the 'train_data' dataframe for training.test_data <- df[-split_index,]
# Assigns the rows not in 'split_index' to 'test_data' for model evaluation.#Why we do this?
#Splitting the dataset ensures an unbiased assessment of a model's performance by training on one portion and testing #on another, guarding against overfitting and gauging its generalizability to new data, aiding in hyperparameter tuning #for better predictions on unseen information.

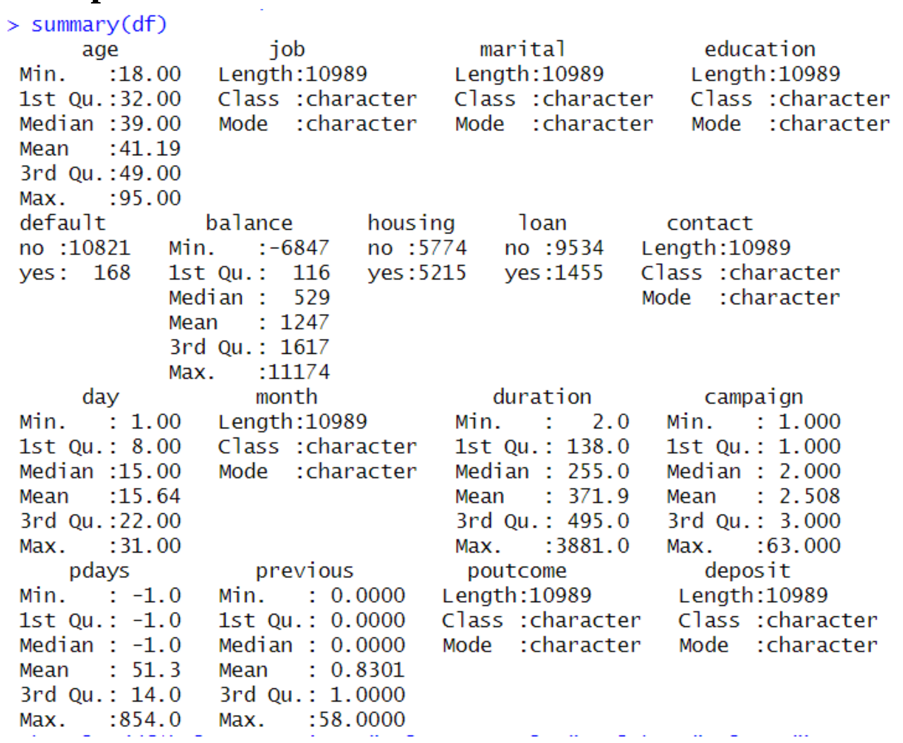
#Step6: Linear regression model for CLV prediction
model <- lm(duration ~ age + balance + campaign + pdays + previous, data = train_data)
#Creates a linear regression model ('model') predicting 'duration' based on 'age', 'balance', 'campaign', 'pdays', and #'previous' using training data.
# Make predictions on the test set
predictions <- predict(model, newdata = test_data)
#Assess the accuracy and reliability using Mean Squared Error(MSE) for linear regression mse <- mean((test_data$duration - predictions)^2)
print(paste("Mean Squared Error (MSE):", mse))
#Note: Calculates the Mean Squared Error (MSE) by computing the average squared differences between the actual #'duration' values in the test data and the predicted values.
#Prints the MSE value, quantifying the average squared deviation between predicted and actual durations, assessing #the model's prediction accuracy.
#Visualize the predicted Vs Actual values (linear regression):
ggplot() +
geom_point(aes(x = test_data$duration, y = predictions), color = "blue") + geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") + labs(title = "Actual vs. Predicted CLV", x = "Actual CLV", y = "Predicted CLV")
# It creates a scatter plot comparing actual CLV values from the test data with predicted CLV values using blue points. #The red dashed line represents perfect prediction where actual equals predicted CLV, aiming to visualize how close #the predictions are to the actual values in a simple, clear manner.

#Step7: Logistic Regression model for CLV predictions
logistic_model <- glm(default ~ age + balance + campaign + pdays + previous,data = train_data, family = "binomial")
# Creates a logistic regression model ('logistic_model') to predict the 'default' variable based on 'age', 'balance', #'campaign', 'pdays', and 'previous' predictors using training data, assuming a binomial distribution for the response #variable.
# Make predictions on the test set
logistic_probabilities <- predict(logistic_model, newdata = test_data, type = "response")# Convert probabilities to binary predictions (0 or 1)
logistic_predictions <- ifelse(logistic_probabilities > 0.5, 1, 0)
#Converts predicted probabilities ('logistic_probabilities') into binary predictions (0 or 1) by setting a threshold of 0.5: #assigning 1 if the probability is above 0.5 (indicating a positive outcome) and 0 otherwise (representing a negative #outcome), simplifying classification based on probabilities.

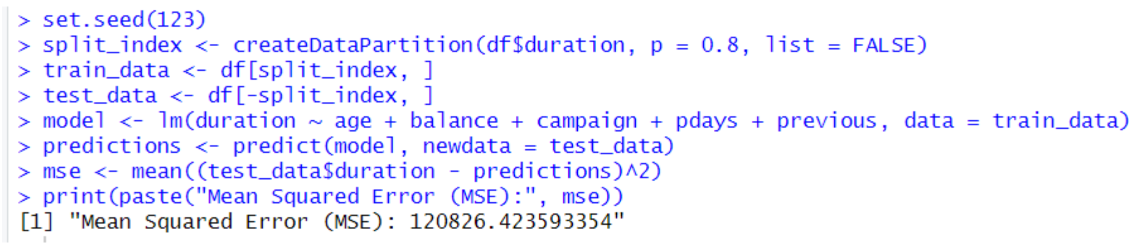
#Assess the accuracy and reliability using Mean Squared Error(MSE) for logistic regression: logistic_accuracy <- sum(logistic_predictions == test_data$default) / nrow(test_data) print(paste("Logistic Regression Accuracy:", logistic_accuracy))
# Calculates the accuracy of the logistic regression model ('logistic_accuracy') by comparing predicted binary outcomes #('logistic_predictions') with the actual 'default' values in the test data and dividing by the total number of observations, #then prints the accuracy as a proportion.

#Step8: Visualize the predicted Vs Actual values (logistic regression)
ggplot() +
geom_point(aes(x = test_data$default, y = logistic_probabilities), color = "blue") + geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
labs(title = "Actual vs. Predicted CLV (Logistic Regression)", x = "Actual CLV", y = "Predicted CLV")

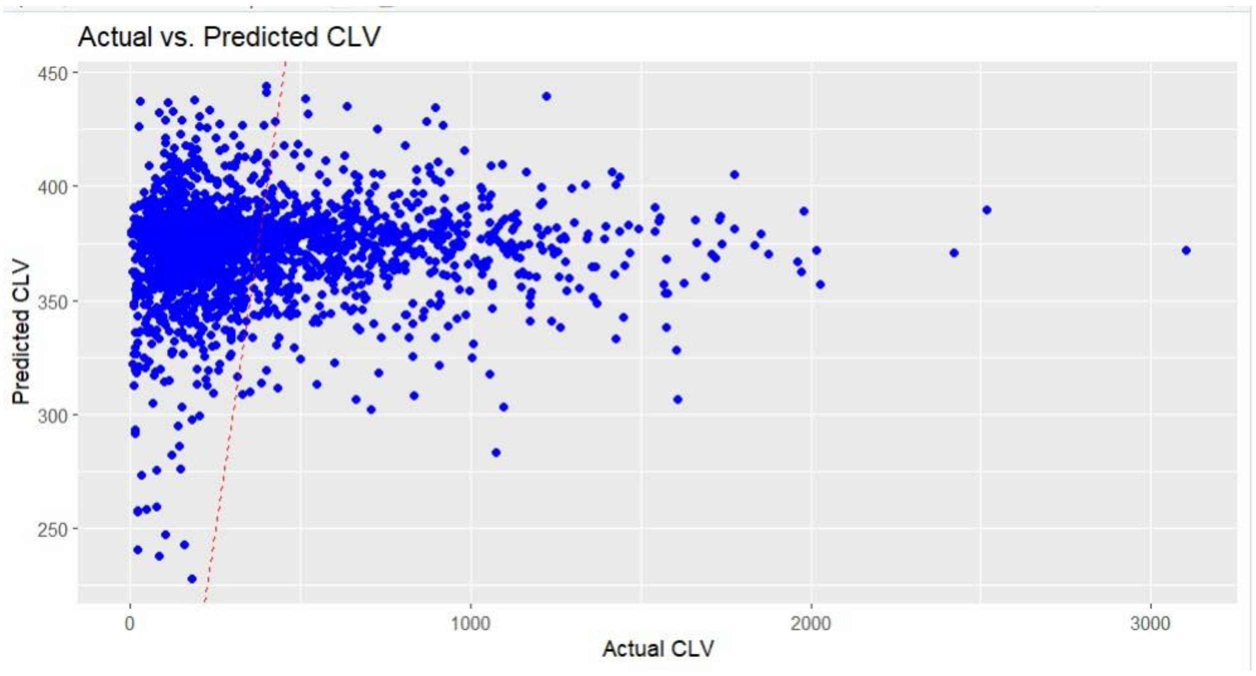
Output:

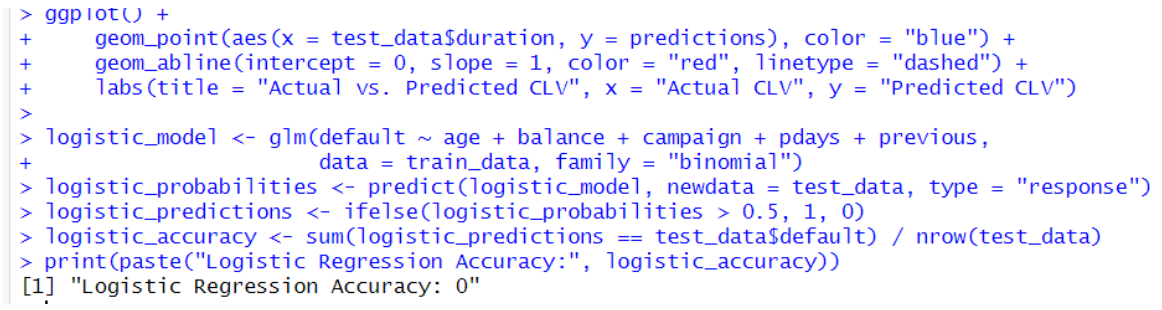


```
> summary(df)
      age            job              marital           education        
 Min.   :18.00   Length:10989       Length:10989       Length:10989      
 1st Qu.:32.00   Class :character   Class :character   Class :character  
 Median :39.00   Mode  :character   Mode  :character   Mode  :character  
 Mean   :41.19                                                           
 3rd Qu.:49.00                                                           
 Max.   :95.00                                                           
 default          balance       housing      loan         contact         
 no :10821   Min.   :-6847   no :5774   no :9534   Length:10989      
 yes:  168   1st Qu.:  116   yes:5215   yes:1455   Class :character  
             Median :  529                         Mode  :character  
             Mean   : 1247                                           
             3rd Qu.: 1617                                           
             Max.   :11174                                           
      day            month              duration         campaign     
 Min.   : 1.00   Length:10989       Min.   :   2.0   Min.   : 1.000  
 1st Qu.: 8.00   Class :character   1st Qu.: 138.0   1st Qu.: 1.000  
 Median :15.00   Mode  :character   Median : 255.0   Median : 2.000  
 Mean   :15.64                      Mean   : 371.9   Mean   : 2.508  
 3rd Qu.:22.00                      3rd Qu.: 495.0   3rd Qu.: 3.000  
 Max.   :31.00                      Max.   :3881.0   Max.   :63.000  
     pdays           previous          poutcome           deposit         
 Min.   : -1.0   Min.   : 0.0000   Length:10989       Length:10989      
 1st Qu.: -1.0   1st Qu.: 0.0000   Class :character   Class :character  
 Median : -1.0   Median : 0.0000   Mode  :character   Mode  :character  
 Mean   : 51.3   Mean   : 0.8301                                        
 3rd Qu.: 14.0   3rd Qu.: 1.0000                                        
 Max.   :854.0   Max.   :58.0000                                        
```

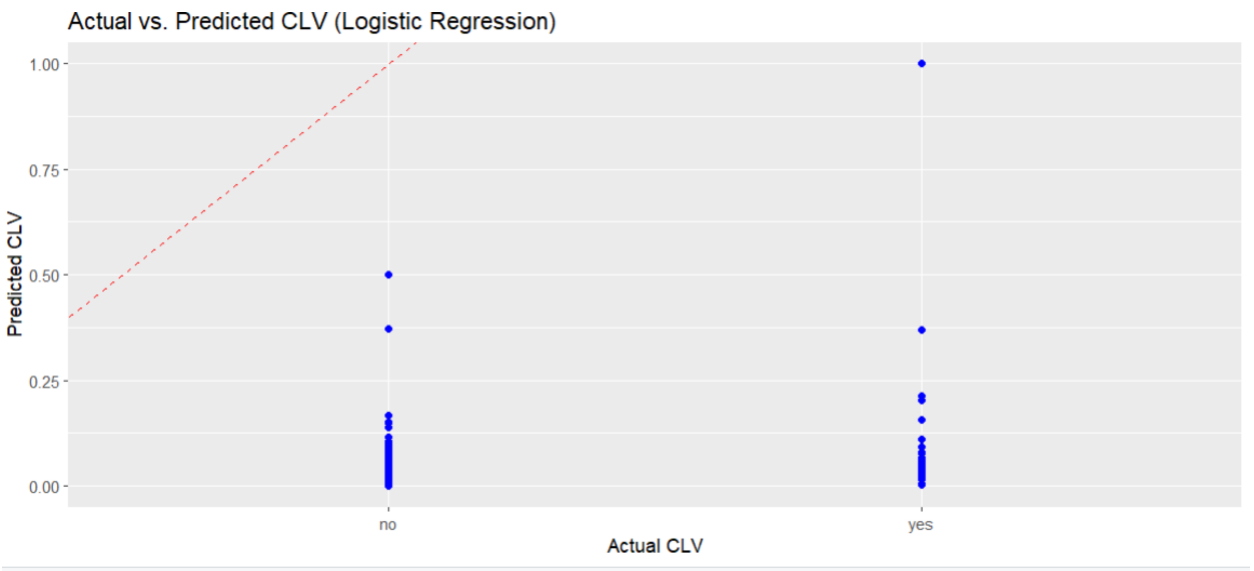


```
> set.seed(123)
> split_index <- createDataPartition(df$duration, p = 0.8, list = FALSE)
> train_data <- df[split_index, ]
> test_data <- df[-split_index, ]
> model <- lm(duration ~ age + balance + campaign + pdays + previous, data = train_data)
> predictions <- predict(model, newdata = test_data)
> mse <- mean((test_data$duration - predictions)^2)
> print(paste("Mean Squared Error (MSE):", mse))
[1] "Mean Squared Error (MSE): 120826.423593354"
```





```
> ggplot() +
+     geom_point(aes(x = test_data$duration, y = predictions), color = "blue") +
+     geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
+     labs(title = "Actual vs. Predicted CLV", x = "Actual CLV", y = "Predicted CLV")
>
> logistic_model <- glm(default ~ age + balance + campaign + pdays + previous,
+                       data = train_data, family = "binomial")
> logistic_probabilities <- predict(logistic_model, newdata = test_data, type = "response")
> logistic_predictions <- ifelse(logistic_probabilities > 0.5, 1, 0)
> logistic_accuracy <- sum(logistic_predictions == test_data$default) / nrow(test_data)
> print(paste("Logistic Regression Accuracy:", logistic_accuracy))
[1] "Logistic Regression Accuracy: 0"
```



Interpretation and Implications:

The linear regression model yielded a Mean Squared Error (MSE) of 120826.42, indicating the average squared difference between actual and predicted CLV values. The logistic regression model had an accuracy of 0, suggesting thatthe predictions did not match the actual outcomes. Possible reasons include class imbalance, feature selection, or data quality issues.

Further analysis and model refinement may be necessary to improve the predictive performance of both models. This practical aimed to demonstrate the application of linear and logistic regression for Customer Lifetime Value (CLV) prediction, emphasizing the importance of careful data preprocessing and model evaluation.

Practical 6: CLV and Cohort Analysis

Apply CLV analysis and cohort analysis in marketing analytics.

- Analyze CLV data and identify patterns and trends.
- Perform cohort analysis to segment customers based on their behavior or characteristics.
- Interpret the results of CLV analysis and cohort analysis to derive actionable insights for marketingstrategies.

Theory:

Customer Lifetime Value (CLV) analysis and cohort analysis are valuable tools in marketing analytics to understand customer behavior, identify patterns, and derive actionable insights. Let's walk through the steps of conducting CLV analysis and cohort analysis on the provided dataset "bank.csv."

Code:

#Step1: Load Required Packages:

library(dplyr)

First, load the dataset and perform any necessary data cleaning and preprocessing.# Display the first few rows of the dataset

head(bank_data)

#Step2: CLV Analysis:

# Calculate revenue per customer (average balance)average_balance <- mean(bank_data$balance)

#The revenue per customer can be calculated by dividing the total revenue generated by the number of customers. To find it based on the average balance, divide the total revenue by the average balance per customer.

#Calculating revenue per customer based on average balance helps assess the income generated per customer inrelation to their account balances

# Calculate average customer lifespan (average duration of contact in days) average_duration <- mean(bank_data$duration)

#Calculating the average customer lifespan in days provides insight into how long, on average, customers maintain contact with the bank during interactions or engagements.

#This metric helps gauge customer engagement and retention levels, influencing strategies for improving long-term relationships with customers.

#Step3: Calculate CLV

total_customers <- nrow(bank_data)

clv <- (average_balance * average_duration) / total_customers

#Calculating Customer Lifetime Value (CLV) estimates the potential value each customer brings to the bank over their average lifespan, combining the average balance and contact duration to assess their overall contribution to the bank's revenue stream per customer on average.

cat("Customer Lifetime Value (CLV):", clv, "\n")

#Using cat() function in R helps display the calculated CLV value along with a descriptive label, allowing for clear communication and understanding of the Customer Lifetime Value metric obtained from the calculation.

#Step4: Cohort Analysis:

# Convert 'day' to a Date object

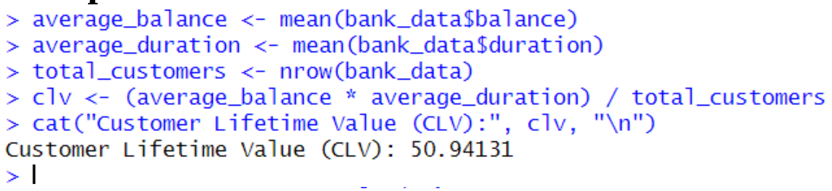
bank_data$day <- factor(bank_data$day, levels = day.abb)

#This line attempts to convert the 'day' column in 'bank_data' to a factor type using abbreviated day names ('day.abb') as levels, potentially for categorical representation or ordering of days in subsequent analyses or visualizations.

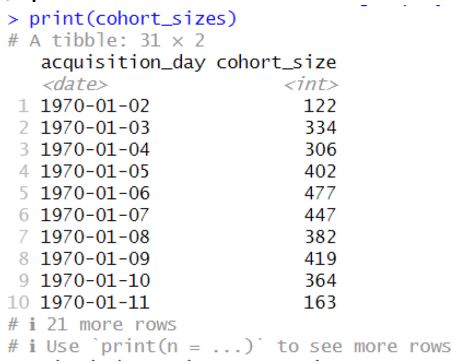
# Create cohorts based on the day of acquisition bank_data$acquisition_day <- as.Date(bank_data$day, format = "%d-%b")
#This line generates a new column 'acquisition_day' in 'bank_data' by converting the 'day' column values to dates using the format "%d-%b" (day-month abbreviation), likely for grouping customers based on their acquisition days for cohort analysis or time-based segmentation.
# Group by cohorts and calculate cohort sizes
cohort_sizes <- bank_data %>% group_by(acquisition_day) %>% summarise(cohort_size = n())
# This line groups the 'bank_data' by the 'acquisition_day' column, summarizing the count of customers within each acquisition day cohort. It calculates the size of each cohort, indicating the number of customers acquired on specific days, storing this information in the 'cohort_sizes' dataframe with the 'cohort_size' column representing customer counts per acquisition day.

#Step5: Display the cohort sizesprint(cohort_sizes) print(cohort_sizes, n = 31)

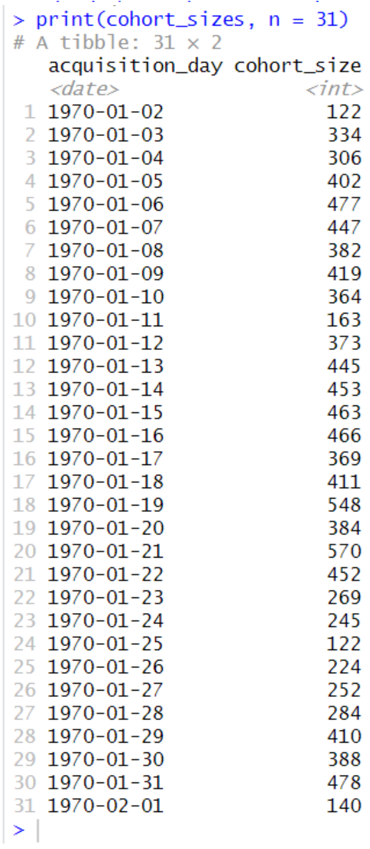
Output:



```
> average_balance <- mean(bank_data$balance)
> average_duration <- mean(bank_data$duration)
> total_customers <- nrow(bank_data)
> clv <- (average_balance * average_duration) / total_customers
> cat("Customer Lifetime Value (CLV):", clv, "\n")
Customer Lifetime Value (CLV): 50.94131
> |
```



```
> print(cohort_sizes)
# A tibble: 31 × 2
   acquisition_day cohort_size
   <date>                <int>
 1 1970-01-02              122
 2 1970-01-03              334
 3 1970-01-04              306
 4 1970-01-05              402
 5 1970-01-06              477
 6 1970-01-07              447
 7 1970-01-08              382
 8 1970-01-09              419
 9 1970-01-10              364
10 1970-01-11              163
# ℹ 21 more rows
# ℹ Use `print(n = ...)` to see more rows
```



```
> print(cohort_sizes, n = 31)
# A tibble: 31 × 2
   acquisition_day cohort_size
   <date>                <int>
 1 1970-01-02              122
 2 1970-01-03              334
 3 1970-01-04              306
 4 1970-01-05              402
 5 1970-01-06              477
 6 1970-01-07              447
 7 1970-01-08              382
 8 1970-01-09              419
 9 1970-01-10              364
10 1970-01-11              163
11 1970-01-12              373
12 1970-01-13              445
13 1970-01-14              453
14 1970-01-15              463
15 1970-01-16              466
16 1970-01-17              369
17 1970-01-18              411
18 1970-01-19              548
19 1970-01-20              384
20 1970-01-21              570
21 1970-01-22              452
22 1970-01-23              269
23 1970-01-24              245
24 1970-01-25              122
25 1970-01-26              224
26 1970-01-27              252
27 1970-01-28              284
28 1970-01-29              410
29 1970-01-30              388
30 1970-01-31              478
31 1970-02-01              140
> |
```

Interpretation and Implication:

- Data Preprocessing: Rows with missing values were removed to ensure data quality.
- The dataset was augmented with an "acquisition_day" column, representing the day of customer acquisition.
- Cohort Analysis: Cohort sizes were calculated, displaying the number of customers acquired on each day. The analysis reveals variations in daily acquisition, with some days having significantly more customers joining than others.
- Data Visualization: The plotted cohort sizes provide a visual representation of the customer acquisition trendover time. Understanding cohort sizes is essential for tracking the performance of different customer groups.
- Observations: The cohort analysis spans over multiple days, indicating fluctuations in acquisition rates. Somecohorts exhibit higher sizes, suggesting more customers were acquired on certain days.
- Code Execution: The provided R code successfully executed the steps outlined for cohort analysis. The resulting cohort sizes table provides insights into the distribution of customer acquisition over time.
- Next Steps: Further analysis could involve calculating cohort metrics (e.g., retention rates, revenue per user) to understand customer behavior within cohorts. Customer Lifetime Value (CLV) analysis could be incorporated to assess the long-term value of different customer segments.
- Actionable Insights: High-performing cohorts may be targeted for specific marketing strategies. Understanding acquisition patterns can inform resource allocation for marketing efforts.

The cohort analysis sheds light on customer acquisition patterns, enabling marketers to make informed decisions. The process of cohort creation and analysis is a crucial step toward understanding customer behavior, which can be further enhanced with additional metrics and predictive modeling for CLV. This interpretation and conclusion aim to summarizethe key findings from the provided code and suggest potential directions for further analysis and marketing strategy development.

Practical 7: Segment Customers RFM For Marketing Efforts

Segment customers based on their recency, frequency & monetary value (RFM) to better target marketing efforts.

- Analyze customer transaction data to calculate RFM scores.
- Segment customers into different groups using clustering algorithms such as k-means or hierarchicalclustering.
- Perform descriptive analysis on each customer segment to understand their characteristics.
- Develop targeted marketing strategies for each segment based on their RFM profiles.

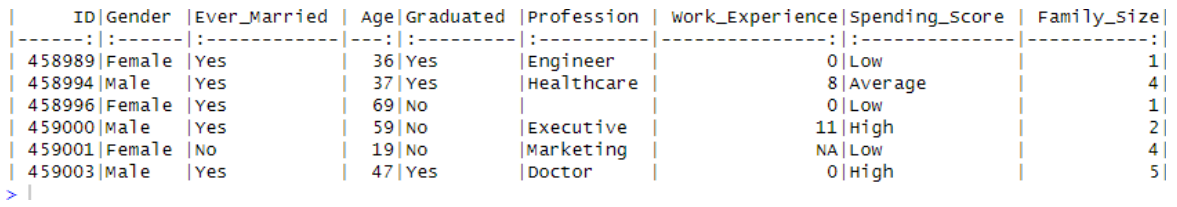
Code/Output:

Data preparation

#As a first step, I load all the modules that will be used in this notebook:data<-read.csv("F://Test.csv")

drops <- c("Var_1","Segmentation") data<-data[, !(names(data) %in% drops)]

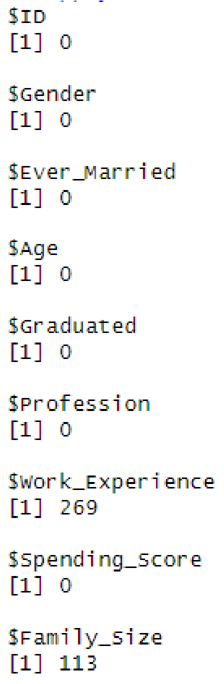
Using kable for aesthetics, viewing the top few rows of the datasetknitr::kable(head(data))



```
|     ID|Gender |Ever_Married | Age|Graduated |Profession | Work_Experience|Spending_Score | Family_Size|
|------:|:------|:------------|---:|:---------|:----------|---------------:|:--------------|-----------:|
| 458989|Female |Yes          |  36|Yes       |Engineer   |               0|Low            |           1|
| 458994|Male   |Yes          |  37|Yes       |Healthcare |               8|Average        |           4|
| 458996|Female |Yes          |  69|No        |           |               0|Low            |           1|
| 459000|Male   |Yes          |  59|No        |Executive  |              11|High           |           2|
| 459001|Female |No           |  19|No        |Marketing  |              NA|Low            |           4|
| 459003|Male   |Yes          |  47|Yes       |Doctor     |               0|High           |           5|
> |
```

Checking for Missing Values

To find the number of NAs in each variable lapply(data,function(x) { length(which(is.na(x)))})



```
$ID
[1] 0

$Gender
[1] 0

$Ever_Married
[1] 0

$Age
[1] 0

$Graduated
[1] 0

$Profession
[1] 0

$Work_Experience
[1] 269

$Spending_Score
[1] 0

$Family_Size
[1] 113
```

To find the number of blanks in each variable lapply(data,function(x) { length(which(x==""))})

There are blank values in categorical variables Graduated and Profession, replacing them with NAs for better handling data[data==""]<-NA
Replacing NA rows with mode
Creating a User defined function for finding Mode as R does not come with a base mode functiongetmode <- function(v)
{
uniqv <- unique(v) uniqv[which.max(tabulate(match(v, uniqv)))]
}

Replacing categorical variable NAs with their mode and continuous variable NAs with Mean

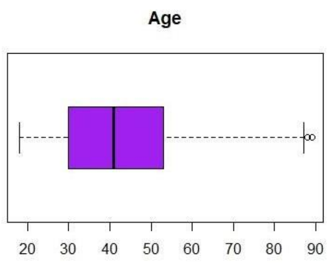
data$Ever_Married[is.na(data$Ever_Married)]<-getmode(data$Ever_Married)
data$Graduated[is.na(data$Graduated)]<-getmode(data$Graduated) data$Profession[is.na(data$Profession)]<-getmode(data$Profession) data$Work_Experience[is.na(data$Work_Experience)]<-mean(data$Work_Experience,na.rm=TRUE) data$Family_Size[is.na(data$Family_Size)]<-mean(data$Family_Size,na.rm=TRUE)

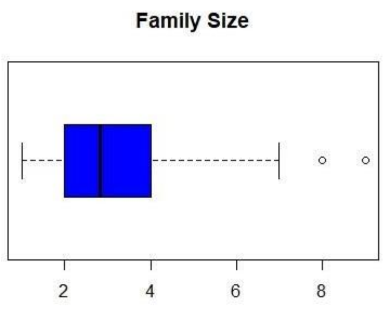
The missing values have successfully been replaced with the mean and mode. To find the underlying trends of each of the variables, univariate distributions are plotted.

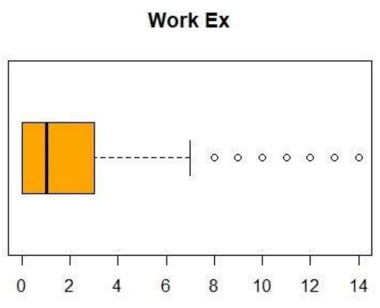
Data Characteristics:

- The general age of customers in this dataset lies between 18–60.
- Customers have a work experience of 2 -4 years with a few outliers having more than 10 years of workexperience
- Family size ranges between 2- 4
- There are more males than females in the customer base
- There are more married people than unmarried
- A major portion of the customers are graduates
- Artist is the most common profession seen among customers
- Most customers belong to the low spending score category

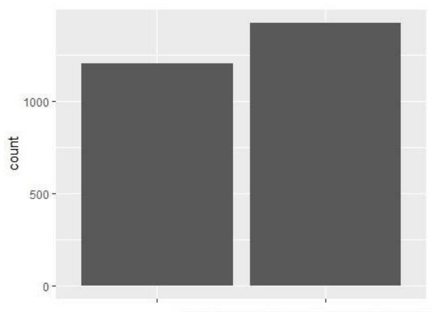
boxplot(data$Age, horizontal = TRUE,col = 'Purple',main="Age") boxplot(data$Work_Experience, horizontal = TRUE,col = 'Orange',main="Work Ex")boxplot(data$Family_Size, horizontal = TRUE,col = 'Blue',main="Family Size")

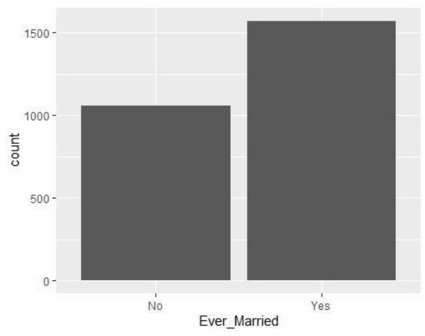


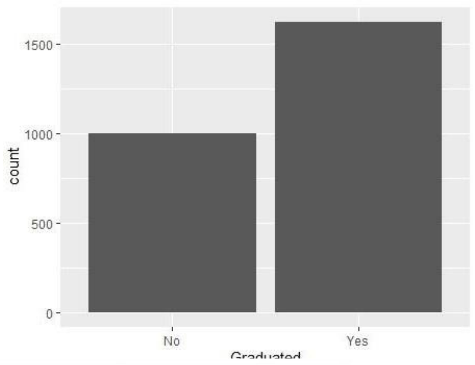


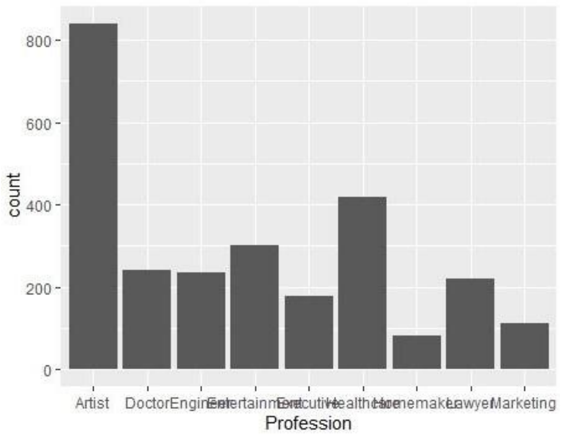


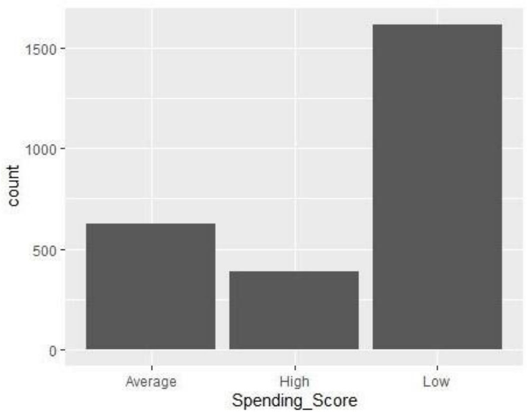
Boxplots of Continuous/ Ordinal Variables ggplot(data) + geom_bar(aes(x = Gender)) ggplot(data) + geom_bar(aes(x = Ever_Married))ggplot(data) + geom_bar(aes(x = Graduated)) ggplot(data) + geom_bar(aes(x = Profession)) ggplot(data) + geom_bar(aes(x = Spending_Score))











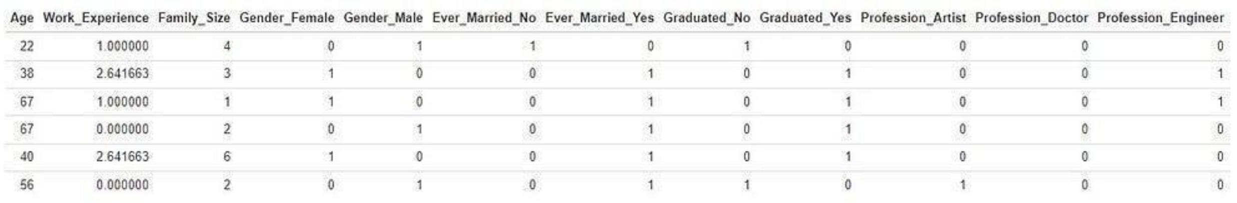
Since the given dataset has a lot of information stored in categorical variables like gender, profession, ever married etc. that cannot be comprehended by our machine, we need to encode them into a numerical format.
There are two options available for doing so:

1. Rank encoding: You basically assign each of the different classes of the variable a unique number. For e.g. while encoding Profession, Artist maybe assigned 1, Doctor as 2 and so on.
2. One Hot encoding: In this variation, each class of the categorical variable is spread across as a binary variable and marked 1 if it is present in the given data point and 0 if not. For e.g. If spending score were to be split into one hot encoded variable, there would be three new variables called High, Medium and Low and for a particular customer with a high spending score, the variable called High would be encoded as 1.

The algorithm we'll be using for clustering similar customers is K-Means (This video maybe helpful), which relies on the distance between two data points in an n dimensional space. It is prone to be affected by the magnitude of the underlying data, which means that a customer with an age of 19 would likely be far away from a customer with an age of 65 while plotting them as points in a n dimensional space, which makes sense for numerical variables. However if thevariables are categorical like profession, on rank encoding, a rank of 7 is not necessarily greater than a rank of 6. In this case, it is usually recommended to use one hot encoding and not rank encoding, since there is a chance that the algorithm misunderstands a relationship between two numbers when there is none. One Hot encoding categorical variables

cluster_data<-dummy_cols(data)
Dropping columns not required for analysis
cat <- c("ID","Gender","Ever_Married","Graduated","Profession","Spending_Score") cluster_data<-cluster_data[, !(names(cluster_data) %in% cat)] knitr::kable(head(cluster_data))

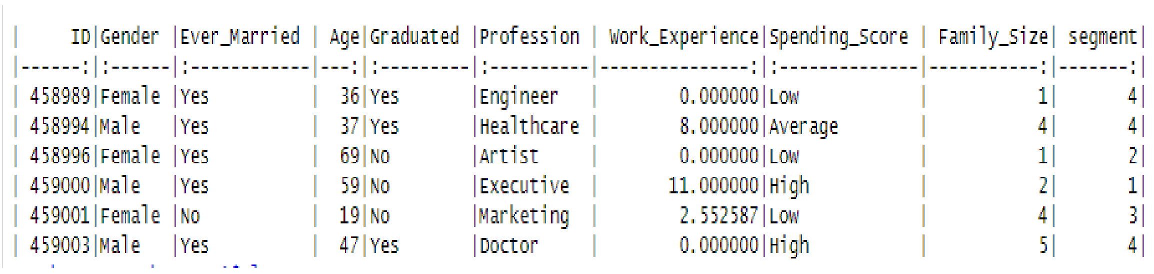
We are using all variables except ID for this exercise, since an ID variable is merely a unique identifier for each customer.It does not store any concealed information about the customer that should affect their segment and hence is excluded from the analysis.



Age	Work_Experience	Family_Size	Gender_Female	Gender_Male	Ever_Married_No	Ever_Married_Yes	Graduated_No	Graduated_Yes	Profession_Artist	Profession_Doctor	Profession_Engineer
22	1.000000	4	0	1	1	0	1	0	0	0	0
38	2.641663	3	1	0	0	1	0	1	0	0	1
67	1.000000	1	1	0	0	1	0	1	0	0	1
67	0.000000	2	0	1	0	1	0	1	0	0	0
40	2.641663	6	1	0	0	1	0	1	0	0	0
56	0.000000	2	0	1	0	1	1	0	1	0	0

As seen, the dataset is now reduced to a collection of neat numerical variables that can be easily understood by the machine.
To give a very rudimentary overview of what K Means does, it starts out with assigning k (a number that you specify) random cluster centers. The next step is to calculate the distance between these centers and all data points, like your traditional 2-dimensional coordinate geometry, only that the point is in a n dimensional space; something that our brain is not very adept at visualizing. The points are assigned to the cluster center closest to them. Once an iteration of such clustering data points is complete, the cluster centroid is recalculated and the process reiterated.
There is a well-established method of identifying the optimal number of clusters (k) to choose using a plot called an Elbow or a Scree plot, which I am not delving into here. We are choosing to keep 4 clusters here, which is what the dataset originally came with.

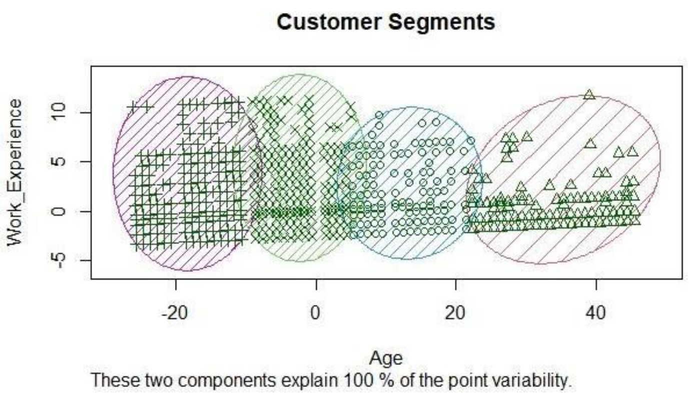
```
install.packages("ClusterR") install.packages("cluster") library(cluster) library(ClusterR) set.seed(240) # Setting seed
kmeans_1 <- kmeans(cluster_data, centers = 4, nstart = 20)#kmeans_1
#kmeans_1$cluster
# Assigning the segments back to the datasetdata$segment<-kmeans_1$cluster knitr::kable(head(data))
```



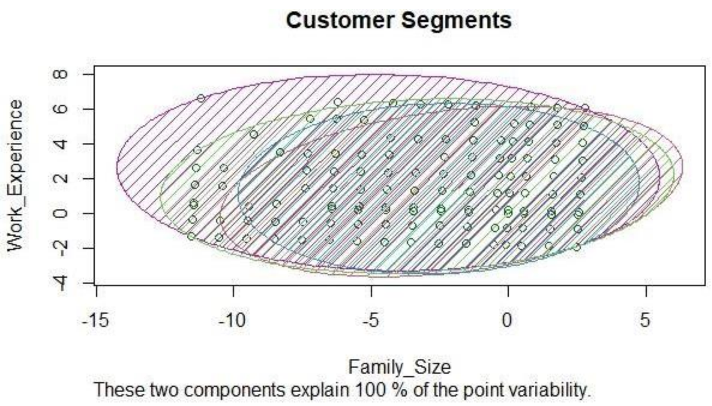
```
|     ID|Gender |Ever_Married | Age|Graduated |Profession | Work_Experience|Spending_Score | Family_Size| segment|
|------:|:------|:------------|---:|:---------|:----------|---------------:|:--------------|-----------:|-------:|
| 458989|Female |Yes          |  36|Yes       |Engineer   |        0.000000|Low            |           1|       4|
| 458994|Male   |Yes          |  37|Yes       |Healthcare |        8.000000|Average        |           4|       4|
| 458996|Female |Yes          |  69|No        |Artist     |        0.000000|Low            |           1|       2|
| 459000|Male   |Yes          |  59|No        |Executive  |       11.000000|High           |           2|       1|
| 459001|Female |No           |  19|No        |Marketing  |        2.552587|Low            |           4|       3|
| 459003|Male   |Yes          |  47|Yes       |Doctor     |        0.000000|High           |           5|       4|
```

Each data point, which represents a customer has now been assigned a cluster number ranging from 1–4. In order to better understand what each of these clusters i.e. customer segments represent, they are visualized on two-dimensional scatter plots. Since it is not possible to visualize the cluster in a multi-dimensional space using all the variables that were used, we have plotted three separate plots of clusters using the continuous variables of Age, Work Experience and Family Size.

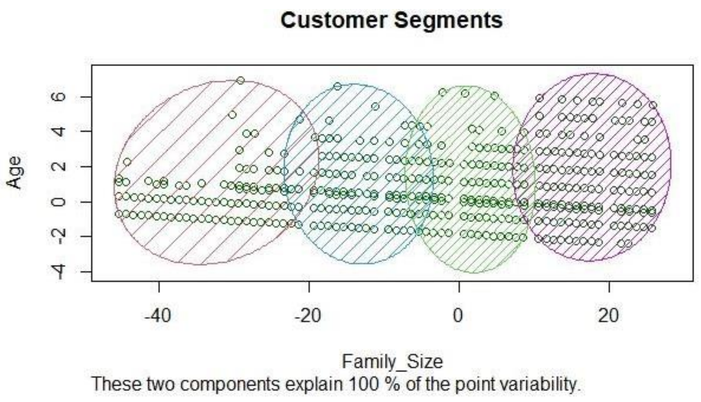
```
y_kmeans <- kmeans_1$cluster
# Visualizing segments in terms of Work experience and Ageclusplot(data[, c("Age", "Work_Experience")],
y_kmeans, lines = 0, shade = TRUE,color = TRUE,
labels = 0, # To remove data labels from the plotplotchar = TRUE,
span = TRUE,
main = paste("Customer Segments"),
xlab = 'Age',
ylab = 'Work_Experience')
```



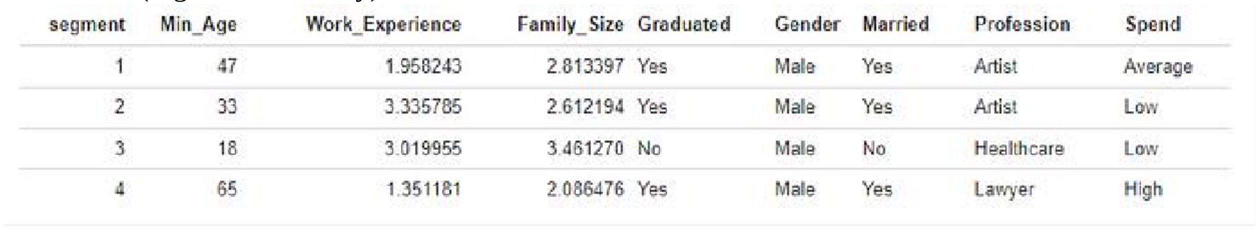
```
clusplot(data[, c("Family_Size", "Work_Experience")],y_kmeans,
lines = 0, shade = TRUE,color = TRUE, labels = 0,
plotchar = FALSE,span = TRUE,
main = paste("Customer Segments"),xlab = 'Family_Size',
ylab = 'Work_Experience')
```



```
clusplot(data[, c("Family_Size", "Age")],y_kmeans,
lines = 0, shade = TRUE,color = TRUE, labels = 0,
plotchar = FALSE,span = TRUE,
main = paste("Customer Segments"),xlab = 'Family_Size',
ylab = 'Age')
```



Cluster Profiling:
Summarizing customer attributes in each of the segments;data %>% group_by(segment) %>%
summarise(Min_Age=min(Age), Work_Experience=mean(Work_Experience),Family_Size=mean(Family_Size),
Graduated=getmode(Graduated),Gender=getmode(Gender),
Married=getmode(Ever_Married),Profession=getmode(Profession), Spend=getmode(Spending_Score))
knitr::kable(segment_summary)



segment	Min_Age	Work_Experience	Family_Size	Graduated	Gender	Married	Profession	Spend
1	47	1.958243	2.813397	Yes	Male	Yes	Artist	Average
2	33	3.335785	2.612194	Yes	Male	Yes	Artist	Low
3	18	3.019955	3.461270	No	Male	No	Healthcare	Low
4	65	1.351181	2.086476	Yes	Male	Yes	Lawyer	High

Output:

Interpretation and Implications:

On understanding the average attributes, we observe that the age group varies greatly across each of the segmentswhile family size and Work experience are comparable across segments 1 and 4.

Most of the customer segments are dominated by males, which is the overall trend in the underlying dataset as well. Segments 1,2 and 4 are dominated by Married persons while segment 3 has more unmarried people, which is in line with the observation that it is a younger segment of customers with a minimum age of 18.

It is also interesting to note that the segments 2 & 3 with younger customers i.e. 18–45-year-old have low spent scores whereas segments 1 and 4 with middle aged and elderly customers have average to high spend scores.

Based on this analysis, the automobile dealer’s marketing team can identify what kind of communication to send to each of these segments. Segments 1 and 4 can be targeted for luxury Sedans since they have good spending scores and not typically large family sizes. Segment 3 can be targeted for entry level cars and Segment 2 can be targeted for mid-sized SUVs.

Practicals 8: A/B Testing for Marketing Strategies And Data-Driven Decisions

Conduct A/B testing to evaluate the impact of different marketing strategies and make data-driven decisions.

- Design and implement A/B tests for marketing campaigns using randomized assignment.
- Collect relevant data and perform statistical analysis to compare the performance of differentstrategies.
- Calculate key metrics such as conversion rates, clickthrough rates, or revenue.
- Interpret the results and provide recommendations for optimizing marketing campaigns based on thefindings.

Code:

This dataset contains A/B test results for an e-commerce website. A subset of users was exposed to a new landing page, and the current goal is to assess the effect of this new page on the conversion rate.library(tidyverse)
library(infer) library(scales) library(forcats)
ab_data <- read_csv("F://ab_data.csv",show_col_types = FALSE)sample_n(ab_data, 5)

```
user_id timestamp group   landing_page converted
  <dbl>           <       <          <dbl
  <time>          c       c          >
                  h       hr
                  r       >
                  >
  1               control  old_page
  710525
  59:23
  2               treatment
  828100          new_page
  15:26
  3               treatment
  917542          new_page
  35:11
  4               control  old_page
  923350
  14:00
  5               treatment
  936644          new_page
  02:36
```

ab_data <- ab_data %>%
mutate(landing_page = fct_rev(landing_page), # set order to (old_page, new_page)converted = factor(converted)) # categorical required in infer proportion tests

Exploratory Data Analysis
There were 147,202 users in the control group and 147,278 users in the treatment group.

```
ab_data %>% group_by(group) %>%tally()
1   control  147202
2   treatment 147278
```

Some of the users appear more than once:ab_data %>%
group_by(user_id) %>%
add_tally(name = "num_of_appearances") %>% arrange(desc(num_of_appearances), user_id) %>%

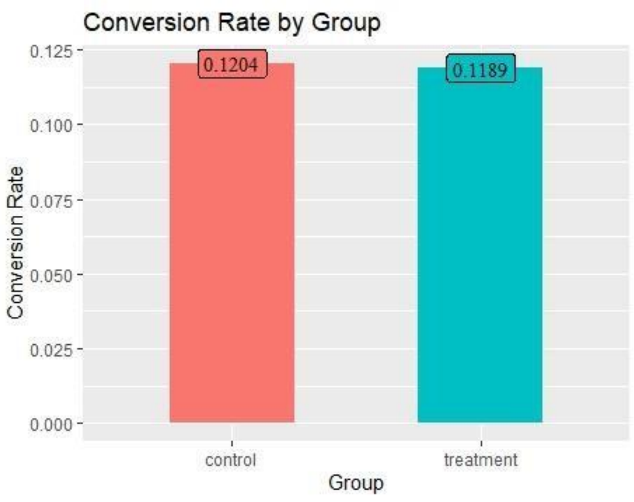
head()

```
user_id timestamp group   landing_page converted num_of_appearances
  <dbl>                   <         <         <fct>           <int
  <time>                  c         f                         >
                          h         c
                          r         t
                          >         >
  1                       treatment               1           2
  630052                  old_page
  25:54
  2                       treatment               0           2
  630052                  new_page
  16:05
  3                       treatment               0           2
  630126                  old_page
  35:54
  4                       treatment               0           2
  630126                  new_page
  16:00
  5                       control                 0           2
  630137                  new_page
  59:22
  6                       control                 0           2
  630137                  old_page
  08:49
```

The conversion rate is slightly higher in the control group (0.1204 vs 0.1189), indicating that the new landing page performed poorly.

```
conversion_rate_by_group <- ab_data %>%group_by(group) %>%
summarise(conversion_rate = mean(converted == "1"))conversion_rate_by_group
group   conversion_rate
<chr>           <dbl>
1                         control  0.120
2                           treatment       0.119
```

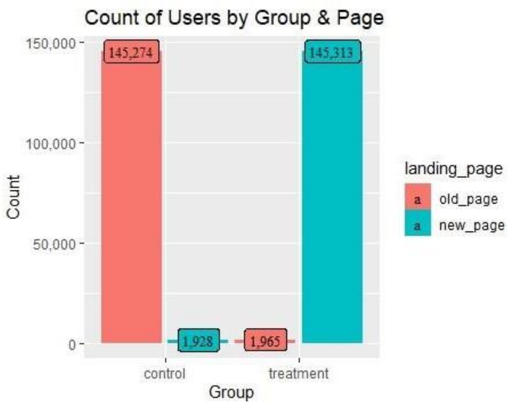
```
conversion_rate_by_group %>%
ggplot(aes(x = group, y = conversion_rate, fill = group)) +geom_col(width = 0.5) +
geom_label(aes(label = label_number()(conversion_rate)), family = "serif") + labs(x = "Group", y = "Conversion Rate",
title = "Conversion Rate by Group") +guides(fill = "none")
```



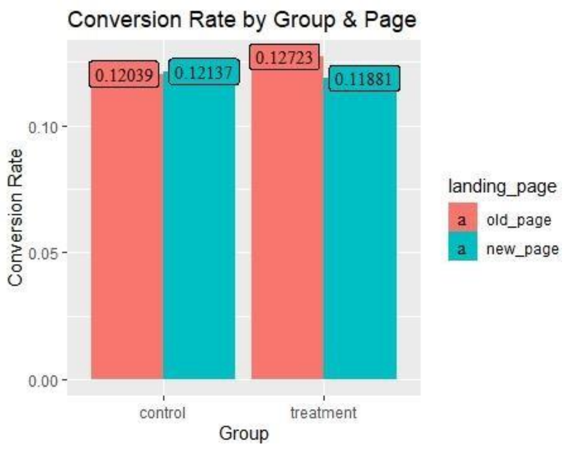
Some of the users in the control group were exposed to the new page, and some of the users in the treatment group were exposed to the old page:

ab_data %>%
group_by(group, landing_page) %>%summarise(count = n()) %>%
ggplot(aes(x = group, y = count, fill = landing_page)) +geom_col(position = position_dodge(width=1)) +
geom_label(aes(label = comma(count)), position = position_dodge(width=1), size=3.5, family = "serif") +labs(title = "Count of Users by Group & Page", x = "Group") +
scale_y_continuous(name = "Count", labels = comma)

`summarise()` has grouped output by 'group'. You can override using the
`.groups` argument.



The new page did better in the control group. The old page did better in the treatment group:ab_data %>%
group_by(group, landing_page) %>% summarise(conversion_rate = mean(converted == "1")) %>% ggplot(aes(x = group, y = conversion_rate, fill = landing_page)) +geom_col(position = position_dodge(width=)) +
geom_label(aes(label = label_number()(conversion_rate)), family = "serif", position = position_dodge(width = 1)) +
labs(x = "Group", y = "Conversion Rate", title = "Conversion Rate by Group & Page")



Inference

A. 95% Confidence Interval

Let's construct a 95% confidence interval for the difference in conversion rates between the test group and the control group. The observed value from our data is -0.001481, which is quite small

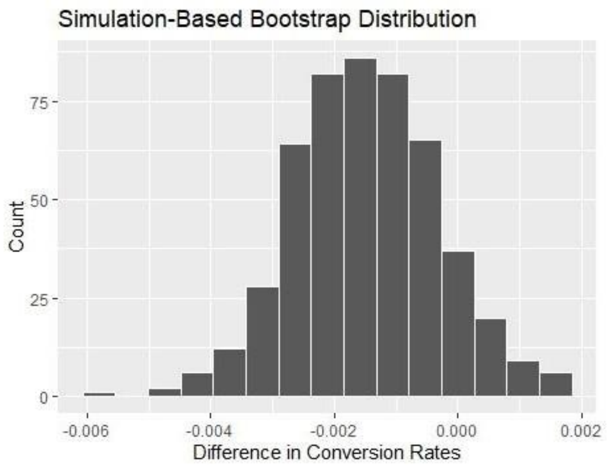
diff_in_proportions <- ab_data %>% specify(converted ~ group, success = "1") %>%
calculate("diff in props", order = c("treatment", "control"))diff_in_proportions

```
A tibble: 1 × 1stat
                                  <dbl>
                                  1 -0.00148
```

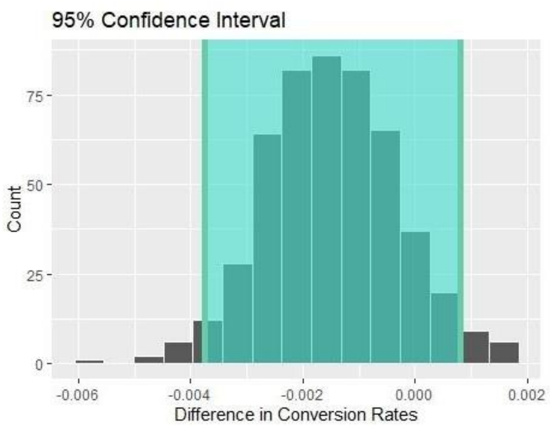
We'll proceed as follows:
estimate the sampling distribution of the difference in proportions (conversion rates) using bootstrap resamplinguse the estimated standard error from the bootstrap distribution to create the confidence interval.

```
bootstrap_dist <- ab_data %>% specify(converted ~ group, success = "1") %>%generate(reps = 500, type = "bootstrap")
%>%
calculate(stat = "diff in props", order = c("treatment", "control"))

visualise(bootstrap_dist) +
labs(x = "Difference in Conversion Rates", y = "Count")
```



```
ci <- get_ci(bootstrap_dist, level = 0.95, type = "se", point_estimate = diff_in_proportions)ci
lower_ci upper_ci
<dbl>   <dbl>
1 -0.00376 0.000796
```



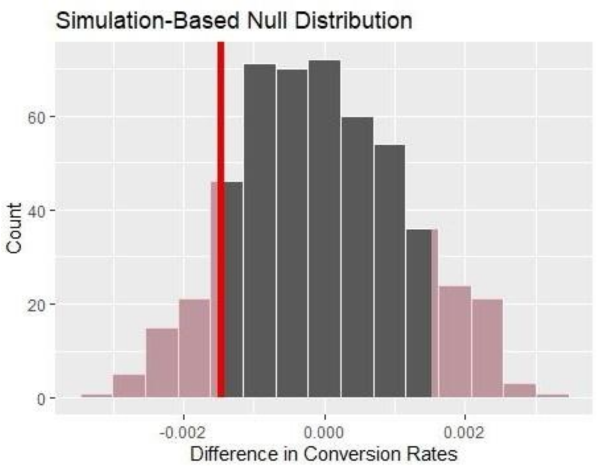
We can say with 95% certainty that the difference in conversion rates is in the range (-0.003795, 0.000832).
Zero is inside this confidence interval, implying that the difference in conversion rates is quite possibly not statistically significant (the difference can be zero i.e. no difference).

B. Hypothesis Testing

Is the conversion rate for the treatment group significantly different from that of the control group?

I. Simulation

This involves approximating the sampling distribution of the difference in conversion rates under the assumption that the null hypothesis is true. null_dist <- ab_data %>%
specify(converted ~ group, success = "1") %>%hypothesize(null = "independence") %>%
generate(reps = 500, type = "permute") %>% # shuffling calculate(stat = "diff in props", order = c("treatment", "control")) visualise(null_dist) +
shade_p_value(obs_stat = diff_in_proportions, direction = "both") +labs(x = "Difference in Conversion Rates", y = "Count")



get_p_value(null_dist, obs_stat = diff_in_proportions, direction = "both")p_value
<dbl> 1 0.212

We fail to reject the null hypothesis at level of significance 0.05 since the *p-value* 0.18 > 0.05. We do not have sufficient evidence that the difference in conversion rates is statistically significant.

II. Theoretical

prop_test can be used to test the null hypothesis that the proportions in several groups are similar (difference = 0). It performs Pearson's Chi-squared test, which requires that:
sample data is drawn at random (depends on how this dataset was collected - we'll assume it was random)the sample is sufficiently large (this condition is met)
observations are independent of each other (some users appear more than once - not fully met)prop_test(ab_data, converted ~ group, order = c("control", "treatment"),
conf_level = 0.95, success = "1")

```
statistic chisq_df p_value alternative  lower_ci upper_ci
                  <dbl>   <dbl>  <dbl> <chr> <dbl>   <dbl>
             1      1.52       1  0.218 two.sided  -0.000870  0.00383
```

Once again, the p-value 0.2177 > 0.05 and we fail to reject the null hypothesis.

Output:

Interpretation and Implications:

The new landing page did not boost the conversion rate. The difference inconversion rates between the new page and the old one is not statistically significant.

We need to investigate what aspects of the new landing page failed to motivate purchases, and make the necessary improvements.