**Retail Marketing Analytics Practical**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Code PSDSP506b** | **Course Title** | **DATE** | **REMARK** |
| 1 | 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. |  |  |
| 2 | Gain proficiency in visualizing marketing data using R.  Understand the key elements of data visualization.   * Create various visualizations such as histograms,scatterplots, line plots, and bar charts using the ggplot() function in R. * Apply appropriate visualization techniques to effectively communicate marketing insights. |  |  |
| 3 | 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. |  |  |
| 4 | 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. |  |  |
| 5 | 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 CLV prediction. * Assess the accuracy and reliability of CLV predictions. |  |  |
| 6 | 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 marketing strategies. |  |  |
| 7 | Extract data from social media platforms and perform analysis to gain insights into customer behavior and preferences.   * Utilize Python libraries like Beautiful Soup and requests to scrape data from social media platforms. * Clean and preprocess the scraped data. * Analyze the data to identify trends, sentiment analysis, or customer engagement metrics. * Visualize the findings using appropriate charts or graphs. |  |  |
| 8 | Analyze customer purchasing patterns and build a recommender system based on market basket analysis.   * Use transactional data to identify frequently occurring item sets using association rule mining algorithms. * Calculate support, confidence, and lift for the identified item sets. * Build a recommendation engine using collaborative filtering techniques. * Evaluate the performance of the recommender system and make recommendations based on customer preferences. |  |  |
| 9 | Segment customers based on their recency, frequency, and 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 kmeans or hierarchical clustering. * Perform descriptive analysis on each customer segment to understand their characteristics. * Develop targeted marketing strategies for each segment based on their RFM profiles |  |  |
| 10 | 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 different strategies. * Calculate key metrics such as conversion rates, click through rates, or revenue. * Interpret the results and provide recommendations for optimizing marketing campaigns based on the findings |  |  |

**PRACTICAL NO:01**

**AIM:**

**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.

**INPUT :-**

library(dplyr)

set.seed(123)

transaction\_data <- data.frame(

CustomerID = rep(1:100, each = 5),

TransactionDate = sample(seq(as.Date('2022-01-01'), as.Date('2023-01-01'), by = "day"),

500, replace = TRUE),

PurchaseAmount = round(runif(500, min = 10, max = 200), 2)

)

write.csv(transaction\_data, "transaction\_data.csv", row.names = FALSE)

transaction\_data <- read.csv("transaction\_data.csv")

head(transaction\_data)

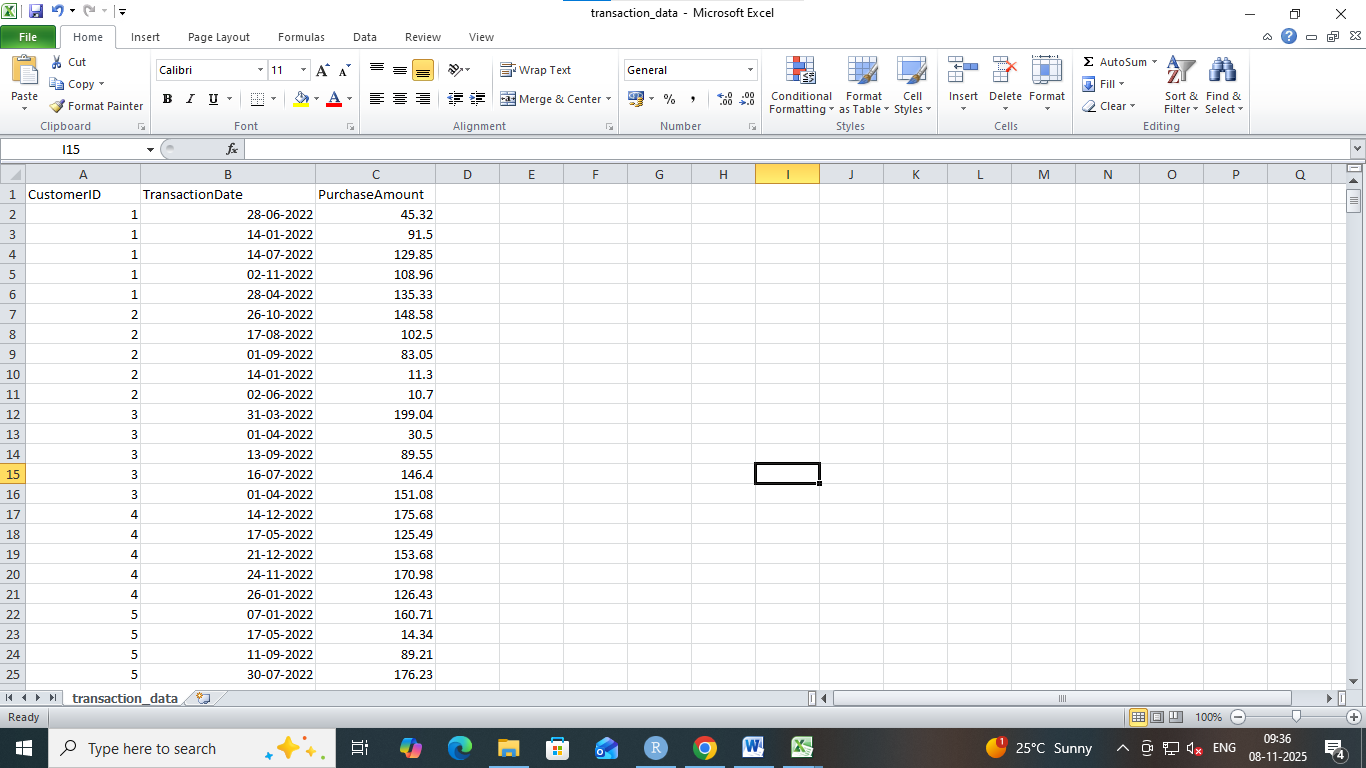
transaction\_data <- read.csv("transaction\_data.csv")

transaction\_data$TransactionDate <- as.Date(transaction\_data$TransactionDate)

str(transaction\_data)

**OUTPUT :-**





**INPUT :-**

head(airquality)

mean(airquality$Wind)

mean(airquality$Solar.R, na.rm = TRUE)

summary(airquality)

x1 <- c(3, 5, 3, 7, 9, 4, 6)

x2 <- c(-1, -4, 2.4, 6, -7)

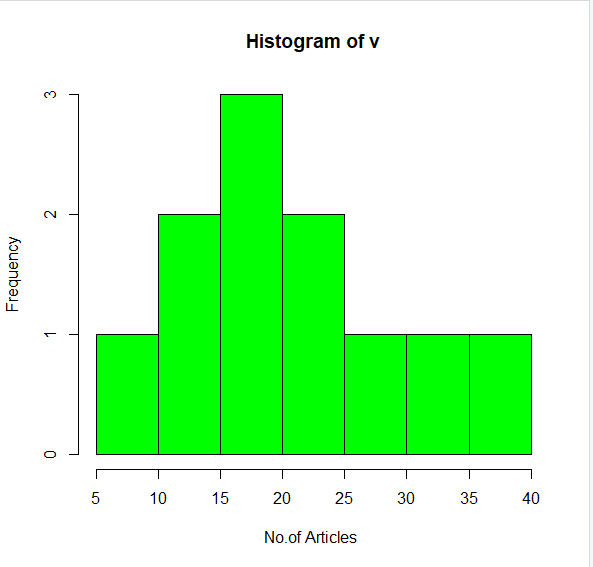
tabulate(x1)

tabulate(x2)

v <- c(19, 23, 11, 5, 16, 21, 32, 14, 19, 27, 39)

hist(v, xlab = "No.of Articles ", col = "green", border = "black")

**OUTPUT :-**



**INPUT :-**

library(readr)

library(dplyr)

library(ggplot2)

salary\_data <- read.csv("C:\\Users\\LAB PC\\Downloads\\salary.csv")

print(salary\_data)

salary <- na.omit(salary\_data)

print(salary)

count\_edu <- table(salary$EDU)

print(count\_edu)

avg\_edu <- mean(salary$EDU, na.rm = TRUE)

print(avg\_edu)

median\_edu <- median(salary$EDU, na.rm = TRUE)

print(median\_edu)

highest\_edu <- max(salary$EDU, na.rm = TRUE)

lowest\_edu <- min(salary$EDU, na.rm = TRUE)

print(paste("Highest Education Level:", highest\_edu))

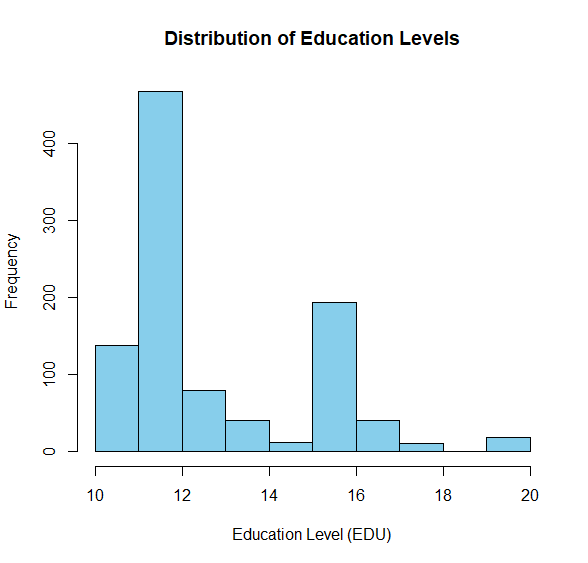
print(paste("Lowest Education Level:", lowest\_edu))

quartiles\_edu <- quantile(salary$EDU)

print(quartiles\_edu)

hist(salary$EDU, main = "Distribution of Education Levels", xlab = "Education Level (EDU)", ylab = "Frequency", col = "skyblue", border = "black")

**OUTPUT :-**



**PRACTICAL NO:02**

**AIM:**

**Gain proficiency in visualizing marketing data using R.**

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

**Bar Plot**

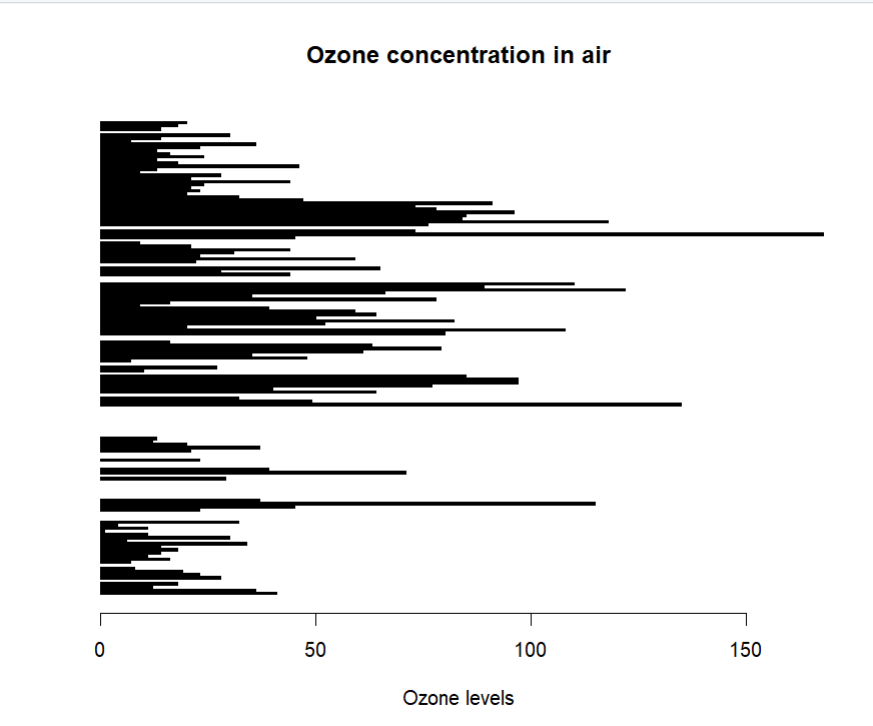
**INPUT :-**

# Horizontal Bar Plot for

# Ozone concentration in air

barplot(airquality$Ozone, main='Ozone concentration in air', xlab='Ozone levels',col='black', horiz=TRUE)

OUTPUT :-



**INPUT :-**

# Vertical Bar Plot for

# Ozone concentration in air

barplot(airquality$Ozone, main='Ozone concentration in air', xlab='Ozone levels', col='blue',horiz=FALSE)

**OUTPUT :-**



**Histogram**

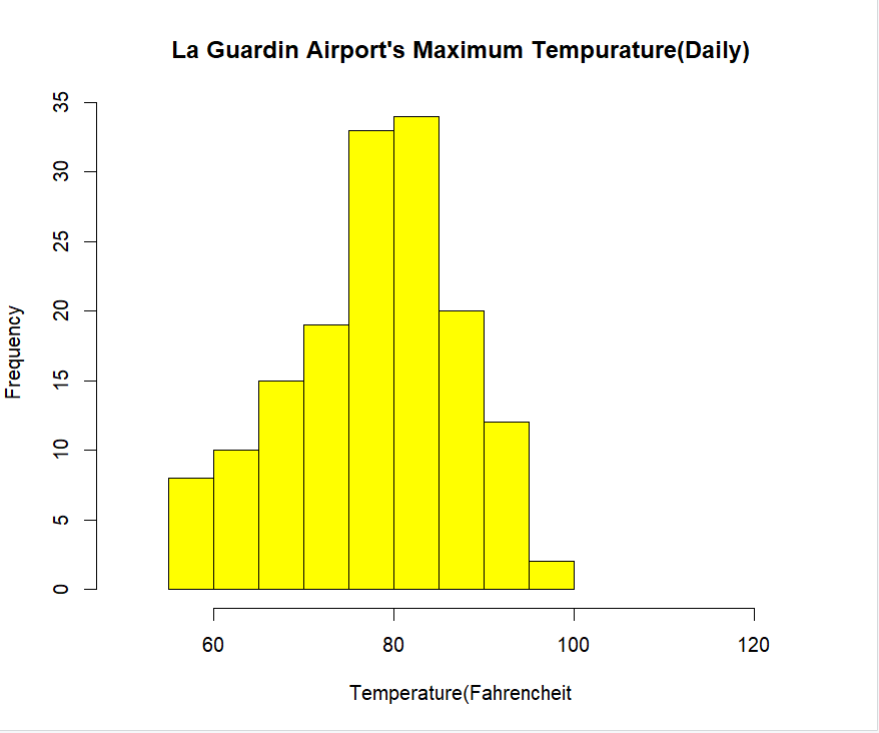
**INPUT :-**

# Histogram for Maximum Daily Temperature

data(airquality)

hist(airquality$Temp, main="La Guardin Airport's\ Maximum Tempurature(Daily)", xlab = "Temperature(Fahrencheit", xlim = c(50,125), col = "yellow", freq = TRUE)

**OUTPUT :-**



**Box Plot**

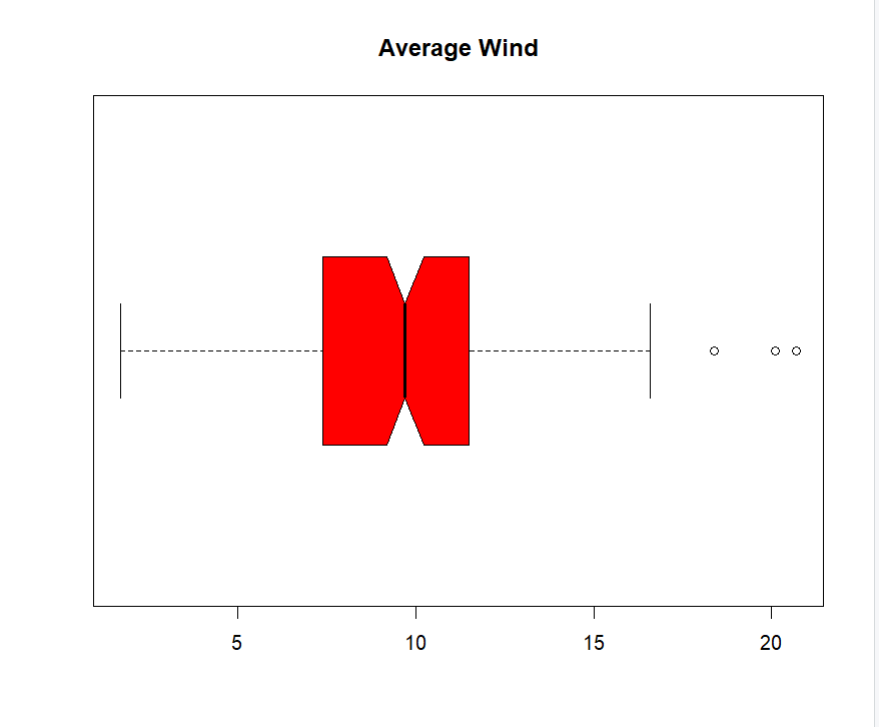
**INPUT :-**

# Box plot for average wind speed

data(airquality)

boxplot(airquality$Wind, main="Average Wind", col = "red", border = "black", horizontal = TRUE, notch=TRUE)

**OUTPUT :-**



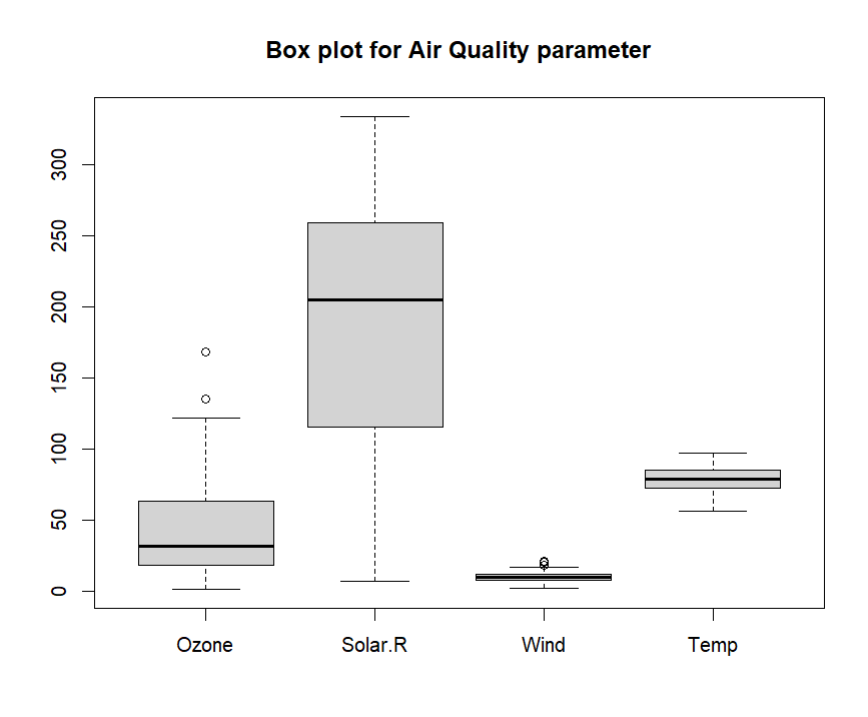
**INPUT :-**

#Multiple box plots, each representing

# an Air Quality Parameter

boxplot(airquality[,0:4], main = 'Box plot for Air Quality parameter')

**OUTPUT :-**



**Practical 2.2: Create various visualizations such as histograms, scatter plots, line plots, and bar charts using the ggplot() function in R.**

**INPUT :-**

**Bar Graph using ggplot2**

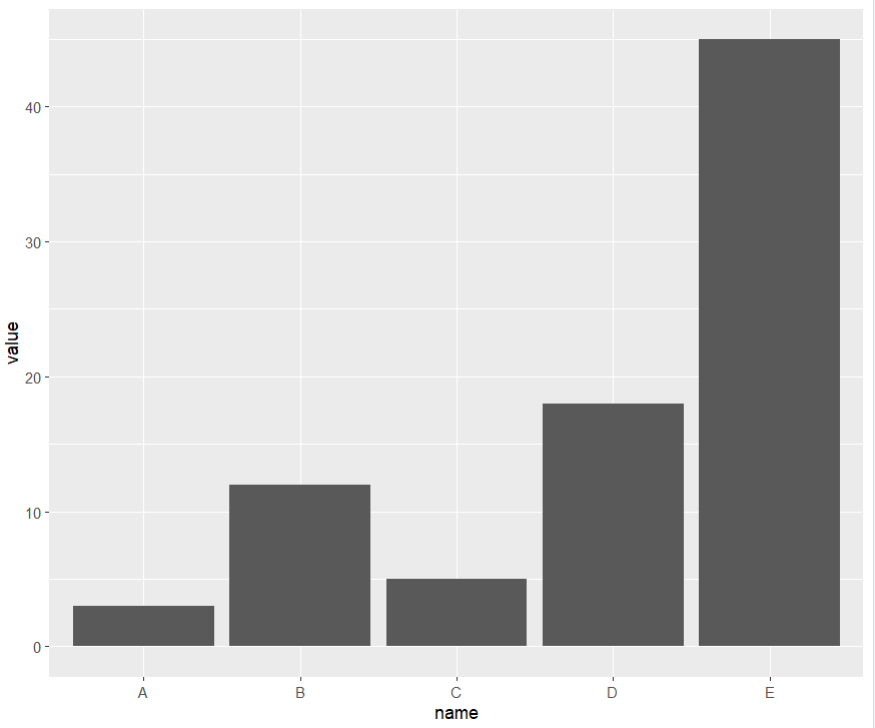
# Loading ggplot2

library(ggplot2)

data <- data.frame(name = c ("A","B","C","D","E"), value = c (3,12,5,18,45))

ggplot(data, aes(x=name, y=value)) + geom\_bar(stat = "identity")

**OUTPUT :-**



**INPUT :-**

**Histogram using ggplot2**

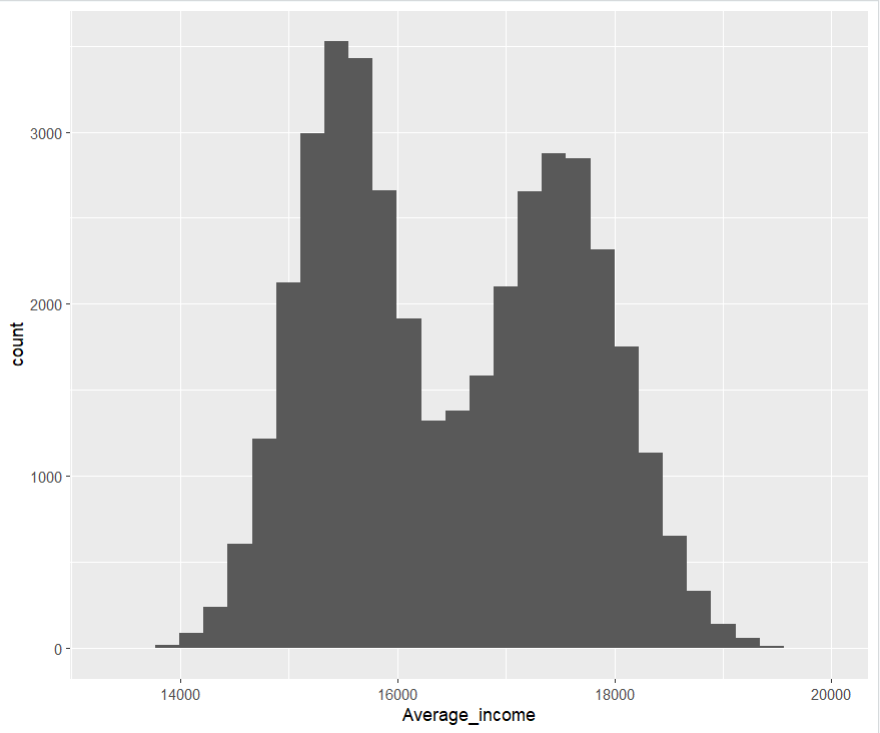
df <- data.frame(gender=factor(rep(c("Average Female income","Average Male income"), each=20000)), Average\_income=round(c(rnorm(20000, mean=15500, sd=500), rnorm(20000, mean=17500, sd=600))))

head(df)

library(ggplot2)

ggplot(df, aes(x=Average\_income)) + geom\_histogram()

**OUTPUT :-**



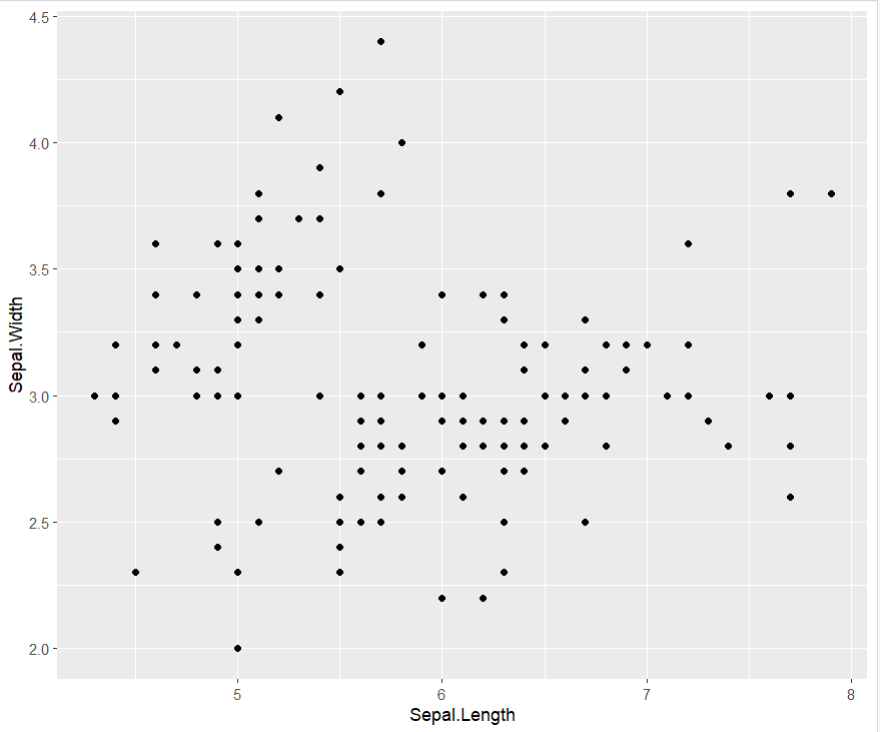
**INPUT :-**

**Scater diagram using ggplot2**

data("airquality")

ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) + geom\_point()

**OUTPUT :-**



**INPUT :-**

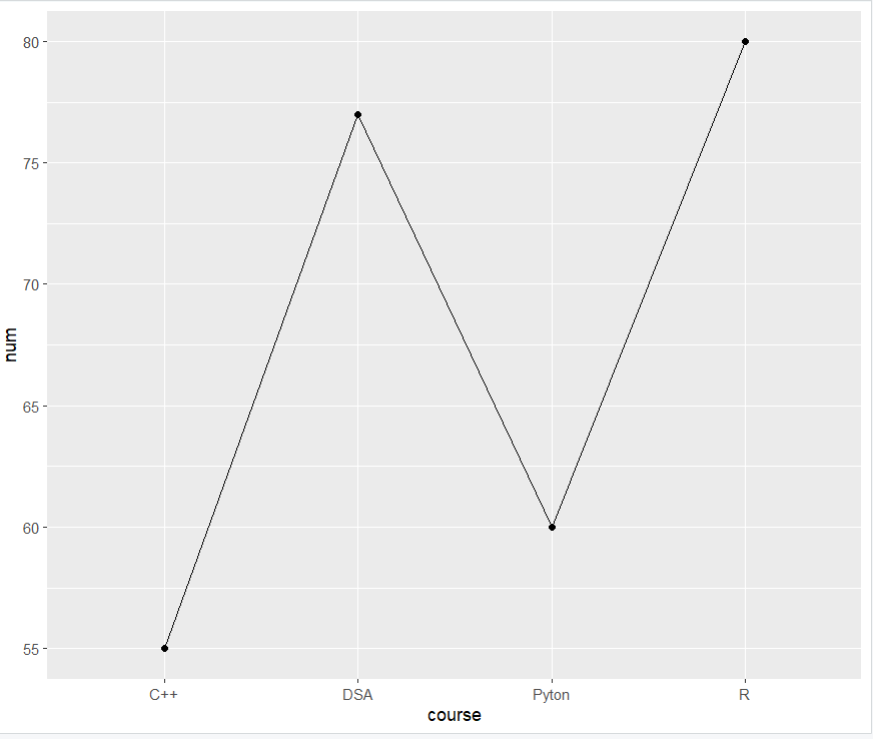
**Line plot using ggplot2**

# Create data for chart

val <- data.frame(course=c('DSA','C++','R','Pyton'), num=c(77,55,80,60))

ggplot(data = val, aes(x=course, y=num, group=1)) + geom\_line() + geom\_point()

**OUTPUT :-**



**PRACTICAL NO:03**

**AIM:**

**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.

**Method 1: Using base R**

**The sampling method has the following documentation in R :**

**INPUT :-**

# Creating the dataset

mat <- matrix(

1:21,

nrow = 7,

ncol = 3,

byrow = TRUE

)

print("Dataset")

print(mat)

# Divide the matrix into training (70%) and testing (30%)

split\_index <- sample(

c(TRUE, FALSE),

nrow(mat),

replace = TRUE,

prob = c(0.7, 0.3)

)

# Creating training dataset

train\_dataset <- mat[split\_index, ]

# Creating testing dataset

test\_dataset <- mat[!split\_index, ]

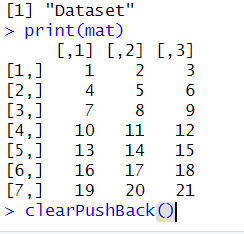
print("Training Dataset")

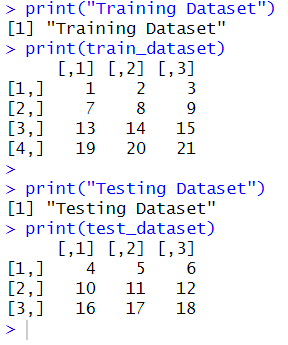
print(train\_dataset)

print("Testing Dataset")

print(test\_dataset)

**OUTPUT :-**





**Method 2: Using dplyr package in R**

**INPUT :-**

library(dplyr)

# Creating a data frame

data\_frame <- data.frame(

col1 = 1:15,

col2 = letters[1:15],

col3 = c(0,1,1,1,0,0,0,0,0,1,1,0,1,1,0)

)

print(data\_frame)

# 70% training data

training\_dataset <- data\_frame %>% sample\_frac(0.7)

print("Training Dataset")

print(training\_dataset)

# Remaining 30% testing data

testing\_dataset <- anti\_join(

data\_frame,

training\_dataset,

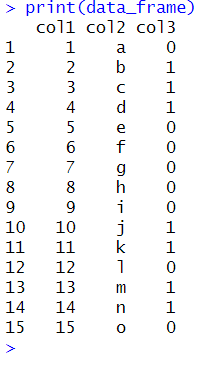
by = "col1"

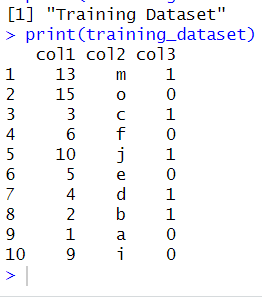
)

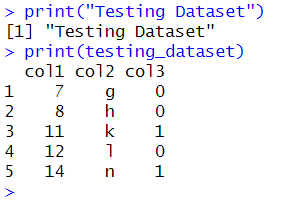
print("Testing Dataset")

print(testing\_dataset)

**OUTPUT :-**







**Method 3: Using catools package in R**

**INPUT :-**

library(caTools)

data\_frame <- data.frame(

col1 = 1:15,

col2 = letters[1:15],

col3 = c(0,1,1,1,0,0,0,0,0,1,1,0,1,1,0)

)

print(data\_frame)

# Creating a sample split (60:40)

split <- sample.split(data\_frame$col3, SplitRatio = 0.6)

print("Training Dataset")

training\_dataset <- subset(data\_frame, split == TRUE)

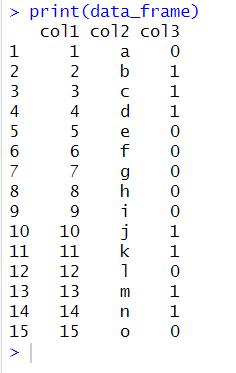
print(training\_dataset)

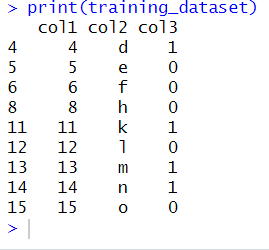
print("Testing Dataset")

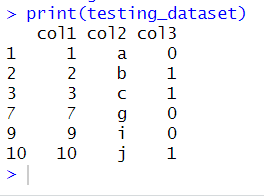
testing\_dataset <- subset(data\_frame, split == FALSE)

print(testing\_dataset)

**OUTPUT :-**







**PRACTICAL NO:04**

**AIM:**

**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.

**INPUT :-**

pwr.anova.test(k=4,f=.25,sig.level=.05,power=.8)

pwr.t.test(n=12,d=0.75,sig.level=.05,alternative="greater")

pwr.2p.test(n=20,sig.level=0.05,power=0.75)

if(!require(lsr)){install.packages("lsr")}

binom.test(5, 100, 0.5)

Input =("

+ Classroom Passed Failed

+ A 8 2

+ B 3 7

+ ")

Matrix = as.matrix(read.table(textConnection(Input),

+ header=TRUE,

+ row.names=1))

Matrix

boxplot(cbind(Class.C, Class.D))

**OUTPUT :-**



**PRACTICAL NO:05**

**AIM:**

**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 CLV prediction.
* Assess the accuracy and reliability of CLV predictions.

**INPUT :-**

library(dplyr) library(ggplot2) set.seed(123)

customers <- data.frame( customer\_id = 1:100, total\_spend = runif(100, min = 100, max = 1000), total\_orders = sample(1:10, 100, replace = TRUE), tenure\_months = sample(12:120, 100, replace = TRUE)

)

print(customers) customers$avg\_spend\_per\_order <- customers$total\_spend / customers$total\_orders print(customers$avg\_spend\_per\_order) lm\_model <- lm(avg\_spend\_per\_order ~ total\_orders + tenure\_months, data = customers) summary(lm\_model)

ggplot(customers, aes(x = total\_orders, y = avg\_spend\_per\_order)) + geom\_point(color = "blue") + # Actual data points geom\_smooth(method = "lm", se = FALSE, color = "red") print(head(customers)) avg\_spend\_per\_order <- customers$total\_spend / customers$total\_orders print(avg\_spend\_per\_order)

print(head(avg\_spend\_per\_order))

customers$high\_spender <- ifelse(avg\_spend\_per\_order > 200, 1, 0) print(customers$high\_spender) print(head(customers$high\_spender))

customers$high\_spender <- factor(customers$high\_spender, levels = c(0, 1), labels = c("Not HighSpender", "High Spender"))

logit\_model <- glm(high\_spender ~ total\_orders + tenure\_months, data = customers, family =

"binomial") print(logit\_model) summary(logit\_model) ggplot(customers, aes(x = total\_orders, y = tenure\_months, color = high\_spender)) + geom\_point() + geom\_smooth(method = "glm", method.args = list(family = "binomial"), se = FALSE) + labs(title = "Logistic Regression: Predicting High Spenders", x = "Total Orders",

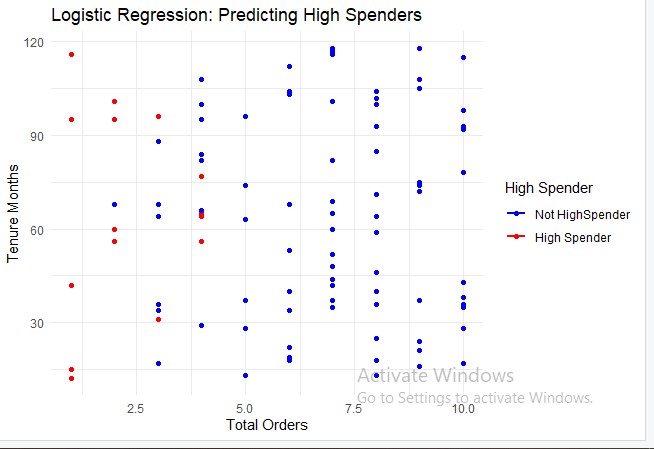
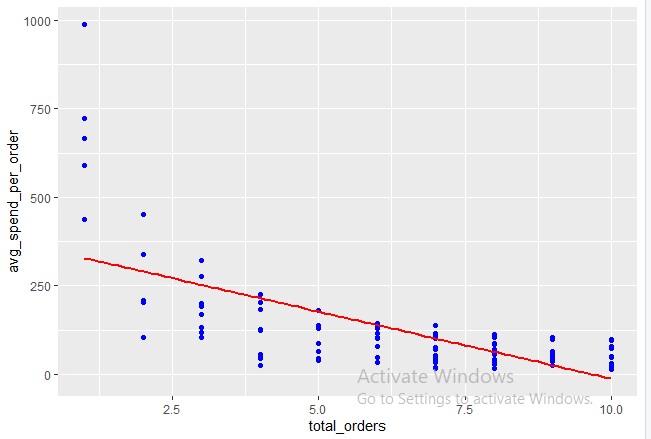
y = "Tenure Months", color = "High Spender") + scale\_color\_manual(values = c("blue", "red")) + # Customizing color scale theme\_minimal() # Minimal theme

install.packages('ggplot2')

#customers$high\_spender <- as.factor(customers$high\_spender) print(customers$high\_spender) ggplot(customers, aes(x = total\_orders, y = as.numeric(high\_spender) - 1, color = high\_spender)) + geom\_point() + geom\_smooth(method = "glm", method.args = list(family = "binomial"), se = FALSE) + labs(title = "Logistic Regression: Predicting High Spenders", x = "Total Orders",

y = "Probability of Being a High Spender", color = "High Spender") + scale\_color\_manual(values = c("blue", "red")) + theme\_minimal()

**OUTPUT :-**



**PRACTICAL NO:06**

**AIM:**

**Apply CLV analysis and cohort analysis in marketing analytics.**

* Analyze CLV data and identify patterns and trends.
* Perform cohort analysis to segment customers based ontheir behavior or characteristics.
* Interpret the results of CLV analysis and cohort analysis to derive actionable insights for marketing strategies.

**INPUT :-**

library(dplyr) library(ggplot2) install.packages('dplyr')

install.packages('ggplot2') set.seed(123) avg\_purchase\_value = customers$total\_spend /customers$total\_orders print(avg\_purchase\_value) purchase\_frequency = customers$total\_orders / customers$tenure\_months print(purchase\_frequency) clv = avg\_purchase\_value \* purchase\_frequency \* customers$tenure\_months head(clv) ggplot(customers, aes(x = clv)) + geom\_histogram(binwidth = 50, fill = "blue", color = "black") + labs(title = "Distribution of CLV", x = "Customer Lifetime Value", y = "Frequency") +

theme\_minimal() summary(customers$clv)

**OUTPUT :-**



**PRACTICAL NO :07**

**AIM:**

**Extract data from social media platforms and perform analysis to gain insights into customer behavior and preferences.**

* Utilize Python libraries like Beautiful Soup and requests toscrape data from social media platforms.
* Clean and preprocess the scraped data.
* Analyze the data to identify trends, sentiment analysis, or customer engagement metrics.
* Visualize the findings using appropriate charts or graphs.

**INPUT :-**

install.packages("dplyr")

library(dplyr)

install.packages("ggplot2")

library(ggplot2)

install.packages("lubridate")

library(lubridate)

transactions <- data.frame(

CustomerID = c(1, 1, 2, 2, 3, 3, 3),

TransactionDate = as.Date(c('2023-01-15', '2023-02-20', '2023-01-25', '2023-03-10', '202302-05', '2023-03-15', '2023-04-01')),

Amount = c(100, 150, 200, 250, 300, 350, 400)

)

transactions <- transactions %>% group\_by(CustomerID) %>% mutate(CohortMonth = floor\_date(min(TransactionDate), "month")) %>% ungroup() transactions <- transactions %>%

mutate(MonthsSinceCohort = as.numeric(difftime(TransactionDate, CohortMonth, units = "days"))

%/% 30) %>%

group\_by(CohortMonth, MonthsSinceCohort) %>% summarise(NumCustomers = n\_distinct(CustomerID), .groups = 'drop')

print(transactions) cohort\_sizes <- transactions %>% group\_by(CohortMonth) %>% summarise(TotalCustomers = sum(NumCustomers), .groups = 'drop') print(cohort\_sizes) cohort\_analysis <- transactions %>% left\_join(cohort\_sizes, by = "CohortMonth") %>% mutate(RetentionRate = NumCustomers / TotalCustomers) print(cohort\_analysis) ggplot(cohort\_analysis, aes(x = MonthsSinceCohort, y = CohortMonth, fill =

RetentionRate)) +

geom\_tile() + scale\_fill\_gradient(low = "red", high = "green") + labs(title = "Cohort Analysis Heatmap", x = "Months Since Cohort", y = "Cohort Month", fill = "Retention Rate") +

theme\_minimal() ggplot(cohort\_analysis, aes(x = MonthsSinceCohort, y = RetentionRate, color =

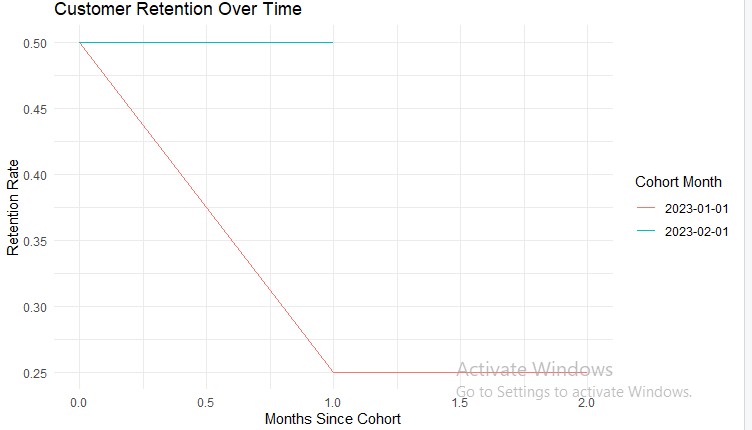
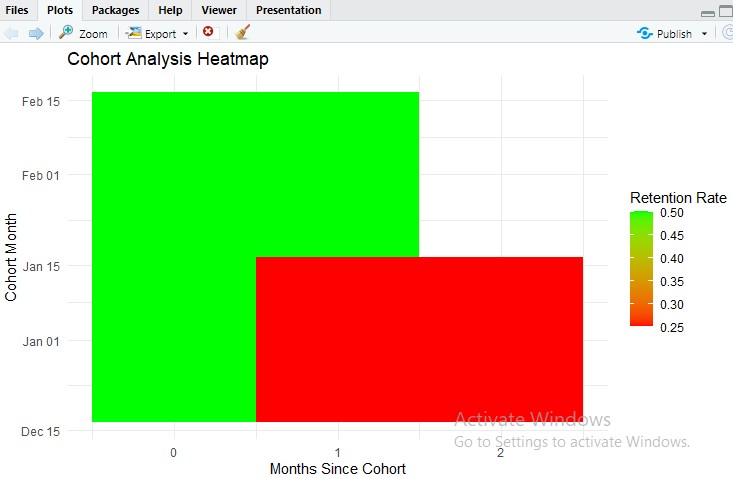
as.factor(CohortMonth))) +

geom\_line() + labs(title = "Customer Retention Over Time", x = "Months Since Cohort", y = "Retention Rate",

color = "Cohort Month") + theme\_minimal()

**OUTPUT :-**

:



**PRACTICAL NO:08**

**AIM:**

**Analyze customer purchasing patterns and build a recommender system based on market basket analysis**

* Use transactional data to identify frequently occurring itemsets using association rule mining algorithms.
* Calculate support, confidence, and lift for the identified item sets.
* Build a recommendation engine using collaborative filtering techniques.
* Evaluate the performance of the recommender system and make recommendations based on customer preferences.

**INPUT :-**

install.packages("tidyverse")

library(tidyverse)

install.packages("arules")

library("arules")

install.packages("arulesViz")

library("arulesViz")

library(dplyr)

search() archive <- read\_csv("C:/Users/LAB PC/Downloads/archive.zip") View(archive) head(archive) str(archive) colnames(archive)

data <- data %>% dplyr::rename(transaction\_id = 'TransactionId', item\_id = 'ItemCode') names(archive)[names(archive) == 'TransactionId'] <- 'transaction\_id' names(archive)[names(archive) == 'ItemCode'] <- 'item\_id' transactions\_list<-split(archive$item\_id,archive$transaction\_id)

transactions <- as(transactions\_list, "transactions") summary(transactions)

frequent\_itemsets <- apriori(transactions, parameter = list(support = 0.01, target = "frequent itemsets"))

print(frequent\_itemsets)

inspect(frequent\_itemsets)

rules <- apriori(transactions, parameter= list(support = 0.01,

target = "frequent itemsets"))

inspect(rules)

rules <- sort(rules, by = "lift", decreasing = TRUE) inspect(rules[1:10])

rule <- rules[1]

support(rule)

rules <- sort(rules, by = "lift", decreasing = TRUE) inspect(rules[1:5])

inspect(rules[1:5])

rule\_metrics <- quality(rules)

print(rule\_metrics[1, ])

support\_value <- rule\_metrics$support[1] print(paste("Support: ", support\_value))

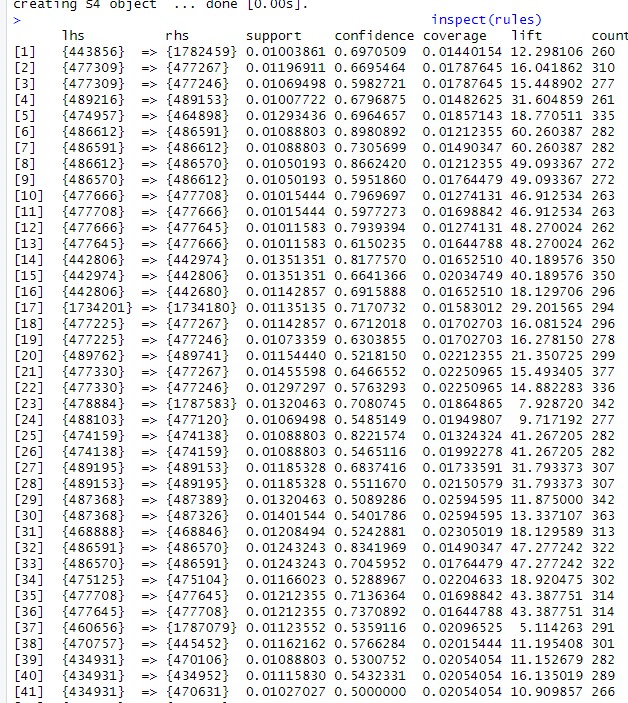
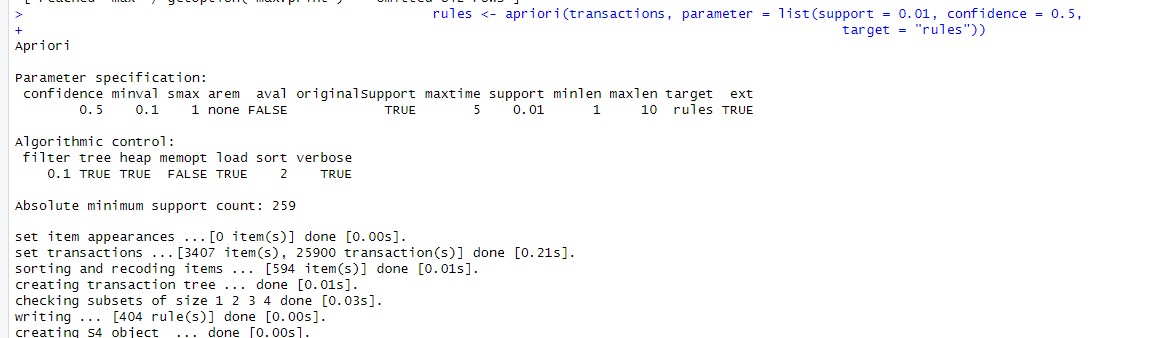
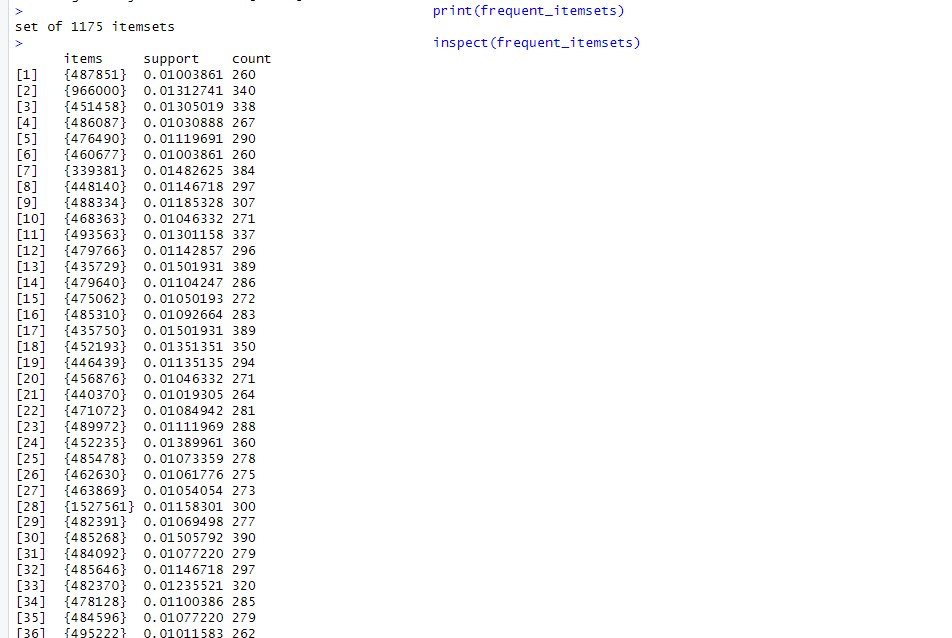
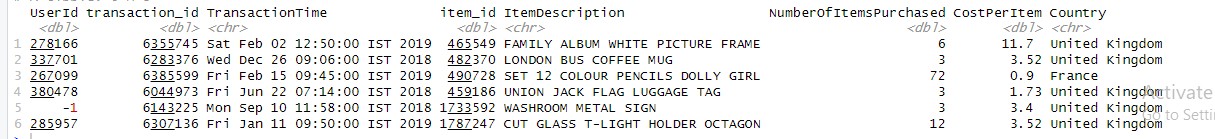
confidence\_value <- rule\_metrics$confidence[1]

lift\_value <- rule\_metrics$lift[1]

print(paste("Confidence: ", confidence\_value)) print(paste("Lift: ", lift\_value))

**OUTPUT**

:



**PRACTICAL No-09**

**AIM:**

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

* Analyze customer transaction data to calculate RFMscores.
* Segment customers into different groups using clusteringalgorithms such as k-means or hierarchical clustering.
* Perform descriptive analysis on each customer segment tounderstand their characteristics.
* Develop targeted marketing strategies for each segmentbased on their RFM profiles.

**INPUT :-**

library(dplyr) set.seed(123) transaction\_data <- data.frame(

CustomerID = rep(1:100, each = 5), # 100 customers with 5 transactions each

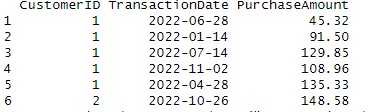
TransactionDate = sample(seq(as.Date('2022-01-01'), as.Date('2023-01-01'), by = "day"),

500, replace = TRUE),

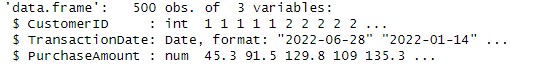
PurchaseAmount = round(runif(500, min = 10, max = 200), 2) # Random purchase amounts

)

write.csv(transaction\_data, "transaction\_data.csv", row.names = FALSE) transaction\_data <- read.csv("transaction\_data.csv") head(transaction\_data)



transaction\_data <- read.csv("transaction\_data.csv") transaction\_data$TransactionDate <- as.Date(transaction\_data$TransactionDate) str(transaction\_data)



current\_date <- as.Date("2024-08-31") current\_date <- as.Date("2023-12-09") recency\_data <- transaction\_data %>% group\_by(CustomerID) %>%

summarize(Recency = as.numeric(difftime(max(TransactionDate), current\_date, units =

"days")))

rfm\_data <- transaction\_data %>% group\_by(CustomerID) %>% summarize(

Recency = as.numeric(difftime(current\_date, max(TransactionDate), units = "days")),

Frequency = n(), # Count of transactions

Monetary = sum(PurchaseAmount), # Total amount spent

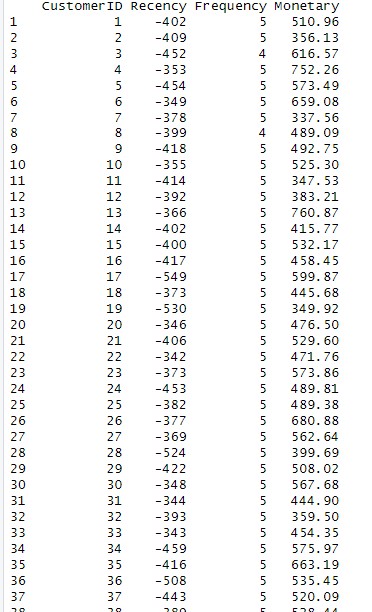
.groups = 'drop'

)

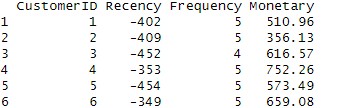
recency\_data <- transaction\_data %>% group\_by(CustomerID) %>%

summarize(Recency = as.numeric(difftime(max(TransactionDate), current\_date, units =

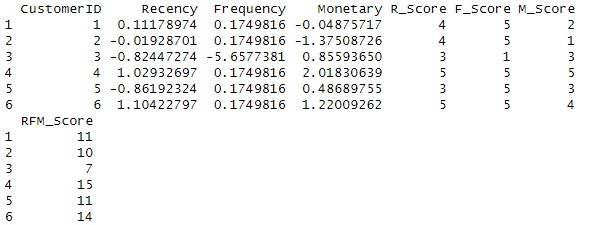
"days"))) frequency\_data <- transaction\_data %>% group\_by(CustomerID) %>% summarize(Frequency = n\_distinct(TransactionDate)) monetary\_data <- transaction\_data %>% group\_by(CustomerID) %>% summarize(Monetary = sum(PurchaseAmount)) rfm\_data <- merge(merge(recency\_data, frequency\_data, by = "CustomerID", all = TRUE), monetary\_data, by = "CustomerID", all = TRUE) print(rfm\_data)



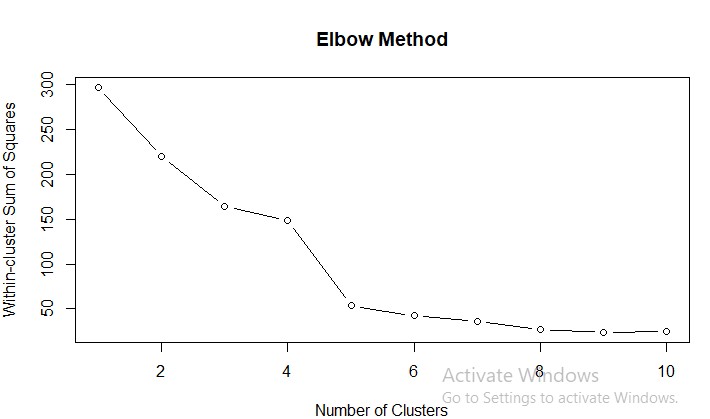
head(rfm\_data)



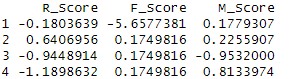
rfm\_data$Recency <- scale(rfm\_data$Recency) rfm\_data$Frequency <- scale(rfm\_data$Frequency) rfm\_data$Monetary <- scale(rfm\_data$Monetary) rfm\_data$R\_Score <- as.integer(cut(rfm\_data$Recency, breaks = 5, labels = FALSE)) rfm\_data$R\_Score <- as.integer(cut(rfm\_data$Recency, breaks = 5, labels = FALSE)) rfm\_data$F\_Score <- as.integer(cut(rfm\_data$Frequency, breaks = 5, labels = FALSE)) rfm\_data$M\_Score <- as.integer(cut(rfm\_data$Monetary, breaks = 5, labels = FALSE)) rfm\_data$RFM\_Score <- rfm\_data$R\_Score + rfm\_data$F\_Score + rfm\_data$M\_Score head(rfm\_data)



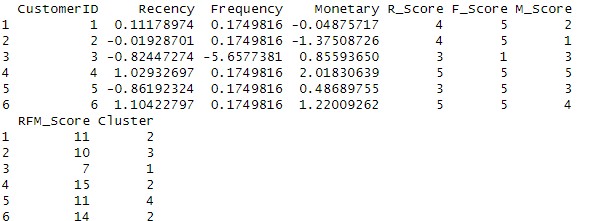
rfm\_for\_clustering <- rfm\_data[, c("R\_Score", "F\_Score", "M\_Score")] rfm\_for\_clustering <- scale(rfm\_for\_clustering) wss <- sapply(1:10, function(k) { kmeans(rfm\_for\_clustering, centers = k)$tot.withinss }) plot(1:10, wss, type = "b", xlab = "Number of Clusters", ylab = "Within-cluster Sum of Squares", main = "Elbow Method") k <- 4



set.seed(123) kmeans\_model <- kmeans(rfm\_for\_clustering, centers = k) rfm\_data$Cluster <- as.factor(kmeans\_model$cluster) print(kmeans\_model$centers)



head(rfm\_data)



cluster\_summary <- rfm\_data %>% group\_by(Cluster) %>% summarise(

Avg\_Recency = mean(R\_Score, na.rm = TRUE),

Avg\_Frequency = mean(F\_Score, na.rm = TRUE), Avg\_Monetary = mean(M\_Score, na.rm = TRUE),

Total\_Customers = n()

)

cluster\_summary <- summarise( group\_by(rfm\_data, Cluster),

Avg\_Recency = mean(R\_Score, na.rm = TRUE),

Avg\_Frequency = mean(F\_Score, na.rm = TRUE),

Avg\_Monetary = mean(M\_Score, na.rm = TRUE),

Total\_Customers = n()

)

cluster\_summary <- rfm\_data %>% group\_by(Cluster) %>% summarise(

Avg\_Recency = mean(R\_Score),

Avg\_Frequency = mean(F\_Score),

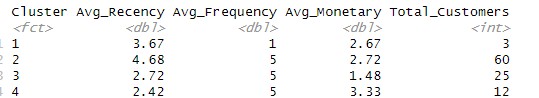
Avg\_Monetary = mean(M\_Score), Total\_Customers = n() )

print(cluster\_summary)

assign\_strategy <

-

function(cluster) {



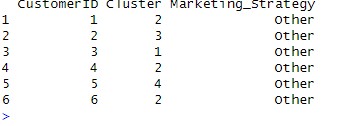
if (cluster == "Cluster 1") {

return("Re-Engagement Campaign") } else if (cluster == "Cluster 2") { return("Loyalty Rewards Program") } else if (cluster == "Cluster 3") { return("VIP or Exclusive Membership")

} else { return("Other")

} }

rfm\_data$Marketing\_Strategy <- sapply(rfm\_data$Cluster, assign\_strategy) head(rfm\_data[, c("CustomerID", "Cluster", "Marketing\_Strategy")])



**PRACTICAL No-10**

**AIM:**

**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 different strategies.
* Calculate key metrics such as conversion rates, click-through rates, or revenue.
* Interpret the results and provide recommendations for optimizing marketing campaigns based on the findings.

**INPUT :-**

1. **Design and implement A/B tests for marketing campaigns using randomized assignment.**

**INPUT :-**

install.packages(c("dplyr", "ggplot2"))

library(dplyr)

library(ggplot2) set.seed(123)

control\_data <- data.frame(user\_id = 1:500, group = "Control")

test\_data <- data.frame(user\_id = 501:1000, group = "Test")

control\_data$conversion <- rbinom(500, 1, 0.05)

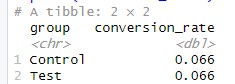
test\_data$conversion <- rbinom(500, 1, 0.08)

ab\_test\_data <- bind\_rows(control\_data, test\_data)

ab\_test\_data <- ab\_test\_data[sample(nrow(ab\_test\_data)), ]

conversion\_rates <- ab\_test\_data %>% group\_by(group) %>% summarise(conversion\_rate = mean(conversion))

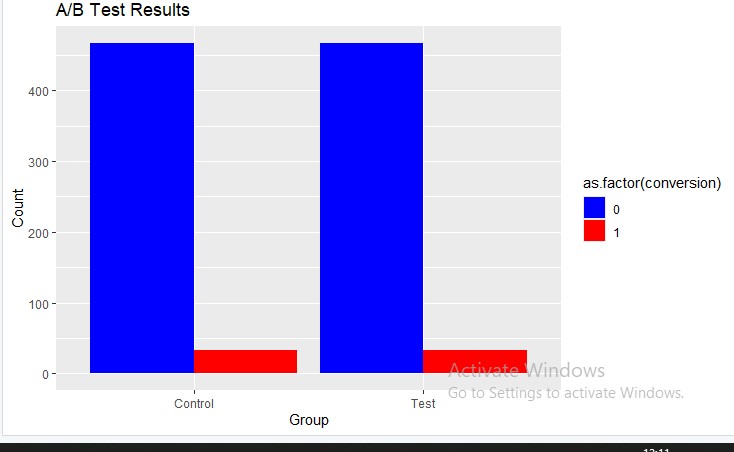
print(conversion\_rates)



ggplot(ab\_test\_data, aes(x = group, fill = as.factor(conversion))) + geom\_bar(position = "dodge") + labs(title = "A/B Test Results", x = "Group", y = "Count") +

scale\_fill\_manual(values = c("0" = "blue", "1" = "red"))

**OUTPUT :-**



**2. Collect relevant data and perform statistical analysis to compare the performance of different strategies.**

**INPUT :-**

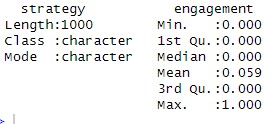
set.seed(123)

strategy\_data <- data.frame( strategy = rep(c("A", "B"), each = 500),

engagement = rbinom(1000, 1, c(0.05, 0.08))

)

summary(strategy\_data)

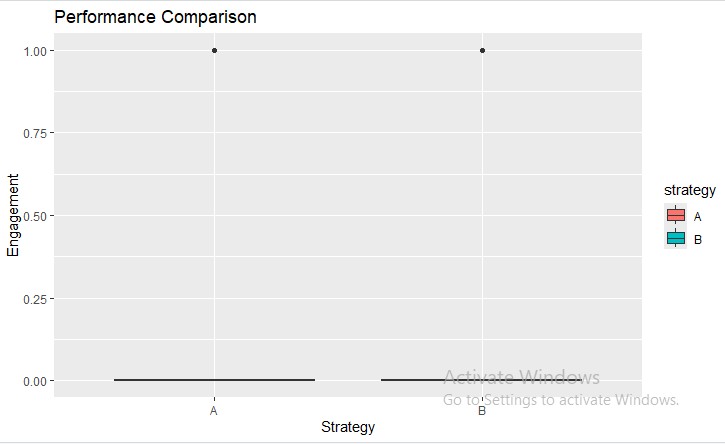


library(ggplot2)

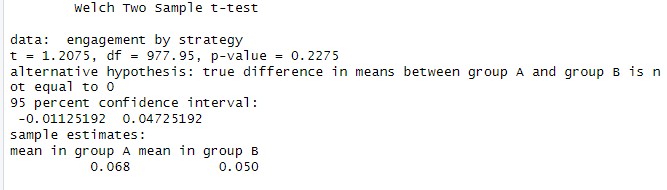
ggplot(strategy\_data, aes(x = strategy, y = engagement, fill = strategy)) + geom\_boxplot() +

labs(title = "Performance Comparison", x = "Strategy", y = "Engagement")

**OUTPUT :-**



t\_test\_result <- t.test(engagement ~ strategy, data = strategy\_data) print(t\_test\_result)



if (t\_test\_result$p.value < 0.05) {

print("The difference in engagement is statistically significant.")

} else {

print("There is no statistically significant difference in engagement.") }



3. Calculate key metrics such as conversion rates, click-through rates, or revenue.

set.seed(123)

data <- data.frame( user\_id = 1:1000,

strategy = rep(c("A", "B"), each = 500),

clicks = rbinom(1000, 1, c(0.2, 0.3)), # Simulated click data

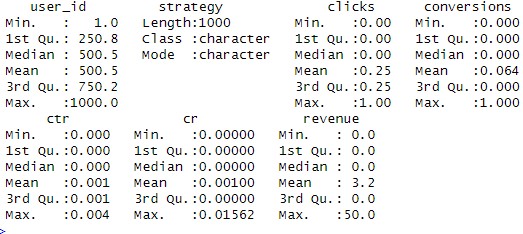
conversions = rbinom(1000, 1, c(0.05, 0.08)) # Simulated conversion data

)

data$ctr <- data$clicks / sum(data$clicks)

data$cr <- data$conversions / sum(data$conversions) revenue\_per\_conversion <- 50

data$revenue <- data$conversions \* revenue\_per\_conversion summary(data)



4. Interpret the results and provide recommendations for optimizing marketing campaigns based on the findings.

set.seed(123) campaign\_performance <- data.frame(Marketing\_Strategy = c("Re-Engagement

Campaign", "Loyalty Rewards Program", "VIP or Exclusive Membership"),

Customers = c(500, 700, 300),

Average\_Purchase = c(3.5, 5.2, 8.9), Repeat\_Purchase\_Rate = c(25, 40, 60) ) print(campaign\_performance)

library(ggplot2)



ggplot(campaign\_performance, aes(x = Marketing\_Strategy)) +

geom\_bar(aes(y = Customers), stat = "identity", fill = "skyblue", alpha = 0.8) + geom\_line(aes(y = Average\_Purchase \* 10, group = 1), color = "red", size = 1.5) + geom\_point(aes(y = Average\_Purchase \* 10, group = 1), color = "red", size = 3) + geom\_line(aes(y = Repeat\_Purchase\_Rate \* 5, group = 1), color = "green", size =

1.5) + geom\_point(aes(y = Repeat\_Purchase\_Rate \* 5, group = 1), color = "green", size =

3) +

labs(title = "Campaign Performance Metrics by Strategy",

y = "Count / Metric Value", x = "Marketing Strategy" ) +

scale\_y\_continuous(sec.axis = sec\_axis(~./10, name = "Average Purchase / 10\nRepeat

Purchase Rate / 5")) + theme\_minimal()

**OUTPUT :-**

