- **Create a data frame employee_data with the following columns and Extract employees who have more than 5 years of experience and display the result. 1. Employee_ID (Integer): 101, 102, 103, 104, 105 2. Name (Character): (Enter five employee names of your choice) 3. Department (Character): "HR", "Finance", "IT", "Marketing", "Sales" 4. Salary (Numeric): (Enter salaries for each employee) 5. Experience (Integer): (Enter years of experience for each employee)**

Answer: employee_data <- data.frame(

Employee_ID = c(101, 102, 103, 104, 105),

Name = c("Alice", "Bob", "Charlie", "David", "Emma"),

Department = c("HR", "Finance", "IT", "Marketing", "Sales"),

Salary = c(50000, 60000, 75000, 55000, 52000),

Experience = c(3, 7, 5, 8, 6)

)

Filtering employees with more than 5 years of experience

experienced_employees <- subset(employee_data, Experience > 5)

Display the result

print(experienced_employees)

- **Perform a Regression Analysis in R to predict house prices in Pune using relevant factors such as house size, number of bedrooms, and house age.**
  Answer:
  # Performing Regression Analysis in R to predict house prices in Pune
  # Load necessary library
  library(ggplot2)
  # Creating a sample dataset
  house_data <- data.frame(
  House_Size = c(1500, 1800, 2100, 2500, 1300, 1700, 2000, 2200, 1600, 2400),
  Bedrooms = c(3, 4, 3, 5, 2, 3, 4, 3, 2, 5),
  House_Age = c(10, 15, 8, 20, 5, 12, 18, 7, 6, 25),
  Price = c(5000000, 7000000, 6500000, 8500000, 4000000, 6000000, 7500000, 7200000, 4300000, 8000000)
)
  # Performing linear regression model
  model <- lm(Price ~ House_Size + Bedrooms + House_Age, data = house_data)
  # Display model summary
  summary(model)
  # Predicting house prices
  predicted_prices <- predict(model, house_data)
  # Adding predictions to the dataset
  house_data$Predicted_Price <- predicted_prices
  # Displaying dataset with predicted prices
  print(house_data)

- **Perform Association Rule Mining in R using the Apriori algorithm on a retail store transaction dataset. Identify frequent itemsets, generate association rules,**
  Answer:

```
library(arules)
library(arulesViz)

# Load the transaction dataset (example dataset used here)
data(Groceries)

# Display summary of the dataset
summary(Groceries)

# Applying the Apriori algorithm
rules <- apriori(Groceries, parameter = list(supp = 0.01, conf = 0.5))

# Displaying top rules
inspect(head(sort(rules, by = "lift"), 10))

# Visualizing the association rules
plot(rules, method = "graph", control = list(type = "items"))
```

- **Perform data manipulation using data frames and strings, then visualize the data using a bar plot in R. (Student Name, Marks, Age)**
  Answer:

```
# Performing data manipulation and visualization in R

# Load necessary library
library(ggplot2)

# Creating a student data frame
student_data <- data.frame(
  Student_Name = c("Alice", "Bob", "Charlie", "David", "Emma"),
  Marks = c(85, 78, 92, 88, 76),
  Age = c(20, 21, 19, 22, 20)
)

# Display the data frame
print(student_data)

# Manipulating strings: Converting student names to uppercase
student_data$Student_Name <- toupper(student_data$Student_Name)

# Displaying modified data
print(student_data)

# Visualizing the data using a bar plot
ggplot(student_data, aes(x = Student_Name, y = Marks, fill = Student_Name)) +
  geom_bar(stat = "identity") +
```

```
  theme_minimal() +
  labs(title = "Student Marks", x = "Student Name", y = "Marks")
```

- **Perform K-Means and Clustering on a manually created cricket score dataset.(Player_ID,Matches,Runs,Strike_rate,Centuries)**
  Answer:

```
# Performing K-Means Clustering on Cricket Score Dataset in R

# Load necessary library
library(ggplot2)
library(cluster)

# Creating a cricket score dataset
cricket_data <- data.frame(
  Player_ID = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
  Matches = c(100, 80, 120, 90, 110, 95, 85, 130, 105, 115),
  Runs = c(5000, 4200, 6200, 4500, 5800, 4900, 4600, 6800, 5300, 6000),
  Strike_rate = c(85, 78, 92, 88, 80, 83, 77, 95, 82, 89),
  Centuries = c(10, 8, 15, 9, 12, 10, 7, 18, 11, 14)
)

# Scaling the data for better clustering
scaled_data <- scale(cricket_data[, -1])

# Performing K-Means clustering
set.seed(123)  # Setting seed for reproducibility
kmeans_result <- kmeans(scaled_data, centers = 3, nstart = 10)

# Adding cluster labels to the dataset
cricket_data$Cluster <- as.factor(kmeans_result$cluster)

# Display clustered data
print(cricket_data)

# Visualizing clusters
ggplot(cricket_data, aes(x = Runs, y = Strike_rate, color = Cluster)) +
  geom_point(size = 4) +
  theme_minimal() +
  labs(title = "K-Means Clustering of Cricket Players", x = "Runs", y = "Strike Rate")
```

- **A retail company maintains records of its customer purchases. a dataset that includes Customer ID, Name, Product Purchased, Quantity, Price per Unit, and City. Your task is to manipulate the data using data frames, vectors, and string operations in R.**
  Answer :

```
# Manipulating Retail Customer Purchase Data in R

# Load necessary library
```

```
library(dplyr)

# Creating a retail customer purchase dataset
retail_data <- data.frame(
  Customer_ID = c(101, 102, 103, 104, 105),
  Name = c("Alice Johnson", "Bob Smith", "Charlie Brown", "David White", "Emma Davis"),
  Product_Purchased = c("Laptop", "Smartphone", "Tablet", "Smartwatch", "Headphones"),
  Quantity = c(1, 2, 1, 3, 2),
  Price_per_Unit = c(70000, 50000, 30000, 15000, 8000),
  City = c("New York", "Los Angeles", "Chicago", "Houston", "San Francisco")
)

# Display the dataset
print(retail_data)

# Manipulating strings: Extracting first names from customer names
retail_data$First_Name <- sapply(strsplit(retail_data$Name, " "), `[`, 1)

# Calculating total price for each purchase
retail_data$Total_Price <- retail_data$Quantity * retail_data$Price_per_Unit

# Displaying modified dataset
print(retail_data)

# Summarizing total revenue by city
total_revenue_by_city <- retail_data %>%
  group_by(City) %>%
  summarise(Total_Revenue = sum(Total_Price))

print(total_revenue_by_city)

# Visualizing total revenue by city
library(ggplot2)
ggplot(total_revenue_by_city, aes(x = City, y = Total_Revenue, fill = City)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Total Revenue by City", x = "City", y = "Total Revenue")
```

- **implement a classification algorithm in R The goal is to classify customers as "Frequent Buyer" or "Occasional Buyer" based on their purchasing behavior.**
  Answer:

```
# Implementing a Classification Algorithm in R
# Load necessary libraries
library(dplyr)
library(caret)
# Creating a retail customer purchase dataset
retail_data <- data.frame(
```

```
  Customer_ID = c(101, 102, 103, 104, 105, 106, 107, 108, 109, 110),
  Name = c("Alice Johnson", "Bob Smith", "Charlie Brown", "David White", "Emma Davis",
           "Frank Taylor", "Grace Hall", "Hannah King", "Ian Scott", "Jack Lee"),
  Total_Purchases = c(15, 5, 20, 7, 25, 10, 3, 30, 12, 6),
  Total_Spend = c(150000, 45000, 200000, 70000, 250000, 100000, 25000, 300000, 120000,
60000),
  City = c("New York", "Los Angeles", "Chicago", "Houston", "San Francisco",
           "Miami", "Dallas", "Seattle", "Boston", "Denver"),
  Buyer_Type = c("Frequent Buyer", "Occasional Buyer", "Frequent Buyer", "Occasional
Buyer",
                 "Frequent Buyer", "Occasional Buyer", "Occasional Buyer", "Frequent Buyer",
                 "Occasional Buyer", "Occasional Buyer")
)

# Display the dataset
print(retail_data)

# Splitting dataset into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(retail_data$Buyer_Type, p = 0.8, list = FALSE)
train_data <- retail_data[trainIndex, ]
test_data <- retail_data[-trainIndex, ]

# Training a classification model (Logistic Regression)
model <- train(Buyer_Type ~ Total_Purchases + Total_Spend, data = train_data, method =
"glm", family = "binomial")

# Making predictions
predictions <- predict(model, test_data)

# Evaluating the model
conf_matrix <- confusionMatrix(predictions, test_data$Buyer_Type)
print(conf_matrix)

# Displaying test data with predictions
test_data$Predicted_Buyer_Type <- predictions
print(test_data)
```

- **Perform Data Pre-processing in R using the manual dataset by handling missing values, encoding categorical variables and displaying the cleaned dataset.**
  Answer:

```
# Performing Data Pre-processing in R

# Load necessary libraries
library(dplyr)
library(caret)

# Creating a retail customer purchase dataset with missing values
```

```
retail_data <- data.frame(
  Customer_ID = c(101, 102, 103, 104, 105, 106, 107, 108, 109, 110),
  Name = c("Alice Johnson", "Bob Smith", "Charlie Brown", "David White", "Emma Davis",
           "Frank Taylor", "Grace Hall", NA, "Ian Scott", "Jack Lee"),
  Total_Purchases = c(15, 5, 20, 7, 25, NA, 3, 30, 12, 6),
  Total_Spend = c(150000, 45000, 200000, 70000, 250000, 100000, 25000, 300000, 120000,
NA),
  City = c("New York", "Los Angeles", "Chicago", "Houston", "San Francisco",
           "Miami", "Dallas", "Seattle", "Boston", "Denver"),
  Buyer_Type = c("Frequent Buyer", "Occasional Buyer", "Frequent Buyer", "Occasional
Buyer",
                 "Frequent Buyer", "Occasional Buyer", "Occasional Buyer", "Frequent Buyer",
                 "Occasional Buyer", "Occasional Buyer")
)

# Handling missing values
retail_data <- retail_data %>% mutate(
  Name = ifelse(is.na(Name), "Unknown", Name),
  Total_Purchases = ifelse(is.na(Total_Purchases), mean(Total_Purchases, na.rm = TRUE),
Total_Purchases),
  Total_Spend = ifelse(is.na(Total_Spend), median(Total_Spend, na.rm = TRUE), Total_Spend)
)

# Encoding categorical variables
retail_data$Buyer_Type <- as.factor(retail_data$Buyer_Type)
retail_data$City <- as.factor(retail_data$City)

# Displaying the cleaned dataset
print(retail_data)
```

- **A supermarket wants to analyze customer purchase patterns to increase sales. a dataset of customer purchases. Perform Association Rule Mining using the Apriori algorithm in R**
  Answer:

```
# Performing Association Rule Mining in R using the Apriori algorithm

# Load necessary libraries
library(arules)
library(arulesViz)

# Creating a transaction dataset for a supermarket
transactions <- list(
  c("Milk", "Bread", "Butter"),
  c("Bread", "Butter", "Eggs"),
  c("Milk", "Eggs"),
  c("Bread", "Milk", "Butter", "Eggs"),
  c("Butter", "Eggs"),
  c("Milk", "Bread"),
  c("Bread", "Eggs"),
```

```
  c("Milk", "Butter"),
  c("Bread", "Butter"),
  c("Milk", "Eggs", "Bread")
)

# Converting list to transaction format
trans <- as(transactions, "transactions")

# Display summary of transactions
summary(trans)

# Applying the Apriori algorithm
rules <- apriori(trans, parameter = list(supp = 0.3, conf = 0.6))

# Displaying the top association rules
inspect(head(sort(rules, by = "lift"), 10))

# Visualizing association rules
plot(rules, method = "graph", control = list(type = "items"))
```

- **A company wants to analyze employee performance based on their department, salary, and experience. Perform data manipulation using data frames and string operations, then visualize the average salary per department using a bar plot.**
  Answer:

```
# Analyzing Employee Performance and Visualizing Salary by Department in R

# Load necessary libraries
library(dplyr)
library(ggplot2)

# Creating an employee performance dataset
employee_data <- data.frame(
  Employee_ID = c(101, 102, 103, 104, 105, 106, 107, 108, 109, 110),
  Name = c("Alice Johnson", "Bob Smith", "Charlie Brown", "David White", "Emma Davis",
           "Frank Taylor", "Grace Hall", "Hannah King", "Ian Scott", "Jack Lee"),
  Department = c("HR", "Finance", "IT", "Marketing", "Sales", "HR", "Finance", "IT",
"Marketing", "Sales"),
  Salary = c(50000, 60000, 75000, 55000, 58000, 52000, 62000, 77000, 53000, 59000),
  Experience = c(5, 7, 10, 6, 8, 4, 9, 12, 5, 7)
)

# Displaying the dataset
print(employee_data)

# Manipulating strings: Extracting first names from employee names
employee_data$First_Name <- sapply(strsplit(employee_data$Name, " "), `[`, 1)

# Calculating average salary per department
```

```
avg_salary <- employee_data %>%
  group_by(Department) %>%
  summarise(Average_Salary = mean(Salary))

# Displaying average salary per department
print(avg_salary)

# Visualizing the average salary per department
ggplot(avg_salary, aes(x = Department, y = Average_Salary, fill = Department)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Average Salary by Department", x = "Department", y = "Average Salary")
```

- **A company wants to predict its sales revenue based on the amount spent on advertising. You are required to implement Simple Linear Regression using manual calculations in R. (Manually compute Slope (m) and Intercept (b).)**

  Answer:

```
# Implementing Simple Linear Regression Manually in R

# Load necessary library
library(ggplot2)

# Creating a dataset for advertising and sales
advertising_data <- data.frame(
  Advertising_Spend = c(10, 15, 20, 25, 30, 35, 40, 45, 50, 55), # in thousand dollars
  Sales_Revenue = c(25, 30, 40, 45, 50, 55, 65, 70, 75, 80) # in thousand dollars
)

# Displaying the dataset
print(advertising_data)

# Manually calculating slope (m) and intercept (b)
x <- advertising_data$Advertising_Spend
y <- advertising_data$Sales_Revenue

n <- length(x)
mean_x <- mean(x)
mean_y <- mean(y)

m <- sum((x - mean_x) * (y - mean_y)) / sum((x - mean_x)^2)
b <- mean_y - m * mean_x

cat("Calculated Slope (m):", m, "\n")
cat("Calculated Intercept (b):", b, "\n")

# Predicting sales revenue based on the linear regression model
advertising_data$Predicted_Sales <- m * advertising_data$Advertising_Spend + b
```

```
# Visualizing the regression line
ggplot(advertising_data, aes(x = Advertising_Spend, y = Sales_Revenue)) +
  geom_point(color = "blue", size = 3) +
  geom_abline(intercept = b, slope = m, color = "red", linetype = "dashed") +
  theme_minimal() +
  labs(title = "Sales Revenue Prediction", x = "Advertising Spend (in $1000s)", y = "Sales
Revenue (in $1000s)")
```

- **A university wants to analyze student exam scores. The dataset includes Student ID, Name, Subject, Marks Obtained, Total Marks, and Grade.**
  i) Extract first names from the Name column.
  ii) Convert subject names to uppercase

```
# Analyzing Student Exam Scores in R

# Load necessary library
library(dplyr)

# Creating a student exam dataset
student_data <- data.frame(
  Student_ID = c(101, 102, 103, 104, 105),
  Name = c("Alice Johnson", "Bob Smith", "Charlie Brown", "David White", "Emma Davis"),
  Subject = c("Mathematics", "Physics", "Chemistry", "Biology", "English"),
  Marks_Obtained = c(85, 78, 92, 88, 76),
  Total_Marks = c(100, 100, 100, 100, 100),
  Grade = c("A", "B", "A+", "A", "B")
)

# Displaying the dataset
print(student_data)

# Extracting first names from the Name column
student_data$First_Name <- sapply(strsplit(student_data$Name, " "), `[`, 1)

# Converting subject names to uppercase
student_data$Subject <- toupper(student_data$Subject)

# Displaying the modified dataset
print(student_data)
```

A bank wants to classify customers as "Eligible" or "Not Eligible" for a loan based on their income and credit score.

I. Customers with Income ≥ 50000 AND Credit Score ≥ 700 are classified as "Eligible".

II. Others are classified as "Not Eligible".

Answer:

```
# Classifying Customers for Loan Eligibility in R

# Load necessary library
```

```
library(dplyr)

# Creating a customer dataset
customer_data <- data.frame(
  Customer_ID = c(201, 202, 203, 204, 205, 206, 207, 208, 209, 210),
  Name = c("Alice Johnson", "Bob Smith", "Charlie Brown", "David White", "Emma
Davis",
           "Frank Taylor", "Grace Hall", "Hannah King", "Ian Scott", "Jack Lee"),
  Income = c(50000, 30000, 80000, 45000, 60000, 25000, 90000, 70000, 40000,
55000),
  Credit_Score = c(700, 550, 750, 620, 680, 500, 780, 730, 580, 640)
)

# Applying classification criteria
customer_data <- customer_data %>%
  mutate(Loan_Eligibility = ifelse(Income >= 50000 & Credit_Score >= 700, "Eligible",
"Not Eligible"))

# Displaying the updated dataset
print(customer_data)
```

- following dataset containing information about **customer purchases** in a retail store Create Data Frame and Handle Missing Values:

Customer_ID	Name	Age	Purchase_Amount	City
101	Raj	21	2000	Mumbai
102	Kunal	NA	1000	Pune
103	Rajesh	27	NA	Kolkata
104	Omkar	NA	4000	Pune
105	Sandeep	20	7000	Nashik

Answer:

```
# Handling Missing Values in Customer Purchase Data in R

# Load necessary library
library(dplyr)

# Creating a customer purchase dataset
customer_data <- data.frame(
  Customer_ID = c(101, 102, 103, 104, 105),
  Name = c("Raj", "Kunal", "Rajesh", "Omkar", "Sandeep"),
  Age = c(21, NA, 27, NA, 20),
  Purchase_Amount = c(2000, 1000, NA, 4000, 7000),
```

```
  City = c("Mumbai", "Pune", "Kolkata", "Pune", "Nashik")
)

# Handling missing values by replacing them with the mean of the column
customer_data$Age[is.na(customer_data$Age)] <- mean(customer_data$Age, na.rm = TRUE)
customer_data$Purchase_Amount[is.na(customer_data$Purchase_Amount)] <-
mean(customer_data$Purchase_Amount, na.rm = TRUE)

# Displaying the cleaned dataset
print(customer_data)
```

- **Perform Data Pre-processing in R using the manual dataset by handling missing values, encoding categorical variables and displaying the cleaned dataset.**

  Answer:

```
# Handling Missing Values and Encoding Categorical Variables in Customer Purchase Data in
R

# Load necessary library
library(dplyr)
library(caret)

# Creating a customer purchase dataset
customer_data <- data.frame(
  Customer_ID = c(101, 102, 103, 104, 105),
  Name = c("Raj", "Kunal", "Rajesh", "Omkar", "Sandeep"),
  Age = c(21, NA, 27, NA, 20),
  Purchase_Amount = c(2000, 1000, NA, 4000, 7000),
  City = c("Mumbai", "Pune", "Kolkata", "Pune", "Nashik")
)

# Handling missing values by replacing them with the mean of the column
customer_data$Age[is.na(customer_data$Age)] <- mean(customer_data$Age, na.rm = TRUE)
customer_data$Purchase_Amount[is.na(customer_data$Purchase_Amount)] <-
mean(customer_data$Purchase_Amount, na.rm = TRUE)

# Encoding categorical variable 'City' using factor levels
customer_data$City <- as.factor(customer_data$City)
customer_data$City <- as.numeric(customer_data$City)

# Displaying the cleaned dataset
print(customer_data)
```