1

```
shopping_cart <- data.frame(
  Name = c("Apple","Ball"),
  Quantity = c(1,3),
  Price = c(40,50)
)

total_before_tax <- shopping_cart$Quantity * shopping_cart$Price

reciept <- cbind(shopping_cart,total_before_tax)
print(reciept)

total_after_tax <- total_before_tax * 1.08

reciept <- cbind(reciept,total_after_tax)
print(reciept)

total_price <- sum(total_after_tax)
cat("total price\n")
print(total_price)
```

OR

```
 # List of products
# List of products
products <- list(
list(name = "Apple", price = 0.5),
list(name = "Banana", price = 0.3),
list(name = "Milk", price = 2),
list(name = "Bread", price = 1.5),
list(name = "Eggs", price = 2.5)
)
# Initialize shopping cart as an empty list
shopping_cart <- list()
# Define items to be added to the cart
cart_items_to_add <- list(
list(name = "Apple", quantity = 3),
list(name = "Milk", quantity = 2)
)
# Add items to the shopping cart
for (item in cart_items_to_add) {
product_name <- item$name
quantity <- item$quantity
# Find the product in the list
product <- NULL
for (p in products) {
if (p$name == product_name) {
product <- p
break
}
}
if (!is.null(product)) {
cart_item <- list(name = product$name, price = product$price, quantity = quantity)
shopping_cart <- c(shopping_cart, list(cart_item))
cat("Item added to cart.\n")
} else {
cat("Product not found.\n")
}
}
# Calculate and display receipt
subtotal <- 0
cat("\nReceipt:\n")
for (item in shopping_cart) {
item_subtotal <- item$price * item$quantity
cat(sprintf("%s (%d units) - Price: $%.2f - Subtotal: $%.2f\n", item$name, item$quantity,
item$price, item_subtotal))
subtotal <- subtotal + item_subtotal
}
tax_rate <- 0.08
tax_amount <- subtotal * tax_rate
total_cost_before_tax <- subtotal
total_cost <- total_cost_before_tax + tax_amount
cat("\nSubtotal: $%.2f\n", subtotal)
cat("Tax Amount (8%): $%.2f\n", tax_amount)
cat("Total Cost: $%.2f\n", total_cost)
```

2

```
num_students <- 5
num_courses <- 5
sample_names <- c("Alice", "Bob", "Charlie", "David", "Eva")
sample_marks <- matrix(c(
  85, 90, 88, 92, 95,   # Alice
  78, 82, 80, 85, 88,   # Bob
  65, 70, 68, 72, 75,   # Charlie
  55, 60, 58, 62, 65,   # David
  92, 95, 93, 97, 98    # Eva
), nrow = num_students, byrow = TRUE)


get_grade <- function(avg_marks) {
  if (avg_marks >= 90) {
    return("A")
  } else if (avg_marks >= 80) {
    return("B")
  } else if (avg_marks >= 70) {
    return("C")
  } else if (avg_marks >= 60) {
    return("D")
  } else {
    return("F")
  }
}
students <- list()
for (i in 1:num_students) {
  student_name <- sample_names[i]
  marks <- sample_marks[i,]
  total_marks <- sum(marks)
  avg_marks <- total_marks / num_courses
   grade <- get_grade(avg_marks)

  students[[i]] <- list(
name = student_name,
marks = marks,
total_marks = total_marks,
avg_marks = avg_marks,
grade = grade
  )
}
cat("\nStudent Information:\n")
for (i in 1:num_students)
{
  student <- students[[i]]
  cat("\nName:", student$name)
  cat("\nMarks:", paste(student$marks, collapse = ", "))
  cat("\nTotal Marks:", student$total_marks)
  cat("\nAverage Marks:", round(student$avg_marks, 2))
  cat("\nGrade:", student$grade, "\n")
}
```

3

```
calculate_fine <- function(days_overdue)
{
 if (days_overdue <= 7)
{
   fine <- 0
 } else if (days_overdue <= 30)
{
   fine_per_day <- 2
```

```
    fine <- (days_overdue - 7) * fine_per_day
  } else
{
    fine_cap <- 50
    fine <- fine_cap
  }
  return(fine)
}
days_overdue <- as.integer(readline("Enter the number of days the book is overdue: "))
fine_amount <- calculate_fine(days_overdue)
cat("Fine Amount:", fine_amount, "\n")
if (fine_amount == 0) {
  cat("No fine. Thank you for returning the book on time!\n")
} else {
  if (days_overdue > 30) {
    cat("Fine exceeds the maximum cap. Please contact the library.\n")
  } else {
    cat("Please pay the fine within the specified period.\n")
  }
}
```

4

```
# Initialize arrays for inventory items and quantities
inventory_items <- character(0)
inventory_quantities <- numeric(0)
# Function to add a new item with quantity
add_item <- function(item, quantity) {
inventory_items <<- c(inventory_items, item)
inventory_quantities <<- c(inventory_quantities, quantity)
cat("Item added to inventory.\n")
}
# Function to update quantity of an existing item
update_quantity <- function(item, new_quantity) {
if (item %in% inventory_items) {
item_index <- which(inventory_items == item)
inventory_quantities[item_index] <<- new_quantity
cat("Quantity updated.\n")
} else {
cat("Item not found in inventory.\n")
}
}
# Function to display inventory
display_inventory <- function() {
```

```
cat("Inventory Items and Quantities:\n")
for (i in 1:length(inventory_items)) {
cat(sprintf("%s: %d\n", inventory_items[i], inventory_quantities[i]))
}
}
# Main program
while (TRUE) {
cat("\n1. Add Item\n2. Update Quantity\n3. Display Inventory\n4. Exit\n")
choice <- as.integer(readline("Enter your choice: "))
if (choice == 1) {
item <- readline("Enter item name: ")
quantity <- as.integer(readline("Enter quantity: "))
add_item(item, quantity)
} else if (choice == 2) {
item <- readline("Enter item name: ")
new_quantity <- as.integer(readline("Enter new quantity: "))
update_quantity(item, new_quantity)
} else if (choice == 3) {
display_inventory()
} else if (choice == 4) {
cat("Exiting the program. Goodbye!\n")
break
} else {
cat("Invalid choice. Please try again.\n")
}
}
```

5.

```
students <- data.frame(Name=character(), Math_Score=numeric(), Science_Score=numeric(),
History_Score=numeric(), Attendance=numeric(), stringsAsFactors=FALSE)
add_student <- function(name, math, science, history, attendance) {
  students <<- rbind(students, data.frame(Name=name, Math_Score=math,
Science_Score=science, History_Score=history, Attendance=attendance))
}
generate_report <- function() {
  students$Average_Score <- rowMeans(students[2:4])
  cat("Student Report:\n"); print(students)
  cat("\nLow Attendance:\n"); print(subset(students, Attendance < 75))
}
add_student("Alice", 85, 90, 88, 80)
add_student("Bob", 75, 70, 65, 60)
add_student("Charlie", 95, 100, 98, 90)
generate_report()
```

OR

```
# Load the 'dplyr' package for data manipulation
library(dplyr)
# Create a data frame to store student information
student_data <- data.frame(
Name = character(0),
Math_Score = numeric(0),
Science_Score = numeric(0),
History_Score = numeric(0),
Attendance = numeric(0)
)
# Function to add student information
add_student <- function(name, math_score, science_score, history_score, attendance) {
new_student <- data.frame(
Name = name,
Math_Score = math_score,
Enter your choice:
Science_Score = science_score,
History_Score = history_score,
Attendance = attendance
)
student_data <<- bind_rows(student_data, new_student)
cat("Student information added.\n")
}
# Function to calculate average scores
calculate_average_scores <- function() {
avg_scores <- student_data %>%
mutate(Average_Score = (Math_Score + Science_Score + History_Score) / 3) %>%
select(Name, Average_Score)
return(avg_scores)
}
# Function to identify students with low attendance
identify_low_attendance <- function(threshold) {
low_attendance <- student_data %>%
filter(Attendance < threshold) %>%
select(Name, Attendance)
return(low_attendance)
}
# Function to generate a performance report
generate_report <- function() {
avg_scores <- calculate_average_scores()
low_attendance <- identify_low_attendance(70)
report <- merge(avg_scores, low_attendance, by = "Name", all = TRUE)
report$Attendance[is.na(report$Attendance)] <- 100
cat("Performance Report:\n")
print(report)
}
# Main program
while (TRUE) {
cat("\n1. Add Student\n2. Generate Report\n3. Exit\n")
choice <- as.integer(readline("Enter your choice: "))
if (choice == 1) {
name <- readline("Enter student name: ")
math_score <- as.numeric(readline("Enter math score: "))
science_score <- as.numeric(readline("Enter science score: "))
history_score <- as.numeric(readline("Enter history score: "))
attendance <- as.numeric(readline("Enter attendance percentage: "))
add_student(name, math_score, science_score, history_score, attendance)
} else if (choice == 2) {
generate_report()
} else if (choice == 3) {
cat("Exiting the program. Goodbye!\n")
break
} else {
cat("Invalid choice. Please try again.\n")
}
}
```

6.

```
library(forecast)
sales_data <- data.frame(
  Month = seq(as.Date("2023-01-01"), as.Date("2023-06-01"), by = "months"),
  Sales = c(12000, 15000, 18000, 16000, 20000, 22000)
)
sales_ts <- ts(sales_data$Sales, frequency = 12)
arima_model <- auto.arima(sales_ts)
forecast_result <- forecast(arima_model, h = 3)
print(forecast_result)
plot(forecast_result, main="Sales Forecast for the Next 3 Months", xlab="Month",ylab="Sales")
```

7

```
library(dplyr)
library(ggplot2)
```

```
purchase_data <- data.frame(
  CustomerID = c(101, 102, 103, 104, 105),
  PurchaseAmount = c(150, 200, 120, 300, 80)
)

mean_purchase <- mean(purchase_data$PurchaseAmount)
median_purchase <- median(purchase_data$PurchaseAmount)
sd_purchase <- sd(purchase_data$PurchaseAmount)
q1_purchase <- quantile(purchase_data$PurchaseAmount, probs = 0.25)
q3_purchase <- quantile(purchase_data$PurchaseAmount, probs = 0.75)

cat("Mean Purchase Amount:", mean_purchase, "\n")
cat("Median Purchase Amount:", median_purchase, "\n")
cat("Standard Deviation of Purchase Amounts:", sd_purchase, "\n")
cat("1st Quartile of Purchase Amounts:", q1_purchase, "\n")
cat("3rd Quartile of Purchase Amounts:", q3_purchase, "\n")

ggplot(purchase_data, aes(x = PurchaseAmount)) +
geom_histogram(binwidth = 50, fill = "blue", color = "black") +
labs(title = "Distribution of Purchase Amounts", x = "Purchase Amount", y = "Frequency")
```

8

```
matrix_A <- matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9), nrow = 3, ncol = 3, byrow = TRUE)
matrix_B <- matrix(c(9, 8, 7, 6, 5, 4, 3, 2, 1), nrow = 3, ncol = 3, byrow = TRUE)
sum_matrix <- matrix_A + matrix_B
scaled_matrix <- matrix_A * 2
transposed_A <- t(matrix_A)
product_matrix <- matrix_A %*% matrix_B
row_sums <- rowSums(matrix_B)
row_names <- paste("Row", 1:3)
barplot_data <- data.frame(Row = row_names, Sum = row_sums)
barplot_plot <- ggplot(barplot_data, aes(x = Row, y = Sum)) +
geom_bar(stat = "identity", fill = "green") +
labs(title = "Sums of Rows in Matrix B", x = "Row", y = "Sum")
print(barplot_plot)
```

9

```
library(ggplot2)
library(gridExtra)
student_data <- data.frame(
  Name = c("Alice", "Bob", "Charlie", "David", "Eva"),
  Score = c(85, 90, 78, 92, 88),
  Attendance = c(95, 90, 85, 93, 87),
  Date = as.Date(c("2023-01-10", "2023-02-10", "2023-03-10", "2023-04-10","2023-05-10"))
)

scatter_plot <- ggplot(student_data, aes(x = Attendance, y = Score)) +
  geom_point(color = 'blue', size = 3) +
  labs(title = "Scatter Plot: Scores vs Attendance", x = "Attendance (%)", y = "Score") +
  theme_minimal()

bar_plot <- ggplot(student_data, aes(x = Name, y = Score, fill = Name)) +
  geom_bar(stat = 'identity', color = 'black') +
  labs(title = "Bar Plot: Distribution of Scores", x = "Student Name", y = "Score") +
  theme_minimal()

line_plot <- ggplot(student_data, aes(x = Date, y = Score)) +
  geom_line(color = 'green', size = 1) +
  geom_point(color = 'green', size = 3) +
  labs(title = "Line Plot: Trend of Scores Over Time", x = "Date", y = "Score") +
  theme_minimal()

histogram <- ggplot(student_data, aes(x = Score)) +
  geom_histogram(binwidth = 5, fill = 'orange', color = 'black') +
  labs(title = "Histogram: Distribution of Scores", x = "Score", y = "Frequency") +
  theme_minimal()

grid.arrange(scatter_plot, bar_plot, line_plot, histogram, ncol = 2)
```

10

```
library(dplyr)
data <- data.frame(Name=c("John", "Jane"), Score=c(85, 90), Attendance=c(95, 90))
filtered <- filter(data, Score > 80)
selected <- select(data, Name, Score)
mutated <- mutate(data, UpdatedScore = Score + 5)
grouped <- data %>% group_by(Score) %>% summarize(Count = n())
arranged <- arrange(data, desc(Score))
joined <- inner_join(data, data, by="Name")
print(filtered)
print(selected)
print(mutated)
print(grouped)
print(arranged)
print(joined)
```

11

```
library(readr)
library(dplyr)
library(ggplot2)
purchase_data <- read_csv("customer_purchases.csv")
total_records <- nrow(purchase_data)
print(paste("Total number of records:", total_records))
total_unique_customers <- n_distinct(purchase_data$`Customer ID`)
print(paste("Total number of unique customers:", total_unique_customers))
mean_purchase <- mean(purchase_data$`Purchase Amount`, na.rm = TRUE)
print(paste("Mean (average) purchase amount:", mean_purchase))
median_purchase <- median(purchase_data$`Purchase Amount`, na.rm = TRUE)
print(paste("Median purchase amount:", median_purchase))
sd_purchase <- sd(purchase_data$`Purchase Amount`, na.rm = TRUE)
print(paste("Standard deviation of purchase amounts:", sd_purchase))
purchase_data <- purchase_data %>%
  mutate(Segment = ifelse(`Purchase Amount` < median_purchase, "Low Spender", "High
Spender"))
histogram_plot <- ggplot(purchase_data, aes(x = `Purchase Amount`)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Purchase Amounts", x = "Purchase Amount", y = "Frequency")
+theme_minimal()
print(histogram_plot)
```

OR

```
library(readr)
library(dplyr)
library(ggplot2)
purchase_data <- data.frame(
  CustomerID = c(101, 102, 103, 104, 105),
  PurchaseAmount = c(150, 200, 120, 300, 80)
)

total_records <- nrow(purchase_data)
print(paste("Total number of records:", total_records))

total_unique_customers <- n_distinct(purchase_data$CustomerID)
print(paste("Total number of unique customers:", total_unique_customers))

mean_purchase <- mean(purchase_data$PurchaseAmount, na.rm = TRUE)
```

```
print(paste("Mean (average) purchase amount:", mean_purchase))

median_purchase <- median(purchase_data$PurchaseAmount, na.rm = TRUE)
print(paste("Median purchase amount:", median_purchase))

sd_purchase <- sd(purchase_data$PurchaseAmount, na.rm = TRUE)
print(paste("Standard deviation of purchase amounts:", sd_purchase))

purchase_data <- purchase_data %>%
  mutate(Segment = ifelse(PurchaseAmount < median_purchase, "Low Spender", "High
Spender"))

histogram_plot <- ggplot(purchase_data, aes(x = PurchaseAmount)) +
  geom_histogram(binwidth = 50, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Purchase Amounts", x = "PurchaseAmount", y = "Frequency")
+theme_minimal()

print(histogram_plot)
```

12

```
library(dplyr)
ipl_data <- read_csv("ipl_data.csv")
str(ipl_data)
summary(ipl_data)
cat("Total matches:", nrow(ipl_data), "\n")
cat("Unique teams:", length(unique(c(ipl_data$Team1, ipl_data$Team2))), "\n")
matches_won <- ipl_data %>% group_by(Winner) %>% summarize(Wins = n())
cat("Matches won by each team:\n", matches_won, "\n")
cat("Average runs:", mean(ipl_data$Total.Runs), "\n")
cat("Average wickets:", mean(ipl_data$Total.Wickets), "\n")
most_frequent_venue <- ipl_data %>% group_by(Venue) %>% summarize(Frequency = n())
%>% arrange(desc(Frequency)) %>% head(1)
cat("Most frequent venue:", most_frequent_venue$Venue, "\n")
ggplot(matches_won, aes(x=Winner, y=Wins)) + geom_bar(stat="identity") +
labs(title="Matches Won by Each Team", x="Team", y="Wins")
```

OR

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

```
# Create the sample dataset
ipl_data <- data.frame(
  Team1 = c("Mumbai Indians", "Chennai Super Kings", "Royal Challengers Bangalore",
"Kolkata Knight Riders", "Sunrisers Hyderabad"),
  Team2 = c("Chennai Super Kings", "Royal Challengers Bangalore", "Kolkata Knight
Riders", "Sunrisers Hyderabad", "Delhi Capitals"),
  Venue = c("Wankhede Stadium", "M. A. Chidambaram Stadium", "M. Chinnaswamy
Stadium", "Eden Gardens", "Rajiv Gandhi Intl. Cricket Stadium"),
  Winner = c("Chennai Super Kings", "Chennai Super Kings", "Kolkata Knight Riders",
"Sunrisers Hyderabad", "Delhi Capitals"),
  Total.Runs = c(300, 280, 250, 275, 290),
  Total.Wickets = c(8, 7, 9, 10, 6)
)
str(ipl_data)
summary(ipl_data)

cat("Total matches:", nrow(ipl_data), "\n")
cat("Unique teams:", length(unique(c(ipl_data$Team1, ipl_data$Team2))), "\n")

matches_won <- ipl_data %>% group_by(Winner) %>% summarize(Wins = n())
cat("Matches won by each team:\n")
print(matches_won)

cat("Average runs:", mean(ipl_data$Total.Runs), "\n")
cat("Average wickets:", mean(ipl_data$Total.Wickets), "\n")

most_frequent_venue <- ipl_data %>%
  group_by(Venue) %>%
  summarize(Frequency = n()) %>%
  arrange(desc(Frequency)) %>%
  head(1)
cat("Most frequent venue:", most_frequent_venue$Venue, "\n")
bar_plot <- ggplot(matches_won, aes(x = Winner, y = Wins)) +
  geom_bar(stat = "identity") +
  labs(title = "Matches Won by Each Team", x = "Team", y = "Wins") +
  theme_minimal()
print(bar_plot)
```