# Experiment 1 - Billing System and Customer Behavior Analysis using Functions in R

## Details:

Name: Sparsh Karna

Registration Number: 23BDS1172

Slot: L21 + L22

## 1. Objective:

To simulate and analyze customer billing data using only vectors and user-defined functions in R.

## 2. Methodology:

- Data created using individual vectors  
- Billing computed using custom functions  
- Summary and filtering logic implemented without data frames

## 3. R Code with Comments:

# Billing System and Customer Behavior Analysis using Functions in R

customer\_names <- c("Sparsh Karna", "Arkita Barua", "Diraq Girach", "Aryan Risi", "Lavanaya Malhotra", "Stuti Handa", "Divyansh Aggarwal", "Mishti Mattu", "Akshat Majila", "Rose Priya"); # This defines the 10 customer names into the vector

customer\_type <- c("Regular", "New", "New", "New", "Regular", "Regular", "Regular", "New", "New", "New"); # This defines the type of customer mapped to each customer name

item\_name <- c("Rice", "Wheat", "Sugar", "Oil", "Rice", "Tea", "Coffee", "Sugar", "Oil", "Wheat", "Rice", "Tea"); # This defines some of the items present in the store

price\_per\_unit <- c(50, 40, 45, 120, 50, 200, 150, 45, 120, 40, 50, 200); # This defines the price per unit of each given item

quantity\_purchased <- c(5, 10, 2, 3, 8, 1, 2, 4, 1, 5, 6, 3); # This defines the the quantity of each item purchased by the customer

#' @title: Function to calculate the total bill of the user

#' @description: function calculate\_total\_bill() takes two vectors (price and quantity) and returns a vector of total bill values for each customer.

#' @param price: The vector defining the price per unit of each item

#' @param quantity: The vector defining the quantity purchased by the customer of each item

#' @returns total\_bill: Contains the total bill of each customer

#' @example

#' # Calculate the total bill of each customer

#' total\_bill <- calculate\_total\_bill(price, quantity)

#' print(total\_bill)

calculate\_total\_bill <- function(price, quantity){

total\_bill <- price \* quantity; # Calculate the total price by multiply quantity of each by its respective price

return(total\_bill); # Return the total bill amount

}

#' @title: Function to calculate the total revenue

#' @description: functions to compute and print the total revenue of the shop

#' @param total\_bill: The vector defining the bill of each customer

#' @returns revenue: Contains the total revenue of the shop

#' @example

#' # Calculate the total bill of each customer

#' total\_bill <- calculate\_total\_bill(price, quantity)

#' # Calculate the total revenue of the shop

#' revenue <- calculate\_total\_revenue(total\_bill)

calculate\_total\_revenue <- function(total\_bill){

revenue <- sum(total\_bill); # Calculate the sum of bill of each customer

return(revenue); # Return the revenue

}

#' @title: Function to calculate Average bill amount

#' @description: functions to compute and print the Average bill amount of the shop

#' @param total\_bill: The vector defining the bill of each customer

#' @returns average\_bill: Contains the average bill amount

#' @example

#' # Calculate the total bill of each customer

#' total\_bill <- calculate\_total\_bill(price, quantity)

#' # Calculate the average bill amount

#' average\_bill <- calculate\_average\_bill(total\_bill)

calculate\_average\_bill <- function(total\_bill){

total\_revenue <- calculate\_total\_revenue(total\_bill); # Calculates the total revenue of shop

total\_customer <- length(total\_bill) # Calculates the total number of customers

average\_bill <- total\_revenue / total\_customer; # Calculates the average bill amount of the shop

return(average\_bill); # Returns the average bill amount

}

#' @title: Function to calculate Maximum and minimum bill amount

#' @description: functions to compute and print the Maximum and minimum bill amount of the shop

#' @param total\_bill: The vector defining the bill of each customer

#' @returns max\_bill: Contains the maximum bill amount

#' @returns min\_bill: Contains the minimum bill amount

#' @example

#' # Calculate the total bill of each customer

#' total\_bill <- calculate\_total\_bill(price, quantity)

#' # Calculate the minimum bill amount

#' min\_bill <- calculate\_min\_max\_bill\_amount(total\_bill)[2]

#' # Calculate the maximum bill amount

#' max\_bill <- calculate\_min\_max\_bill\_amount(total\_bill)[1]

calculate\_min\_max\_bill\_amount <- function(total\_bill){

max\_bill <- max(total\_bill) # Calculates the maximum bill amount

min\_bill <- min(total\_bill) # Calculates the minimum bill amount

return(c(max\_bill, min\_bill)); # Return the maximum and minimum bill maount

}

#' @title: Function to calculate Total number of "Regular" vs "New" customers

#' @description: functions to compute and print the Total number of "Regular" vs "New" customers of the shop

#' @param customer\_type: The vector defining the type of each customer

#' @returns reg: Contains the total number of regular customer

#' @returns new: Contains the total number of new customer

#' @example

#' # Calculate the total number of regular customer

#' reg <- calculate\_reg\_new\_customers(customer\_type)[1]

#' # Calculate the total number of new customer

#' new <-calculate\_reg\_new\_customers(customer\_type)[2]

calculate\_reg\_new\_customers <- function(customer\_type){

count\_reg = 0; # Keeps the count of the regular customers

count\_new = 0; # Keeps the count of the new customers

# Iterate over the customer\_type vector

for(customer in customer\_type){

if(customer == "Regular"){ # Check if the customer is regular

count\_reg = count\_reg + 1; # If regular then increase the count of the regular customer

}

else{ # If the customer is not regular then must be new

count\_new = count\_new + 1; # If new then increase the count of the new customer

}

}

return(c(count\_reg, count\_new)) # Return the count of new customer and regular customer

}

#' @title: Function to figure out highest spender

#' @description: functions hat takes the total bill vector and customer names and returns names of customers who spent more than Rs. 500.

#' @param customer\_name: Names of all the customers

#' @param total\_bill: Total bill of each customer

#' @returns high\_spender\_names: The vector containing the customer name who spend more than 500

#' @example

#' # Names of customer who spend more than 500

#' names <- get\_high\_spenders(customer\_name, total\_bill)

get\_high\_spenders <- function(customer\_name, total\_bill){

high\_spender\_names <- c();

# Iterate over the customers vector using indexes

for(i in 1:length(customer\_name)){

if(total\_bill[i] > 500){ # Check if the total bill of that customer is more than 500

high\_spender\_names <- c(high\_spender\_names, customer\_name[i]); # If it is then append the name of that customer to the vector

}

}

return(high\_spender\_names); # Return the names of the high spender vector

}

#' @title: Function to calculate item-wise total quantity sold

#' @description: Calculate total quantity sold for each unique item

#' @param item\_name: Names of all the items

#' @param quantity\_purchased: Quantity of each item purchased

#' @returns total\_quantities: named numeric vector with total quantity for each item

#' @example

#' # Get unique counts of each item

#' unique\_quantity <- item\_wise\_quantity(item\_name, quantity\_purchased)

item\_wise\_quantity <- function(items\_vector, quantities\_vector) {

# Get unique items

unique\_items <- unique(items\_vector)

# Initialize vector to store total quantities

total\_quantities <- numeric(length(unique\_items))

names(total\_quantities) <- unique\_items

# Calculate total quantity for each item

for(item in unique\_items) {

# Find all indices where this item appears

item\_indices <- which(items\_vector == item)

# Sum quantities for this item

total\_quantities[item] <- sum(quantities\_vector[item\_indices])

}

return(total\_quantities)

}

# Main execution - Demonstrating all functions with the given data

cat("=== Billing System and Customer Behavior Analysis ===\n\n")

# Calculate total bill for each customer

total\_bill <- calculate\_total\_bill(price\_per\_unit, quantity\_purchased)

cat("1. Total bill for each customer:\n")

for(i in 1:length(customer\_names)){

cat(" ", customer\_names[i], ": Rs.", total\_bill[i], "\n")

}

cat("\n")

# Calculate total revenue

total\_revenue <- calculate\_total\_revenue(total\_bill)

cat("2. Total revenue of the shop: Rs.", total\_revenue, "\n\n")

# Calculate average bill amount

average\_bill <- calculate\_average\_bill(total\_bill)

cat("3. Average bill amount: Rs.", round(average\_bill, 2), "\n\n")

# Calculate maximum and minimum bill amounts

min\_max\_bill <- calculate\_min\_max\_bill\_amount(total\_bill)

max\_bill <- min\_max\_bill[1]

min\_bill <- min\_max\_bill[2]

cat("4. Maximum bill amount: Rs.", max\_bill, "\n")

cat(" Minimum bill amount: Rs.", min\_bill, "\n\n")

# Calculate regular vs new customers

reg\_new\_count <- calculate\_reg\_new\_customers(customer\_type)

regular\_count <- reg\_new\_count[1]

new\_count <- reg\_new\_count[2]

cat("5. Customer analysis:\n")

cat(" Regular customers:", regular\_count, "\n")

cat(" New customers:", new\_count, "\n\n")

# Get high spenders (customers who spent more than Rs. 500)

high\_spenders <- get\_high\_spenders(customer\_names, total\_bill)

cat("6. High spenders (customers who spent more than Rs. 500):\n")

if(length(high\_spenders) > 0) {

for(spender in high\_spenders) {

cat(" -", spender, "\n")

}

} else {

cat(" No customers spent more than Rs. 500\n")

}

cat("\n")

# Calculate item-wise total quantity sold

item\_quantities <- item\_wise\_quantity(item\_name, quantity\_purchased)

cat("7. Item-wise total quantity sold:\n")

for(i in 1:length(item\_quantities)) {

cat(" ", names(item\_quantities)[i], ":", item\_quantities[i], "units\n")

}

cat("\n")

cat("=== Analysis Complete ===\n")

## 4. Output Summary:

## === Billing System and Customer Behavior Analysis ===

## 1. Total bill for each customer:

## Sparsh Karna : Rs. 250

## Arkita Barua : Rs. 400

## Diraq Girach : Rs. 90

## Aryan Risi : Rs. 360

## Lavanaya Malhotra : Rs. 400

## Stuti Handa : Rs. 200

## Divyansh Aggarwal : Rs. 300

## Mishti Mattu : Rs. 180

## Akshat Majila : Rs. 120

## Rose Priya : Rs. 200

## 2. Total revenue of the shop: Rs. 3400

## 3. Average bill amount: Rs. 283.33

## 4. Maximum bill amount: Rs. 600

## Minimum bill amount: Rs. 90

## 5. Customer analysis:

## Regular customers: 4

## New customers: 6

## 6. High spenders (customers who spent more than Rs. 500):

## No customers spent more than Rs. 500

## 7. Item-wise total quantity sold:

## Rice : 19 units

## Wheat : 15 units

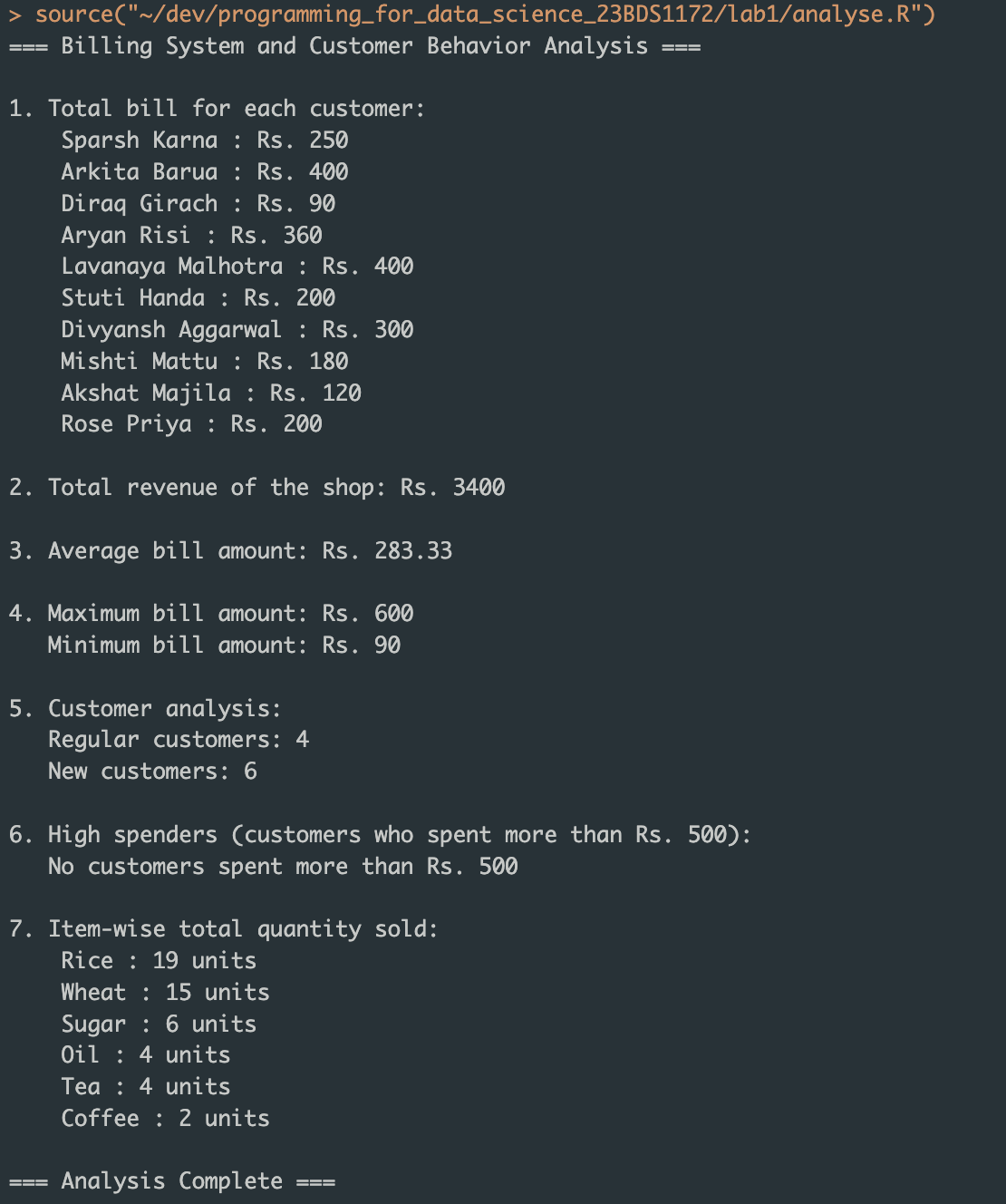
## Sugar : 6 units

## Oil : 4 units

## Tea : 4 units

## Coffee : 2 units

## === Analysis Complete ===



## 5. Interpretations:

- The most sold item is Rice with 19 units.  
- Regular customers contributed Rs. 1150 to the revenue, while new customers contributed Rs. 1350. Thus, new customers generated more revenue.

## 6. Conclusion:

Through this experiment, we learned how to perform data analysis using only vectors and basic functions in R. This approach reinforces the understanding of fundamental programming constructs and avoids reliance on advanced data structures, which is beneficial for foundational learning.