UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

(FSKTM)

SEMESTER II 2024/2025

DATA MINING

BIT 33603

SECTION 03

LAB ASSIGNMENT 02

TITLE

BASICS OF R, LOOPS IN R, ARRAYS IN R, AND READ AND WRITE FILES IN R

LECTURER'S NAME

DR. ROZITA BINTI ABDUL JALIL

NAME	TUAN KHALIDAH SYAZWANA BINTI TUAN MOHD KASMAWI
MATRIC NUMBER	AI220118
DATE SUBMISSION	March 28, 2025

Assessment Questions:

1) **What are the key differences between vectors and arrays in R?**

   Vectors in R are one-dimensional and store elements of the same data type, such as numbers or characters. Arrays, on the other hand, can have multiple dimensions (e.g., 2D or 3D) but still store elements of the same type. While vectors are used for simpler data storage, arrays are useful for handling structured data like matrices.

2) **How do loops help in data processing in R?**

   Loops help in data processing by automating repetitive tasks, such as iterating over datasets, applying calculations, or filtering data. For loops are used when the number of iterations is known, while loops run until a condition is met. This makes loops useful for handling large datasets efficiently without manual repetition.

3) **Write an R script to create an array of size 2x3 and perform matrix multiplication.**

   R script for creating a 2x3 array and performing matrix multiplication:

```r
arr1 <- array(1:6, dim = c(2,3))  # Creating a 2x3 array
arr2 <- array(7:12, dim = c(3,2)) # Creating a 3x2 array
result <- arr1 %*% arr2  # Performing matrix multiplication
print(result)
```

4) **How can you read and write CSV files in R?**

   To read a CSV file in R, use read.csv("file.csv"), which loads the data into a dataframe for analysis. To write data to a CSV file, use write.csv(data, "file.csv"), which saves the dataframe into a new file. This is useful for importing and exporting structured data, making it easier to analyze and share.

Activity 1: Getting Started with R

Instructions:

1. Open RStudio and create a new script file.

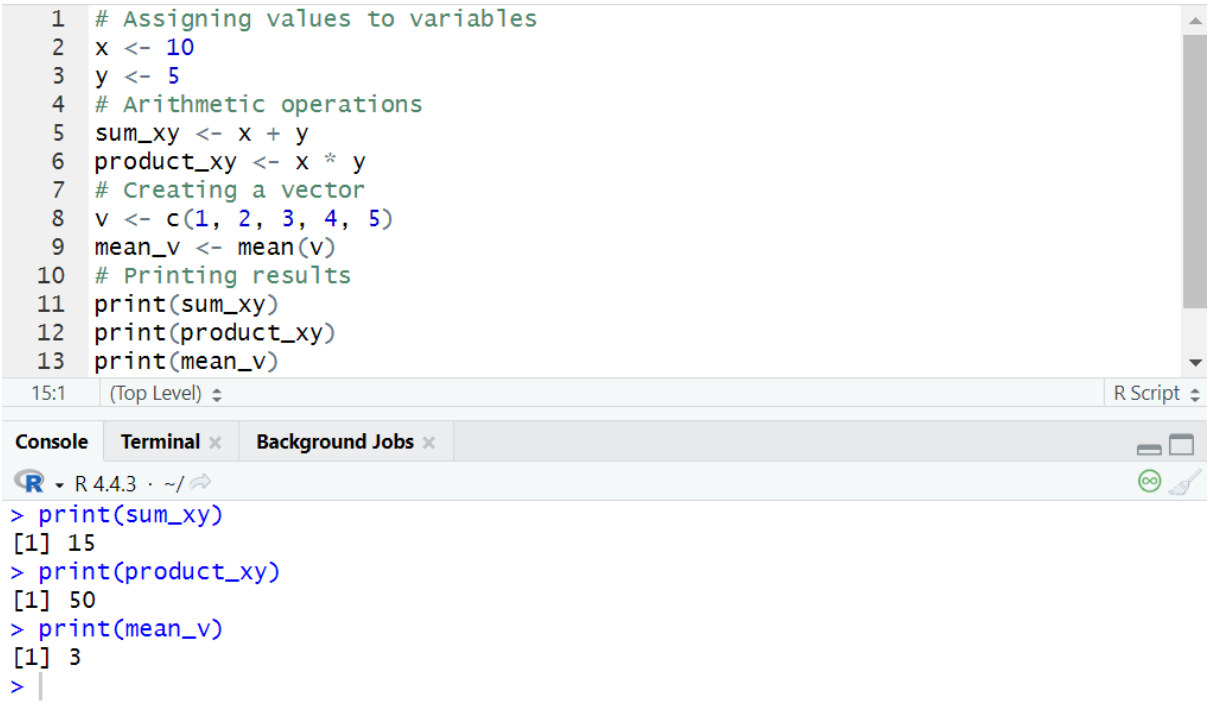
2. Write R code to perform the following tasks:

    o Assign values to variables and print them.

    o Perform basic arithmetic operations (+, -, *, /, ^).

    o Create and manipulate vectors using c() function.

    o Use built-in functions like sum(), mean(), max(), min().



Activity 2: Using Loops in R

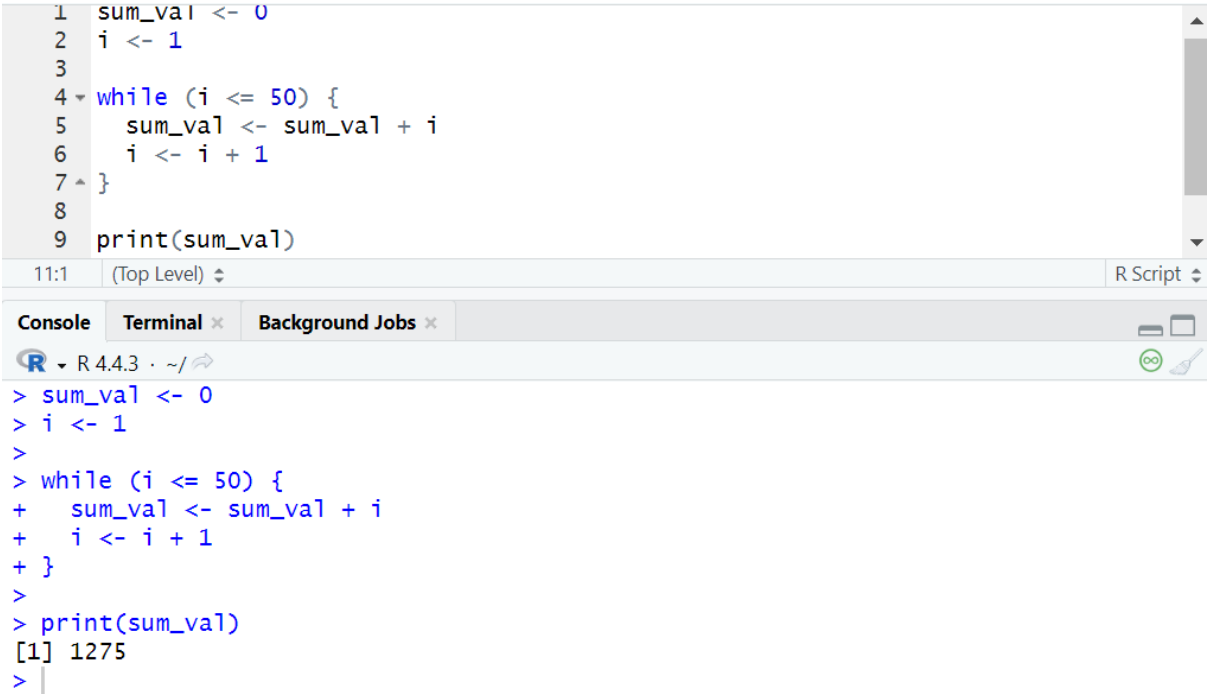
Instructions:

1. Implement a for loop to print numbers from 1 to 10.

```
# For loop example
for (i in 1:10) {
  print(i)
}
```

```
> # For loop example
> for (i in 1:10) {
+   print(i)
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
>
```

2. Use a while loop to compute the sum of numbers from 1 to 50.



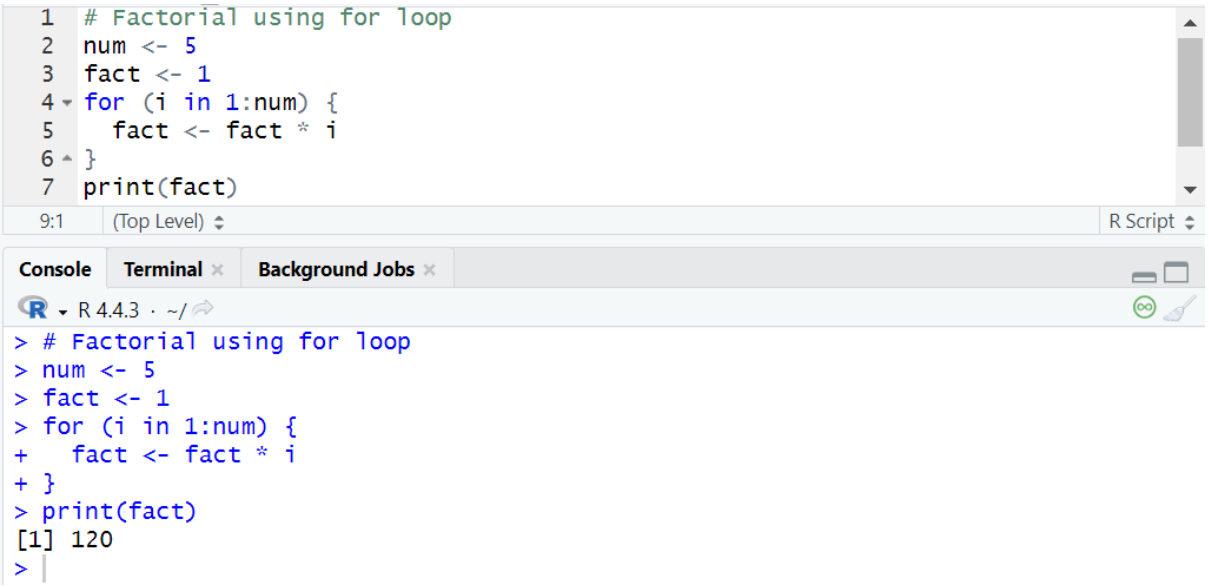
```
sum_val <- 0
i <- 1

while (i <= 50) {
  sum_val <- sum_val + i
  i <- i + 1
}

print(sum_val)
```

```
> sum_val <- 0
> i <- 1
>
> while (i <= 50) {
+   sum_val <- sum_val + i
+   i <- i + 1
+ }
>
> print(sum_val)
[1] 1275
>
```

3. Write a loop to find the factorial of a given number.



```r
# Factorial using for loop
num <- 5
fact <- 1
for (i in 1:num) {
  fact <- fact * i
}
print(fact)
```

```
9:1  (Top Level)  R Script

Console  Terminal  Background Jobs

R 4.4.3 · ~/

> # Factorial using for loop
> num <- 5
> fact <- 1
> for (i in 1:num) {
+   fact <- fact * i
+ }
> print(fact)
[1] 120
>
```

Activity 3: Working with Arrays in R

Instructions:

1. Create a 3x3 matrix using an array.

2. Perform element-wise addition and multiplication on two arrays.

3. Extract specific rows and columns from the array.

```r
# Creating an array
arr1 <- array(1:9, dim = c(3,3))
arr2 <- array(10:18, dim = c(3,3))
# Performing operations
sum_arr <- arr1 + arr2
prod_arr <- arr1 * arr2
# Extracting rows and columns
row1 <- arr1[1, ]
col2 <- arr1[, 2]
# Printing results
print(sum_arr)
print(prod_arr)
print(row1)
print(col2)
```

15:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.4.3 · ~/

```
> col2 <- arr1[, 2]
>
> # Printing results
> print(sum_arr)
     [,1] [,2] [,3]
[1,]   11   17   23
[2,]   13   19   25
[3,]   15   21   27
> print(prod_arr)
     [,1] [,2] [,3]
[1,]   10   52  112
[2,]   22   70  136
[3,]   36   90  162
> print(row1)
[1] 1 4 7
> print(col2)
[1] 4 5 6
>
```

Activity 4: Reading and Writing Files in R

Instructions:

1. Create a sample CSV file with the following content and save it as data.csv:

Name, Age, Score
Ali, 22, 85
Siti, 21, 90
John, 23, 78

2. Write an R script to read the CSV file and display its contents.

3. Modify the dataset by adding a new row and save the updated data to a new file.

```r
# Reading CSV file
data <- read.csv("data.csv")
print(data)  # Display the dataset

# Adding a new row
new_data <- data.frame(Name="Ahmad", Age=24, Score=88)
data <- rbind(data, new_data)

# Writing the modified data to a new file
write.csv(data, "updated_data.csv", row.names=FALSE)
print("Updated data saved successfully.")
```

13:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.4.3 · C:/Users/user/Downloads/R/

```
> # Reading CSV file
> data <- read.csv("data.csv")
> print(data)  # Display the dataset
  Name Age Score
1  Ali  22    85
2 Siti  21    90
3 John  23    78
>
> # Adding a new row
> new_data <- data.frame(Name="Ahmad", Age=24, Score=88)
> data <- rbind(data, new_data)
>
> # Writing the modified data to a new file
> write.csv(data, "updated_data.csv", row.names=FALSE)
> print("Updated data saved successfully.")
[1] "Updated data saved successfully."
>
```