

NAME: JUWAIIRIAH BIN MUHAMMAD SIN

MATRIK NUMBER: AI220365

LAB NAME: LAB 2

Assessment Questions:

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

| Feature | Vectors | Arrays |
|------------|--|--|
| Definition | A one-dimensional collection of elements of the same data type | A multi-dimensional data structure (2D, 3D, or more) containing elements of the same data type |
| Dimensions | 1D (single row or column) | 2D (matrix) or more (3D, 4D, etc.) |
| Data Type | Must contain elements of the same type (numeric, character, logical, etc.) | Must contain elements of the same type |
| Creation | <code>c(1, 2, 3, 4, 5, 700, 1000)</code> | <code>array(1:9, dim = c(3,3))</code> |
| Indexing | Uses single index <code>[]</code> | Uses multi-dimensional index <code>[,]</code> |
| Operations | Element-wise operations are simpler | Multi-dimensional calculations and slicing possible |
| Use Case | Used for simple lists of numbers, strings, or logical values | Used for multi-dimensional data like 2D tables (matrices) or 3D structures |

Vectors:

```
# Creating a vector
v <- c(1, 2, 3, 4, 5, 700, 1000)
v
mean_v <- mean(v)
mean_v
sum_v <- sum(v)
sum_v

> # Creating a vector
> v <- c(1, 2, 3, 4, 5, 700, 1000)
> v
[1] 1 2 3 4 5 700 1000
> mean_v <- mean(v)
> mean_v
[1] 245
```

Arrays:

```
# 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
arr1
arr2
```

```
> # 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
> arr1
      [,1] [,2] [,3]
[1,]     1     4     7
[2,]     2     5     8
[3,]     3     6     9
```

2. How do loops help in data processing in R?

By iterating across datasets, applying transformations, and carrying out calculations without manually duplicating code, loops in R increase the efficiency of data processing by automating repetitive processes.

1. **Automate Repetitive Tasks** – Apply the same operation to multiple rows or columns without manual repetition.
2. **Efficient Data Cleaning** – Remove missing values, standardize data formats, or perform conditional modifications.
3. **Batch Processing** – Process multiple files, datasets, or elements efficiently.
4. **Flexible Data Manipulation** – Apply functions to subsets of data dynamically.

```
# Activity Two

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

# While loop example
sum_val <- 0
i <- 1
while (i <= 50) {
  sum_val <- sum_val + i
  i <- i + 1
}
print(sum_val)

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

```
> # 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
> # while loop example
> sum_val <- 0
> i <- 1
> while (i <= 50) {
+   sum_val <- sum_val + i
+   i <- i + 1
+ }
> print(sum_val)
[1] 1275
> # Factorial using for loop
> num <- 5
> fact <- 1
> for (i in 1:num) {
+   fact <- fact * i
+ }
> print(fact)
[1] 120
```

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

```
# Creating a 2x3 array (acts as a matrix)
A <- array(c(1, 2, 3, 4, 5, 6), dim = c(2, 3))
print("Matrix A (2x3):")
print(A)

# Creating a 3x2 matrix (compatible for multiplication with A)
B <- matrix(c(7, 8, 9, 10, 11, 12), nrow = 3, ncol = 2)
print("Matrix B (3x2):")
print(B)

# Performing matrix multiplication (A %*% B)
C <- A %*% B
print("Result of A %*% B (2x2):")
print(C)
```

```
> # Creating a 2x3 array (acts as a matrix)
> A <- array(c(1, 2, 3, 4, 5, 6), dim = c(2, 3))
> print("Matrix A (2x3):")
[1] "Matrix A (2x3):"
> print(A)
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> # Creating a 3x2 matrix (compatible for multiplication with A)
> B <- matrix(c(7, 8, 9, 10, 11, 12), nrow = 3, ncol = 2)
> print("Matrix B (3x2):")
[1] "Matrix B (3x2):"
> print(B)
      [,1] [,2]
[1,]    7   10
[2,]    8   11
[3,]    9   12
> # Performing matrix multiplication (A %*% B)
> C <- A %*% B
> print("Result of A %*% B (2x2):")
[1] "Result of A %*% B (2x2):"
> print(C)
      [,1] [,2]
[1,]   76  103
[2,]  100  136
> |
```

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

```
# Reading CSV file
#data <- read.csv("Rdata.csv") #same directory
data <- read.csv("C:/Users/User/Desktop/DATA MINING/LAB/Lab2/data.csv")
print(data)
#file.exists("data.csv")(nak tau kalau file tu exist atau tidak)(false tak exist)(true exist)
#getwd() (nak tau current directory)
#list.files()(nak tau file yang ada dekat dalam directory)
```

```
> # Activity Four
> # Reading CSV file
> #data <- read.csv("Rdata.csv") #same directory
> data <- read.csv("C:/Users/User/Desktop/DATA MINING/LAB/Lab2/data.csv")
> print(data)
```

| | Name | Age | Score |
|---|------|-----|-------|
| 1 | Ali | 22 | 85 |
| 2 | Siti | 21 | 90 |
| 3 | John | 23 | 78 |