| **Name of Student: Pushkar Sane** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Roll Number: 45** | | | **Lab Assignment Number: 1** | | | |
| **Title of Lab Assignment: Write the commands for the following Data acquisition, install packages, loading packages, Data types, checking type of variables, printing variables and objects (Vector, matrix, list, factor, data frame, tables).** | | | | | | |
| **DOP: 15-09-2023** | | | **DOS: 16-09-2023** | | | |
| **CO Mapped:**  **CO1** | **PO Mapped:**  **PO1, PO2, PO3, PO4,**  **PO6, PO7, PSO1, PSO2** | | | **Signature:** | | |

**Practical No. 1**

**Aim:** Write the commands for the following Data acquisition, install packages, loading packages, Data types, checking type of variables, printing variables and objects (Vector, matrix, list, factor, data frame, tables).

**Theory:**

1. **Data acquisition:** In R programming, there are various ways to acquire data, whether from local files, databases, APIs, or other sources. Here are some commonly used data acquisition commands in R:

Reading Data from Files:

Reading CSV files: read.csv(), read.csv2(), read.delim(), read.table()

Reading Excel files: read.xlsx() (using the openxlsx package) or readxl package

Reading text files: readLines(), scan()

**Example:** # Reading a CSV file data

<- read.csv("data.csv") # Reading an Excel file library(openxlsx) data

<- read.xlsx("data.xlsx")

1. **Install Packages:** In R programming, the install.packages() command is used to install packages from CRAN (Comprehensive R Archive Network) or other repositories. Packages are collections of functions, data sets, and documentation that provide additional functionality to R.

**Example:** install.packages("package\_name")

1. **Loading Packages:** In R programming, you can load packages using the library() function. Loading a package makes its functions, datasets, and other features available for use in your R script or session.

**Example:** library(package\_name)

1. **Data Types:** In R programming, there are several data types you can work with. Some common data types in R include:
   1. **Numeric**: Used for representing numerical values, such as integers or decimals. For example, 5, 3.14.
   2. **Character:** Used for representing text or strings. Strings are enclosed in double or single quotes. For example, "Hello", 'World'.
   3. **Logical:** Used for representing boolean values, which can be either TRUE or FALSE.
   4. **Integer:** A special type of numeric data that represents whole numbers without decimals. For example, 10, -3.
   5. **Matrix:** A matrix is a 2-dimensional data structure with rows and columns.

Example: mat <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3)

* 1. **Data Frame:** A data frame is a 2-dimensional table-like structure where each column can have a different data type. Eg: df <- data.frame(name = c("Alice", "Bob"), age = c(25, 30)).
  2. **List:** A list is a collection of elements that can be of different data types.

Eg: my\_list <- list(name = "Alice", age = 25, is\_student = TRUE)

1. **Checking types of variables:** In R programming, you can use several functions to determine the data type of an object or variable. Here are some common functions you can use to find data types:
   1. **class():** The class() function returns the class or data type of an R object.

Function:

x <- 5

y <- “Hello”

print(class(x)) # Output: “numeric”

print(class(y)) # Output: “character”

* 1. **typeof():** The typeof() function returns the fundamental object of an R object.

Function:

x <- 5

y <- “Hello”

print(typeof(x)) # Output: “double”

print(class(y)) # Output: “character”

**Output:**

**Code:**

# Commands to install packages from source editor

install.packages("openxlsx")

install.packages("csvread")

install.packages("XLS")

# Data Acquisition examples

# Loading packages "readxl" and "csvread'

library(readxl)

# Importing excel file to perform operations

SalesData <- read\_excel("F:/Pushkar/MCA/Sem-1/DAR/SalesData.xlsx")

View(SalesData)

library(csvread)

# Importing .csv file to perform operations

SalesData1 <- read.csv("F:/Pushkar/MCA/Sem-1/DAR/SalesData1.csv")

SalesData1

# Performing arithmetic operations using R

# Printing variables and objects

# Addition

a <- 5

b <- 10

sum\_result <- a + b

print(sum\_result)

# Subtraction

difference <- b - a

print(difference)

# Multiplication

product <- a \* b

print(product)

# Division

quotient <- b / a

print(quotient)

# Exponentiation

power <- a^2

print(power)

# Modulus (remainder)

remainder <- b %% a

print(remainder)

# Data Types in R

# Identifying data types using class()

x <- 5.7 # Numeric(Double)

class(x)

y <- 10L # The 'L' suffix indicates an integer

class(y)

name <- "Alice" # Character

class(name)

is\_valid <- TRUE #Logical(Boolean)

class(is\_valid)

z <- 3 + 2i # complex

class(z)

gender <- factor(c("Male", "Female", "Male", "Female")) #Factor class(gender)

today <- as.Date("2023-08-30") #Date and Time

class(today)

nums <- c(1, 2, 3, 4, 5) #Vector

class(vector())

mat <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3) #Matrix

class(mat)

df <- data.frame(name = c("Alice", "Bob"), age = c(25, 30)) #Data Frame

class(df)

my\_list <- list(name = "Alice", age = 25, is\_student = TRUE) #List

class(my\_list)

**Console Output:**

> # Data Acquisition examples

> # Loading packages "readxl" and "csvread'

> library(readxl)

>

> # Importing excel file to perform operations

> SalesData <- read\_excel("F:/Pushkar/MCA/Sem-1/DAR/SalesData.xlsx")

> View(SalesData)

>

> library(csvread)

> # Importing .csv file to perform operations

> SalesData1 <- read.csv("F:/Pushkar/MCA/Sem-1/DAR/SalesData1.csv")

> SalesData1

month\_number facecream facewash toothpaste bathingsoap shampoo moisturizer total\_units total\_profit

1 1 2500 1500 5200 9200 1200 1500 21100 211000

2 2 2630 1200 5100 6100 2100 1200 18330 183300

3 3 2140 1340 4550 9550 3550 1340 22470 224700

4 4 3400 1130 5870 8870 1870 1130 22270 222700

5 5 3600 1740 4560 7760 1560 1740 20960 209600

>

> # Performing arithmetic operations using R

> # Printing variables and objects

> # Addition

> a <- 5

> b <- 10

> sum\_result <- a + b

> print(sum\_result)

[1] 15

>

> # Subtraction

> difference <- b - a

> print(difference)

[1] 5

>

> # Multiplication

> product <- a \* b

> print(product)

[1] 50

>

> # Division

> quotient <- b / a

> print(quotient)

[1] 2

>

> # Exponentiation

> power <- a^2

> print(power)

[1] 25

>

> # Modulus (remainder)

> remainder <- b %% a

> print(remainder)

[1] 0

>

> # Data Types in R

> # Identifying data types using class()

> x <- 5.7 # Numeric(Double)

> class(x)

[1] "numeric"

> y <- 10L # The 'L' suffix indicates an integer

> class(y)

[1] "integer"

> name <- "Alice" # Character

> class(name)

[1] "character"

>

> is\_valid <- TRUE #Logical(Boolean)

> class(is\_valid)

[1] "logical"

> z <- 3 + 2i # complex

> class(z)

[1] "complex"

> gender <- factor(c("Male", "Female", "Male", "Female")) #Factor class(gender)

> today <- as.Date("2023-08-30") #Date and Time

> class(today)

[1] "Date"

> nums <- c(1, 2, 3, 4, 5) #Vector

> class(vector())

[1] "logical"

> mat <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3) #Matrix

> class(mat)

[1] "matrix" "array"

> df <- data.frame(name = c("Alice", "Bob"), age = c(25, 30)) #Data Frame

> class(df)

[1] "data.frame"

> my\_list <- list(name = "Alice", age = 25, is\_student = TRUE) #List

> class(my\_list)

[1] "list"

**Conclusion:** Successfully performed the commands required to perform the data acquisition, install packages, loading packages, data types, checking the type of variables, print variables and objects like vector, matrix, list, factor, data frame, tables.