| **Course Name:** | **Introduction to Data Science** | **Semester:** | **IV** |
| --- | --- | --- | --- |
| **Date of Performance:** | **6 / 1 / 2025** | **DIV/ Batch No:** | **IDS\_6** |
| **Student Name:** | **Om Satish Thanage** | **Roll No:** | **16010123217** |

**Experiment No: 1**

**Title: Exploring R for Data Science**

| **Aim and Objective of the Experiment:** |
| --- |
| Understand the basic data types, operations and visualization using R. |

| **COs to be achieved:** |
| --- |
| **CO1:** Understand the significance of Data Science processes in real world applications  CO3: Learn data cleaning, transformation, and feature engineering techniques. |

| **Theory:** |
| --- |
| R can be downloaded from<http://cran.r-project.org/> .We recommend that you run R within an integrated development environment (IDE) such as RStudio, which can be freely downloaded from<http://rstudio.com>RStudio Pane LayoutThe RStudio user interface has 4 primary panes:Source paneConsole paneEnvironment pane, containing the Environment, History, Connections, Build, VCS , and Tutorial tabsOutput pane, containing the Files, Plots, Packages, Help, Viewer, and Presentation tabsEach pane can be minimized or maximized within the column by clicking the minimize/maximize buttons.Basic Arithmetic OperatorsArithmetic operations: +, -, \*, /, ^, %%, %/%Assigning values: <- , =Basic Data TypesVectors and Basic Vector OperationsR uses functions to perform operations.To run a function called *funcname*, we type *funcname(input1, input2)*, where the inputs (or arguments) *input1* and *input2* tell R how to run the function. A function can have any number of inputs. For example, to create a vector of numbers, we use the function c() (for concatenate).Any numbers inside the parentheses are joined together.The following command instructs R to join together the numbers 1, 3, 2, and 5, and to save them as a vector named x.When we type x, it gives us back the vector. Other ways to create vectors are using seq() and rep()We can tell R to add two sets of numbers together. It will then add the first number from x to the first number from y, and so on. However, x and y should be the same length. We can check their length using the length() function.The ls() function allows us to look at a list of all of the objects, such as data and functions, that we have saved so far. The rm() function can be used to delete any that we don’t want.It’s also possible to remove all objects at once:The matrix() function can be used to create a matrix of numbers. Before we use the matrix() function, we can learn more about it:The help file reveals that the matrix() function takes a number of inputs, but for now we focus on the first three: the data (the entries in the matrix), the number of rows, and the number of columns. First, we create a simple matrix.Note that we could just as well omit typing data=, nrow=, and ncol= in the matrix() command above: that is, we could just typeand this would have the same effect. However, it can sometimes be useful to specify the names of the arguments passed in, since otherwise R will assume that the function arguments are passed into the function in the same order that is given in the function’s help file. As this example illustrates, by default R creates matrices by successively filling in columns. Alternatively, the byrow = TRUE option can be used to populate the matrix in order of the rows.Notice that in the above command we did not assign the matrix to a value such as x. In this case the matrix is printed to the screen but is not saved for future calculations.The sqrt() function returns the square root of each element of a vector or matrix. The command x^2 raises each element of x to the power 2; any powers are possible, including fractional or negative powers.Creating a listCreating dataframesWriting dataframe to csv file  Reading dataframe from a csv file  Basic Visualizationplot()  hist()The hist() function in R is used to create histograms, a graphical representation of the distribution of a dataset. It divides the data into bins (intervals) and counts the frequency of data points in each bin.hist(x, breaks, main, xlab, ylab, xlim, ylim, col, border, ...) **Key Arguments**   * **x**: A numeric vector of data values. * **breaks**: Specifies the number of bins or the breakpoints for the histogram. It can be:   + A single number (number of bins).   + A vector specifying the bin boundaries. * **main**: Title for the histogram. * **xlab**: Label for the x-axis. * **ylab**: Label for the y-axis. * **xlim, ylim**: Limits for the x and y axes, respectively. * **col**: Color of the bars. * **border**: Color of the bar borders.    Scatterplot, using ggplot2     **Students have to perform the above tasks and add their code and screenshots of the output here :** |

| **Problem Statements:** |
| --- |
|  |

| **Code :** |
| --- |
| 1.Declaring variables and performing arithmetic operations  > n1 = 10  > n2 = 20  > q <- n2%/%n1  > r = n2 %% n1  > q  [1] 2  > r  [1] 0  2. Check data types using typeof and class  > x = 1000L  > class(x)  [1] "integer"  > typeof(x)  [1] "integer"  3. Vector Operations  > v1 = c(1,6,8,9,10)  > v1  [1] 1 6 8 9 10  > length(v1)  [1] 5  > odd\_vec = seq(1, 20, by=2)  > odd\_vec  [1] 1 3 5 7 9 11 13 15 17 19  > rep\_vec = rep(5, times = 10)  > rep\_vec  [1] 5 5 5 5 5 5 5 5 5 5  > odd\_vec + rep\_vec  [1] 6 8 10 12 14 16 18 20 22 24  > rm(rep\_vec)  > ls(rep\_vec)  Error in as.environment(pos) :  no item called "rep\_vec" on the search list  In addition: Warning message:  In ls(rep\_vec) : ‘rep\_vec’ converted to character string  4. Matrix operations  > mat = matrix( c(1,23,4,7), 2, 2)  > mat  [,1] [,2]  [1,] 1 4  [2,] 23 7  > sqrt(mat)  [,1] [,2]  [1,] 1.000000 2.000000  [2,] 4.795832 2.645751  5. List  > my\_list = list(name = "Om", age = "19")  > my\_list  $name  [1] "Om"  $age  [1] "19"  > my\_list$age  [1] "19"  6. Data frames  > df = data.frame( Name= c("Om", "Omkar", "Omsiddh", "Paaras", "Omik"), Age = rep(19, times= 5), City = c("Thane", "Panvel", "Nerul", "Thane", "Powai"))  > df  Name Age City  1 Om 19 Thane  2 Omkar 19 Panvel  3 Omsiddh 19 Nerul  4 Paaras 19 Thane  5 Omik 19 Powai  > rep\_df = cbind(df, rep(row.names(df), each = 2))  > rep\_df  Name Age City rep(row.names(df), each = 2)  1 Om 19 Thane 1  2 Omkar 19 Panvel 1  3 Omsiddh 19 Nerul 2  4 Paaras 19 Thane 2  5 Omik 19 Powai 3  6 Om 19 Thane 3  7 Omkar 19 Panvel 4  8 Omsiddh 19 Nerul 4  9 Paaras 19 Thane 5  10 Omik 19 Powai 5  Writing dataframe to csv file  > write.csv(rep\_df, "output.csv")  Reading dataframe from a csv file  > new\_df = read.csv("output.csv")  > new\_df  X Name Age City rep.row.names.df...each...2.  1 1 Om 19 Thane 1  2 2 Omkar 19 Panvel 1  3 3 Omsiddh 19 Nerul 2  4 4 Paaras 19 Thane 2  5 5 Omik 19 Powai 3  6 6 Om 19 Thane 3  7 7 Omkar 19 Panvel 4  8 8 Omsiddh 19 Nerul 4  9 9 Paaras 19 Thane 5  10 10 Omik 19 Powai 5  Basic Visualization:   1. Plot()   > x = 1:10  > y = x^3  > plot(x, y, type="o", col="green")   1. Hist()   > hist(c(1,2,2,3,4,5,6))   1. Scatter Plot using ggplot2   > d1 = data.frame(x = 1: 10, y = c(2,4,6,8,10,12,14,16,18,20))  > ggplot(d1, aes(x = x, y = y))+ geom\_point(color = "blue", size = 3) + ggtitle("SCATTERPLOT using ggplot2")+ xlab("X-axis") + ylab("Y-axis") |

| **Output:** |
| --- |
| Data Frames:      Graph: |

| **Post Lab Subjective/Objective type Questions:** |
| --- |
| 1. Compare the additive and multiplicative model of time series   ANS: |

| **Conclusion:** |
| --- |
| Got basic introduction of R and it installation |

| **Signature of faculty in-charge with Date:** |
| --- |