SANTDNYANESHWAR SHIKSHANSANSTHA'S
ANNASAHEB DANGE COLLEGE OF ENGINEERING & TECHNOLOGY, ASHTA

DEPARTMENT OF Artificial Intelligence and Data Science

LABORATORY MANUAL

Data Science using R (1ADPE306)

Prepared by
Mrs. AmrutaAwati

THIRD YEAR

AcademicYear

2024-25

INDEX

Experiment No:01

Title : Installing R and R studio, Create a folder DS_R and make it a working directory, installing the "ggplot2","caTools","CART" packages.

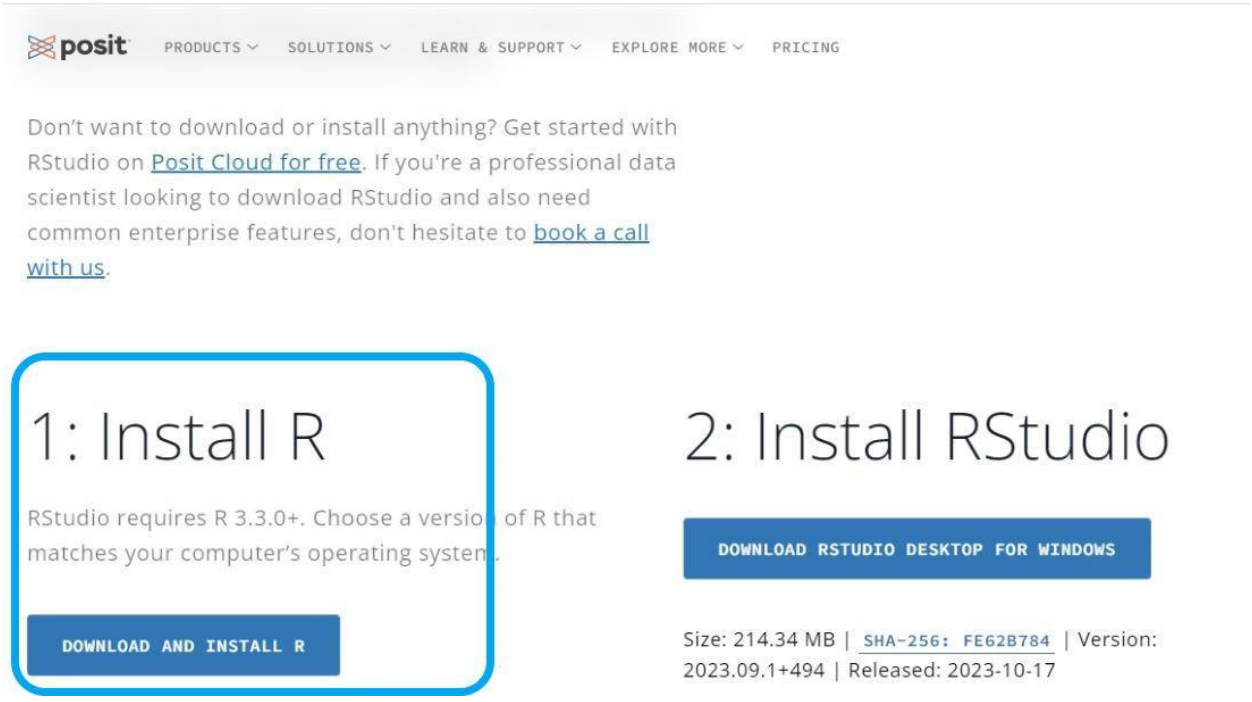
Objective :

- Installing R and RStudio
- Creating a Folder DS_R and Making It the Working Directory
- Installing Packages ("ggplot2", "caTools", "CART")

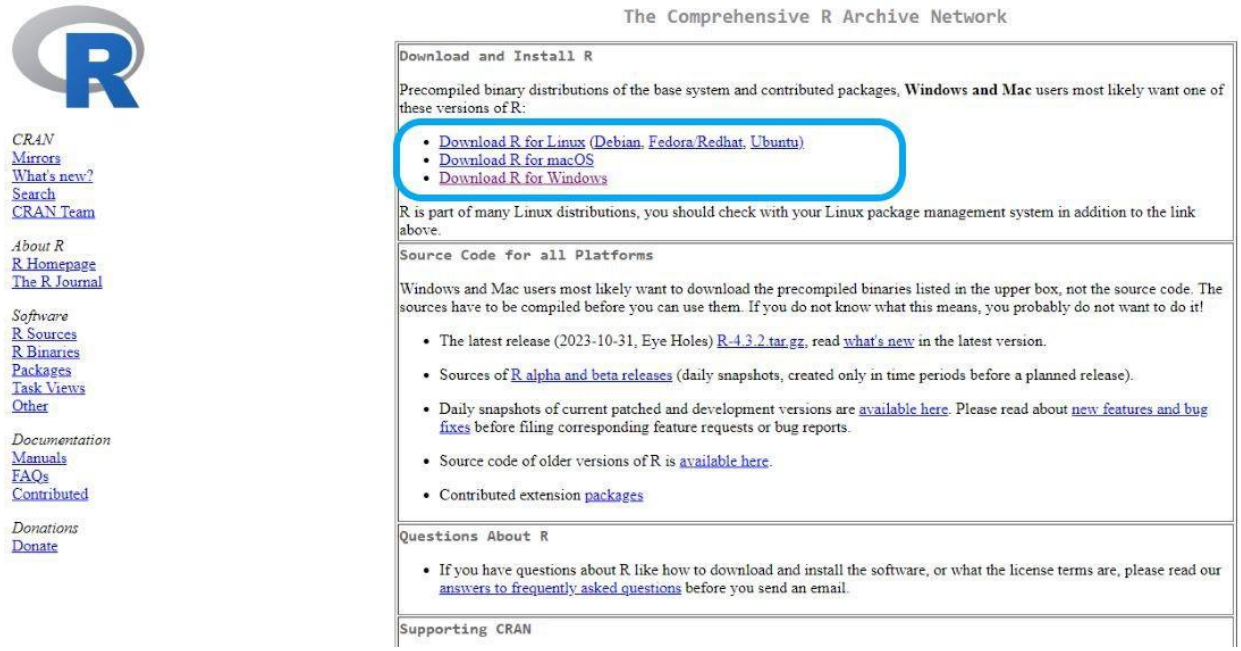
Theory

R programming language is a language and free software environment, available under GNU license, supported by R Foundation for Statistical Computing. The language is most widely known for its powerful statistical and data interpretation capabilities.

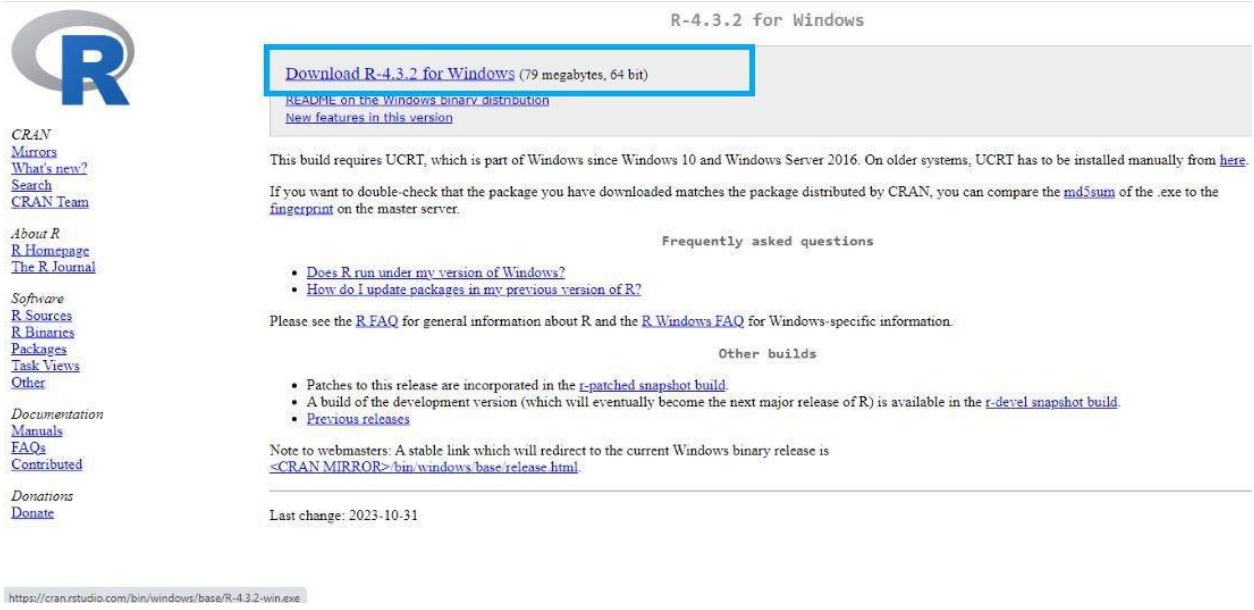
Step 1: First, we need to set up an R environment in our local machine. We can download the same from r-project.org.



We have to download both the applications first go with R Base and then install RStudio. After click on install R we will get a new page like this.



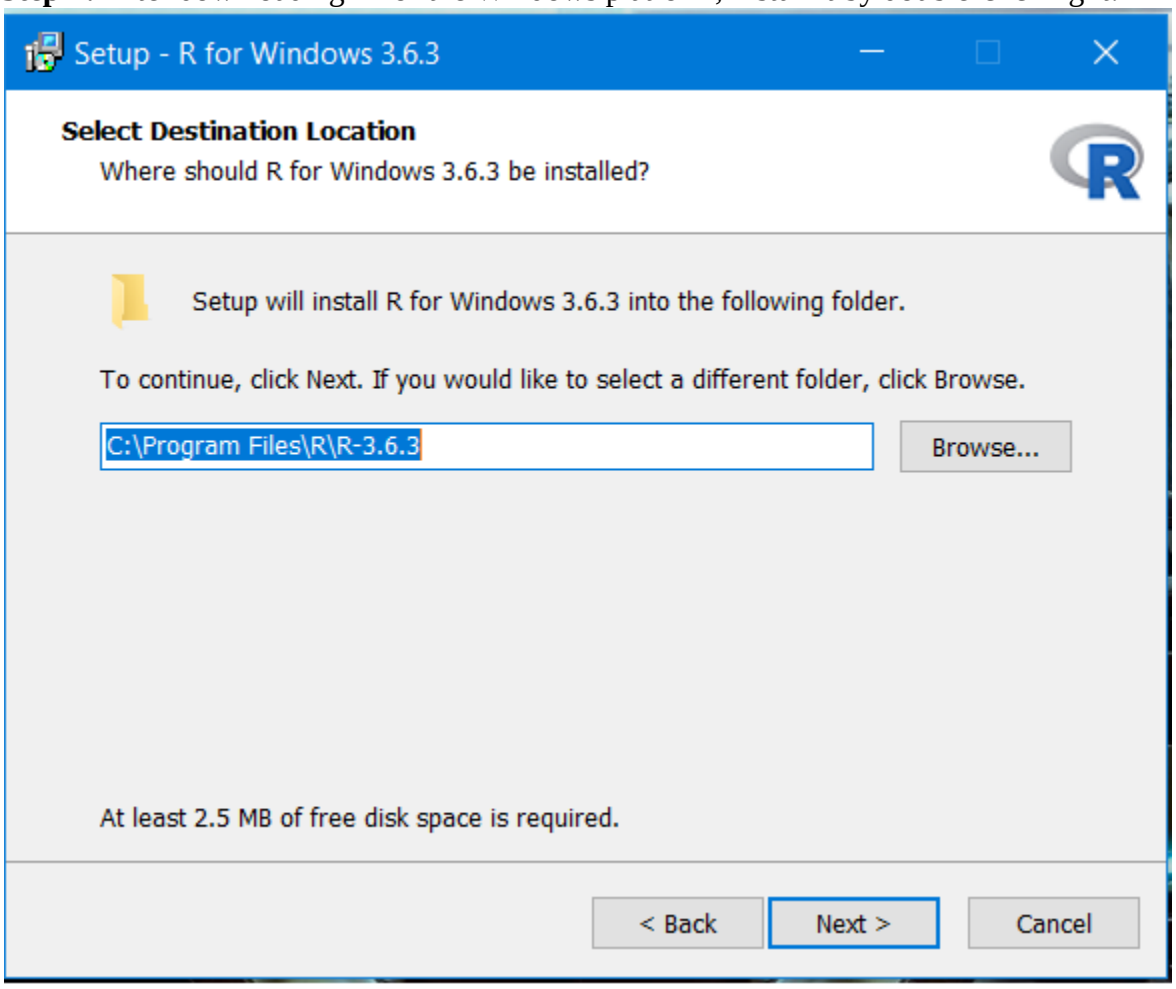
Here we can select the linux,mac or windows any one according to users system. We have to click on for which we want to install.



now click on the link show above in image so R base start downloading and after again go to main page and download and click on Install RStudio.

Steps to Install R and R Studio

Step 1: After downloading R for the Windows platform, install it by double-clicking it.



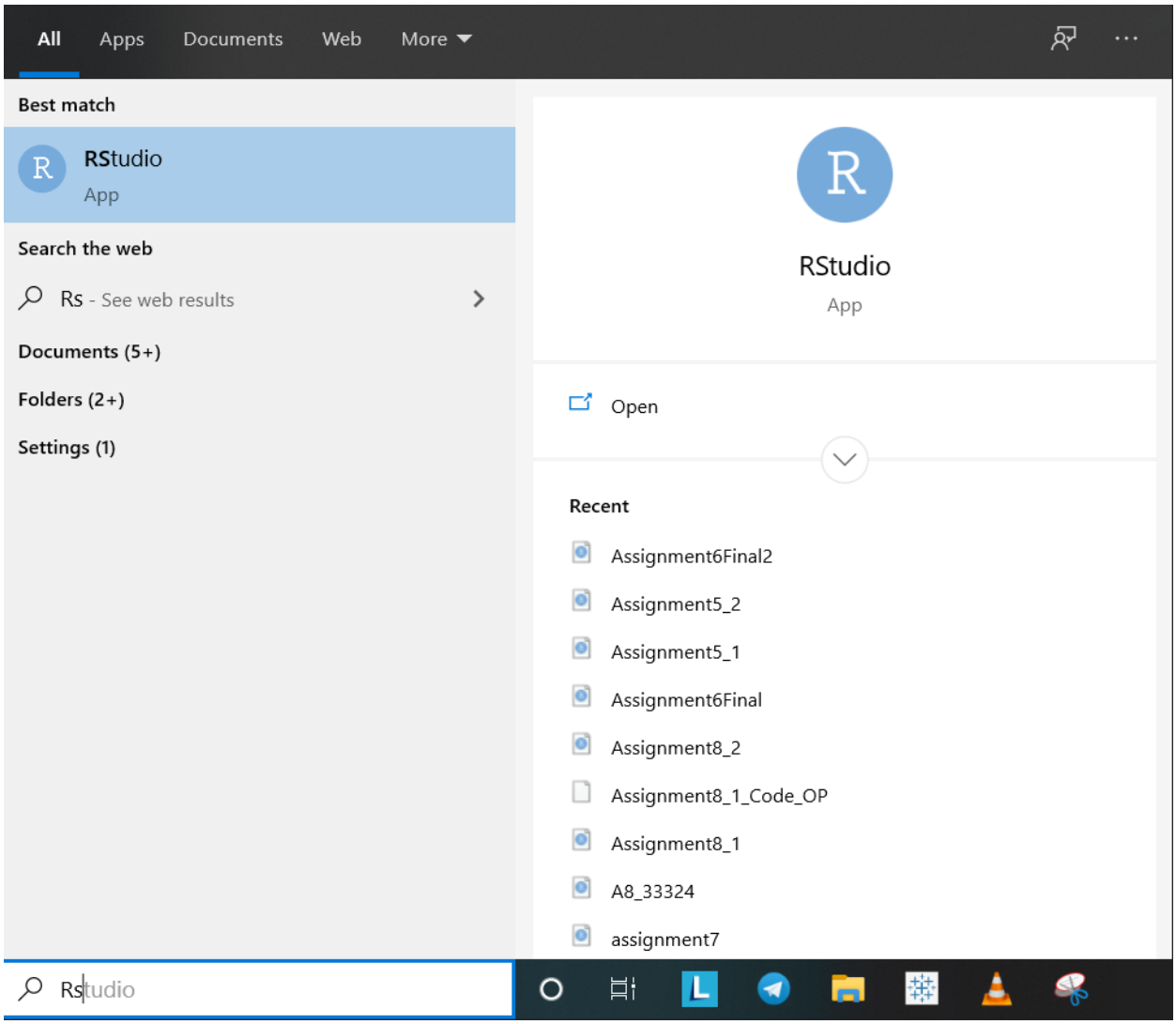
Step 2: Download R Studio from their official page.

Step 3: After downloading, we will get a file named "RStudio-1.x.xxxx.exe" in our Downloads folder.

Step 4: Double-click the installer, and install the software.

Step 5: Test the R Studio installation

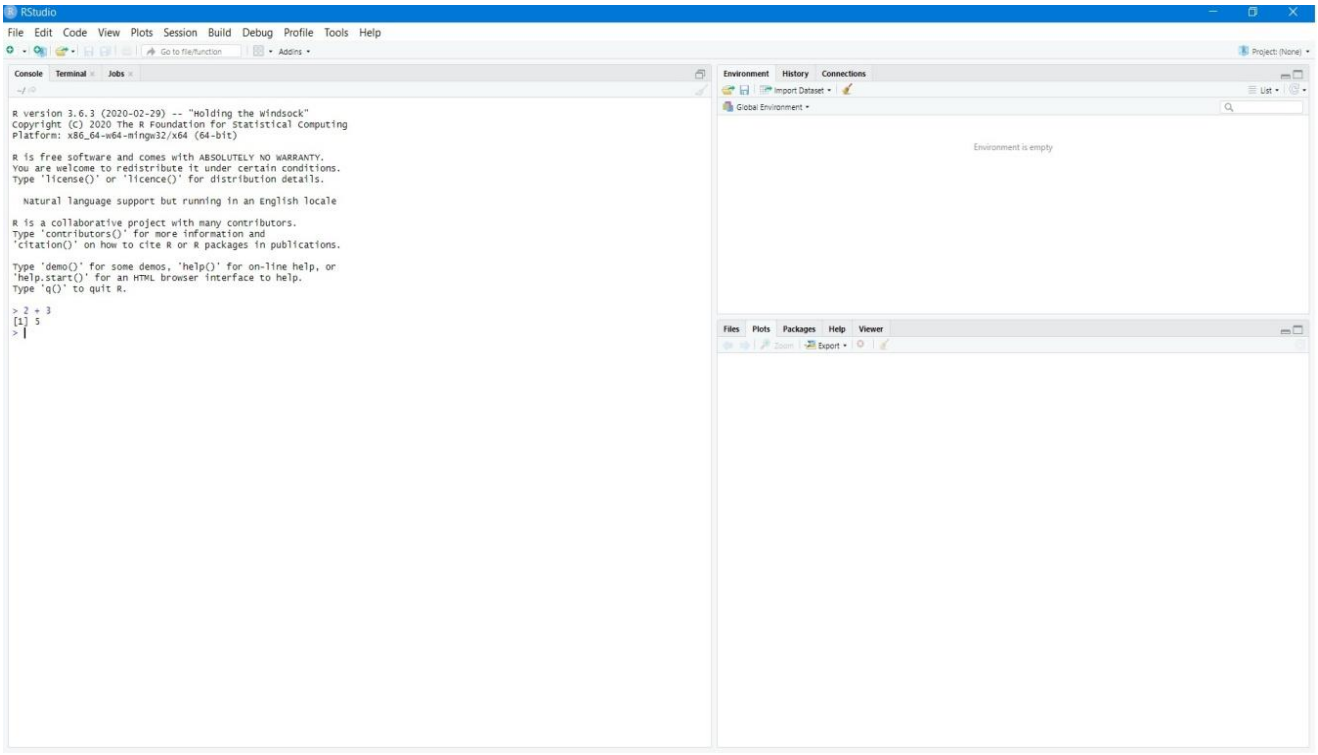
- Search for RStudio in the Window search bar on Taskbar.



- Start the application.

- Insert the following code in the console.

Input :print('Hello world!')
Output :[1] "Hello world!"



Step 6: Our installation is successful.

Set Working directory in Rstudio

setwd(): This method is used to set the specified pathname as the current working space directory of the R console.
Syntax: setwd(dir)
Where, dir is specify a working directory.

The setwd() function allows us to move a hierarchy level up in the folder domain using the ".." argument in this method.

Install a Package in R

Method 1: Using application options

1. Open R studio.
2. Select tools
3. After selecting the tools we need to press install packages.
4. Here we need to give the package name we need to install.

Installing packages in RGui:
1. Open RGui
2. Select packages
3. Select install packages.
4. Select required package and click ok.

Method 2: Using command
In this method, simply pass the package to be installed as an argument to install.packages() function
Syntax:
install.packages("package name")

Loading a package
A package can be loaded once it has been installed using **library()** command.
Syntax;
library("package_name")

Outcome:
- R and RStudio Installed and Configured
- Project Folder `DS_R` Created and Set as Working Directory
- Essential R Packages Installed

Conclusion :

This practical successfully set up an R environment for data science. We have installed R and RStudio, organized our project files with a dedicated working directory, and installed essential packages for data visualization and modeling. We are now ready to start analyzing data and building models using R.

Experiment No:02

Title: Learn all the basics of R –Programming (Data types, Variables, Operators etc)

Objectives :

1. Understand Basic Data Types in R
2. Learn About Variables and Assignment
3. Get Familiar with Operators
4. Explore Data Structures (Vectors, Lists, Matrices, Data Frames)
5. Understand Basic Functions and Control Structures

Theory:

R divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators
- Miscellaneous operators

R Arithmetic Operators

Arithmetic operators are used with numeric values to perform common mathematical operations:

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
^	Exponent	x ^ y
%%	Modulus (Remainder from division)	x %% y
%/%	Integer Division	x%/%y

R Assignment Operators

Assignment operators are used to assign values to variables

R Comparison Operators

Comparison operators are used to compare two values

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

R Logical Operators

Logical operators are used to combine conditional statements

Operator	Description
&	Element-wise Logical AND operator. It returns TRUE if both elements are TRUE
&&	Logical AND operator - Returns TRUE if both statements are TRUE
|	Elementwise- Logical OR operator. It returns TRUE if one of the statement is TRUE
||	Logical OR operator. It returns TRUE if one of the statement is TRUE.
!	Logical NOT - returns FALSE if statement is TRUE

R Miscellaneous Operators

Miscellaneous operators are used to manipulate data:

Operator	Description	Example
:	Creates a series of numbers in a sequence	x <- 1:10
%in%	Find out if an element belongs to a vector	x %in% y
%*%	Matrix Multiplication	x <- Matrix1 %*% Matrix2

Variables in R

R Programming Language is a dynamically typed language, i.e. the R Language Variables are not declared with a data type rather they take the data type of the R-object assigned to them. This feature is also shown in languages like Python and PHP.

Declaring and Initializing Variables in R Language

R supports three ways of variable assignment:

- Using equal operator- operators use an arrow or an equal sign to assign values to variables.
- Using the leftward operator- data is copied from right to left.
- Using the rightward operator- data is copied from left to right.

R Variables Syntax

Types of Variable Creation in R:

- *Using equal to operators*
  variable_name = value

- *using leftward operator*
  variable_name<- value

- *using rightward operator*
  value ->variable_name

Data Types in R Programming Language

Basic Data Types	Values	Examples
Numeric	Set of all real numbers	"numeric_value<- 3.14"

Basic Data Types	Values	Examples
Integ	Set of all integers, Z	"integer_value<- 42L"
Logical	TRUE and FALSE	"logical_value<- TRUE"
Complex	Set of complex numbers	"complex_value<- 1 + 2i"
Character	“a”, “b”, “c”, …, “@”, “#”, “$”, …., “1”, “2”, …etc	"character_value<-"Hello Geeks"
raw	as.raw()	"single_raw<- as.raw(255)"

We can use the class() function to check the data type of a variable.

Problem Statement:

Note: The students should perform for following statements using variables, datatypes and operators.

1. Create variables a and b with values of your choice (e.g., a = 15 and b = 4).
2. Calculate the following using these variables:

- Sum of a and b
- Difference between a and b
- Product of a and b
- Quotient of a divided by b
- Remainder when a is divided by b
- a raised to the power of b
- Print all the results with appropriate labels.

Problem 2: Data Type Conversion

1. Create a numeric variable num with a value (e.g., num = 123.456).
2. Convert num to an integer and a character. Store these conversions in new variables num_int and num_char, respectively.
3. Create a character variable char_val with a value (e.g., "789").
4. Convert char_val to numeric and store it in a variable char_to_num.
5. Print the original values and their converted counterparts.

Problem 3: Logical Operations and Conditions

1. Create two variables x and y (e.g., x = 10 and y = 20).
2. Check and print whether x is greater than y.
3. Check and print whether x is less than or equal to y.
4. Create a logical variable is_even that is TRUE if x is even and FALSE otherwise.
5. Print the value of is_even.

Problem 4: Operations on Vectors

1. Create two numeric vectors vec1 and vec2 with arbitrary values (e.g., vec1 = c(1, 2, 3) and vec2 = c(4, 5, 6)).
2. Perform and print the following operations:
    - Element-wise addition of vec1 and vec2
    - Element-wise multiplication of vec1 and vec2
    - Sum of all elements in vec1
    - Mean of all elements in vec2
3. Verify the types of vec1 and vec2.

Problem 6: Creating and Using Lists

1. Create a list named personal_info with the following elements:
    - A numeric vector age with values 25, 30, 35
    - A character vector names with values "Alice", "Bob", "Charlie"
    - A logical vector is_member with values TRUE, FALSE, TRUE
2. Print the entire list.
3. Extract and print:
    - The names vector
    - The second element of the age vector
    - The first element of the is_member vector

Outcome:

The students will learn basic idea of working with variables, datatypes and operators in R.

R programming Exercises, Practice, Solution: To encourage students to participate actively ,students will use W3 resource(Practice programs) and to share their results and experiences with others. It helps learner to better understand the theoretical information.

Link:

- https://www.w3resource.com/r-programming-exercises/basic/index.php

Viva Questions:

1.	**What is R programming and what are main feature of R?**
2.	**What Are Some Advantages and drawbacks of R?**
3.	**Explain different data types in R.**
4.	**How to Concatenate Strings in R?**
5.	**Write R program to input number from user and calculate its absolute value.**
6.	**Explain how type casting will be done in R Programming.**
7.	**Write a R program to create a list named s containing sequence of 15 capital letters, starting from 'E'.**

Conclusion: Basics Knowledge helps students to manage data and perform tasks in R, setting students up for more advanced work.

Experiment 3

Title: Implement R loops with different examples.

Objectives:

1. Understand the syntax of different types of loops in R.
2. Learn how to use loops to perform repetitive tasks efficiently.
3. Explore common applications and best practices for using loops in R.

Theory:

In R programming, we require a control structure to run a block of code multiple times. Loops come in the class of the most fundamental and strong programming concepts. A loop is a control statement that allows multiple executions of a statement or a set of statements. The word 'looping' means cycling or iterating.

A loop asks a query, in the loop structure. If the answer to that query requires an action, it will be executed. The same query is asked again and again until further action is taken. Any time the query is asked in the loop, it is known as an iteration of the loop. There are two components of a loop, the control statement, and the loop body. The control statement controls the execution of statements depending on the condition and the loop body consists of the set of statements to be executed.

In order to execute identical lines of code numerous times in a program, a programmer can simply use a loop.

There are three types of loops in R programming:

- For Loop
- While Loop
- Repeat Loop

For Loop in R

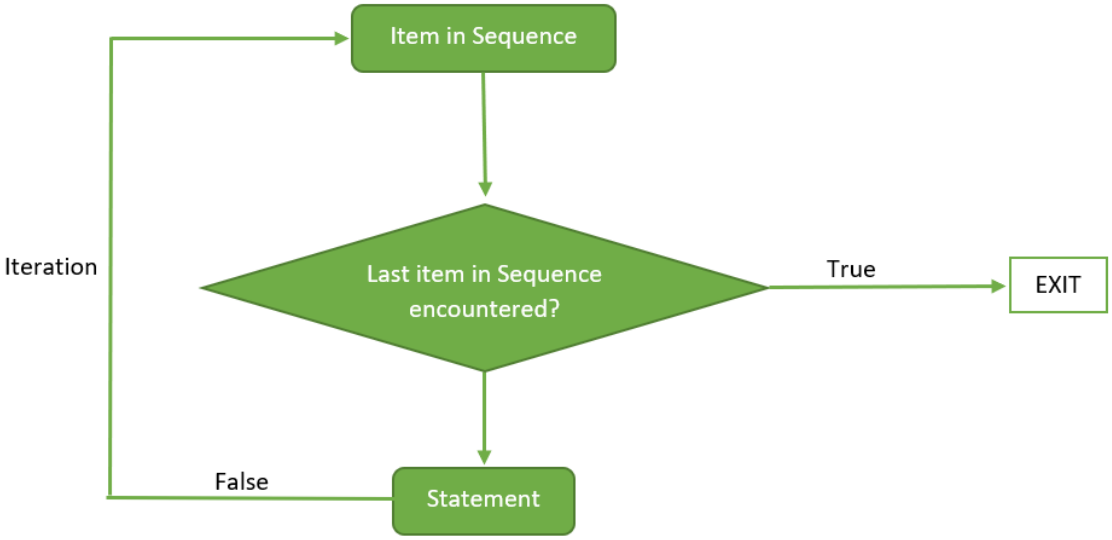
It is a type of control statement that enables one to easily construct an R loop that has to run statements or a set of statements multiple times. For R loop is commonly used to iterate over items of a sequence. It is an entry-controlled loop, in this loop, the test condition is tested first, then the body of the loop is executed, the loop body would not be executed if the test condition is false.

R – For loop Syntax:

```
for (value in sequence)
{
```

```
statement
}
```

For Loop Flow Diagram:



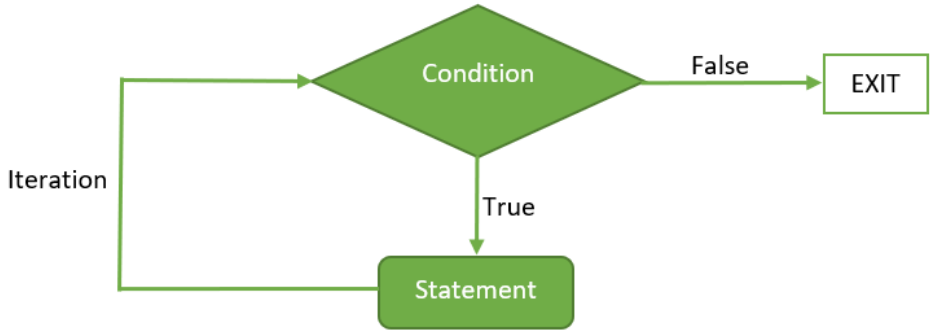
While Loop in R

It is a type of control statement that will run a statement or a set of statements repeatedly unless the given condition becomes false. It is also an entry-controlled loop, in this loop, the test condition is tested first, then the body of the loop is executed, the loop body would not be executed if the test condition is false.

R – While loop Syntax:

```
while ( condition )
{
statement
}
```

While loop Flow Diagram:



Repeat Loop in R

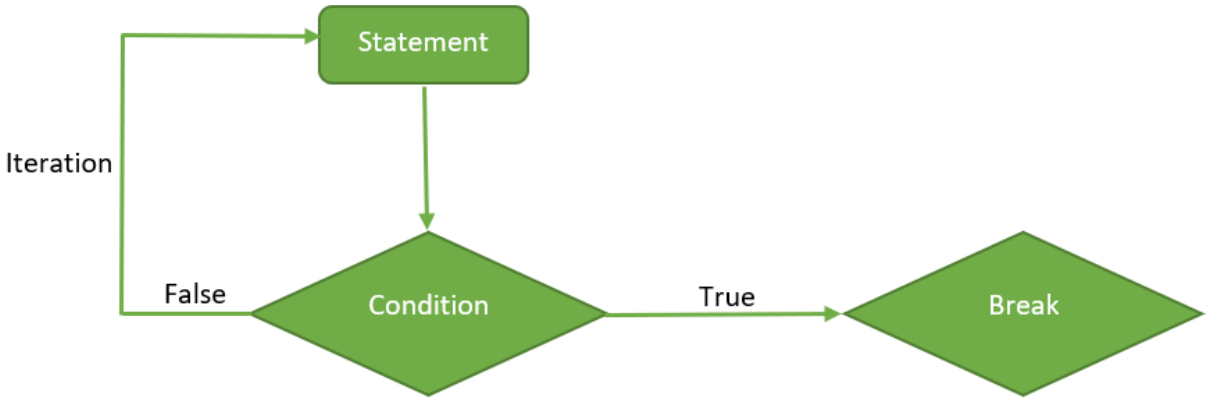
It is a simple loop that will run the same statement or a group of statements repeatedly until the stop condition has been encountered. Repeat loop does not have any condition to terminate the loop, a programmer must specifically place a condition within the loop's body and use the declaration of a break statement to terminate this loop. If no condition is present in the body of the repeat loop then it will iterate infinitely.

R – Repeat loop Syntax:

```
repeat
{
statement


if( condition )
   {
break
   }
}
```

Repeat loop Flow Diagram:



To terminate the **repeat** loop, we use a jump statement that is the **break** keyword. Below are some programs to illustrate the use of repeat loops in R program.

Problem Statement:

Note: The students should perform for following statements using For loop,while loop and Repeat loop.

1. Program to display days of the week
2. **Program to calculate Fibinocci series of given number.**
3. **Program to calculate Factorial of given number.**

Outcome:

The students will learn basic idea of working of R loops.The students will able to perform programs on R loops.

Online Reference Websites:

1. **https://www.geeksforgeeks.org/loops-in-r-for-while-repeat/**
2. **https://www.w3schools.com/r/r_for_loop.asp**
3. **https://www.javatpoint.com/r-tutorial**

Viva Questions:

1.	Explain for loop and while loop in R.
2.	What types of loops exist in R, and what is the syntax of each type
3.	How to aggregate data in R?

4.	How to transpose two-dimensional data in R?
5.	How do you get the name of the current working directory in R?

Extra Learning:

The students can do the extra activity to increase their learning ability.The following are some of the R programming exercise and Quiz.The students should solve exercise and quiz to test their basic skills in R programming.

1. **https://www.w3schools.com/r/r_exercises.asp**
2. **https://www.w3schools.com/r/r_quiz.asp**

Conclusion:

In order to execute identical lines of code numerous times in a program, a programmer can simply use a loop. By this experiment student should understand basic concept of R loops and student will able to apply R loops in real time examples.