Experiment 4

Title: Learn the basic of functions in R and implement with examples.

Objective: The objective of this practical is to understand how to define, use, and implement basic functions in R. We will learn how functions take input, process it, and return output, making our code more efficient and reusable.

Theory:

A set of statements which are organized together to perform a specific task is known as a function. R provides a series of in-built functions, and it allows the user to create their own functions. Functions are used to perform tasks in the modular approach.

Functions are used to avoid repeating the same task and to reduce complexity. To understand and maintain our code, we logically break it into smaller parts using the function. A function should be

1. Written to carry out a specified task.
2. May or may not have arguments
3. Contain a body in which our code is written.
4. May or may not return one or more output values.

"An R function is created by using the keyword function." There is the following syntax of R function:

```
func_name <- function(arg_1, arg_2, ...) {
  Function body
}
```

Components of Functions

There are four components of function, which are as follows:

Function Name

The function name is the actual name of the function. In R, the function is stored as an object with its name.

Arguments

In R, an argument is a placeholder. In function, arguments are optional means a function may or may not contain arguments, and these arguments can have default values also. We pass a value to the argument when a function is invoked.

Function Body

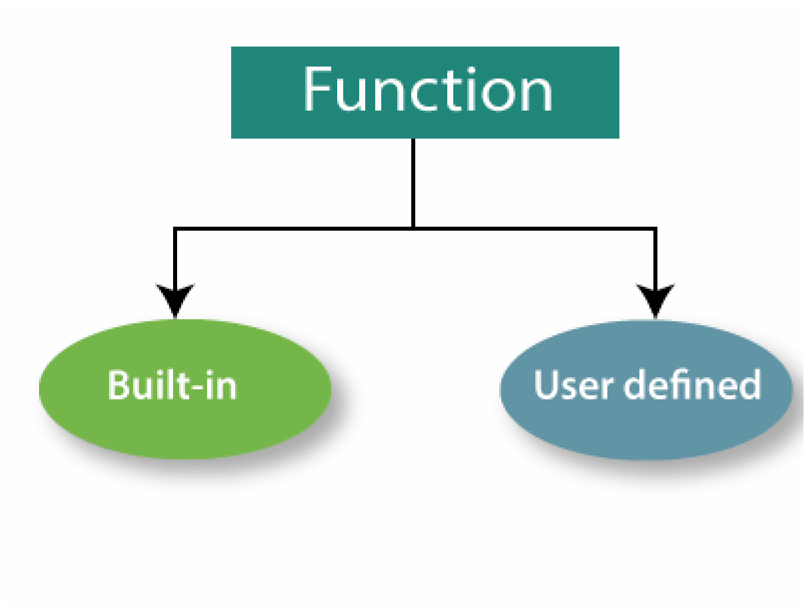
The function body contains a set of statements which defines what the function does.

Return value

It is the last expression in the function body which is to be evaluated.

Function Types

Similar to the other languages, R also has two types of function, i.e. **Built-in Function** and **User-defined Function**. In R, there are lots of built-in functions which we can directly call in the program without defining them. R also allows us to create our own functions.



Built-in function

The functions which are already created or defined in the programming framework are known as built-in functions. User doesn't need to create these types of functions, and these functions are built into an application. End-users can access these functions by simply calling it. R have different types of built-in functions such as seq(), mean(), max(), and sum(x) etc.

User-defined function
R allows us to create our own function in our program. A user defines a user-define function to fulfill the requirement of user. Once these functions are created, we can use these functions like in-built function.

Parameters or Arguments in R Functions:

Parameters and arguments are same term in functions.
Parameters or arguments are the values passed into a function.
A function can have any number of arguments, they are separated by comma in paranthesis.

Example:

```
# function to add 2 numbers

add_num <- function(a,b)

{

  sum_result <- a+b

  return(sum_result)

  }

# calling add_num function

 sum = add_num(35,34)

#printing result

print(sum)
```

No. of Parameters:

Function should be called with right no. of parameters, neither less nor more or else it will give error.

Default Value of Parameter:

Some functions have default values, and we can also give default value in our user-defined functions.
These values are used by functions if user doesn't pass any parameter value while calling a function.

Calling a Function in R:

After creating a Function, we have to call the function to use it.
Calling a function in R is very easy, we can call a function by writing it's name and passing possible parameters value.

Passing Arguments to Functions in R Programming Language
There are several ways we can pass the arguments to the function:

- **Case 1**: Generally in R, the arguments are passed to the function in the same order as in the function definition.
- **Case 2**: If we do not want to follow any order what we can do is we can pass the arguments using the names of the arguments in any order.
- **Case 3**: If the arguments are not passed the default values are used to execute the function.

Recursive Function in R:

Recursion is when the function calls itself. This forms a loop, where every time the function is called, it calls itself again and again and this technique is known as recursion. Since the loops increase the memory we use the recursion. The recursive function uses the concept of recursion to perform iterative tasks they call themselves, again and again, which acts as a loop. These kinds of functions need a stopping condition so that they can stop looping continuously. Recursive functions call themselves. They break down the problem into smaller components. The function() calls itself within the original function() on each of the smaller components. After this, the results will be put together to solve the original problem.

Problem Statement:

Note: The students should perform for following statements using functions and recursive functions in R.

1. A function that returns both the mean and standard deviation of a vector.
2. A function with arguments and without arguments.
3. Calculate reverse of given number using functions.
4. Calculate Fibinocci Series using Recursion function in R
5. Calculate factorial of given number using recursion in R.

Outcome:

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

Online Reference Websites:

- https://www.geeksforgeeks.org/functions-in-r-programming/
- https://www.w3schools.com/r/r_functions.asp
- **https://www.javatpoint.com/r-built-in-functions**

Viva Questions:

1.	How to create a user-defined function in R?
2.	What is the difference between the subset() and sample() functions n R?
3.	What is the use of the switch() function in R?
4.	

	What is the difference between the functions apply(), lapply(), sapply(), and tapply()?
5.	Write R program to find reverse of given number.

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.

- https://www.w3schools.com/r/r_quiz.asp
- https://www.w3resource.com/r-programming-exercises/#google_vignette
- https://www.evamariakiss.de/tutorial/rquiz/

Conclusion:

In R, functions help make our code more organized, reusable, and efficient. They allow us to break down tasks, handle different types of arguments, and simplify complex operations. Using functions leads to cleaner and more manageable code.