Loops in R Programming

1. Introduction to Loops

Loops are used in R to execute a block of code multiple times. They help automate repetitive tasks and process large datasets efficiently.

2. Types of Loops in R

a. for Loop

Used when the number of iterations is known.

Syntax:

```
for (variable in sequence) {
  # Code to execute
}
```

Example:

```
for (i in 1:5) {
  print(paste("Iteration:", i))
}
```

b. while Loop

Executes a block of code while a condition remains TRUE.

Syntax:

```
while (condition) {
  # Code to execute
}
```

Example:

```
x <- 1
while (x <= 5) {
  print(paste("Value of x:", x))
  x <- x + 1
}
```

c. repeat Loop

Executes a block of code indefinitely until explicitly stopped using break.

Syntax:

```
repeat {
  # Code to execute
  if (condition) {
    break
```

```
  }
}
```

Example:

```
x <- 1
repeat {
  print(paste("Value of x:", x))
  x <- x + 1
  if (x > 5) {
    break
  }
}
```

3. Controlling Loop Execution

a. break Statement

Exits a loop immediately.

Example:

```
for (i in 1:10) {
  if (i == 6) {
    break
  }
  print(i)
}
```

b. next Statement

Skips the current iteration and proceeds to the next.

Example:

```
for (i in 1:10) {
  if (i %% 2 == 0) {
    next
  }
  print(i)
}
```

4. Looping Over Data Structures

a. Looping Through a Vector

```
vec <- c(10, 20, 30, 40)
for (value in vec) {
```

```
  print(value)
}
```

b. Looping Through a List

```
lst <- list(a = 1, b = 2, c = 3)
for (item in lst) {
  print(item)
}
```

c. Looping Through a Matrix

```
mat <- matrix(1:9, nrow = 3)
for (i in 1:nrow(mat)) {
  for (j in 1:ncol(mat)) {
    print(mat[i, j])
  }
}
```

5. Practice Problems

Problem 1: Write a loop to print the first 10 natural numbers.

Problem 2: Use a while loop to calculate the sum of numbers from 1 to 100.

Problem 3: Write a for loop to print all even numbers from 1 to 50.

Problem 4: Write a loop to compute the factorial of a number (n!).

Problem 5: Iterate through a dataframe and print the names of all columns.