If/else Statements and For Loops

Boolean Expressions

Boolean expressions: expressions that are either true or false

Commands: use ==, != , > , < , >= , or <= to compare two expressions.

Example 1: Suppose you defined the following vectors:

```
s <- 5
m <- 5
```

What are the outputs if I ran the following lines of codes?

Code	Output
`s == m`	
`s < m`	
`s != m`	
`s >= m`	

Example 2: Suppose you defined the following vectors:

```
x <- c(4,5,6,7)
y <- 4:7
z <- c(4, 5, 7, 9)
```

What are the outputs if I ran the following lines of codes?

Code	Output
`x == y`	
`x == z`	
`x != z`	
`x > 6`	
`x <= 5`	

Conditional Statements

Conditional statements are an essential part of programming that allow you to control the flow of your code based on certain conditions.

In R, there are primarily three ways to implement conditional statements: "if", "if else", and the "ifelse()" function.

If Statements

The "if" statement allows you to execute a block of code only if a given condition is true.

Example 3: What is the output when you run this code?

```
sky <- "sunny"
if (sky == "sunny"){
  print("Leave your umbrella at home!")
}
```

Output:

What is the output when you change the first line to `sky <- "cloudy"`?

Output:

Example 4: Define a one element vector named *number* whose one element is 5. Write an if statement that prints "It's a positive number" if the value of vector *number* is greater than 0. What is the output once you run these lines?

Code:

Output:

What is the output when you change the first line to `a <- -5`?

Output:

If-Else Statements

The “if else” statement extends the “if” statement to execute different blocks of code based on whether a condition is true or false.

Example 5: What is the output when you run this code?

```
number <- -3

if (number > 0) {
  print("The number is positive.")
} else if (number < 0) {
  print("The number is negative.")
} else {
  print("The number is zero.")
}
```

Output:

Example 6: Define a one element vector named age whose one element is 18. Write an if-else statement about driving in Illinois. The statement should print “you are too young to drive” if the age is less than 16, “you need written consent from parents” if the age is between 16 and 17, and “no age restriction to get a license” if age is 18 or greater.

Code:

Output:

Ifelse Function

This is for when you want to use if else statements repeatedly to return a vector.

Example 7: What is the output when you run this code?

```
numbers <- c(-2, 3, 0, -5, 7)
positive_indicator <- ifelse(numbers > 0, "Positive", "Not Positive")
numbers
positive_indicator
```

Output:

Example 8: Define a vector named numbers_2 with elements 5, -2, 0. Then define a new vector named eval_5 using the ifelse function that tells you whether the numbers in numbers_2 are greater or equal to 5 or smaller than 5.

Code:

Output:

For Loops

For Loops: it's a fundamental programming construct that allows you to repeat a block of code a specified number of times. It's especially useful when you need to perform the same operation multiple times or iterate over a sequence of elements, such as elements in a list, vector, or data frame.

The basic structure of a for loop looks like this:

```
for (variable in sequence) {
   code to be executed in each iteration
}
```

Here's a breakdown of the components:

- variable: This is a variable that takes on each value in the sequence during each iteration of the loop.
- sequence: This defines the range or collection of values that the loop will iterate over. It could be a sequence of numbers (1:10), a vector, a list, or any other iterable data structure.
- The code block within the curly braces {} contains the operations you want to perform in each iteration of the loop.

Example 9: What is the output when you run this code?

```
for (i in 1:5){
  print(i)
}
```

Output:

Example 10: What is the output when you run this code?

```
for (i in c(2, 3, 5)){
  print(i+1)
}
```

Output:

Example 11: What is the output when you run this code?

```
for (i in c("Mike", "Mary", "Tom")){
  print(paste(i, "works at Loyola", sep = " - "))
}
```

Output:

Example 12: Create a character vector called student_names that contains the elements “Rohan”, “Ana”, and “Amir”. The create a for loop to print the following information:

Student 1 is Rohan

Student 2 is Ana

Student 3 is Amir

Code:

Output:

Example 13: Doing computations with for loops. What is the output when you run this code?

```
x <- matrix(c(9, 0, 3, 17, 5, 2), ncol = 3, byrow = FALSE)
y<-NA
for(i in 1:3){
  y[i] <- mean(x[,i])
}
y
```

Output: