BIO CODE

CODE OF LIFE

LEARN PYTHON & R FOR BIOINFORMATICS

Prerequisite Terminologies:

In order to have better understanding of the main topic, you should have the basic concept of the following terms:

- ➢ **Introduction to R language**

Introduction:

Decision making is an important part of programming. This can be achieved in R programming utilizing the *'If Else'* statements. '*If Else'* statements are the conditional statements that can be utilized to return an output based on a condition. It allows you to run logical conditions, statements or anything that is correlated with the logic.

- ❑ **If Statement:**

  The syntax of if statement is:

```
If (expression) {
Statement
}
```

Here how this works, if the expression provided within the brackets is TRUE, the statement within curly brackets will be executed. But if it's FALSE, nothing will happen. Here the expression can be a logical or numeric vector.

- ❑ **If Else Statement:**

  The syntax of *'if else'* statement is:

```
If (expression) {
Statement1
} else {
Statement2
}
```

The else part is optional that particularly provides an extension to the *if else* statement and is only executed when the expression provided after the '*if*' keyword is FALSE.

Steps:

- For the understanding of *if else* statements, we'll use an example to retrieve and read a particular file from the working directory. You can utilize *if else* statement as:

```
getwd()
if (file.exists("filename")) {
   myData <- read ("filename")
   View (myData)
} else {
  Print ("No such file, please create it.")
}
```

[If the file exists in the working directory then the *if* condition will TRUE, so the *if* statement will be executed and it'll print out that particular file.

But if the file does not exist, the *if* condition will be FALSE then the *else* statement will be executed and it'll print out "No such file, please create it".**]**

Summary:

In this video, we learned about the *if else* statements and how these conditional statements can be utilized in R programming.