R ifelse()

Here, we are going to have a look at the method "ifelse()" provided by R language.

We'll cover the following

Most of the programs written in R language involve some kind of manipulation of **vectors**. Vectors are basic R Objects. We have already covered vectors in the previous chapter.

Use of `ifelse()` Function

Now, imagine a case where we have to write `if...else` statement for each element present in a vector. This task can be easily and efficiently coded using `ifelse()`. It is the vector equivalent form of the `if…else` statement in R.

Syntax

```
ifelse(expression, condition1, condition2)

# Here "expression" is an object which can be coerced to logical mode.
# Condition1 is returned for elements that satisfy the expression (return TRU
E)
# Condition2 is returned for elements that do not satisfy the expression (retu
rn FALSE)
```

> The output of `ifelse()` function is always a vector! Each element of the output vector tells whether the test (expression) passed for that element of the input vector.

Let’s revisit our original positive and negative numbers problem using vectors this time.

```
x <- c(5, -5)
ifelse(x > 0, "positive", "negative") # if the expression is satisfied return condition1 else retu
```

Using `ifelse()`

Notice that the size of the output vector is the same as the input vector. Also, remember that the `expression` parameter must return a logical vector. In our code, $x > 0$ returns a logical vector `[TRUE, FALSE]`. The `TRUE` value corresponds to the first condition and `FALSE` to the second condition.

In the next lesson, we have a small exercise for you to test your ability to use `ifelse()`.