R ifelse() Function (With Example)

R ifelse() Function (With Example)

In this article, you'll learn about ifelse() function. This is a shorthand function to the traditional if...else statement.

Vectors form the basic building block of R programming.

Most of the functions in R take vector as input and output a resultant vector.

This vectorization of code, will be much faster than applying the same function to each element of the vector individually.

Similar to this concept, there is a vector equivalent form of the if...else statement in R, the `ifelse()` function.

Syntax of ifelse() function

```
ifelse(test_expression, x, y)
```

Here, `test_expression` must be a logical vector (or an object that can be coerced to logical). The return value is a vector with the same length as `test_expression`.

This returned vector has element from `x` if the corresponding value of `test_expression` is `TRUE` or from `y` if the corresponding value of `test_expression` is `FALSE`.

This is to say, the `i-th` element of result will be `x[i]` if `test_expression[i]` is `TRUE` else it will take the value of `y[i]`.

The vectors `x` and `y` are recycled whenever necessary.

Example: ifelse() function

```
> a = c(5,7,2,9)
> ifelse(a %% 2 == 0,"even","odd")
[1] "odd"  "odd"  "even" "odd"
```

In the above example, the `test_expression` is `a %% 2 == 0` which will result into the vector `(FALSE,FALSE,TRUE ,FALSE)` .

Similarly, the other two vectors in the function argument gets recycled to `("even","even","even","even")` and `("odd","odd","odd","odd")` respectively.

And hence the result is evaluated accordingly.