Lecture 5 Module 5: Conditional Branching

Exercises

```
load( "Module 5 R Objects.Rdata" )
```

Exercise

Exercise 1: Vectorized conditional branching

The R object `exercise.1.data` is a numeric vector.

Create a new character string vector:

- If an element of `exercise.1.data` is less than 0, then the corresponding element in the new vector should be the character string "Cold".
- If an element of `exercise.1.data` is greater than or equal to 0, then the corresponding element in the new vector should be the character string "Hot".

Directly display the vector `exercise.1.data`, and try to determine what the new vector will be, based on this data.

Then write R code to create the new vector.

Ideally, you should do this using two approaches:

- Using vectorized operations with `ifelse()`.
- Using a `for()` loop and an `if()` statement.

Which method do you prefer?

Solution

Solution to the Exercise

Exercise 1: Vectorized conditional branching

The R object `exercise.1.data` is a numeric vector.

Create a new character string vector:

- If an element of `exercise.1.data` is less than 0, then the corresponding element in the new vector should be the character string "Cold".

- If an element of `exercise.1.data` is greater than or equal to 0, then the corresponding element in the new vector should be the character string “Hot”.

Directly display the vector `exercise.1.data`, and try to determine what the new vector will be, based on this data.

Then write R code to create the new vector.

Ideally, you should do this using two approaches:

- Using vectorized operations with `ifelse()`.
- Using a `for()` loop and an `if()` statement.

Which method do you prefer?

Solution

First, let’s directly display the vector:

```
exercise.1.data
```

```
## [1]  9  5 -8  2  7 -3  4 -1
```

In `exercise.1.data`, positive numbers occur in positions 1, 2, 4, 5, and 7, while negative numbers occur in positions 3, 6, and 8.

Thus, the new vector should have the character string “Hot” in positions 1, 2, 4, 5, and 7, and the character string “Cold” in positions 3, 6, and 8.

We can do this with `ifelse()`:

```
new.vector <-
  ifelse(
    exercise.1.data >= 0,
    "Hot",
    "Cold"
  )

new.vector
```

```
## [1] "Hot"  "Hot"  "Cold" "Hot"  "Hot"  "Cold" "Hot"  "Cold"
```

This is exactly what we had predicted.

To do this using a `for()` and an `if()` statement is a little more work:

```
number.of.elements <-
  length( exercise.1.data )

new.vector <-
  character( number.of.elements )

for( i in 1:number.of.elements ) {

  if( exercise.1.data[ i ] >= 0 ) {
    new.vector[ i ] <- "Hot"
  } else {
    new.vector[ i ] <- "Cold"
  }
}

new.vector
```

```
## [1] "Hot"  "Hot"  "Cold" "Hot"  "Hot"  "Cold" "Hot"  "Cold"
```

This is exactly the same result as before, but it was a little bit more work to get there using a `for()` loop and an `if()` statement, compared to doing this with an `ifelse()` statement.