More Complex Vector Indexing & Manipulations

Sam Mason

Introduction

This assignment will first introduce the logical object class along with boolean operators. We'll use these two tools to filter and transform vectors of data prior to making inference. The data you'll be working with are a collection of vectors describing the amount of time (in seconds) that individual butterflies spent exhibiting certain behaviors. These behaviors include perching with wings closed (`pwc`), perching with wings open (`pwo`), flying (`flight`), and nectaring (`nectar`). This is all real data collected by me in Virginia during the summer of 2019.

Logical Objects & Boolean Operators

```
var1 <- TRUE
var2 <- FALSE
class(var1)
```

```
## [1] "logical"
```

```
class(var2)
```

```
## [1] "logical"
```

Logical objects are those that take on the values `TRUE` or `FALSE`. In and of themselves, logical objects are not particularly useful (or at least I don't use them very frequently when modeling). The logical construct is, however, very useful to us when indexing vectors using Boolean operators.

```
1 == 1 # Is 1 equal to 1?
```

```
## [1] TRUE
```

```
1 != 2 # Is 1 not equal to 2?
```

```
## [1] TRUE
```

```
1 > 2 # Is 1 greater than 2?
```

```
## [1] FALSE
```

```r
5 < 10 # Is 5 less than 10?
```

```
## [1] TRUE
```

```r
2 >= 2 # Is 2 greater than or equal to 2?
```

```
## [1] TRUE
```

```r
2 <= 4 # Is 2 less than or equal to 4?
```

```
## [1] TRUE
```

Boolean operators ask true-or-false questions, and R returns the solutions.

Indexing Vectors with Boolean Operators

Task 1

Create a vector of 100 values randomly drawn from a normal distribution with a mean of 5 and a standard deviation of 1. Assign this vector to a sensibly named object (from here on out object creation and naming is up to you).

Task 2

Just like arithmetic operators are applied to each value of a vector, Boolean operators ask the true-or-false question of every value. Ask R whether or not each value of the vector created in **Task 1** is greater than or equal to 5. You'll want to save the consequent logical output to a new object.

Task 3

We talked about indexing a vector using a vector of indices to select only certain values. What happens when you run the following code (`task1` is the vector you created in **Task 1** and `task2` is the vector your created in **Task 2**?

```r
task1[task2]
```

Logical objects acts like light switches when indexing vectors. A value of `TRUE` flips the light on (keeps the corresponding value from `task1`), and a value of `FALSE` flips the light off (omits the corresponding value from `task1`).

Task 4

Create a new vector containing only those values from `task1` that are less than the first quartile of the vector.

Task 5

Create a new vector containing only those values from `task1` that are greater than or equal to the 95th quantile of the vector. You may need to look up the documentation for the `quantile()` function. How long is the resulting vector? `?length()` should help.

Cleaning Up & Working with Butterfly Behavior Data

Task 6

Clear your environment and load in the "butterfly behaviors.RData" file. Notice how `pwo` is slightly longer than any of the other vector. The last 3 observations were included erroneously. Create a new object that omits them.

Task 7

A couple of the vectors include NA data where there should be zeros (i.e., the butterfly spent no time exhibiting that particular behavior). Convert those NA values to zeros. You'll need to use `NA` in your Boolean statement.

Task 8

Create a vector that includes the proportion of time that each of the 71 butterflies spent nectaring. That is, the time spent nectaring divided by the total time spent exhibiting any of the four behaviors.

Task 9

We can generally group these four behaviors into two categories, active behaviors and inactive behaviors. When a butterfly is perching (regardless of wing orientation), it is inactive. When it is flying or nectaring it is active. Create two new vectors, one that contains the total active time for each butterfly, and another that includes the total inactive time. Are these butterflies generally more active, or more inactive? How could you determine this?

Task 10

Select the 5 most inactive butterflies. How could you do this? Perhaps take a peek at the `order()` function.

Task 11

Select all the butterflies that spent some amount of time nectaring. When butterflies are nectaring, do they tend to spend much time perching?