Class 6: R Functions

Tyler (A16651122)

2024-01-25

R Functions

Functions are how we get stuff done. We call functions to do everything useful in R.

One cool thing about R is that it makes writing your own functions comparatively easy.

All functions in R have at least three things:

- A **name** (we get to pick this)
- One or more **input arguments** (the input to our function)
- The **body** (lines of code that do the work)

Can use #| eval: false to make an entire coding block not run

```
funname <- function(input1, input2) {
  The body with R code
}
```

Let's write a silly first function to add two numbers:

```
x <- 5
y <- 1
x + y
```

```
[1] 6
```

```
addme <- function(x, y=1) {
  x + y
}
```

```
addme(1,1)
```

```
[1] 2
```

```
addme(10)
```

```
[1] 11
```

Lab for today

Write a function to grade student work from class. Start with a simplified version of the problem:

```
# Example input vectors to start with
student1 <- c(100, 100, 100, 100, 100, 100, 100, 90)
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
```

Come back to this NA problem. But things worked for `student1`.

We want to drop the lowest score before getting the `mean()`.

How do I find the lowest (minimum) score?

```
min(student1)
```

```
[1] 90
```

Found which.min().

```
which.min(student1)
```

```
[1] 8
```

Cool - it is the 8th element of the vector that has the lowest score. Can I remove this one?

```
# Find the lowest score
ind <- which.min(student1)
# remove lowest score and find the mean
```

```
  mean(student1[-ind])
```

```
[1] 100
```

```
  mean(student1[-which.min(student1)])
```

```
[1] 100
```

Use a common shortcut and use `y` as my input

```
  y <- student2
  mean(y[-which.min(y)])
```

```
[1] NA
```

```
  is.na(student3)
```

```
[1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
  sum(is.na(student3))
```

```
[1] 7
```

How can I remove the NA elements from the vector?

```
  z <- student1

  z[is.na(z)] <- 0
  mean( z[-which.min(z)])
```

```
[1] 100
```

Last step now that I have my working code snippets is to make my `grade()` function.

```r
grade <- function(x) {
  x[is.na(x)] <- 0
  mean(x[-which.min(x)], na.rm = TRUE)
}
```

```r
grade(student1)
```

```
[1] 100
```

```r
grade(student2)
```

```
[1] 91
```

```r
grade(student3)
```

```
[1] 12.85714
```

`apply()` takes any function and applies it over a dataset.

- x = array
- margin = 1 (rows) or 2 (columns)
- fun = function

Read the online gradebook (CSV file)

```r
gradebook <- read.csv("https://tinyurl.com/gradeinput", row.names = 1)

head(gradebook)
```

```
          hw1 hw2 hw3 hw4 hw5
student-1 100  73 100  88  79
student-2  85  64  78  89  78
student-3  83  69  77 100  77
student-4  88  NA  73 100  76
student-5  88 100  75  86  79
student-6  89  78 100  89  77
```

Lab Questions

Q1

```
grade <- function(x) {
  # Sets all NA values equal to 0
  x[is.na(x)] <- 0
  # Removes the lowest value score, which includes 0, and averages the scores
  mean(x[-which.min(x)], na.rm = TRUE)
}
```

```
# uses grade function on gradebook dataset, but only on the rows
results <- apply(gradebook, 1, grade)
results
```

```
 student-1  student-2  student-3  student-4  student-5  student-6  student-7
     91.75      82.50      84.25      84.25      88.25      89.00      94.00
 student-8  student-9 student-10 student-11 student-12 student-13 student-14
     93.75      87.75      79.00      86.00      91.75      92.25      87.75
student-15 student-16 student-17 student-18 student-19 student-20
     78.75      89.50      88.00      94.50      82.75      82.75
```

Q2

```
max(results)
```

```
[1] 94.5
```

```
which.max(results)
```

```
student-18
        18
```

The top student is student-18.

Q3

```
# uses mean function on gradebook dataset but only on columns (hw assignments)
hw <- apply(gradebook, 2, mean, na.rm = TRUE)
```

```
  hw
```

```
     hw1      hw2      hw3      hw4      hw5
89.00000 80.88889 80.80000 89.63158 83.42105
```

```
  min(hw)
```

```
[1] 80.8
```

```
  which.min(hw)
```

```
hw3
  3
```

Homework 3 was the toughest.

Q4

```
  # make all (or mask) NA equal 0
  mask <- gradebook
  mask[is.na(mask)] <- 0
  #mask
```

We can use the `cor()` function for correlation analysis.

```
  # correlation for grade (after being masked) and each homework assignment
  cor(mask$hw1, results)
```

```
[1] 0.4250204
```

```
  cor(mask$hw2, results)
```

```
[1] 0.176778
```

```
  cor(mask$hw3, results)
```

```
[1] 0.3042561
```

```
cor(mask$hw4, results)
```

```
[1] 0.3810884
```

```
cor(mask$hw5, results)
```

```
[1] 0.6325982
```

Need to use `apply()` function to run analysis over the whole masked gradebook.

```
apply(mask, 2, cor, results)
```

```
      hw1       hw2       hw3       hw4       hw5
0.4250204 0.1767780 0.3042561 0.3810884 0.6325982
```