class06 R functions

amy (pid A16962111)

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 **body** (lines of code that do the work)

```
funname <- function() {
  # 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(100,100)
```

```
[1] 200
```

```r
addme(10)
```

```
[1] 11
```

lab for today

```r
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)
```

```r
mean(student1)
```

```
[1] 98.75
```

```r
mean(student2, na.rm = TRUE)
```

```
[1] 91
```

```r
mean(student3, na.rm = TRUE)
```

```
[1] 90
```

This is not fair - there is no way that student 3 should have a mean of 90.

We also want to let students drop their lowest grade.

How do I remove the lowest score?

```r
min(student1)
```

```
[1] 90
```

I found the `which.min` function. Maybe this is more useful?

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

```
[1] 8
```

Cool - the eighth element of the vector has the lowest score. Can I remove this one?

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

```
[1] 100
```

We still have the problem of missing values.

One idea is to replace NA values with zero.

```
y <- c(1, 2, 3, 4, 5)
y[y==3] <- 0
y
```

```
[1] 1 2 0 4 5
```

```
x <- student2

# change NA values to zero
x[is.na(x)] <- 0
# find and remove lowest score, and find the mean
mean(x[-which.min(x)])
```

```
[1] 91
```

Last step is take make the `grade()` function.

```
grade <- function(x) {
  # change NA values to zero
  x[is.na(x)] <- 0
  # find and remove lowest score, and find the mean
  mean(x[-which.min(x)])
}
```

```r
grade(student3)
```

```
[1] 12.85714
```

Now read the online gradebook.

```r
url <- "https://tinyurl.com/gradeinput"
gradebook <- read.csv(url, 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
```

Grade the students.

```r
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 
```

Who is the top-scoring student?

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

```
student-18 
        18 
```

Which homework was the hardest?

```
which.min(apply(gradebook, 2, sum, na.rm=T))
```

```
hw2
  2
```

Which homework was most predictive of overall score?

```
#mask NAs as zeros
gradebook[is.na(gradebook)] <- 0
```

We can use the `cor()` function for correlation analysis, and use `apply()` to run the analysis over the whole course.

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

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

```
which.max(apply(gradebook, 2, cor, results))
```

```
hw5
  5
```