Class 6: R Functions

Renny Ng (PID: A98061553)

2024-01-26

#Our first simple silly function

All functions in R have 3 parts. They have:

- A name
- Input arguments (none, one, or more)
- Body

A function to add two numbers

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

Let's try out this function (after executing the code chunk and seeing it under "Functions" in Environment)

```
sillyadd(10)
```

```
[1] 11
```

```
sillyadd(100,100)
```

```
[1] 200
```

```
sillyadd(100,)
```

```
[1] 101
```

To get the code from any function: just type out the function without parentheses.

#Let’s do something more useful.

Write a function grade() to determine an overall grade from a vector of student homework assignment scores, dropping the lowest single assignment score.

Loaded in input vectors from the Class 6 R Functions Lab

```r
# 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)
```

Strategy:

1. Sort Student score, so that first value is the lowest value.
2. Set NA to 0
3. Take the mean from the 2nd to 8th scores.

Trying out the function writing now:

```r
sort(student1)
```

```
[1]  90 100 100 100 100 100 100 100
```

I will begin by getting a skateboard solutio nto my tesla problem.

```r
mean(student1)
```

```
[1] 98.75
```

```r
min(student1)
```

```
[1] 90
```

```r
student1
```

```
[1] 100 100 100 100 100 100 100  90
```

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

```
[1] 8
```

```
student1[-8]
```

```
[1] 100 100 100 100 100 100 100
```

mean(student1-which.min(student1))

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

```
[1] 90.75
```

```
#still not correct
```

```
student1[-8]
```

```
[1] 100 100 100 100 100 100 100
```

```
student1_minus_lowest <- student1[-8]
```

```
mean(student1_minus_lowest)
```

```
[1] 100
```

```
x <- student1
#Find lowest value
ind <- which.min(x)
#Exclude lowest value and find mean
mean(x[-ind])
```

```
[1] 100
```

```
x <- student2
#Find lowest value
ind <- which.min(x)
ind
```

```
[1] 8
```

```
#Exclude lowest value and find mean
mean(x[-ind],na.rm=T)
```

```
[1] 92.83333
```

```
x <- student3
#Find lowest value
x
```

```
[1] 90 NA NA NA NA NA NA NA
```

```
ind <- which.min(x)
ind
```

```
[1] 1
```

```
#Exclude lowest value and find mean
mean(x[-ind],na.rm=T)
```

```
[1] NaN
```

Find and replace NA value with 0 is.na function designates value NA with any different value. == asks for whether something is true

```
x <- 1:5
x
```

```
[1] 1 2 3 4 5
```

```r
x[x==3] <-10000
x
```

```
[1]     1     2 10000     4     5
```

```r
x <- student3
x
```

```
[1] 90 NA NA NA NA NA NA NA
```

```r
x[is.na(x)] <- 0
x
```

```
[1] 90  0  0  0  0  0  0  0
```

```r
x
```

```
[1] 90  0  0  0  0  0  0  0
```

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

```
[1] 12.85714
```

fname <- function(arg1, arg2) {paste(arg1, arg2)}

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

```r
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

#Question 1: Read a class gradebook CSV file from here: “https://tinyurl.com/gradeinput”

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

Reassigning names to rows argument: row.names=#

Now use our `grade()`function to grade the whole class. But currently it will not work, because our format is different, because we are working with data frame (not a one-dimensional vector).

We need can loop the function: for each student, loop and save. However, there is something easier: the `apply` function.

- apply(X, MARGIN, FUN, …, simplify = TRUE)
- In our case: X = gradebook, FUN = grade()
- What is MARGIN?

MARGIN dictates whether the function is applied to rows or columns.

- Rows = 1
- Columns = 2

We can “apply” our new `grades()` function over rows or columns of the gradebook.

```
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
```

#Question 2: Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook?

```
sorted_results <- sort(results)
sorted_results
```

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

Student-18 appears to be the highest-scoring.

```
which_max <- which.max(results)
which_max
```

```
student-18 
        18 
```

Using `which.max` confirms that student-18 is doing the best. #Question 3: From your analysis of the gradebook, which homework was toughest on students (i.e. obtained the lowest scores overall?

```
averaged_assignments <- apply(gradebook, 2, mean)
averaged_assignments
```

```
 hw1  hw2  hw3  hw4  hw5 
89.0   NA 80.8   NA   NA 
```

lowest_score <- function(x) { x[is.na(x)] <- 0}

```
lowest_score <- function(x) {
  x[is.na(x)] <- 0}
```

```
 apply(gradebook, 2, mean, na.rm=T)
```

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

```
#na.rm removes NA before applying
```

It appears that homework 3 was the toughest assignment.

#Question 4: From your analysis of the gradebook, which homework was most predictive of overall score (i.e. highest correlation with average grade score)

Consider the correlation between individual assignment scores with overall performance.

Look at the individual columns in the gradebook, with the results of grades. We want to see higher correlation scores for each assignment.

Use Pearson correlation (0 means no correlation, 1 is perfect correlation). Using the `cor` function.

```
gradebook$hw5
```

```
 [1]  79  78  77  76  79  77 100 100  77  76 100 100  80  76  NA  77  78 100  79
[20]  76
```

```
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 
```

```
#Are the trends correlated?
cor(x=gradebook$hw5, y=results)
```

```
[1] NA
```

```
#This yields an error, because there are still missing homeworks (NA values).
#We need to mask NA values to 0.
```

```
mask <- gradebook
mask[is.na(mask)] <- 0
mask
```

```
           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   0  73 100  76
student-5   88 100  75  86  79
student-6   89  78 100  89  77
student-7   89 100  74  87 100
student-8   89 100  76  86 100
student-9   86 100  77  88  77
student-10  89  72  79   0  76
student-11  82  66  78  84 100
student-12 100  70  75  92 100
student-13  89 100  76 100  80
student-14  85 100  77  89  76
student-15  85  65  76  89   0
student-16  92 100  74  89  77
student-17  88  63 100  86  78
student-18  91   0 100  87 100
student-19  91  68  75  86  79
student-20  91  68  76  88  76
```

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

```
[1] 0.6325982
```

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

```
[1] 0.6325982
```

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

```
[1] 0.3042561
```

```r
apply(mask, 2, cor, y=results)
```

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

It appears that homework 2 is the least predictive of overall student success.