Lab 6 R Functions

Thoi Tran (A17035545)

Today we are going to explore R functions and begin to think about writting our own functions.

Let's start simple and write out first function to add some numbers.

Every function in R has at least 3 things:

- a **name**, we pick this
- one or more input **arguments**
- a **body**, where the work gets done

```
add <-function(x, y) {
  x + y
}
```

Now let's try it out.

```
add(1, 1)
```

```
[1] 2
```

```
add(10,1)
```

```
[1] 11
```

```
add(c(10,1,1,10), 1)
```

```
[1] 11  2  2 11
```

Lab Exercise

Import the gradebook

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

> Q1. Write a function grade() to determine an overall grade from a vector of student homework assignment scores dropping the lowest single score. If a student misses a homework (i.e. has an NA value) this can be used as a score to be potentially dropped. Your final function should be adquately explained with code comments and be able to work on an example class gradebook such as this one in CSV format: "https://tinyurl.com/gradeinput"

Let's try it out with a simple dataset

```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, na.rm = T)
```

```
[1] 98.75
```

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

```
[1] 91
```

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

```
[1] 90
```

We also want to drop the the lowest score from a given students set of scores.

```r
student1[-8]
```

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

We can try the `min()` function to find the lowest score

```r
min(student1)
```

```
[1] 90
```

I want to find the location of the min value not the value itself. For this I can use `which.min()`.

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

```
[1] 8
```

Let's put these two things together.

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

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

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

```
[1] 100
```

Next, let's fix the NA problem. We can find the NAs and replace them with 0.

```r
## Find NAs in 'x' and make them 0
student2[is.na(student2)] <- 0
student2
```

```
[1] 100   0  90  90  90  90  97  80
```

```r
## Find the min value and remove it before getting mean
student2[-which.min(student2)]
```

```
[1] 100  90  90  90  90  97  80
```

```r
mean(student2[-which.min(student2)])
```

```
[1] 91
```

Let's try it with student3. So far we have a working snippet.

```r
student3[is.na(student3)] <- 0
student3
```

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

```r
student3[-which.min(student3)]
```

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

```r
mean(student3[-which.min(student3)])
```

```
[1] 12.85714
```

Now turn it into a function.

```r
grade <- function(x){
  ## Find NAs in 'x' and make them 0
  x[is.na(x)] <- 0

  ## Drop lowest score and find mean
  mean(x[-which.min(x)])
}
```

Apply the `grade()` function for student1, student2, and student3.

```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)
grade(student1)
```

```
[1] 100
```

```r
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Now `apply()` to our class gradebook. To use the `apply()` function on the `gradebook` dataset I need to decide whether I want to "apply" the `grade()` function over the rows (1) or columns (2) of the `gradebook`.

```
q1_ans <- apply(gradebook, 1, grade)
q1_ans
```

```
 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. Using your `grade()` function and the supplied `gradebook`, Who is the top scoring student overall in the gradebook?

```
q2_ans <- which.max(q1_ans)
q2_ans
```

```
student-18 
        18 
```

```
q1_ans[q2_ans]
```

```
student-18 
      94.5 
```

student-18 has the highest score in the gradebook, which is 94.5

> Q3. From your analysis of the gradebook, which homework was toughest on students (i.e. obtained the lowest scores overall?

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

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

```r
masked_gradebook <- gradebook
masked_gradebook[is.na(masked_gradebook)] <- 0
q3_ans <- apply(masked_gradebook, 2, mean)
q3_ans
```

```
  hw1   hw2   hw3   hw4   hw5 
89.00 72.80 80.80 85.15 79.25 
```

```r
which.min(q3_ans)
```

```
hw2 
  2 
```

```r
q3_ans[which.min(q3_ans)]
```

```
 hw2 
72.8 
```

hw2 was toughest and had the lowest score overall, 72.8.

I could modify the `grade()` function to do this too - not drop the lowest options

```r
grade2 <- function(x, drop.low = TRUE) {
  x[is.na(x)] <- 0

  if(drop.low) {
    out <- mean(x[-which.min(x)])
  }
  else {
    out <- mean(x)
  }
  return(out)
}
```

```
apply(gradebook, 2, grade2, FALSE)
```

```
  hw1   hw2   hw3   hw4   hw5 
89.00 72.80 80.80 85.15 79.25 
```

> Q4. From your analysis of the gradebook, which homework was most predictive of overall score (i.e. highest correlation with average grade score)?

The function to calculate correlation in R is called cor()

```
x <- c(100, 90, 80, 100)
y <- c(100, 90, 80, 100)
z <- c(80, 90, 100, 10)

cor(x,y)
```

```
[1] 1
```

```
cor(x,z)
```

```
[1] -0.6822423
```

Now apply() the cor() function over the masked_gradebook and use the q1_ans scores for the class.

```
q4_ans <- apply(masked_gradebook, 2, cor, q1_ans)
q4_ans
```

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

hw5 had the highest correlation with the average grade score.