Class 6: R functions

Saba Heydari Seradj (PID: A17002175)

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

Let's start simple and write our 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**
- the **body**, where the work gets done

Writing a simple 'add' function. Setting y as zero for default.

```
add <- function(x,y=0){
  x+y
}
```

Now let's try it out

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

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

```
add(10)
```

```
[1] 10
```

Lab sheet work

> **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" [**3pts**]

```
# 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)
```

Begin by calculating the average for student1.

```
student1
```

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

```
mean(student1)
```

```
[1] 98.75
```

But, we want to stop the lowest score from a given student's set of scores.

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

```
[1] 100
```

This code will set the NA values to 0. Then we can drop the lowest value.

```
student2
```

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

```
# make an alias so we don't override the original vector.
student2_clean <- student2
# find NAs in 'x' and make them 0
student2_clean[is.na(student2_clean)] <- 0
student2_clean
```

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

```
# finds the min value and removes it before getting mean
mean(student2_clean)
```

```
[1] 79.625
```

```
mean(student2_clean[-which.min(student2_clean)])
```

```
[1] 91
```

So far we have a working snippet,

```
student3
```

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

```
# make an alias so we don't override the original vector.
x <- student3
# find NAs in 'x' and make them 0
x[is.na(x)] <- 0
x
```

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

```
# finds the min value and removes it before getting mean
mean(x)
```

```
[1] 11.25
```

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

```
[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
  # finds the min value and removes it before getting mean
  mean(x[-which.min(x)])
}
```

```r
grade(student1)
```

```
[1] 100
```

```r
grade(student2)
```

```
[1] 91
```

```r
grade(student3)
```

```
[1] 12.85714
```

```r
gradebook <- data.frame(student1,student2,student3)
gradebook <- t(gradebook)
final_grades <- apply(gradebook, 1, grade)
final_grades
```

```
 student1  student2  student3
100.00000  91.00000  12.85714
```

I read the csv file for the 'gradebook'.

```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
```

To use the apply() function on this 'gradebook' dataset, I need to decide whether I want to apply the grade function over the rows (1) or columns (2) of the 'gradebook'. Students are rows, so I need to assign margin=1.

```
grades <- apply(gradebook, 1, grade)
head(grades)
```

```
student-1 student-2 student-3 student-4 student-5 student-6
    91.75     82.50     84.25     84.25     88.25     89.00
```

Q2. Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook? [**3pts**]

```
grades[which.max(grades)]
```

```
student-18
      94.5
```

Student 18 is the top scoring student overall in the gradebook.

Q3. From your analysis of the gradebook, which homework was toughest on students (i.e. obtained the lowest scores overall? [**2pts**

```
homework <- apply(gradebook, 2, grade)
homework
```

```
     hw1      hw2      hw3      hw4      hw5
89.36842 76.63158 81.21053 89.63158 83.42105
```

```
homework[which.min(homework)]
```

```
     hw2
76.63158
```

Using this strategy homework 2 is the most difficult.

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

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

Using this strategy homework 3 and 2 are low, with homework 3 being very slightly lower.

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

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

Using this third strategy, homework 2 is lowest.

Modifying our original function so that it takes another argument so we can choose whether or not we want to drop the lowest score.

```
grade2 <- function(x, drop.low=T){
  # find NAs in 'x' and make them 0
  x[is.na(x)] <- 0
  if(drop.low){
    cat('Hello low')
    # finds the min value and removes it before getting mean
    out <- mean(x[-which.min(x)])
  } else{
    out <- mean(x)
    cat('No low')
  }
  return(out)
}
```

```
grade2(student1,drop.low=F)
```

```
No low
```

```
[1] 98.75
```

> **Q4.** Optional Extension: From your analysis of the gradebook, which homework was most predictive of overall score (i.e. highest correlation with average grade score)? [**1pt**]

The function to calculate correlations in R is 'cor()'.

```
cor(grades, masked_gradebook$hw5)
```

```
[1] 0.6325982
```

I want to apply the cor function over the 'masked_gradebook' and use the grades for the class.

```
apply(masked_gradebook,2,cor,y=grades)
```

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

Homework 5 is most predictive of overall score of students, as it had the highest correlation with students' average score.