Class 6: R functions

Samson A16867000

Today we are going to explore R functions and begin to think about writing 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 **arguements**
- the **body**, where the work gets done.

```
add <- function(x,y=1, z=0){
  x+y+z
}
```

Now lets try it out

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

```
[1] 2
```

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

```
[1] 11
```

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

```
[1] 20
```

```
add(10,10,20)
```

```
[1] 40
```

```r
mean(c(10,10,NA),na.rm=T)
```

```
[1] 10
```

Lab sheet work

. Q1. 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]

```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)
```

Begin by calculating the average for student 1

```r
mean(student1)
```

```
[1] 98.75
```

```r
student2
```

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

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

```
[1] 91
```

and student3

```r
student3
```

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

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

[1] 90

Hmm…. this sucks! I need to try something ele and come back to this issue of missing values (Nas).

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

```
student1[-8]
```

[1] 100 100 100 100 100 100 100

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

```
min(student1)
```

[1] 90

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

```
student1
```

[1] 100 100 100 100 100 100 100 90

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

[1] 8

Let's put these two things together

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

[1] 100

```
x <- student2
x
```

[1] 100 NA 90 90 90 90 97 80

```r
x[is.na(x)]=0
x
```

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

So far we have a working snippet.

```r
x <- student1
## Finds NAs in `x` and make them 0
x[is.na(x)] <- 0

# finds the minimum and rm's it before getting mean
mean(x[-which.min(x)])
```

```
[1] 100
```

Now turn it into a function

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

  # finds the minimum and rm's 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
```

NOw `apply()` to our class 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`.

```r
ans <-apply(gradebook,1,grade)
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? [3pts]

```r
which.max(ans)
```

```
student-18
        18
```

```r
ans[which.max(ans)]
```

```
student-18
      94.5
```

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

```
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
```

I could modify the **grade()** function to do this too - i.e not drop the lowest option

```
grade2 <- function(x, drop.low = TRUE) {

  # Finds NAs in`x` and make them 0
  x[is.na(x)] <- 0

  if (drop.low) {
    cat("Hello low")
    # Drop lowest values and find mean
    out <- mean(x[-which.min(x)])
  } else {
    out <- mean(x)
    cat("No low")
  }
  return(out)
}
```

```
grade2(student1, TRUE)
```

```
Hello low
```

```
[1] 100
```

.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 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
```

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

```
[1] 0.6325982
```

I want to apply() the or() function over the masked_gradebook and use the ans scores for the class

```
apply(masked_gradebook, 2, cor, ans)
```

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