Class 6: R Functions

Kalisa Kang (PID: A16741690)

Functions are how we get work done in R. We call functions to do everything, from reading data to doing data analysis and outputting plots and results.

All functions in R have at least 3 things:

- a **name** (you get to pick this)
- input **arguments** (there can be only one or loads - again your call)
- the **body** (where the work gets done, the code between the curly brackets)

A first silly function

Let's write a funct to add some numbers. We can call it `add()`

```
x <- 10
y <- 10
x + y
```

```
[1] 20
```

```
add <- function(x) {
  y <- 10
  x + y
}
```

Can I just use my new function?

```
add(1)
```

```
[1] 11
```

Let's make it a bit more flexible.

```r
add <- function(x, y=1) {
  x + y
}
add(10, 10)
```

```
[1] 20
```

```r
add(10)
```

```
[1] 11
```

```r
add(10, 100)
```

```
[1] 110
```

2nd example grade() function

Write a function to grade student work.

We will start with a simple version of the problem and the following example student vectors:

```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)
```

Start with student1

```r
mean(student1)
```

```
[1] 98.75
```

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

```
[1] 91
```

`na.rm()` na stands for not available (NA) and rm stands for remove. Equating this to TRUE means that you want to remove the NAs.

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

```
[1] 90
```

Let's try to work with student1 and find (and drop) their lowest score.

```r
student1
```

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

Google told me about min() and max().

```r
min(student1)
```

```
[1] 90
```

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

```
[1] 8
```

```r
student1[8]
```

```
[1] 90
```

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

```
[1] 90
```

Minus sign removes the specified value. `which.min()` locates the bin/position of the lowest value

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

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

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

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

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

```
[1] 100
```

Our first working snippet that drops the lowest score and calculates the mean.

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

```
[1] 100
```

```
x <- student3
mean(x[ -which.min(x) ])
```

```
[1] NA
```

Our approach to the NA problem (missing homeworks): we can replace all NA values with zero.

1st task is to find the NA values (i.e., where they are in the vector)

```
x <- student2
is.na(x)
```

```
[1] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
```

I found the NA (TRUE) values from `is.na()` now I want to make them equal to zero (overwrite them/mask them etc.)

```
y <- 1:5
y
```

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

```r
y > 3
```

```
[1] FALSE FALSE FALSE  TRUE  TRUE
```

```r
y[y>3] <- 0
```

Translate this to the student data. I want to combine the `is.na()` with making these elements equal to zero. Then take this "masked" (vector of student scores with NA values as zero) and drop the lowest and get the mean.

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

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

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

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

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

```
[1] 91
```

Student 3

```r
x <- student3
x[is.na(x)] <- 0
mean(x[-which.min(x)])
```

```
[1] 12.85714
```

Now I can turn my snippet into my first function.

```r
grade <- function(x) {
  # Mask NA (missing work) equal to zero
  x[is.na(x)] <- 0
  # Drop lowest score and get mean
  mean(x[-which.min(x)])
}
```

```
grade(student3)
```

```
[1] 12.85714
```

> 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]

```
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
```

The `apply()` function in R is super useful but can be a little confusing to begin with. Let's have a look at how it works.

`apply(x, MARGIN, FUNCTION, extra arguments for fun)`

- x is the dataset (gradebook)
- MARGIN is the row or column or both (1 for row, 2 for column, c(1,2) for both)
  - This is essentially asking whether you want to grade the students (rows) or grade the homework (column)
- FUNCTION is the function you wrote above previously
- extra arguments for fun (ex: y=ans in Q4)

```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
max(ans)
```

```
[1] 94.5
```

Student-18 was the top scoring student.

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

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

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

```r
which.min(apply(gradebook, 2, mean, na.rm=TRUE))
```

```
hw3 
  3 
```

Homework 3 was the toughest on students.

> 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]

```r
#ans
cor(gradebook$hw1, ans)
```

```
[1] 0.4250204
```

```r
cor(gradebook$hw5, ans)
```

```
[1] NA
```

```r
gradebook$hw5
```

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

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

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

```
[1] 0.6325982
```

Now we can use `apply()` to examine the correlation of every assignment in the masked gradebook to the overall score for each student in the class.

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

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

Homework 5 was most predictive of overall score.