Class 6: R functions

Yang Liu (A16629623)

All functions in R have at least 3 things:

-A **name** (youget to pick this) -input **arguments** (there can be only one or loads -again your calls) -the **body** (where the works gets done, this code between the curly brackets{})

Lest's write a function to add some numbers, we can call it 'add'

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

```
[1] 20
```

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

can I just use my function?

```
add(1)
```

```
[1] 11
```

Let's make it a bit more flexible.

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

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

```
[1] 20
```

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

add(x=10, y=10)
```

```
[1] 20
```

```
add(10)
```

```
[1] 11
```

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)
```

Start with student 1

```
mean (student1)
```

```
[1] 98.75
```

#rm means remove, na.rm will be remove the NA

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

```
[1] 91
```

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

```
[1] 90
```

Let's try to work with student1 and find (and drop) the 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
```

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

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

```
mean(student1)
```

```
[1] 100
```

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

```
[1] 100
```

```
x <- student2
mean(x[-which.min(x)], na.rm=TRUE)
```

```
[1] 92.83333
```

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

1st task is find the NA values

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

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

I have find the NA (TRUE) values from 'is.na()' now I want to make them equal to zero(overwrite them/mast them etc.)

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

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

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

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

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

```
[1] 1 2 3 0 0
```

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

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

```
[1] 12.85714
```

```
grade <- function (scores) {
  # Remove one lowest score or NA
  scores[is.na(scores)] <- 0
  if (length(scores) > 1) {
    scores <- scores[-which.min(scores)]
  }
  # Return the average of the remaining scores
  mean(scores)
}

grade(student2)
```

```
[1] 91
```

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

The **apply()** function in R is super useful but can be a little confusing to begin with.

```
gradebook <- read.csv(url("https://tinyurl.com/gradeinput") )
apply(gradebook[,-1], 1, grade)
```

```
 [1] 91.75 82.50 84.25 84.25 88.25 89.00 94.00 93.75 87.75 79.00 86.00 91.75
[13] 92.25 87.75 78.75 89.50 88.00 94.50 82.75 82.75
```

```r
gradebook <- read.csv(url("https://tinyurl.com/gradeinput") )
average_scores <- apply(gradebook[, -1], 1, grade)

# Now find the top scoring student's name
top_student_name <- gradebook$X[which.max(average_scores)]

top_student_name
```

```
[1] "student-18"
```

```r
max(average_scores)
```

```
[1] 94.5
```

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

```r
apply(gradebook[,-1], 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[,-1], 2, mean, na.rm=TRUE))
```

```
hw3 
  3 
```

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
average_scores
```

```
 [1] 91.75 82.50 84.25 84.25 88.25 89.00 94.00 93.75 87.75 79.00 86.00 91.75
[13] 92.25 87.75 78.75 89.50 88.00 94.50 82.75 82.75
```

```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
cor(gradebook$hw1, average_scores)
```

```
[1] 0.4250204
```

```r
cor(gradebook$hw2, average_scores)
```

```
[1] NA
```

```r
cor(gradebook$hw3, average_scores)
```

```
[1] 0.3042561
```

```r
cor(gradebook$hw4, average_scores)
```

```
[1] NA
```

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

```
[1] NA
```

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

```
            X hw1 hw2 hw3 hw4 hw5
1   student-1 100  73 100  88  79
2   student-2  85  64  78  89  78
3   student-3  83  69  77 100  77
4   student-4  88   0  73 100  76
5   student-5  88 100  75  86  79
6   student-6  89  78 100  89  77
7   student-7  89 100  74  87 100
8   student-8  89 100  76  86 100
9   student-9  86 100  77  88  77
10 student-10  89  72  79   0  76
11 student-11  82  66  78  84 100
12 student-12 100  70  75  92 100
13 student-13  89 100  76 100  80
14 student-14  85 100  77  89  76
15 student-15  85  65  76  89   0
16 student-16  92 100  74  89  77
17 student-17  88  63 100  86  78
18 student-18  91   0 100  87 100
19 student-19  91  68  75  86  79
20 student-20  91  68  76  88  76
```

```
cor(mask$hw1, average_scores)
```

```
[1] 0.4250204
```

```
cor(mask$hw2, average_scores)
```

```
[1] 0.176778
```

```
cor(mask$hw3, average_scores)
```

```
[1] 0.3042561
```

```
cor(mask$hw4, average_scores)
```

```
[1] 0.3810884
```

```
cor(mask$hw5, average_scores)
```

```
[1] 0.6325982
```

```
apply(mask[,-1], 2, cor, y=average_scores)
```

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

Q5. Make sure you save your Quarto document and can click the “Render” (or Rmarkdown”Knit”) button to generate a PDF foramt report without errors. Finally, submit your PDF to gradescope. [1pt]