Class 06: R Functions

Jason Hsiao (PID: A15871650)

Every function in R has at least 3 things: 1) Name 2) Arguments (input(s) to your function) 3) The body

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).

```
# 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)
```

Student 1 Average

```
mean(student1)
```

```
[1] 98.75
```

Minimum Grade for Student 1

```
min(student1)
```

```
[1] 90
```

Using `which.min` to determine vector position of minimum

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

```
[1] 8
```

Using `which.min` to remove lowest score

```r
student1_grade <- mean(student1[-which.min(student1)])
student1_grade
```

```
[1] 100
```

What about for student 2?

```r
x <- student2
student2_grade <- mean(x[-which.min(x)], na.rm = TRUE)
student2_grade
```

```
[1] 92.83333
```

Let’s set it so that if you don’t submit an assignment you get zero points:

```r
student2[is.na(student2)] <- 0
student2
```

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

Putting it all together

```r
x <- student3
#Masking NA to zero
x[is.na(x)] <- 0
#Find the mean dropping the lowest score
mean(x[-which.min(x)], na.rm = TRUE)
```

```
[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]

Turn this snippet into a function

```
grade <- function(x) {
  x[is.na(x)] <- 0
  mean(x[-which.min(x)], na.rm = TRUE)
}
```

Using the above function to grade any student

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Making a slightly different grading scheme so that student fails if they fail to submit 2 or more assignments:

```
library(dplyr)
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
y <- student3

score <- function(y) {
  x <- sum(is.na(y)) >= 2
  if(x == TRUE) {print("FAIL!")}
  if(x == FALSE) {print(mean(y[-which.min(y)], na.rm = TRUE)) }
}

score(y)
```

```
[1] "FAIL!"
```

I need to read the gradebook CSV file

```
gradebook <- read.csv("student_homework.csv", row.names = 1)
#Setting NA = 0
gradebook[is.na(gradebook)] <- 0
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   0  73 100  76
student-5   88 100  75  86  79
student-6   89  78 100  89  77
student-7   89 100  74  87 100
student-8   89 100  76  86 100
student-9   86 100  77  88  77
student-10  89  72  79   0  76
student-11  82  66  78  84 100
student-12 100  70  75  92 100
student-13  89 100  76 100  80
student-14  85 100  77  89  76
student-15  85  65  76  89   0
student-16  92 100  74  89  77
```

```
student-17  88  63 100  86  78
student-18  91   0 100  87 100
student-19  91  68  75  86  79
student-20  91  68  76  88  76
```

A very useful function that Barry is forcing us to use here is the `apply()` function. How do we user it to take our `grade()` function and apply it over the full gradebook?

```
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]

##Top Scoring Student

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

```
student-18 
        18 
```

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

##Toughest Homework

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

```
hw2 
  2 
```

> 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]

Take the `apply()` function and the `cor()` function and run over our whole gradebook

```r
which.max(apply(gradebook, 2, cor, ans))
```

```
hw5 
  5 
```