Class 06: R Functions

Nicholas Yousefi

Table of contents

Grade function

> 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]

These are some tests that the function must work with:

```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)
```

Grade function to calculate average of all assignment scores in vector, assignments:

First, we try to find the mean.

```r
mean(student1)
```

```
[1] 98.75
```

Uh oh! That didn't work! It didn't drop the lowest score!

To drop the lowest score, we can use `which.min()`, which tells us the location of the lowest score:

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

```
[1] 8
```

Yup, 8 is the index of the lowest score for `student1`.

We can create a vector without the lowest score as follows, using the `-` operator:

```r
student1WithoutMin <- student1[-which.min(student1)]
student1WithoutMin
```

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

We can combine `which.min()` and the minus index trick to get the student's scores without the lowest score:

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

```
[1] 100
```

Let's try this function with `NA` values:

```r
mean(student2[-which.min(student2)])
```

```
[1] NA
```

Uh oh! It gives NA! We don't want that!

Let's try the `mean()` function with `na.rm=TRUE`.

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

```
[1] 92.83333
```

```r
mean(student3[-which.min(student3)], na.rm=TRUE)
```

```
[1] NaN
```

It gives `NaN` because we are removing all the `NA`'s and then the minimum score (90) is also removed, so we are taking the mean of an empty vector!

What happens when I run `which.min()` on a vector with `NA`?

```r
which.min(student2)
```

```
[1] 8
```

It totally skipped the `NA` value and just said that the min value (80) was at index 8.

Let's try converting the `NA` to a zero. Then the `which.min()` function should work!

```r
is.na(student1)
```

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

```r
is.na(student2)
```

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

```r
is.na(student3)
```

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

We can assign all the `NA` values of the vectors, `student2` and `student3` the value 0 as follows:

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

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

```r
student3[is.na(student3)] <- 0
student3
```

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

Now, we can use what we had before to drop the lowest score and take the mean:

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

```
[1] 12.85714
```

Now, let's convert this whole thing to a function. All this copy-pasting is going to lead to errors.

```r
grade <- function(x) {
  x[is.na(x)] <- 0 # convert all NA values to 0
  mean(x[-which.min(x)]) # drop lowest score, then find mean
}
```

Let's try calling this function on the three students:

```r
grade(student1)
```

```
[1] 100
```

```r
grade(student2)
```

```
[1] 91
```

```r
grade(student3)
```

```
[1] 12.85714
```

Now, we will read a gradebook from online and output the class's grades.

Q2.

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

```r
# First, we will read in the grade book:
url <- "https://tinyurl.com/gradeinput"
gradebook <- read.csv(url, row.names = 1)
```

Let's look at the grades data frame:

```r
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
```

Let's use the `grade()` function to grade all the students:

```r
results <- apply(gradebook, 1, grade)
results
```

```
 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 
```

The `apply()` function is used to apply a function to a data frame. Its arguments are:

1) the data frame (or matrix) to apply the function to
2) a number: 1 means apply it to the rows of the matrix; 2 means apply it to the columns; c(1, 2) means apply it to both rows and columns
3) the function to apply

To find the student that scored the highest in the gradebook, we use `which.max()`:

```
which.max(results)
```

```
student-18
        18
```

This student scored:

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

```
student-18
      94.5
```

Q3.

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

```
lowest_HW <- which.min(apply(gradebook, 2, sum, na.rm=T))
lowest_HW
```

```
hw2
  2
```

Q4.

> 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]

```
mask <- gradebook # mask is the gradebook with zeros for NA
mask[is.na(mask)] <- 0
cor(mask$hw5, results)
```

```
[1] 0.6325982
```

We can use apply to run this function on every homework:

```r
correlations <- apply(mask, 2, cor, y=results)
# the ... in the help page is where you put additional parameters for the function you wan
```

The homework with the highest correlation (i.e. most predictive of overall score) is:

```r
which.max(correlations)
```

```
hw5
  5
```