Class 6: R Functions

Aishwarya R.

Table of contents

Function basics

All functions in R consist of at least three things:

- A **name** (we can pick this but it must start with a character) Input **arguments** (there can be multiple comma separated inputs)
- The **body** (where work actually happens).

```
# 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 by using `mean()` function to calculate an average.

```
mean(student1)
```

```
[1] 98.75
```

Found the `min()` function to find the minimum value in a vector.

```
min(student1)
```

```
[1] 90
```

Looking at See Also for `min()`, found the `which.min()` function which tells location (index) of minimum value in a vector.

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

```
[1] 8
```

Trying to get the vector without the final value explicitly.

```
student1[1:7]
```

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

Can get the same vector without 8th element with the minus index trick.

```
student1[-8]
```

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

So I will combine the output of `which.min()` with the minus index trick to get the student scores without the lowest value

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

```
[1] 100
```

Hmm… For student2 this gives NA

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

```
[1] NA
```

I see there is an `na.rm=FALSE` by default argument to the `mean()` function. Will it help?

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

```
[1] 92.83333
```

For student3:

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

```
[1] 90
```

Did not work! Need another way…

How about we replace all NA (missing values) with zero

```
student3
```

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

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

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

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

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

All the copy pasting is silly and dangerous - need a function.

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

```
[1] 12.85714
```

We now have working snippet code that has been simplified to work with any student `x`.

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

```
[1] 12.85714
```

Now turn into function:

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

```
grade(student1)
```

```
[1] 100
```

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

```
url <- 'https://tinyurl.com/gradeinput'
gradebook <- read.csv(url, row.names=1)
```

Have a look at first 6 rows:

```
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
```

Time to learn about the `apply()` function.

```
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
```

Which student scored the highest score?

```
highest_scorer=which.max(results)
highest_scorer
```

```
student-18
        18
```

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

```
student-18
      94.5
```

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

```
# Using the sum as a metric of the score
minsum <- which.min(apply(gradebook, 2, sum, na.rm=TRUE))
minsum
```

```
hw2
  2
```

```
# Finding the average score of all students for each HW
average <- apply(gradebook, 2, grade)
average
```

```
     hw1      hw2      hw3      hw4      hw5
89.36842 76.63158 81.21053 89.63158 83.42105
```

```
# Indexing average scores by index of minimum average
# Result is the column label and score of the toughest HW
toughest <- average[which.min(average)]
toughest
```

```
     hw2
76.63158
```

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

cor(mask$hw5, results)
```

```
[1] 0.6325982
```

Can do manually for each HW:

```r
cor(mask$hw1, results)
```

```
[1] 0.4250204
```

Or just use apply!

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

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