

class06

Joshua Lau

R Functions

In this class we will work on developing our own functions for calculating average grades for fictional student

We will start with a simplified version of the problem, grading some vectors of student scores. We want to drop the lowest score and get the average

```
# 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)
```

Finding the average of student1 dropping the lowest value

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

```
[1] 100
```

Doing the same on student 2 returns NA

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

```
[1] NA
```

One solution is to use the na.rm argument in the mean function to remove the NAs

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

```
[1] 92.83333
```

However this does not work with student 3 as there is only one numeric value, does not seem fair

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

```
[1] NaN
```

We can use a logical vector to index the vectors for na

```
student2[is.na(student2)]
```

```
[1] NA
```

```
student3[is.na(student3)]
```

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

and assign a 0 to these values

```
#we make copies so that we do not overwrite the original vectors
student2_copy <- student2
student3_copy <- student3

student2_copy[is.na(student2_copy)] <- 0
student3_copy[is.na(student3_copy)] <- 0
student2_copy
```

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

```
student3_copy
```

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

```
#and calculate the mean (with lowest removed
mean(student2_copy[-which.min(student2_copy)])
```

```
[1] 91
```

```
mean(student3_copy[-which.min(student3_copy)])
```

```
[1] 12.85714
```

We can write this all into a function

Q1. Write a function `grade()` to determine an overall grade from a vector of student homework assignment scores dropping the lowest single score.

```
grade <- function(x){  
  #mask NA to zero  
  x[is.na(x)] <- 0  
  #drop the lowest value  
  x <- x[-which.min(x)]  
  #return the mean  
  return(mean(x))  
}
```

Trying it out

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Can we do this with a gradebook csv? First we load the data

```
gradebook <- read.csv("student_homework.csv", row.names = 1)
```

We can use the super useful but complicated `apply()` function to use our existing `grade()` function on the whole class gradebook

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

```
grades <- apply(gradebook, MARGIN = 1, FUN = grade)
```

```
names(which.max(grades))
```

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

The Top Scoring Student overall is Student 18

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

```
apply(gradebook, MARGIN = 2, FUN = sum, na.rm = TRUE)
```

```
hw1 hw2 hw3 hw4 hw5
1780 1456 1616 1703 1585
```

```
which.min(apply(gradebook, MARGIN = 2, FUN = sum, na.rm = TRUE))
```

```
hw2
2
```

The toughest homework assignment was HW2

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]

Create a mask variable with NAs replaced by 0

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

```
Q4 <- apply(mask, MARGIN = 2, FUN = cor, y = grades, use="everything")
```

```
Q4
```

| hw1 | hw2 | hw3 | hw4 | hw5 |
|-----------|-----------|-----------|-----------|-----------|
| 0.4250204 | 0.1767780 | 0.3042561 | 0.3810884 | 0.6325982 |

```
names(which.max(Q4))
```

```
[1] "hw5"
```

```
max(Q4)
```

```
[1] 0.6325982
```

HW 5 has the highest correlation with the final grade with a correlation coefficient **0.63**