class06: functions in R

jack olmstead

In this session you will work through the process of developing your own function for calculating average grades for fictional students in a fictional class. The process will involve starting slowly with small defined input vectors (where you know what the answer should be). Then building up to work with more complex input vectors (with multiple missing elements).

```
# 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)
```

To get the average, we can use the function `mean()`

```
mean(student1)
```

```
[1] 98.75
```

The `min()` and `which.min()` functions will return the smallest score and its index

```
min(student1)
```

```
[1] 90
```

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

```
[1] 8
```

```
x <- 1:5
x
```

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

```
x[-3]
```

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

Put these bits together to find the average score after dropping the lowest grade.

```
# get the final grade after dropping the lowest score
best_grades <- student1[-which.min(student1)]
final_grade <- mean(best_grades)
final_grade
```

```
[1] 100
```

```
# or, all on one line
mean( student1[-which.min(student1)] )
```

```
[1] 100
```

Let's do the other students!

```
student2
```

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

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

```
[1] 92.83333
```

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

```
[1] 12.85714
```

We now have a working code snippet that can be the body of our function. Recall that all functions in R have at least 3 things:

- name (we pick this)
- argumments (input to the function)
- body (where the work gets done)

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]

```
grade <- function(raw_scores) {
  # preserve input
  scores = raw_scores

  # replace NA scores with 0s
  scores[ is.na(scores) ] <- 0

  # drop lowest score
  scores <- scores[-which.min(scores)]

  # get average score
  avg_grade = mean(scores)

  return(avg_grade)
}
```

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

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

```
url <- "https://tinyurl.com/gradeinput"
gradebook = read.csv(url, row.names=1)
results <- apply(gradebook, 1, grade)
results[which.max(results)]
```

```
student-18
      94.5
```

```
gradebook$average <- results
```

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

```
# replace NAs with 0s before calculating average score for each HW
mask <- gradebook
mask[is.na(mask)] <- 0
x <- apply(mask, 2, mean)
x
```

```
    hw1     hw2     hw3     hw4     hw5 average
 89.000  72.800  80.800  85.150  79.250  87.425
```

The toughest homework was homework 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]

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

```
      hw1       hw2       hw3       hw4       hw5   average
0.4250204 0.1767780 0.3042561 0.3810884 0.6325982 1.0000000
```

The homework that is best correlated with overall class performance is homework 5!

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]