

Week 3 R functions

Zainab Ashir

2023-04-23

This week we are introducing **R functions** and how to write our own functions.

questions to answer:

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 adequately 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]

```
# 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)
```

```
# Student1 mean() to find overall grade
student1 <-c(100, 100, 100, 100, 100, 100, 100, 90)

mean(student1)
```

```
## [1] 98.75
```

However... First we need to identify the lowest score in order to drop the lowest score.

```
# Which element of the vector is the lowest?
which.min(student1)
```

```
## [1] 8
```

Now I want to drop the lowest score (exclude it) from my `mean()` calculation

```
# This will return everything except the eighth element within this vector
student1[-8]
```

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

Now we can use the answer from `which.min()` to return all other elements of the vector

```
# This is our first working portion
mean(student1[-which.min(student1)])
```

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

Now, make a function that works with NA case for the other two students

```
# Finding the overall grade of student2 with lowest score (NA) dropped
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
x <- student2
x[is.na(x)] <- 0
mean(x[-which.min(x)])
```

```
## [1] 91
```

```
# Finding the overall grade of student3 with lowest score (NA) dropped
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
y <- student3
y[is.na(y)] <- 0
mean(y[-which.min(y)])
```

```
## [1] 12.85714
```

Now, we make our own function to find student's overall grade with lowest homework dropped, with comments added for our future selves.

```
#' Calculate the average score for a vector of
#' student score dropping the lowest score
#' Missing values (NA) will be treated as zero.
#'
#' @param x A numeric vector of homework scores
#'
#' @return Average score
#' @export
#'
#' @examples student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
#' grade(student2)
#'

grade <- function(x) {
  # mask NA with zero to treat missing values as zero
  x[is.na(x)] <- 0
  # Exclude lowest score from mean
  mean(x[-which.min(x)])
}
```

Now we can test if it works correctly!!

```
# Using the function grade() on example student input vectors from previously
grade(student1)
```

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

```
grade(student2)
```

```
## [1] 91
```

```
grade(student3)
```

```
## [1] 12.85714
```

Now, we can finally answer the Q1 with applying the function to the whole grass in the CSV file. First, need to insert the data of the class. CSV format: "https://tinyurl.com/gradeinput"

```
# Data inserted, and fixed row so that the first colum is the hw scores
url <- "https://tinyurl.com/gradeinput"
gradebook <- read.csv(url, row.names=1)
```

```
# Now, run the function grade() for our gradebook data to find each student overall grade in class
apply(gradebook, 1, grade)
```

```
## 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]

```
# store overall grades function for each student in a variable
students_scores <- apply(gradebook, 1, grade)
# find the max score out of these students and name it
top_score <- which.max(students_scores)
# print the variable to get the answer!
top_score
```

```
## student-18
##          18
```

Awesome, we just figured out that the highest scoring student is student-18!!!

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

```
# make a function with apply() to show average hw scores and name it
avergae_hw <- apply(gradebook, 2, mean, na.rm=TRUE)
# print its name
avergae_hw
```

```
##      hw1      hw2      hw3      hw4      hw5
## 89.00000 80.88889 80.80000 89.63158 83.42105
```

```
# find the minimum average
which.min(avergae_hw)
```

```
## hw3
##    3
```

We have found that hw 3 has the minimum avergae, thus it was the toughest on the students.

Q4. From your analysis of the gradebook, which homework was most predictive of overall score (i.e. highest correlation with average grade score)? [1pt]

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

# test finding the correlation of students overall grade in hw5
cor(students_scores, masked.gradebook$hw5)
```

```
## [1] 0.6325982
```

```
# Now, we do it for all homeworks
apply(masked.gradebook, 2, cor, x=students_scores)
```

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

Thus, we have found that hw5 has the highest correlation!!!

Q5. Make sure you save your Quarto document and can click the “Render” (or Rmark- down”Knit”) button to generate a PDF format report without errors. Finally, submit your PDF to gradescope.
[1pt]

Knit documnt to make a PDF