

# Class 6: Writing R Functions

Julia Ainsworth

10-14-22

## Table of contents

Input vectors . . . . .	1
<b>Q1</b>	<b>1</b>
Notes for finishing Q1 . . . . .	2
Getting a formula with one easy vector - student 1 . . . . .	2
Notes from class: . . . . .	2
Answer Q1 . . . . .	4
<b>Q2</b>	<b>5</b>
<b>Q3</b>	<b>6</b>
<b>Q4</b>	<b>6</b>

## Input vectors

```
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)
```

## 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]

## Notes for finishing Q1

### Getting a formula with one easy vector - student 1

```
mean(student1)
```

```
[1] 98.75
```

```
#so far so easy  
which.min(student1)
```

```
[1] 8
```

```
student1[-8]
```

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

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

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

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

```
[1] 100
```

### Notes from class:

- A **name** (we determine this)
- Input **arguments** (there can be loads, comma separated)
- A **body** (the R code that does the work)

Student 2 - NAs are a problem

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

[1] NA

```
#NAs are a problem  
  
mean(adjusted.student2, na.rm = TRUE)
```

[1] 92.83333

Student 3

```
adjusted.student3 <- student3[-which.min(student3)]  
adjusted.student3
```

[1] NA NA NA NA NA NA NA

```
mean(adjusted.student3, na.rm = TRUE)
```

[1] NaN

```
#many NAs are also a problem
```

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

[1] FALSE TRUE FALSE FALSE FALSE FALSE FALSE

```
sum(is.na(student2))
```

[1] 1

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

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

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

```
[1] 91
```

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

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

```
adjusted.student3 <- student3[-which.min(student3)]
mean(adjusted.student3)
```

```
[1] 12.85714
```

```
# or alternately
missing_hw <- is.na(student3)
student3[ missing_hw] <- 0
student3
```

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

```
mean(student3[-which.min(student3)])
```

```
[1] 12.85714
```

Putting all of that into a function

## Answer Q1

```
# Combining 1. dropping the lowest score and 2. making NAs 0s
grade <- function(x) {
  x[ is.na(x) ] <- 0;
  mean(x[ -which.min(x) ] )
}
# IT W O R K S !!!!!!!
# Note: absolutely the hardest part was the parentheses. Spaces help with this
```

Testing out the function:

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

## Q2

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

Importing data from csv

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

Now we are going to use the `apply()` function; this will more efficiently apply a function to a matrix; we will apply our function `grade` to the matrix `gradebook`

```
student_results <- apply(gradebook, 1, grade)
#Note: using 1 here for the margin to get the students
which.max(student_results)
```

```
student-18
      18
```

### Q3

```
hardest_hw <- apply(gradebook, 2, sum, na.rm = TRUE)
hardest_hw
```

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

```
#Note: using 2 instead of 1 here returns hw numbers (columns; margin =2 )
which.min(hardest_hw)
```

```
hw2
      2
```

### Q4

```
mask <- gradebook
mask
```

	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
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	NA	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	NA
student-16	92	100	74	89	77
student-17	88	63	100	86	78
student-18	91	NA	100	87	100
student-19	91	68	75	86	79
student-20	91	68	76	88	76

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

	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
```

```
cor(mask$hw5, student_results)
```

```
[1] 0.6325982
```

Applying `cor()` function to gradebook where NAs have been replaced with 0s?

```
q4 <- apply(mask, 2, cor, y = student_results)
q4
```

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