

Class 06: R Functions

Alexander Ko: A17576411

#All about functions in R Functions are the way we get stuff done in R. We call a function to read data, compute stuff, plot stuff, etc.

R makes writing functions accessible but we should always start by trying to get a working snippet of code first before we write our function

##Today's lab

We will grade a whole class of student assignments. We will always try to start with a simplified version of the problem

```
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)
```

```
#If we want the average we can use the `mean` function
mean(student1)
```

```
[1] 98.75
```

Let's be nice instructors and drop the lowest score so the answer is 100.

We can use the `min()` function to find the lowest value

```
min(student1)
```

```
[1] 90
```

I found the `which.min` function that may be useful here. How does it work?

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

```
[1] 8
```

I can use the minus syntax trick to get everything but the min value

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

```
[1] 90
```

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

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

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

```
[1] 100
```

I have my first working snippet of code

Let's test on the other students

```
student2
```

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

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

```
[1] NA
```

where is the problem??

```
mean(student2)
```

```
[1] NA
```

Where is the problem? It's with the NA result

```
mean(student2, trim=0, na.rm=TRUE)
```

```
[1] 91
```

```
student3
```

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

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

```
[1] 90
```

No bueno! We need to fix

I want to stop working with `student1`, `student2`, `student3` and typing it out every time, let's instead work with an input called `x`

```
x <- student2  
x
```

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

We want to overwrite the NA values with zero - if you miss a homework assignment, you score zero.

Google and Claude told me about the `is.na` function. Let's see how it works.

```
is.na(x)
```

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

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

We can use logicals to index a vector

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

```
[1] 91
```

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

```
[1] 100
```

This is my working snippet of code that solves my example

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

```
[1] 12.85714
```

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]

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

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

We need to read the gradebook

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

I can use the `apply()` function if I figure out how to use the damn thing 1 is rows and 2 is columns

```
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	

```
answer <- apply(gradebook,1,grade)
```

```
answer
```

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]

```
which.max(answer)
```

```
student-18  
18
```

Q2: the top scoring student was student18, with a 94.50

Q3: we need to find which hw assignment has the lowest score

```
apply(gradebook,2,grade)
```

	hw1	hw2	hw3	hw4	hw5
	89.36842	76.63158	81.21053	89.63158	83.42105

```
answer2 <- apply(gradebook,2,grade)
```

```
which.min(answer2)
```

```
hw2  
2
```

q3: the lowest score was on hw2