

20231020_BGGN213_Class06

Savannah Bogus A69027475

All about functions in R

Every function in R has at least 3 things: name, arguments (inputs to functions), and the body.

Today's function will be to grade a class of student assignment scores, such as homework.

Initially, I will make a simplified vector input where I know what the answer will be.

```
# 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)
```

In class practice before questions

I defined a student1d vector below to make things easier for me to see so that I could edit easier and visually understand what is going on, but if I were to make it cleaner, I could just put in the mean of what is contained in student1d without defining student1d. For student 1 and student 2, I will not define this student1d vector.

```
mean(student1)
```

```
[1] 98.75
```

```
min(student1)
```

```
[1] 90
```

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

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

```
#student1d means student1dropped  
student1d<-student1[-which.min(student1)]  
mean(student1d)
```

```
[1] 100
```

I found the `which.min(x)` function both in google and the help pages, which I didn't understand before, but I'm getting it now after having seen you using it on the screens. Now, I know I can use the `-` sign to get rid of that min. However, if you look in the code chunk below the NAs are still causing problems. I googled a how to replace NA with 0's in a vector, and I found this function, but there's also a more intuitive (to me) `replace` function that I could use.

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

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

I did manage to replace NA with 0. Jiachen also just told me this function is in our pdf for today, so I googled it for no reason when you provided the right information. Bummer for the amount of work I'm putting in. Now, can I take the mean of that?

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

```
[1] 91
```

Looks like it worked! Now, there's problems with student 3 who has multiple NA's, and that I'm really struggling with, honestly. I need to drop only one 0, and I have a feeling the tactic I'm using above won't work.

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

```
[1] 12.85714
```

Oh, my bad. It really only does drop one min, not all of the mins. That's awesome! Now, I think I need to make a full function for it? But I'll be honest, I'm super lost about how to make a full working function. I'll start to try to do so below, and hopefully I get it right, though.

```
#Need to define x
x<-student1
#Make NA's 0. Barry used the word "Mask"
x[is.na(x)]<-0
#Finding the mean after dropping the lowest score without defining a new vector
mean(x[-which.min(x)])
```

[1] 100

Above, that should work for any student defined x. I probably need to know how to make x be any student so it can go through a list of students, which I'm not totally sure how to d, but I think I need to make x a data frame?

```
grade<-function(x){
  #Make NA's 0. Barry used the word "Mask"
  x[is.na(x)]<-0
  #Finding the mean after dropping the lowest score without defining a new vector
  mean(x[-which.min(x)])
}
```

Okay, so that should be the grade function. Good. Great. But, I need to make x be that gradebook you put in the PDF. That's Q1.

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]

```
# Read grade input CSV into DataFrame
grades_df <- read.csv("https://tinyurl.com/gradeinput",
                      row.names=1)
```

I gotta figure out how to grade this. It's not working when I just try to grade. It keeps telling me I have an error in `which.min(x)`.

```
'grade(grades_df)'
```

That didn't work in R (so I took it out of R code). Apparently, we have to use `apply()` somehow, but I'm not sure how, and the help page isn't super helpful, as you stated. I googled it, though, and found a UChic page that I'm going to try to use myself before you get to it in class. I like to work ahead and fail a lot :)

```
apply(grades_df,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	

```
scores<-apply(grades_df,1,grade)
```

Heck yeah, I did it before you got to it in class! Thanks, UChic page about `apply`! That was way more understandable. Now, I can get to more questions.

Q2

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

I have no real idea if I can apply a `max()` function to an `apply()` function, but I'm going to try. I'll leave my failure below if it fails.

```
max(apply(grades_df,1,grade))
```

```
[1] 94.5
```

Okay, cool, so I know the top score, but not the actual student it was. I forgot but I know there's a way to find the name of it. Is it `row.names`? I'm gonna try that. Me from slightly in the future: it didn't work and neither did `rownames`. I bet it's `which.max()`

```
rownames(max(apply(grades_df,1,grade)))
```

NULL

I'm trying `which.max()`

```
which.max(apply(grades_df,1,grade))
```

```
student-18  
18
```

So, student 18 is our top scoring pal, and I know from something I did before that they got a 94.5.

Q3

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

Okay, so I think I need to not use the grade function where I drop a lowest score because I actually want to look at the lowest score this time.

```
which.min(apply(grades_df,2,mean,na.rm=TRUE))
```

```
hw3  
3
```

I need to be careful about getting rid of the NA's, though. If no one did the HW because it was so hard, then that would be the hardest HW, so I need to look at the same thing but with NA's=0. So, I'll make a new df with NAs masked to 0.

```
grades_df2<-grades_df  
grades_df2[is.na(grades_df2)]<-0  
grades_df2
```

	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

Now, let's find this min.

```
which.min(apply(grades_df2,2,mean))
```

```
hw2
2
```

But, maybe it's not fair to put NA's as 0. Maybe I should do the sum instead.

```
which.min(apply(grades_df2,2,sum))
```

```
hw2
2
```

Oh, but it's the same answer. Either HW2 or HW3 are the hardest HW's. I could also do a muddypoint survey like you do to figure it out from the horse's (student's) mouth.

Q4

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]

That'll be the correlation function.

```
cor(grades_df2$hw2,scores)
```

```
[1] 0.176778
```

```
cor(grades_df2$hw5,scores)
```

```
[1] 0.6325982
```

I'm trying to use `apply()` to get a correlation between each HW. Struggling, though, since `cor()` needs an x and a y. `apply(x, MARGIN, function)`

```
apply(grades_df2,2,cor,y=scores)
```

hw1	hw2	hw3	hw4	hw5
0.4250204	0.1767780	0.3042561	0.3810884	0.6325982

I had to argue to define y when using `cor()`. The HW with the highest correlation is HW5.