

102023 Lab6

Blair (PID A59000602)

All about functions in R

Every function in R has at least 3 things: - name - argument (the input to my function) - body

Today we will write a function to grade a class of student assignment scores

First I will work with a simplified vector input that I know the answer

```
# 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)
```

Let's start slow and find the average for student1

```
mean(student1)
```

```
[1] 98.75
```

How can we drop the lowest score? I can use `min()` function to find the lowest score

```
min(student1)
```

```
[1] 90
```

```
#I found the function `which.min()` that will locate the lowest score
```

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

```
[1] 8
```

```
student1[8]
```

```
[1] 90
```

```
student1[-8] # excludes 8th score
```

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

```
#Drop the lowest score from Student1 and find avg  
mean(student1[-which.min(student1)])
```

```
[1] 100
```

```
# Students have NA?  
d <- student2  
mean(d, na.rm = TRUE)
```

```
[1] 91
```

We can mask NA or change NA to zero. We can use the `is.na()` function to find where the missing hw is in the input vector. And use `x[!is.na(x)]` to exclude NA from vector

```
is.na(student3)
```

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

```
student3[!is.na(student3)]
```

```
[1] 90
```

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

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

I think we are there.

```

z <- student3
# Mask NA to zero
z[ is.na(z)] <- 0
# Find the mean dropping the lowest score
mean(z[ -which.min(z)])

```

```
[1] 12.85714
```

Turn this snippet into a function

```

grade <- function(x){
  # This is the body code lives
  x[ is.na(x)] <- 0
  mean(x[ -which.min(x)])
}

```

We can use this function now to grade student

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Q1 I need to read the gradebook

```

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

```

```
# A very useful function that Barry is forcing us to use here is the `apply()`
```

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

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 Find the top score

```
which.max(ans)
```

```
student-18
18
```

Q3 Find the hardest HW? Depends on how we wanna define difficult

```
hardest_hw <- which.min( apply(gradebook, 2, mean, na.rm=TRUE))
hardest_hw
```

```
hw3
3
```

Let's mask NA to zero!

```
mask <- gradebook
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

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

```
hw2
2
```

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

```
hw2
2
```

So, I would say hw2 is the hardest homework

Q4 Which hw can predict the overall student performance?

```
cor(mask$hw2, ans)
```

```
[1] 0.176778
```

```
cor(mask$hw1, ans)
```

```
[1] 0.4250204
```

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

```
[1] 0.6325982
```

Now take `apply()` function again and the `cor()` run over the whole gradebook

```
apply(mask, 2, cor, y=ans)
```

	hw1	hw2	hw3	hw4	hw5
	0.4250204	0.1767780	0.3042561	0.3810884	0.6325982

```
which.max(apply(mask, 2, cor, y=ans))
```

```
hw5  
5
```

hw5 will be the most predictable hw for students' performance!