

Class 06: R functions

Abzael Verduzco

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. 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 students assignment. We will always try to start with a simplified version of the problem.

```
# 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)
```

If we want the average we can use the `mean` function:

```
mean(student1)
```

```
[1] 98.75
```

Let's be nice instructors drop the lowest score so the answer here should be 100.

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

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

```
[1] 8
```

```
mean(student1[1:7])
```

```
[1] 100
```

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

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

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

```
# or  
student1[-8]
```

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

First working snippet of code

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

```
[1] 100
```

```
mean(student1[-8])
```

```
[1] 100
```

```
mean(mean(student1[1:7]))
```

```
[1] 100
```

Let's test on the other students

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

```
[1] NA
```

Where is the problem?

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

```
[1] 91
```

Let's try student 3

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

```
[1] 90
```

```
mean(student3)
```

```
[1] NA
```

I want to stop working with `student1`, `student2` etc. and typing it out every time so let 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 you score zero on this homework.

Google, ChatGPT, and Claude told me about the `is.na()` function. Lets see how it work.

```
# For student 2  
x[is.na(x)] <- 0  
x
```

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

```
x[x>0]
```

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

```
mean(x)
```

```
[1] 79.625
```

```
# This is another method to do it by dropping the lowest score  
mean(x[x>0])
```

```
[1] 91
```

Then drop the lowest score for all students and calculate the mean, after making NA = 0

The following are working snippets of code that solves the problem for all my example student inputs :-)

```
# For student1  
x <- student1  
x[is.na(x)] <- 0  
mean(x[-which.min(x)])
```

```
[1] 100
```

```
# For student2  
x <- student2  
# Make NA = 0  
x[is.na(x)] <- 0  
# Take the mean of the scores after dropping the lowest value  
mean(x[-which.min(x)])
```

```
[1] 91
```

```
# For student3  
x <- student3  
# Make NA = 0  
x[is.na(x)] <- 0  
# Take the mean of the scores after dropping the lowest value  
mean(x[-which.min(x)])
```

```
[1] 12.85714
```

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) {  
  # Make NA = 0  
  x[is.na(x)] <- 0  
  # Take the mean of the scores after dropping the lowest value  
  mean(x[-which.min(x)])  
}
```

Use this function:

```
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

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

```
ans <- apply(gradebook, 1, grade, simplify = T)
```

```
which.max(apply(gradebook, 1, grade, simplify = T))
```

```
student-18
18
```

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

```
student-18
18
```

Student 18 is the top scoring student.

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

```
ans2 <- apply(gradebook, 2, grade, simplify = T)
```

```
which.min(apply(gradebook, 2, grade, simplify = T))
```

```
hw2  
2
```

```
which.min(ans2)
```

```
hw2  
2
```

The toughest homework on the students was Hw 2.