

Class 6: R Functions

Arshiya Zarmahd (PID: A16247996)

Functions are how we get work done in R. We call functions to do everything from reading data to doing analysis and outputting plots and results.

All functions in R have at least 3 things:

- A **name** (you pick this)
- Input **arguments** (only one or many, between parenthesis)
- A **body** (where work gets done, the code between curly brackets)

First, A Silly Function

Lets write a function to add some numbers. We can call it `add()`.

```
x <- 10  
y <- 10  
x + y
```

```
[1] 20
```

```
add <- function(x) {  
  y <- 10  
  x + y  
}
```

Can I now use my new function?

```
add(1)
```

```
[1] 11
```

Lets make it a bit more flexible.

```
add <- function(x,y=1) {  
  x + y  
}  
  
add(x=10,y=10)
```

```
[1] 20
```

```
add(10)
```

```
[1] 11
```

```
add(10, 100)
```

```
[1] 110
```

My Turn: Example Student Grades

Write a function to grade student work.

We will start with a simple version of the problem and the following example student vectors.

```
# 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)
```

```
mean(student1)
```

```
[1] 98.75
```

```
mean(student2)
```

```
[1] NA
```

```
mean(student3)
```

```
[1] NA
```

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

```
[1] 91
```

What is student 1's lowest score?

Google told me about `min()` and `max()`

```
min(student1)
```

```
[1] 90
```

When did student 1 score their lowest?

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

```
[1] 8
```

How do I remove student 1's lowest score from their dataset?

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

```
[1] 90
```

```
student1[-8]
```

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

```
mean(student1, trim=90)
```

```
[1] 100
```

```
mean(student1, trim=(student1 [which.min(student1)]))
```

```
[1] 100
```

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

```
[1] 100
```

Our first working snippet that drops the lowest score and calculates the mean.

```
x <- student3  
mean(x[ -which.min(x) ], na.rm=T)
```

```
[1] NaN
```

Our approach to the NA problem (missing homework). We can replace all NA values with 0.

1st task is finding the NA values (ie. where they are in the vector).

```
x <- student2  
x
```

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

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

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

I have found the NA value (TRUE) from `is.na()`

Now I want to make the NA value equal to 0 (overwrite/mask them).

```
y <- 1:5  
y
```

```
[1] 1 2 3 4 5
```

```
y[y > 3] <- 0
y
```

```
[1] 1 2 3 0 0
```

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

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

I want to combine the `is.na(x)` with making these elements equal to 0.

And then take this “masked” (vector of student scores with NA values as 0) and drop the lowest and get the mean.

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

```
[1] 91
```

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

```
[1] 12.85714
```

Gradebook Dataset

Now I can turn this snippet into my first function.

```
grade <- function(x) {
  # Make NA (missing work) equal to 0
  x[is.na(x)] <- 0
  # Drop lowest score and get mean
  mean(x [-which.min(x)] )
}
```

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[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].

```
url <- "https://tinyurl.com/gradeinput"
gradebook <- read.csv(url, 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

The `apply` function in R is super useful but can be a little confusing to begin with. Lets have a look at how it works.

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

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

```
student-18  
18
```

```
max(ans)
```

```
[1] 94.5
```

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

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

```
hw3  
3
```

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].

```
#ans  
cor(gradebook$hw1, ans)
```

```
[1] 0.4250204
```

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

```
[1] NA
```

```
cor(gradebook$hw3, ans)
```

```
[1] 0.3042561
```

```
cor(gradebook$hw4, ans)
```

```
[1] NA
```

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

```
[1] NA
```

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

```
[1] NA
```

```
gradebook$hw5
```

```
[1] 79 78 77 76 79 77 100 100 77 76 100 100 80 76 NA 77 78 100 79  
[20] 76
```

Make all NA values into 0.

```
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

```

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

```
[1] 0.6325982
```

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

```

      hw1      hw2      hw3      hw4      hw5
0.4250204 0.1767780 0.3042561 0.3810884 0.6325982

```

Homework 5 had the highest correlation with average grade score.