

Class 6: R Functions

Achyuta (PID: A16956100)

Today we are going to explore R functions and begin to think about writing our own functions.

Let's start simple and write our first function to add some numbers.

Every function in R has at least 3 things:

- a **name**, we pick this
- one or more input **arguments**
- the **body**, where the work gets done

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

Now lets try it out

```
add(c(10, 1, 1, 10),1)
```

```
[1] 11  2  2 11
```

```
add(10, 11)
```

```
[1] 21
```

```
# add('barry')
```

```
mean( c(10, 10, NA), na.rm=T)
```

```
[1] 10
```

```
# 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)
```

Begin by calculating the average for student 1

```
mean(student1)
```

```
[1] 98.75
```

```
student2
```

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

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

```
[1] 91
```

```
student3
```

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

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

```
[1] 90
```

```
min(student1)
```

```
[1] 90
```

```
student1
```

```
[1] 100 100 100 100 100 100 100 90
```

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

```
[1] 100
```

Replace all NA with 0

```
x <- student2
x[is.na(x)] <- 0
mean(x[-which.min(x)])
```

```
[1] 91
```

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]

This is the creation of the `grade` function. First, it defines the function `grade` as having one variable (called `x`). Then in the body of the function, it starts by changing every “NA” present in the gradebook to a 0. Then, it takes the mean of every element in in that specific row of the gradebook, while eliminating the lowest value, as the lowest grade is eliminated in the calculation.

```
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
```

This line of code reads the code line-by-line in order to match every student with their scores. The `head()` function prints the first 6 rows in order to look at the gradebook.

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

Q2. Using your `grade()` function and the supplied gradebook, Who is the top scoring student overall in the gradebook?

The `grade` function from above is applied to the gradebook. The 1 in the `apply` function specifies that the `grade` function should be applied to each row, instead of each column. This allows for the calculation of an average grade for each student, which is stored in **ans**. From this, we can see that student 18 is the highest scoring student in the class.

```
ans <- apply(gradebook, 1, grade)
which.max(ans)
```

```
student-18
18
```

```
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	

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

This code helps calculate the average grades for each of the homework assignments. First, a new **masked_gradebook** is created. Then, all of the na values in **masked_gradebook** are converted to 0, similar to the previous function. Then, the mean function is applied to **masked_gradebook**, using 2 instead of 1 in order to go through each column. Each homework assignment along with its associated average score is saved under **z**. Finally, **z** is printed to show all of the average scores for each of the homework assignments, plus the assignment with the lowest average score. Here, we can see that the assignment with the lowest average score was assignment 2.

```
masked_gradebook <- gradebook
masked_gradebook[is.na(masked_gradebook)] = 0
z<- apply (masked_gradebook, 2, mean)
z
```

```
hw1 hw2 hw3 hw4 hw5
89.00 72.80 80.80 85.15 79.25
```

```
which.min(z)
```

```
hw2
2
```

```
grade2 <- function(x, drop.low = TRUE) {
  x[is.na(x)] <- 0
  if(drop.low) {
    cat("Hello low")
    out <- mean(x[-which.min(x)])
  } else {
    mean(x)
  }
  return(out)
}
grade2(student1)
```

```
Hello low
```

```
[1] 100
```

```
x <- c(100, 90, 80, 100)
y <- c(100, 90, 80, 100)
z <- c(80, 90, 100, 10)
cor(x,y)
```

```
[1] 1
```

```
cor(x,z)
```

```
[1] -0.6822423
```

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)?

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

```
[1] 0.4250204
```

Here, we find the correlation of the grades on each homework assignment with the overall grades in the class. This is done using the **cor** function, applied over the assignment scores generated listed in **masked_gradebook** and the overall average class grades stored in **ans**. Then, we save all of the correlation values as **correlations**.

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

```
[1] 0.176778
```

```
correlations <- apply(masked_gradebook, 2, cor, ans)
correlations
```

	hw1	hw2	hw3	hw4	hw5
	0.4250204	0.1767780	0.3042561	0.3810884	0.6325982

Here, we find the homework assignment with the maximum correlation value. To do this, we simply have to look through the **correlations** variable, until we find the maximum correlation value, at which point the associated homework assignment is saved. Then, the assignment number and its associated correlation are both printed. From here, we can see that the assignment with the highest correlation value is HW 5, at around 0.63.

```
s <- which.max(correlations)
correlations[s]
```

```
hw5
0.6325982
```

```
s
```

```
hw5
5
```