

Lab Class 06: R Functions

Jennifer Quach

Table of contents

An R function is defined with: - A **name** - A set of input **arguments** - Regular R code for the **function body**

Q1. Write a function `grade()` to determine an overall grade from a vector of student homework assignment scores dropping the lowest single score.

```
# 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)
```

I can start by using the ‘`mean()`’ function to calculate an average.

```
mean(student1)
```

```
[1] 98.75
```

I found the ‘`min()`’ function to find the minimum value in a vector.

```
min(student1)
```

```
[1] 90
```

Looking at the “See Also” section of the ‘`min()`’

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

```
[1] 8
```

```
student1[1:7]
```

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

I can get the same vector without the 8th element.

```
student1[-8]
```

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

```
is.na(student1)
```

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

So I will combine the output of ‘which.min()’ with the minus index trick to get the student scores without the lowest value.

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

```
[1] 100
```

For student2 this gives NA

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

```
[1] NA
```

‘I see there is an ‘na.rm=FALSE’ by default argument to the ‘mean()’ function.

```
mean(student2[-which.min(student2)], na.rm=TRUE)
```

```
[1] 92.83333
```

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

```
[1] 90
```

We need another way...

How about we replace all NA (missing values) with zero.

```
student3
```

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

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

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

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

```
[1] 90 0 0 0 0 0 0 0
```

```
mean(student3[-which.min(student3)])
```

```
[1] 12.85714
```

All this copy paste is silly and dangerous - time to write a function.

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

```
[1] 12.85714
```

I now have my working snippet of code that I have simplified to work with any student 'x'.

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

```
[1] 12.85714
```

Now turn into a function:

```
grade <- function(x) {
  x[is.na(x)]<-0
  mean(x[-which.min(x)])
}
```

```
grade(student1)
```

```
[1] 100
```

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

```
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

Time to learn about the ‘apply()’ function.

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

Which student did the best overall?

```
which.max(results)
```

```
student-18
18
```

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

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

hw2

2

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)?

```
mask<-gradebook
mask[is.na(mask)] <- 0

cor(mask$hw5 , results)
```

[1] 0.6325982

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

[1] 0.4250204

Or use apply...

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

	hw1	hw2	hw3	hw4	hw5
	0.4250204	0.1767780	0.3042561	0.3810884	0.6325982