

Class 6: R functions

Yi-Hung Lee (PID: A16587141)

Introduction of R Functions

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

All functions in R have at least three things:

- a **name** (you get to pick this)
- a input **arguments** (there can be only one or loads - again you call)
- a **body** (where the work gets done, this code between curly bracket)

A first silly function

Let's write a function to add numbers. We can call it `add()`

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

```
[1] 20
```

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

Can I just use the function?

```
add(10)
```

```
[1] 20
```

Create a gradebook

Write a function to grade student work. ### 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)
```

Calculate the average of the first student

```
mean(student1)
```

```
[1] 98.75
```

Calculate student 2 average score by removing NA

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

```
[1] 91
```

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

```
[1] 90
```

```
mean(na.omit(student2))
```

```
[1] 91
```

```
mean(na.omit(student3))
```

```
[1] 90
```

Drop the lowest grade by finding the min value for student1

```
# Find the lowest score by index
min1 <- which.min(student1)
```

```
# Drop the lowest score
new_student1 <- student1[-c(min1)]

# Get the new mean
mean(new_student1)
```

```
[1] 100
```

Replace the NA with 0 for student2 and student3

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

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

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

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

Drop the lowest grade by finding the min value for student2 and student3

```
min2 <- which.min(student2)
new_student2 <- student2[-(min2)]
mean(new_student2)
```

```
[1] 91
```

```
min3 <- which.min(student3)
new_student3 <- student3[-(min3)]
mean(new_student3)
```

```
[1] 12.85714
```

Create a grade function

```
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)
```

```
grade <- function(x) {
  # Replace NA to 0
  x[is.na(x)] <- 0
  # Find the minimum score by index and remove it
  x <- x[-(which.min(x))]
  # Find the average
  mean(x)
}
```

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Use the grade function to grade the student HW

Read the student homework cvs file

```
setwd("//Users/alex/Desktop/BIMM 143/Lab6")
student.hw <- read.csv("student_homework.csv", row.names = 1)
head(student.hw)
```

```
      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

```

```

# or I can do
# url <- "https://tinyurl.com/gradeinput"
# student.hw <- read.csv(url, row.names = 1)

```

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]

Grade function

```

grade <- function(x) {
  # Replace NA to 0
  x[is.na(x)] <- 0
  # Find the minimum score by index and remove it
  x <- x[-(which.min(x))]
  # Find the average
  mean(x)
}

```

Apply the grade function to each row The `apply()` function in R is super useful but can be a little confusing to begin with.

```

# Apply the grade() function into 2 to 6 index in each row
mean_score <- apply(student.hw, 1, grade)
mean_score

```

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]

```
# Find the max average score student
which.max(mean_score)
```

```
student-18
18
```

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

```
# Apply grade function into each HW
mean_hw <- apply(student.hw, 2, mean, na.rm = TRUE)
mean_hw
```

```
      hw1      hw2      hw3      hw4      hw5
89.00000 80.88889 80.80000 89.63158 83.42105
```

```
# Find the lowest HW
which.min(mean_hw)
```

```
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]

Create another gradebook that replaces NA with 0

```
copy.student.hw <- student.hw
copy.student.hw[is.na(copy.student.hw)] <- 0
head(copy.student.hw)
```

```
      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
```

Use `cor()` function to find the correlation between each HW and the mean score of the students

```
corr <- apply(copy.student.hw, 2, cor, y = mean_score)
corr
```

hw1	hw2	hw3	hw4	hw5
0.4250204	0.1767780	0.3042561	0.3810884	0.6325982

```
which.max(corr)
```

```
hw5
5
```