

Class 6: R Functions

Varun Durai

Table of contents

Question 1	1
Question 2	3
Question 3	5
Question 4	6

Question 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>”

Below are some example input vectors we can use to test our function:

```
# Some example input vectors;
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)

student1
```

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

From the “See Also” page of the `min()` function help page, we find out about `which.min()`.

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

```
[1] 8
```

If we were to use the `mean` function to find the mean grade after removing the lowest value, we would get:

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

```
[1] 100
```

```
#Applying the same method to student2:  
mean(student2[-which.min(student2)])
```

```
[1] NA
```

```
#Applying the same method to student3:  
mean(student3[-which.min(student3)])
```

```
[1] NA
```

We get a value of NA for student2 and student3, which appears because the vectors contains NA values. the `mean` function has an argument to remove these values from the vector :

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

```
[1] 92.83333
```

```
# Applying the same parameters to student3:  
mean(student3[-which.min(student3)], na.rm = TRUE)
```

```
[1] NaN
```

This results in an unwanted value for student3. Instead we can try replacing all NA values with 0

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

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

If we use all of these steps in a single function, we get:

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

grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Question 2

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

The `read.csv()` function can be used to read the data in a CSV file as a data frame. It is important the the `row.names` argument for the method is set to 1 so that the values in the column with the index 1 will become the name their respective rows. This ensures that they are not used in any operations.

```
gradebook <- read.csv(file = "https://tinyurl.com/gradeinput")
head(gradebook)
```

	X	hw1	hw2	hw3	hw4	hw5
1	student-1	100	73	100	88	79
2	student-2	85	64	78	89	78
3	student-3	83	69	77	100	77
4	student-4	88	NA	73	100	76
5	student-5	88	100	75	86	79
6	student-6	89	78	100	89	77

```
colnames(gradebook)
```

```
[1] "X" "hw1" "hw2" "hw3" "hw4" "hw5"
```

```
# Adding the argument row.names = 1:
gradebook <- read.csv(file = "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

The `apply()` function with the following arguments will apply our `grade` function to all of the rows of the dataframe.

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

In order to find the student that got the highest grade overall, we can use the `which.max()` function:

```
which.max(applyfn)
```

```
student-18
18
```

```
# In order to print the student's grade:  
applyfn[which.max(applyfn)]
```

```
student-18  
94.5
```

So, from the results of the above function, we can see that Student 18 had the highest grade with a grade of 94.5

Question 3

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

Using the `apply()` function with 2 as a parameter in place of 1 will result in the function being iterated over the columns of the dataframe. This will help us find the homework with the lowest mean score.

```
applyfnMin <- apply(gradebook, 2, mean, na.rm = TRUE)  
applyfnMin
```

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

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

```
hw3  
80.8
```

This shows us the hw3 had the lowest mean score at 80.8

Question 4

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)?

We can use the `cor()` function to find the correlation between the final grade of the students and their score on a certain homework. First, all NA scores need to be converted to 0 so that the function can be applied to them.

```
mask <- gradebook
mask[is.na(mask)] <- 0
apply(mask, 2, cor, y=applyfn)
```

	hw1	hw2	hw3	hw4	hw5
	0.4250204	0.1767780	0.3042561	0.3810884	0.6325982

This shows us that hw2 has the lowest correlation between the student's final grade and their grade on the homework.