

Week 5: R functions

Qiyu Chen

11/6/2020

Here we will write a function to grade some student homework.

We will start with a more simple input example - a vector of student homework scores.

```
# 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)
```

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]

```
student1
```

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

The regular average will be returned by the `'mean()'`

```
mean(student1)
```

```
## [1] 98.75
```

To find the position of the smallest value (i.e. `min`) in our vector

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

```
## [1] 8
```

So this will get us the min value

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

```
## [1] 90
```

So to get everything but the min value

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

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

Then we can just take the mean of this vector

```
# this is my first solution/snippet  
mean(student1[-which.min(student1)])
```

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

Use the 'is.na()' function

```
x <- student2  
x
```

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

```
# To assign NA as zero scores  
x[is.na(x)] <- 0  
x
```

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

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

```
## [1] 91
```

Now we can make our function

We will take our working snippet and make it a function

```
#' Calculate average scores for a vector of homework scores  
#' dropping the lowest single score  
#' Missing values will be treated as zero scores  
#'  
#' @param x Numeric vector of homework scores  
#'  
#' @return Average score  
#' @export  
#'  
#' @examples  
#' student <- c(100, NA, 90, 80)  
#' grade(student)  
  
grade <- function(x) {  
  # Map NA missing homework values to zero  
  # Missing homework scores zero  
  x[is.na(x)] <- 0  
  # We exclude lowest score homework  
  mean(x[-which.min(x)])  
}
```

Now use it on simple input examples

```
grade(student1)
```

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

```
grade(student2)
```

```
## [1] 91
```

```
grade(student3)
```

```
## [1] 12.85714
```

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>”

Now we can take the gradebook and **grade the whole class** of multiple students.

```
# import the gradebook CSV file
url <- "https://tinyurl.com/gradeinput"
# make the first column as row names
gradebook <- read.csv(url, row.names=1)
```

```
# grade the whole class
apply(gradebook, 1, grade)
```

```
## 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?

```
# access the grading for the whole class
results <- apply(gradebook, 1, grade)
# find the top scoring student
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)?

```
# Let's try the average
hw.ave <- apply(gradebook, 2, mean, na.rm=TRUE)
# find the homework with the lowest average score
which.min(hw.ave)
```

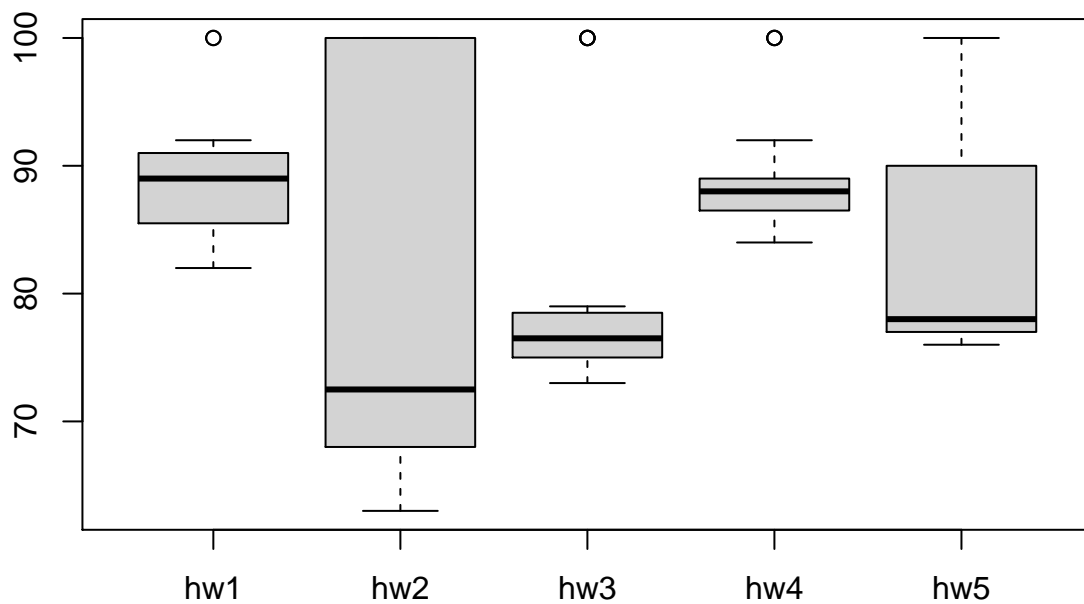
```
## hw3
## 3
```

```
# Let's try the median
hw.med <- apply(gradebook, 2, median, na.rm=TRUE)
# find the homework with the lowest median score
which.min(hw.med)
```

```
## hw2
## 2
```

There is a different result when using mean and median. Plot the data and see

```
boxplot(gradebook)
```



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)?

```
# assign NA to zero scores
gradebook[is.na(gradebook)] <- 0
# see the correlation between average grade scores and homework 1 scores
cor(results, gradebook$hw1)
```

```
## [1] 0.4250204
```

```
# see the correlations between average scores and all homework scores
apply(gradebook, 2, cor, x=results)
```

```
##      hw1      hw2      hw3      hw4      hw5
## 0.4250204 0.1767780 0.3042561 0.3810884 0.6325982
```

Q5. Make sure you save your Rmarkdown document and can click the “Knit” button to generate a PDF format report without errors. Finally, submit your PDF to gradescope