Class 06 Worksheet

Julia Napoli

10/15/2021

Here I will write a function to grade student homework. I will begin with an example of student test scores in a vector.

```
# 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 adquately 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()' function

```
mean(student1)
```

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

To identify the lowest test score to be dropped; to identify the position of the lowest test score to be dropped.

```
min(student1)
```

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

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

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

This will pull the min value by indexing the position back into the original vector.

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

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

ow to pull out every value but the minimum value.

```
student1
```

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

```
student1[-8]
```

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

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

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

Take the mean of the remaining values:

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

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

Let's make sure it works by testing it on another student:

```
student2
```

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

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

```
## [1] NA
```

```
# Need to do some fiddling with the NA value
mean(student2,na.rm = TRUE)
```

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

How do I identify NA values? Use the 'is.na()' function.

```
is.na(student2)
```

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

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

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

```
mean(x)
```

```
## [1] 79.625
```

Time to combine all the useful pieces of script I've written together to find the average score for all students, excluding their lowest score.

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

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

Now, it's time to make the function!

I will take my working snippet and make it a function.

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

Now use the function.

```
grade(student1)
```

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

```
grade(student2)
```

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

```
grade(student3)
```

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

Time to fully **comment the code**.

```
#' Calculate average homework scores (vector form)
#' while dropping the lowest score. Missing values will be
#' treated as zero.
#'
#' @param x Numeric vector of homework scores
#'
#' @return Average score
#' @export
#'
#' @examples
#' student <- c(100,NA,90,85)
#' grade(student)
#'
#'
grade <- function(x) {
  # Turn missing homework scores (NA scores) to zero
  # Missing homework scores zero
  x[is.na(x)] <- 0
  # Exclude the lowest score homework
  mean(x[-which.min(x)])
}
```

Time to take the gradebook and **grade the whole class** of multiple students.

```
url <- "https://tinyurl.com/gradeinput"
gradebook <- read.csv(url, row.names =1)
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
## student-7   89 100  74  87 100
## student-8   89 100  76  86 100
## student-9   86 100  77  88  77
## student-10  89  72  79  NA  76
## student-11  82  66  78  84 100
## student-12 100  70  75  92 100
## student-13  89 100  76 100  80
## student-14  85 100  77  89  76
## student-15  85  65  76  89  NA
## student-16  92 100  74  89  77
## student-17  88  63 100  86  78
## student-18  91  NA 100  87 100
## student-19  91  68  75  86  79
## student-20  91  68  76  88  76
```

```
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?

Let's take a look at the results from Q1 to find the top-scoring student.

```
results <- apply(gradebook,1,grade)
sort(results, decreasing = TRUE)
```

```
## student-18  student-7  student-8 student-13  student-1 student-12 student-16
##      94.50      94.00      93.75      92.25      91.75      91.75      89.50
##  student-6  student-5 student-17  student-9 student-14 student-11  student-3
##      89.00      88.25      88.00      87.75      87.75      86.00      84.25
##  student-4 student-19 student-20  student-2 student-10 student-15
##      84.25      82.75      82.75      82.50      79.00      78.75
```

Alternatively, I can do. . .

```
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 homework did students all around score the lowest on? Let's do a by column analysis. . . and calculate a summary stat for each column of the gradebook. Let's take a look at the averages, and the medians.

```
# Average homework score
hw.av <- apply(gradebook, 2,mean, na.rm = TRUE)
hw.av
```

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

```
which.min(hw.av)
```

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

```
hw.med <- apply(gradebook,2,median, na.rm = TRUE)
hw.med
```

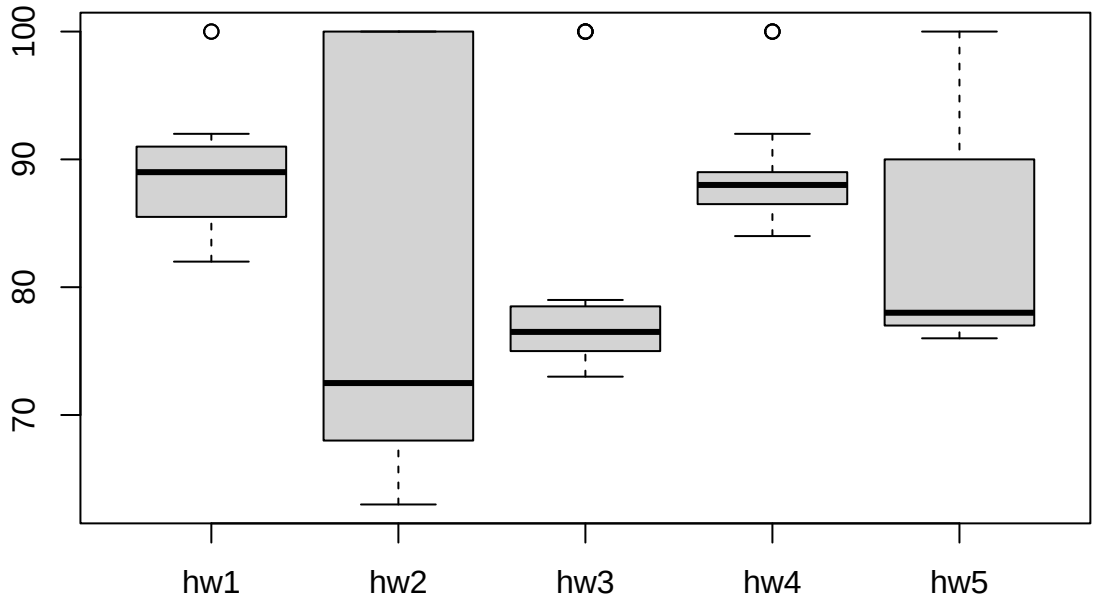
```
##  hw1  hw2  hw3  hw4  hw5
## 89.0 72.5 76.5 88.0 78.0
```

```
which.min(hw.med)
```

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

Interesting. The mean and medians return different results. Let's take a look at the data visually.

```
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)?

```
results
```

```
##  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
```

```
gradebook[is.na(gradebook)] <- 0
cor(results, gradebook$hw1)
```

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

Let's do it on all the homeworks...

```
correlations <- apply(gradebook,2,cor,x=results)
correlations
```

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

```
max(correlations)
```

```
## [1] 0.6325982
```

Q5. Make sure you save your Rmarkdown document and can click the "Knit" button to generatea PDF foramt report without errors. Finally, submit your PDF to gradescope.