Lab 6

Karen Guerrero

Write a function to grade some student homework.

Start by adding 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 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]

```
#view student1 homework scores
student1
```

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

```
#obtain the average of the student1 homework scores by using the following function: 'mean
mean(student1)
```

```
[1] 98.75
```

To find the position of the smalleset value (i.e. min) value in our vector we can use 'which.min()'

```
#obtain the min value of the student1 homework scores by using the following function: 'wh
which.min(student1)
```

```
[1] 8
```

```
#view student1 homework scores
student1
```

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

```
#view specifically student1's homework 8 score
student1[8]
```

```
[1] 90
```

Provides the min value for student1

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

```
[1] 90
```

Code is used to get everything but the min value

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

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

This will get us the mean value of the vector, minus the lowest score

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

```
[1] 100
```

```
#View student2 homework scores
student2
```

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

Not a great way of grading because this will just NOT include the NA value rather than inputing it as a 0

```
#will give average of student2 homework assignments (using mean() function) and will exclu
mean(student2, na.rm = TRUE)
```

```
[1] 91
```

```
#obtaining average of student2 scores without including the second homework assignment sco
mean(student2[-2])
```

```
[1] 91
```

```
#View student 3 homework scores
student3
```

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

```
#obtaining average of student3 scores and excluding NA values
mean(student3, na.rm = TRUE)
```

```
[1] 90
```

Map/Change NA values to zero. How do I identify NA values? Use the 'is.na() function.

```
student2
```

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

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

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

Identify the NA values Changing NA to 0 value

```
#assign student2 scores to x
x <- student2
#view student2 scores
x
```

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

```
#changing all the NA values in student2 homework scores to 0
x[is.na(x)] <- 0
#View student2 scores with NA values changed to 0
x
```

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

```
#obtain the average of the student2 homework scores with 0 replacing the NA values
mean(x)
```

```
[1] 79.625
```

```
#assign student3 scores to x
x <- student3
#changing all the NA values in student3 homework scores to 0
x[is.na(x)] <- 0
#obtain the average of the student3 homework scores with 0 replacing the NA values, minus
mean(x[-which.min(x)])
```

```
[1] 12.85714
```

##Creating a function

Will take the working snippet and make it a function

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

Use the function created

```r
grade(student1)
```

```
[1] 100
```

```r
grade(student2)
```

```
[1] 91
```

```r
grade(student3)
```

```
[1] 12.85714
```

Take gradebook and grade the whole class of multiple students

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

```r
#assigning data fro url link to the name url
url <- "https://tinyurl.com/gradeinput"
gradebook <- read.csv(url, row.names = 1)
```

```r
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? [3pts]

```r
#sorting the grades from decrease -> increasing to identify the student with the highest g
results <- apply(gradebook, 1, grade)
sort (results)
```

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

```r
#change the sorting order to increase -> decrease
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 
```

```r
#will tell us exactly who received the highest score
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? [2pts]

we want to calculate a summary stat for each assignment (column) of the gradebook

```
#First caluclate average
hw.ave <- apply(gradebook, 2, mean, na.rm = TRUE)

#figure out which assignment had the lowest average
which.min(hw.ave)
```

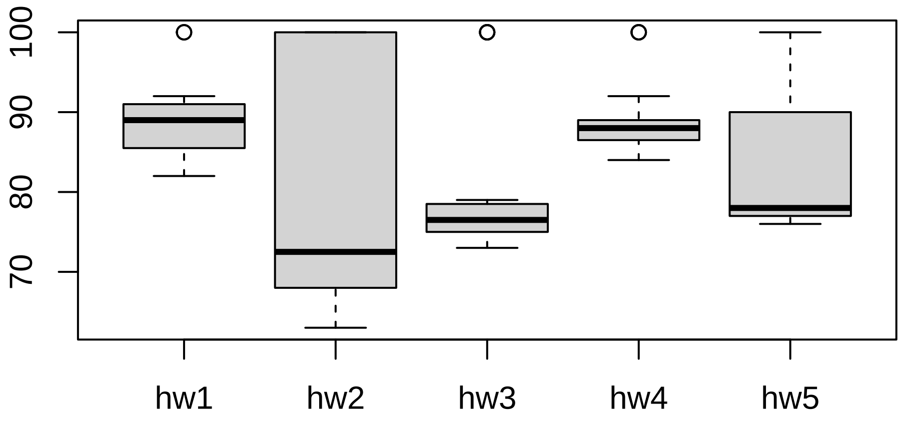
```
hw3
  3
```

```
#First caluclate median
hw.med <- apply(gradebook, 2, median, na.rm = TRUE)
#figure out which assignment had the lowest median
which.min(hw.med)
```

```
hw2
  2
```

Plot the median and mean to see the differences

```
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)? [1pt]

```
#set all the NA values to 0
gradebook [ is.na(gradebook)] <- 0
#obtain cor value of specific hw assigment (column)
cor(results, gradebook$hw5)
```

```
[1] 0.6325982
```

```
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 Quarto document and can click the "Render" (or Rmarkdown"Knit") button to generate a PDF foramt report without errors. Finally, submit your PDF to gradescope. [1pt]