Class06

Yipeng Li

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)
```

drop the lowest score

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

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

```
mean(student1)
```

```
[1] 100
```

change NA to 0

```
#drop the NA from student2 and calculate the mean
student2[is.na(student2)] <- 0
student2
```

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

```
mean(student2)
```

```
[1] 79.625
```

```
#drop all the NA from student 3 and calculate the mean
student3[is.na(student3)] <- 0
student3
```

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

```
mean(student3)
```

```
[1] 11.25
```

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]

```
#my grade function
grade <- function(student) {
  student[is.na(student)] <- 0
  student <- student[-which.min(student)]
  return(mean(student))
}
```

read the csv document

```
data <- read.csv("https://tinyurl.com/gradeinput", row.names = 1)
data
```

```
           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 my grade function

```
students_grade <- apply(data, 1, grade)
students_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]

```
Top_student <- students_grade[which.max(students_grade)]
Top_student
```

```
student-18
      94.5
```

Q3. From your analysis of the gradebook, which homework was toughest on students (i.e. obtained the lowest scores overall? [2pts] #The overall grade for HW1 to HW5

Calculate the mean & sum for homework

```
homework_mean <- apply(data, 2, mean, na.rm = TRUE)
homework_mean
```

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

```
homework_sum <- apply(data, 2, sum, na.rm = TRUE)
homework_sum
```

```
 hw1  hw2  hw3  hw4  hw5
1780 1456 1616 1703 1585
```

So the lowest score

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

```
 hw3
80.8
```

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

```
 hw2
1456
```

Based on the answer, HW2 and HW3 are not having a good score. HW3 have the lowest average, and HW2 have the lowest overall score.

> 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]

Modify the data to fit the requirement

```
masked.gradebook <- data
masked.gradebook[is.na(masked.gradebook)] <- 0
masked.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   0  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   0  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   0
student-16  92 100  74  89  77
student-17  88  63 100  86  78
student-18  91   0 100  87 100
student-19  91  68  75  86  79
student-20  91  68  76  88  76
```

Then calculate the correlation for hw1

```
cor(students_grade, masked.gradebook$hw1)
```

```
[1] 0.4250204
```

Apply to all

```
correlation <- apply(masked.gradebook, 2, cor, x=students_grade)
correlation
```

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

The highest correlation is

```
correlation[which.max(correlation)]
```

```
      hw5
0.6325982
```

Q5. Make sure you save your Quarto document and can click the "Render" (or Rmark- down"Knit") button to generate a PDF foramt report without errors. Finally, submit your PDF to gradescope. [1pt]