Class 6 R Functions

Joshua Mac A17016359

2025-01-27

> 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]

```
# 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)
```

Following the steps from class: Simple problem Step 1

```
student1 <- c(100, 100, 100, 100, 100, 100, 100, 90)

mean(student1)
```

```
[1] 98.75
```

Drop the lowest score

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

```
[1] 8
```

```
student1[-8]
```

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

```
## First working snippet
mean(student1[-which.min(student1)])
```

```
[1] 100
```

What about the other students in the class?

```
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
mean(student2, na.rm=T)
```

```
[1] 91
```

```
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
mean(student3, na.rm=T)
```

```
[1] 90
```

Do these make sense? No, because student 3 did only 1 homework and got an average hw grade of 90

```
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
x<-student2
is.na(x)
```

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

```
which(is.na(x))
```

```
[1] 2
```

Identified NA elements, now want to mask the NA, replacing them with 0s and dropping the lowest

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

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

```
mean(x)
```

```
[1] 79.625
```

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

```
[1] 91
```

Now for student 3, a more extreme case

```
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
y<-student3
y[is.na(y)]<-0
mean(y[-which.min(y)])
```

```
[1] 12.85714
```

##Step 2: Function making time for grade()! 3 Parts to a Function: - Name: name - Input: arguments - Body: working snippet

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

Now we have a function we can run!

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```r
grade(student3)
```

```
[1] 12.85714
```

Let's annotate the function

```r
#' Calculate average score for vector of students, dropping the lowest.
#'Missing values will be treated as zero
#' @param x A numeric vector of HW scores
#'
#' @returns Average score
#' @export
#'
#' @examples
#' student<-c(100, 90, 50, 80)
#' grade(student)
#'
grade <- function(x) {
  #Mask NA with zero
  #Treat missing as zero
  x[is.na(x)]<-0
  #Exclude the lowest score from mean
  mean(x[-which.min(x)])
}
```

Now, to a larger "gradebook" CSV format: "https://tinyurl.com/gradeinput"

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

```r
#Applying to data with margin a function
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?

Apply function, see result using code not eyeballs

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

```
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 
```

```
which.max(results)
```

```
student-18 
        18 
```

So the answer is student 18!

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

```
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
```

```
ave.scores<- apply(gradebook, 2, mean, na.rm=T)
ave.scores
```

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

```
which.min(ave.scores)
```

```
hw3
  3
```

```
median.scores<-apply(gradebook, 2, median, na.rm=T)
median.scores
```

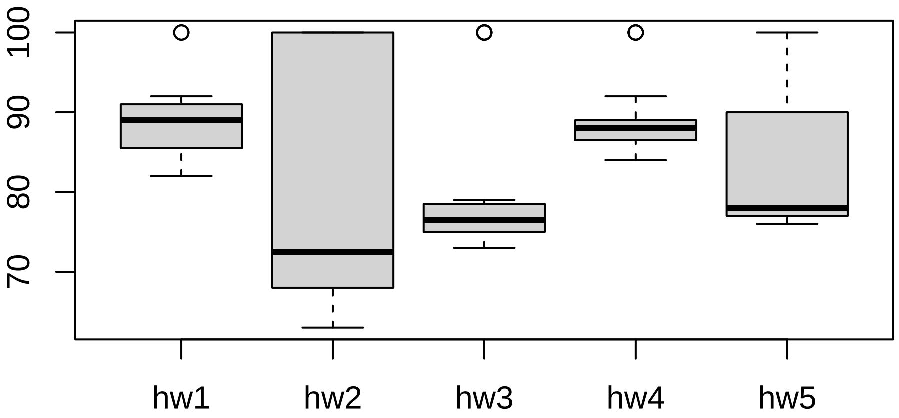
```
 hw1  hw2  hw3  hw4  hw5
89.0 72.5 76.5 88.0 78.0
```

```
which.min(median.scores)
```

```
hw2
  2
```

So homework 3 was, on average, the most challenging for students, while hw 2 had the lowest median score with just a tiny bit higher average score.

```
boxplot(gradebook)
```



Looking at the boxplot of the gradebook, **hw2** (even though average for hw3 was smaller) was the most challenging for students overall.

> 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.