R Functions Lab (Week 5)

Vivian Cai

2/6/2022

Writing a Function to Grade Students' Homework

##1) Start with simple vectors

```
# 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]

```
# identify the lowest score first
min(student1)
```

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

```
min(student2)
```

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

```
min(student3) # min identifies NA as the lowest, that saves work for me!
```

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

```
# finding the location of the smallest value
which.min(student1)
```

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

```
which.min(student2)
```

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

```
which.min(student3)
```

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

```
# which.mean doesn't identify NA as the lowest, bummer
```

```
# exploring base function mean and na.rm argument
mean(student1)
```

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

```
mean(student2)
```

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

```
mean(student2, na.rm = TRUE)
```

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

```
# na.rm works but is not going to be a fair way for us to calculate grades, since any student with more
```

```
# exploring is.na()
is.na(student1)
```

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

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

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

```
# We can use the boolean to select and swap out all the NAs
student2[is.na(student2)]
```

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

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

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

```
student3[is.na(student3)] <- 0
student3
```

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

```
# looking pretty good
```

```
# now to drop the lowest scores, going back to which.min
which.min(student3)
```

```
## [1] 2
```

```
# now that I have NAs changed to 0, I can find the lowest score with no problem!
```

```
# Time to write the function!
grade <- function(x) {
  x[is.na(x)] <- 0
  mean(x[-which.min(x)])
}

# reload example vectors
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)

# try out the function
grade(student1)
```

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

```
grade(student2)
```

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

```
grade(student3)
```

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

Great! That was fun!

“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”

```
# current function
#' Mean Value for Vector (Dropping the lowest value)
#' Calculates the average score after exluding the lowest score. The missing/NA values will be treated 
#'
#' @param x Numeric vector of a set of values/scores
#'
#' @return Average value/score
#' @export
#'
#' @examples
#' student <- c(25, NA, NA, 60, 80, 95)
#‘ grade(student)
```

```r
grade <- function(x) {
  # swapping out missing homework values to 0
  x[is.na(x)] <- 0
  # get the mean after dropping the lowest score
  mean(x[-which.min(x)])
}
```

Time to try out the gradebook!

```r
# reading out the csv file from a link and store it as the gradebook
url <- "https://tinyurl.com/gradeinput"
gradebook <- read.csv(url)
gradebook
```

```
##              X hw1 hw2 hw3 hw4 hw5
## 1    student-1 100  73 100  88  79
## 2    student-2  85  64  78  89  78
## 3    student-3  83  69  77 100  77
## 4    student-4  88  NA  73 100  76
## 5    student-5  88 100  75  86  79
## 6    student-6  89  78 100  89  77
## 7    student-7  89 100  74  87 100
## 8    student-8  89 100  76  86 100
## 9    student-9  86 100  77  88  77
## 10  student-10  89  72  79  NA  76
## 11  student-11  82  66  78  84 100
## 12  student-12 100  70  75  92 100
## 13  student-13  89 100  76 100  80
## 14  student-14  85 100  77  89  76
## 15  student-15  85  65  76  89  NA
## 16  student-16  92 100  74  89  77
## 17  student-17  88  63 100  86  78
## 18  student-18  91  NA 100  87 100
## 19  student-19  91  68  75  86  79
## 20  student-20  91  68  76  88  76
```

```r
# changing row name column
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
```

```
# use the apply() function on the gradebook array
apply(gradebook, 1, grade) # here 1 is for row since we want to calculated across the rows, if we want
```

```
##  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]

```
mean.score <- apply(gradebook, 1, grade)
# two ways!
which.max(mean.score)
```

```
## student-18
##         18
```

```
sort(mean.score, 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
```

Q3. From your analysis of the gradebook, which homework was toughest on students (i.e. obtained the lowest scores overall? [2pts]

```
# calculate summary stats of the gradebook
head(gradebook) # inspect first
```

```
##           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
```

```r
apply(gradebook, 2, mean) # not quite
```

```
##  hw1  hw2  hw3  hw4  hw5
## 89.0   NA 80.8   NA   NA
```

```r
ave <- apply(gradebook, 2, mean, na.rm = TRUE)
which.min(ave)
```

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

```r
# let's try sum
apply(gradebook, 2, sum) # not quite
```

```
##  hw1  hw2  hw3  hw4  hw5
## 1780   NA 1616   NA   NA
```

```r
sum <- apply(gradebook, 2, sum, na.rm = TRUE)
which.min(sum)
```

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

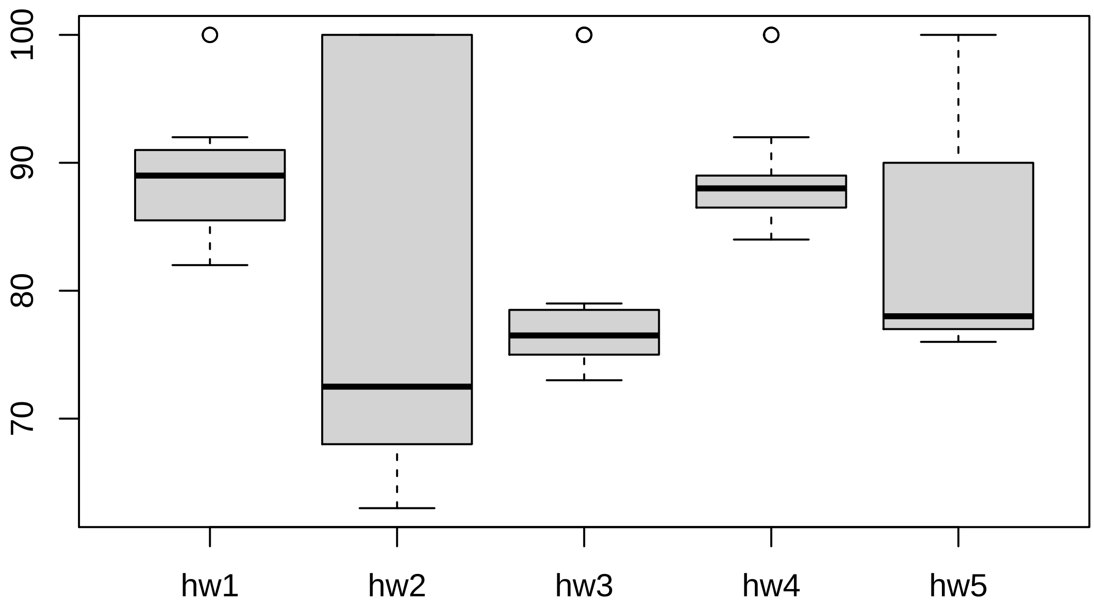
```r
# median should also work
apply(gradebook, 2, median, na.rm = TRUE)
```

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

```r
med <- apply(gradebook, 2, median,na.rm = TRUE)
which.min(sum)
```

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

```r
# plotting things is always good
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]

```
# ?cor
head(mean.score)
```

```
## student-1 student-2 student-3 student-4 student-5 student-6
##     91.75     82.50     84.25     84.25     88.25     89.00
```

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

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

```
cor(mean.score, gradebook$hw5)
```

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

```
apply(gradebook, 2, cor, mean.score)
```

```
##       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 foramt report without errors. Finally, submit your PDF to gradescope. [1pt]