class6 functions

Rachel Kraft

2/6/2022

introduction to R functions/writing them

> 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

```
# 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)
```

- find lowest score

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

#tells which vector element is the lowest
which.min(student1)
```

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

- drop lowest score

```
# returns everything but the 8th element from student 1
student1[-8]
```

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

- use answer from which.min() to return all other vector elements besides lowest, then can take the mean

```
#first working part of code for question 1
mean(student1[-which.min(student1)])
```

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

will this work for other students?

can try na.rm=TRUE for mean, but not a good approach

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

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

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

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

mask (replace) the NAs with zero

- first find NA elements

```
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
```

now have identified NA elements that we want to mask and replace with 0

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

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

```
mean(x)
```

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

still need to drop lowest score

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

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

try for student 3 - here is the working code

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

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

##Now make the function

take code from above and turn into a function need: - name 'grade()' - input arguments, vector of student scores - body (our code)

select 'Code > Extract Function'

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

test

```
grade(student1)
```

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

```
grade(student2)
```

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

```
grade(student3)
```

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

Comments to explain this code for any future users: - insert Roxygen skeleton

```
#' Calculate the average score, after dropping the lowest, for a vector of student scores. Missing scor
#'
#' @param x This is a numeric vector of student homework scores
#'
#' @return Average score
#' @export
#'
#' @examples
#' student <- c(100, 80, NA, 92)
#' grade(student)
#'

grade <- function(x) {
  #Treat missing scores as zero, instead of NA
    x[is.na(x)] <- 0
    #Drop the lowest score and take the mean of all other       values
    mean(x[-which.min(x)])
  }
```

Now, use the grade() function on whole class data from question 1. use read.csv() to read the csv gradebook file https://tinyurl.com/gradeinput

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

use apply() function to work with data frame inputs

```
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?

Run the apply() function and save the results

```
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
```

Find the student with the highest average

```
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?

```
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
```

check by average per column

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

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

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

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

- hw 3 is the worst

check by median per column

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

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

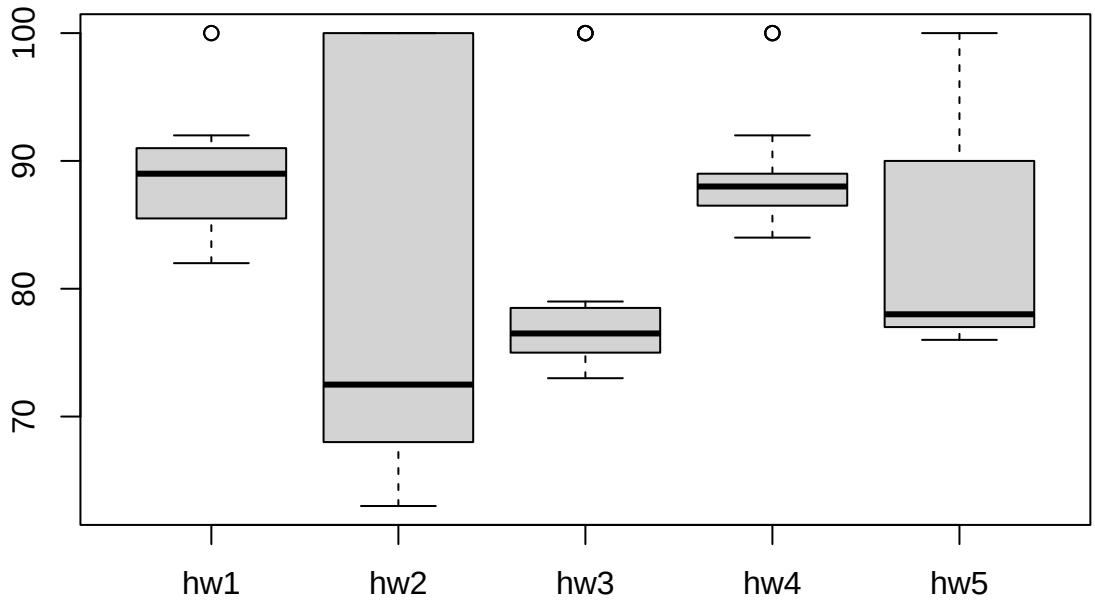
```
which.min(med.scores)
```

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

- hw 2 is the worst

best way? use a boxplot

```
boxplot(gradebook)
```



Q4) Optional

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

Knit the document to make a PDF