Week 5: R Functions Lab

Divya Shetty (A15390408)

2/5/2022

R Functions: How To Write Your Own

QUESTIONS

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 adequately 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)
```

Step 1: Find the average of given vector of scores to determine a student's grade

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

mean(student1)
```

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

Step 2: Determine the lowest score

```
#returns index of the lowest score
which.min(student1)
```

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

Step 3: Remove the lowest score from the vector

```
#returns vector of scores without the lowest score
student1[-8]
```

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

Step 4: NOW determine the average score when lowest score is dropped

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

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

Step 5: Consider other students (edge cases). How should NA scores be dealt with?

Approach #1: remove NA scores

```
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)

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

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

```
mean(student3, na.rm = TRUE)
```

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

HOWEVER, this approach is an unfair grading system…

Approach #2: change NA scores to 0

```
#find the index of NA values
which(is.na(student2))
```

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

```
which(is.na(student3))
```

```
## [1] 2 3 4 5 6 7 8
```

```
#replace the index values to 0
student2[which(is.na(student2))] <- 0
student3[which(is.na(student3))] <- 0

#NOW find the mean grade
mean(student2)
```

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

```
mean(student3)
```

```
## [1] 11.25
```

NICE! Use approach #2 and put it together with the code snippet from Step 4 to calculate the grade…

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

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

Step 6: Use the above code and make it into a function called **grade()**

```
#' Returns the avg grade when lowest score dropped. Missing values treated as 0.
#'
#' @param x numeric vector of student's scores
#' @return avg score
grade <- function(x) {
  #consider missing scores as 0
  x[is.na(x)] <- 0
  #drop the lowest score when calculating avg score
  mean(x[-which.min((x))])
}
```

Testing code…

```
grade(student1)
```

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

```
grade(student2)
```

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

```
grade(student3)
```

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

Step 7: Show that grade() works with given gradebook!

```
#get score(s) from link to gradebook
url <- "https://tinyurl.com/gradeinput"
gradebook <- read.csv(url, row.names = 1)

#use grades() for every student in gradebook
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?

Use grade() on the gradebook and find the maximum of the results. This is the top scoring student.

```
#store results of gradebook
results <- apply(gradebook, 1, grade)

#determine the max avg grade
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?

Step 1: Determine how gradebook is organized

```
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
```

Step 2: For each homework, should the mean score or the median score be used? Identify the homework with the lowest score.

Approach #1: using the mean

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

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

```
which.min(avg_scores)
```

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

Approach #2: using the median

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

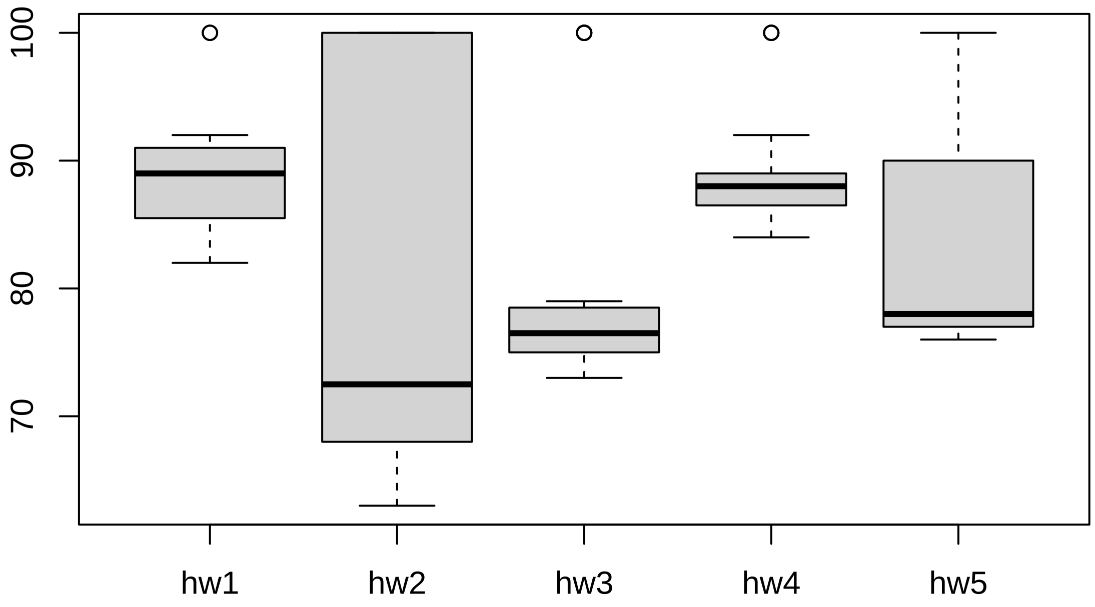
```
##  hw1  hw2  hw3  hw4  hw5 
## 89.0 72.5 76.5 88.0 78.0
```

```
which.min(med_scores)
```

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

Use a boxplot to visualize which statistic is better to use!

```
boxplot(gradebook)
```



The median score for a homework best represents the homework that was toughest on students. Therefore, the hardest homework was **hw2**.

Q4. From your analysis of the gradebook, which homework was most predictive of overall score (i.e. highest correlation with average grade score)?

What is the correlation between the average score for each student and the homework scores? Step 1: replace NA scores with a 0

```
masked <- gradebook
masked[is.na(masked)] <- 0

#check that all NA vvalues have been changed
masked
```

```
##            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
```

Step 2: use the cor() function to calculate the correlation for each hw and determine the highest correlation

```
#correlation for all hw
cors <- apply(masked, 2, cor, x = results)
cors
```

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

```
#find the max correlation
which.max(cors)
```

```
## hw5
##   5
```

The homework that was most predictive of the overall score was **hw5**.

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.

[completed]