

class06

Anjali Hamada

2024-01-28

Please note that the classwork walkthrough is shown at the bottom of this document. The work just below is Q1-Q5.

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> [3pts]

```
#' Calculate the average score for a vector of student scores dropping the lowest score. M
# '
# ' @param x A numeric vector of homework scores
# '
# ' @return Average score
# ' @export
# '
# ' @examples
# ' student <- c(100, NA, 90, 97)
# ' grade(student)
# '
grade <- function(x){
  # mask NA with zero
  # treat missing values as zero
  x[is.na(x)] <- 0
  # exclude lowest score from mean
  mean(x[-which.min(x)])
}
```

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

Q2

Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook? [3pts]

```
results <- apply(gradebook, 1, grade)
#sort(results, decreasing = TRUE)
# OR you can do...
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)? [2pts]

```
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
```

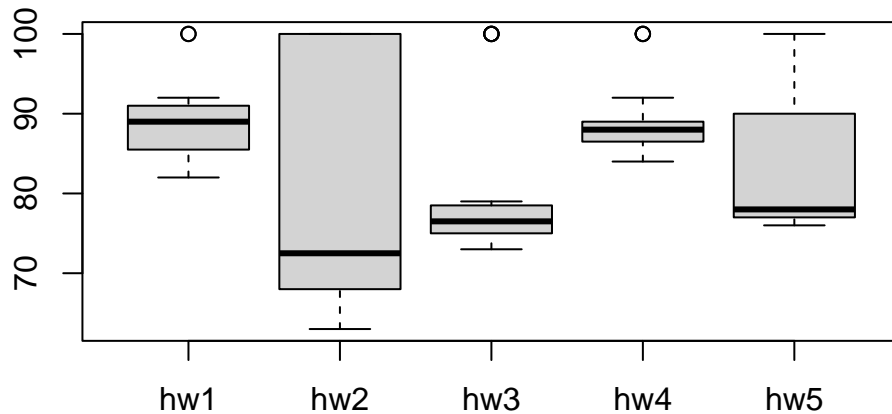
```
med.scores <- apply(gradebook, 2, median, na.rm=TRUE)
med.scores
```

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

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

```
hw2  
2
```

```
boxplot(gradebook)
```



The median/boxplot shows that HW2 had the lowest scores, with over half the class scoring below any of the other assignments. This makes sense, since the mean can't take into account skewed distributions.

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]

```
# Are the final results (i.e. average score for each student) correlated with the results  
masked.gradebook <- gradebook  
masked.gradebook[is.na(masked.gradebook)] <- 0  
cor(results, masked.gradebook$hw5)
```

```
[1] 0.6325982
```

```
apply(masked.gradebook, 2, cor, x=results)
```

	hw1	hw2	hw3	hw4	hw5
	0.4250204	0.1767780	0.3042561	0.3810884	0.6325982

HW5 had the highest correlation of 0.6325982.

Q5

Make sure you save your Quarto document and can click the “Render” (or Rmarkdown ”Knit”) button to generate a PDF format report without errors. Finally, submit your PDF to grade-scope. [1pt]

On it, boss!

In-Class Work

```
# 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)
```

```
# straight forward mean()
# student1 <- c(100, 100, 100, 100, 100, 100, 100, 90)
mean(student1)
```

```
[1] 98.75
```

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

```
[1] 8
```

```
# returns everything but the 8th element of the vector
student1[-8]
```

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

```
# first working snippet! calculates avg grade with lowest  
# score dropped  
mean(student1[-which.min(student1)])
```

```
[1] 100
```

```
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
```

```
# na.rm works but both s2 and s3 end with As!
```

```
# replace all NA values with 0  
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)  
x <- student2  
  
# is.na(x)  
# which(is.na(x))  
# x[which(is.na(x))]  
# mean(x[-which(is.na(x))])  
# all useful solutions, but we like the one below better  
x[is.na(x)] <- 0  
x
```

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

```
mean(x)
```

```
[1] 79.625
```

```
# final working snippet!
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
```

```
# using Code > Extract Function to turn this into a function
gradeClass <- function(x){
  x[is.na(x)] <- 0
  mean(x[-which.min(x)])
}
```

```
gradeClass(student1)
```

```
[1] 100
```

```
gradeClass(student2)
```

```
[1] 91
```

```
gradeClass(student3)
```

```
[1] 12.85714
```