

Lab6

Norman Lee

2025-10-22

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> (<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)
```

Drop lowest score for student 1; first we need to find the lowest score

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

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

result tells us it is the 8th value in the vector, but what is the actual value?

```
student1[8]
```

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

drop the lowest score from mean calculation

```
student1[-8]
```

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

we can use the answer from `which.min()` to return all other elements of the vector

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

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

how about student 2 and 3 we can use na.rm=true but it is not a fair 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
```

Another approach is to replace na with 0

```
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 we identified na and we can replace them with 0

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

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

student3

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

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

turn the snippet into function every function has 3 parts 1. a name, in this case 'grade()' 2. input 3. The body, snippet of our code

```
## calculate average homework score that drops the lowest score
## missing values will be treated as 0
## @param x a numeric vector of homework scores
##
## @returns average score
## @export
##
## @examples
## student <- c(100, NA, 90, 80)
## grade(student)
grade <- function(x) {
  # mark na (missing value) as 0
  x[is.na(x)] <- 0
  # exclude lowest score
  mean(x[-which.min(x)])
}
```

now we can use our function on class data "<https://tinyurl.com/gradeinput>" (<https://tinyurl.com/gradeinput>)"

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

```
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? [3pts]

run apply()and save the results

```
results <- apply(gradebook, 1, grade)
results
```

```
## 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
```

```
sort(results)
```

```
## student-15 student-10 student-2 student-19 student-20 student-3 student-4
##      78.75      79.00      82.50      82.75      82.75      84.25      84.25
## student-11 student-9 student-14 student-17 student-5 student-6 student-16
##      86.00      87.75      87.75      88.00      88.25      89.00      89.50
## student-1 student-12 student-13 student-8 student-7 student-18
##      91.75      91.75      92.25      93.75      94.00      94.50
```

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

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

Student 18 has the highest score

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

```
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
```

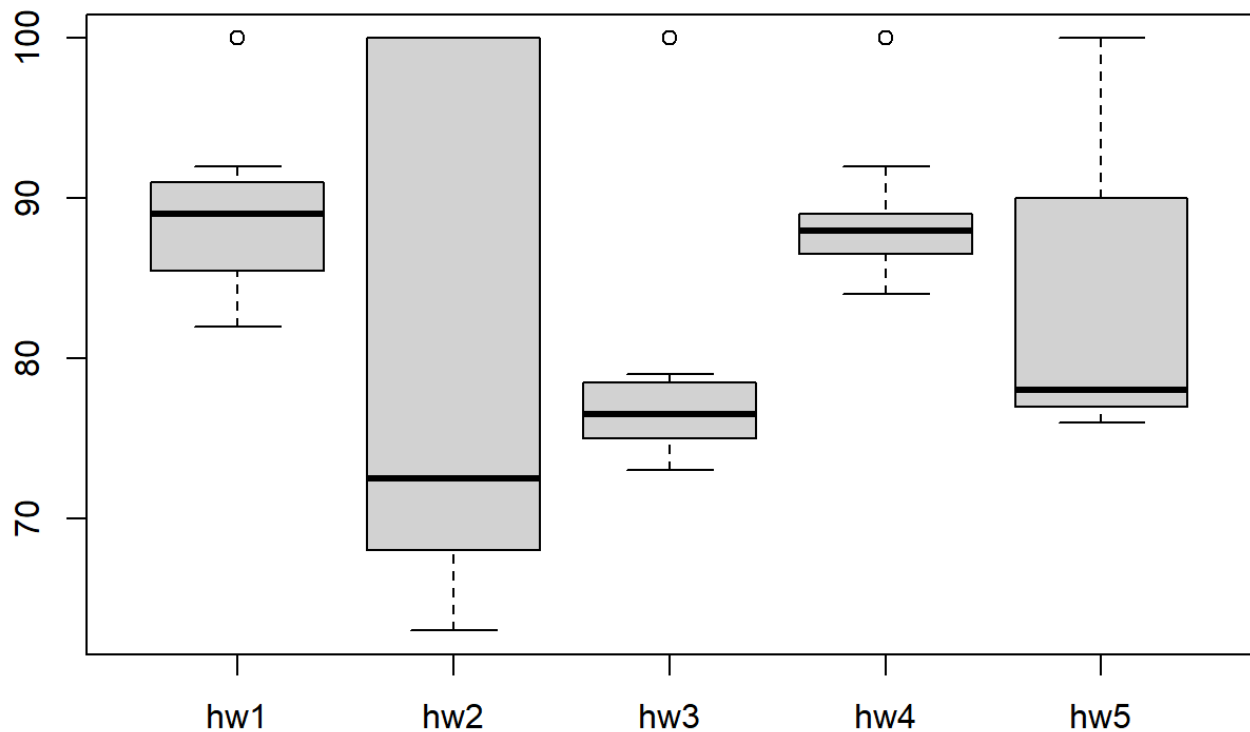
```
avg.scores <- apply(gradebook, 2, mean, na.rm=TRUE)
avg.scores
```

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

```
which.min(avg.scores)
```

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

```
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]

```
masked.gradebook <- gradebook  
masked.gradebook[is.na(masked.gradebook)] <- 0  
cor(results, masked.gradebook$hw5)
```

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

```
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 has the highest value and has the highest correlation with average grade score

Q5. Make sure you save your Quarto document and can click the “Render” (or Rmark- down”Knit”) button to generate a PDF format report without errors. Finally, submit your PDF to gradescope. [1pt]