

Lab6

Ashlyn Murphy

Lab 6

```
# example input 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)
```

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>”

```
student1
```

```
[1] 100 100 100 100 100 100 100 90
```

The regular avg will be returned by the `mean()` function

```
# calculates the average aka the overall grade
mean(student1)
```

```
[1] 98.75
```

find the minimum value to find the dropped score

```
min(student1)
```

```
[1] 90
```

to find the position of the minimum value in the vector, use `which.min()`

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

```
[1] 8
```

```
student1[8]
```

```
[1] 90
```

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

```
[1] 90
```

returns the min value for student 1

to get everything but the min value...

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

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

```
# the minus sign in front of the which.min() function returns everything but the result of
```

the take the mean of this function to get the grade

```
#first solution to drop the lowest value from a student score list  
mean(student1[-which.min(student1)])
```

```
[1] 100
```

```
#student2
```

```
student2
```

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

cant just use `na.rm = True` to deal with this NA value because it would just remove it and not count it as the lowest score – need to find a way to map the NA values to zero.

To identify NA values use the `is.na()` function

```
student2
```

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

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

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

```
student2[is.na(student2)]
```

```
[1] NA
```

set `x = student` to for easier code writing!

to map NA to zero, set instances when `student2` (or `x`) is NA as equal to zero

```
x <- student2  
x
```

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

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

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

then take the mean to get the true average of `student2` scores

```
mean(x)
```

```
[1] 79.625
```

student3

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

```
[1] 12.85714
```

function

use the working snippet to make a function

```
#' Calculate average scores for a vector of homework scores
#' dropping the lowest single score. Missing values will be treated
#' as zero score
#'
#' @param x Numeric vector of homework scores
#'
#' @return Average score
#' @export
#'
#' @examples
#' student <- c(100, NA, 90, 80)
#' grade(student)
#'
grade <- function(x) {
  # missing homework scores zero
  x[is.na(x)] <- 0
  # exclude lowest score homework
  mean(x[-which.min(x)])
}
```

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

now use the gradebook to grade the whole class of multiple students

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

```
#second argument is the margin which is 1 = rows 2 = columns
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	

Q2. Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook?

```
#find max score in gradebook
max(apply(gradebook, 1, grade))
```

```
[1] 94.5
```

```
#find where the max score occurs in the gradebook
which.max(apply(gradebook, 1, grade))
```

```
student-18
18
```

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

```
apply(gradebook, 2, grade)
```

	hw1	hw2	hw3	hw4	hw5
	89.36842	76.63158	81.21053	89.63158	83.42105

```
min(apply(gradebook, 2, grade))
```

```
[1] 76.63158
```

```
which.min(apply(gradebook, 2, grade))
```

```
hw2  
2
```

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

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

```
[1] 0.6325982
```

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

	hw1	hw2	hw3	hw4	hw5
	0.4250204	0.1767780	0.3042561	0.3810884	0.6325982

the most predictive homework is homework 5.

Q5. Make sure you save your Quarto document and can click the “Render” button to generate a PDF format report without errors. Finally, submit your PDF to gradescope.

Finally, make a pdf report.