

R Functions Lab (Class 06)

Jiayi Zhou (PID: A17856751)

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] We first paste the csv

```
csv_txt <- """ , "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'
```

```

gb <- read.csv(
  text = csv_txt,
  stringsAsFactors = FALSE,
  na.strings = c("NA")
)

names(gb)[1] <- "student"
# first column header was blank

```

We Define grade() per the policy

```

grade <- function(x) {
  # Define a function named grade that takes a numeric vector x of one student's scores.
  stopifnot(is.numeric(x)) #Fail if x isn't numeric
  n <- length(x)
  if (n < 2) {
    # if there's only one score, just return it.
    # If that single score is missing (NA), treat it as 0.
    return(ifelse(is.na(x), 0, x))
  }
  # Choose which single value to drop
  if (anyNA(x)) {
    drop_idx <- which(is.na(x))[1] } else {
    drop_idx <- which.min(x) # drop one NA if present
    # otherwise drop the minimum score
  }
  kept <- x[-drop_idx]
  # keep n-1 scores (everything except value chose to drop)
  kept[is.na(kept)] <- 0 # remaining NAs count as 0
  mean(kept)
}

```

Apply to every student (row-wise)

```

is_num <- sapply(gb, is.numeric)
grade_cols <- names(gb)[is_num]
G <- as.matrix(gb[, grade_cols]); storage.mode(G) <- "double"
gb$Overall <- apply(G, 1, grade)

```

Check if it works

```
head(gb)
```

	student	hw1	hw2	hw3	hw4	hw5	Overall
1	student-1	100	73	100	88	79	91.75
2	student-2	85	64	78	89	78	82.50
3	student-3	83	69	77	100	77	84.25
4	student-4	88	NA	73	100	76	84.25
5	student-5	88	100	75	86	79	88.25
6	student-6	89	78	100	89	77	89.00

Q2. Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook? [3pts]

```
gb[which.max(gb$Overall), c("student", "Overall")]
```

	student	Overall
18	student-18	94.5

This returns the row of gb with the highest Overall score (i.e., the top student) and shows only the student and Overall columns.

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

```
G_zeroNA <- as.matrix(gb[, c("hw1", "hw2", "hw3", "hw4", "hw5")])  
# Covert the homework columns to a matrix  
G_zeroNA[is.na(G_zeroNA)] <- 0  
# Treat NA as 0  
hw_means <- colMeans(G_zeroNA)  
# Find class average for each homework (column means)  
names(hw_means)[which.min(hw_means)] # homework name with the lowest average
```

```
[1] "hw2"
```

```
hw_means
```

	hw1	hw2	hw3	hw4	hw5
	89.00	72.80	80.80	85.15	79.25

So hw2 was toughest, it has the lowest class average (72.80).

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]

```
G0 <- gb[, c("hw1","hw2","hw3","hw4","hw5")]
G0[is.na(G0)] <- 0

# Compute correlations with Overall
cors <- sapply(G0, function(col)
  cor(as.numeric(col), gb$Overall, use = "complete.obs")
)

cors
```

	hw1	hw2	hw3	hw4	hw5
	0.4250204	0.1767780	0.3042561	0.3810884	0.6325982

```
names(cors)[which.max(cors)]
```

```
[1] "hw5"
```

So hw5 is most predictive of overall score ($r = 0.633$).

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