# Class 6

Benjie Miao (PID: 69026849)

## Q1

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 adquately 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].

```
# 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)

#' Calculate average scores for a vector of homework score
#' dropping the lowest score. NA will be considered as 0
#'
#' @param x Numeric vector of score
#'
#' @return Average score
#' @export
#'
#' @examples
#'      student1 <- c(100, 100, 100, 100, 100, 100, 100, 90)
#'      grade(student1)
grade <- function(x) {
  x[is.na(x)] <- 0 # view all the NA score as 0 score
  mean_minus_min <- (sum(x) - min(x)) / (length(x) - 1)
  # (score sum - min score) / (N-1)
  mean_minus_min
}
```

```
  grade(student1)
```

```
[1] 100
```

```
  grade(student2)
```

```
[1] 91
```

```
  grade(student3)
```

```
[1] 12.85714
```

## Q2

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

```
  url = "https://tinyurl.com/gradeinput"
  gradebook <- read.csv(url, row.names = 1)
  # gradebook
  results <- apply(gradebook, 1, grade)
  sort(results, decreasing = TRUE)
```

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

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

```
student-18
        18
```

Student 18 has the highest score.

## Q3

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

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

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

```
which.min(homework_mean)
```

```
hw3
  3
```

Homework 3 was toughest when we used the mean score as the criteria.

Or we can try using the median:

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

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

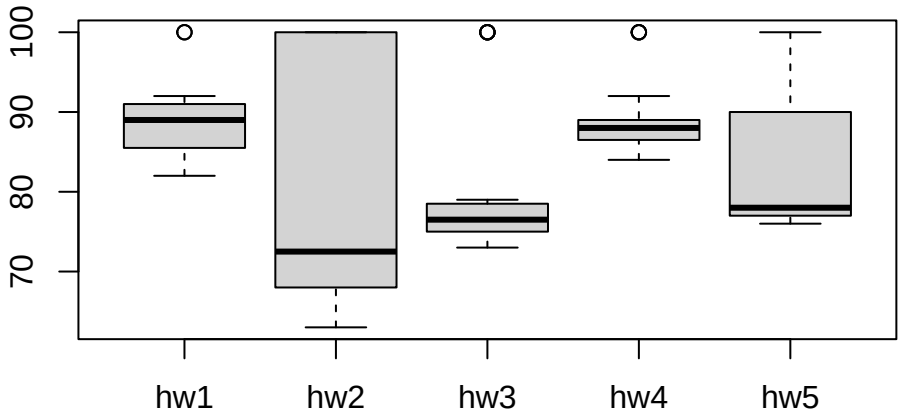
```
which.min(homework_median)
```

```
hw2
  2
```

Hw2 was the toughest when we used the median score.

We can also use boxplot to visualize the score of each hw:

```
boxplot(gradebook)
```




## Q4

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]

```
overall_score <- results # use the Q1 result

corr_with_overall_score <- function(x) {
  cor(x, overall_score, use="complete.obs") # "complete.obs" to handle NA
}

hw_score_corr <- apply(gradebook, 2, corr_with_overall_score)
hw_score_corr
```

```
       hw1         hw2         hw3         hw4         hw5
0.42502036  0.61142768  0.30425610 -0.09644108  0.60398041
```

```
which.max(hw_score_corr)
```

```
hw2
  2
```

Hw2 is the most predictive one, and hw5 is slightly worse but still good.

## Q5

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

Here it is.