

Lab 6 R Functions

Yushi Li (A15639705)

2/5/2022

```
# 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)

# defining function grade()
grade_pt <- function(x) {
  # set NA grades to 0 within each vector.
  x[is.na(x)] <- 0
  print(x)
  print(x[-which.min(x)])
  # calculating average grade while excluding the lowest score.
  overall <- mean(x[-which.min(x)])
  overall
}
grade_pt(student1)
```

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

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

```
grade_pt(student2)
```

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

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

```
grade_pt(student3)
```

```
## [1] 90  0  0  0  0  0  0  0
## [1] 90  0  0  0  0  0  0
```

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

```
# looks like it works!
```

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 grade of vector of scores dropping the lowest one.
#' Missing values are treated as zeros.
#'
#' @param x A numeric vector of scores.
#'
#' @return The average of the scores.
#' @export
#'
#' @examples
#' student <- c(100, NA, 90, 97, 80)
#' grade(student)

grade <- function(x) {
  # set NA grades to 0 within each vector
  x[is.na(x)] <- 0
  # calculating average grade while excluding the lowest score
  overall <- mean(x[-which.min(x)])
  overall
}

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

# fun grade() on gradebook
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]

```
# assign grading results to grades
grades <- apply(gradebook, 1, grade)
# find the maximum in grades
which.max(grades)
```

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

```
# looks like student-18 has the highest grade overall.
```

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

```
# method 1: using mean(), treating NAs as outliers (remove them).  
which.min(apply(gradebook, 2, mean, na.rm = TRUE))
```

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

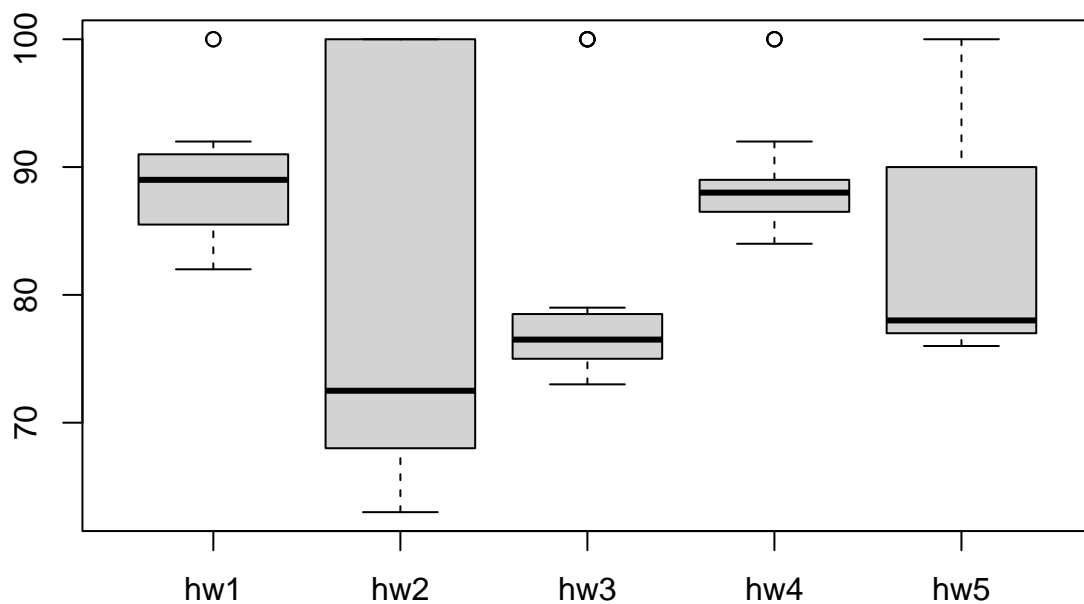
```
# based on the means, hw3 was the most challenging.
```

```
# method 2: using median(), again, treating NAs as outliers.  
which.min(apply(gradebook, 2, median, na.rm = TRUE))
```

```
## hw2  
## 2
```

```
# based on the medians, hw2 was the most challenging.
```

```
# boxplot of results  
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]

```
# assign all NA to 0 on gradebook, save as masked_gradebook
masked_gradebook <- gradebook
masked_gradebook[is.na(masked_gradebook)] <- 0
# assign the graded scores as results
results <- apply(gradebook, 1, grade)
# apply cor() to the processed gradebook
apply(masked_gradebook, 2, cor, x = results)
```

```
##          hw1          hw2          hw3          hw4          hw5
## 0.4250204 0.1767780 0.3042561 0.3810884 0.6325982
```

```
which.max(apply(masked_gradebook, 2, cor, x = results))
```

```
## hw5
##    5
```

```
# hw5 has the highest correlation with average grade.
```

Q5. Make sure you save your Rmarkdown document and can click the “Knit” button to generate a PDF foramt report without errors. Finally, submit your PDF to gradescope. [1pt]