Class 06: R Functions

Zichen “Cardiff” Jiang

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

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**]

`grade()` that takes in a vector of vector and return a vector of grades. All NAs in the grade book are replaced with 0. And for each student, the lowest score is dropped and mean is then calculated.

```r
grade_of_nested_vector <- function(vectorOfvector) {
  means = c()
  for(student in students) {
    # Map/ replace NA values to zero
    student_no_na <- student
    student_no_na[ is.na(student_no_na) ] <- 0

    # Exclude the lowest grade and calculate the mean
    mean <- mean( student_no_na[ -which.min(student_no_na) ] )
    means <- append(means, mean)
  }

  return(means)
}

student1 <- c(100, 100, 100, 100, 100, 100, 100, 90)
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
```

```r
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
students <- list(student1, student2, student3)

means <- grade_of_nested_vector(students)
means
```

```
[1] 100.00000  91.00000  12.85714
```

`grade_from_csv()` can read CSV format

```r
grade_from_csv <- function(filepath) {
  df <- read.csv(filepath, row.names = 1)
  means = c()
  for(i in 1:nrow(df)) {
    # Obtain the scores of each student in the dataframe
    student <- as.numeric(df[i,])

    # Map/ replace NA values to zero
    student_no_na <- student
    student_no_na[ is.na(student_no_na) ] <- 0

    # Exclude the lowest grade and calculate the mean
    mean <- mean( student_no_na[ -which.min(student_no_na) ] )
    means <- append(means, mean)
  }

  return(means)
}

means <- grade_from_csv('student_homework.csv')
means
```

```
 [1] 91.75 82.50 84.25 84.25 88.25 89.00 94.00 93.75 87.75 79.00 86.00 91.75
[13] 92.25 87.75 78.75 89.50 88.00 94.50 82.75 82.75
```

We can also use the `apply()` function to grade all the students in this class with out new `grade()` function.

The `apply()` function allows us to run any function over the rows or columns of a data.frame. Let's see how it works

```
grade <- function(student) {
  # Map/ replace NA values to zero
  student_no_na <- student
  student_no_na[ is.na(student_no_na) ] <- 0

  # Exclude the lowest grade and calculate the mean
  return(mean( student_no_na[ -which.min(student_no_na) ] ))
}

filepath = 'student_homework.csv'
gradebook <- read.csv(filepath, 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]

To use `apply()`,

```
filepath = 'student_homework.csv'
gradebook <- read.csv(filepath, row.names = 1)
which.max(apply(gradebook, 1, grade))
```

```
student-18
        18
```

Student 18

To write a new function

```
find_top_student <- function(filepath) {
  df <- read.csv(filepath, row.names = 1)
  means = c()
  for(i in 1:nrow(df)) {
    # Obtain the scores of each student in the dataframe
    student <- as.numeric(df[i,])

    # Map/ replace NA values to zero
    student_no_na <- student
    student_no_na[ is.na(student_no_na) ] <- 0

    # Exclude the lowest grade and calculate the mean
    mean <- mean( student_no_na[ -which.min(student_no_na) ] )
    means <- append(means, mean)
  }
  # Return the name of the student with the max mean
  # df["mean"] <- means
  # print(df)
  # top_student <- df[which.max(df$mean),]
  # return(top_student)
  return(which.max(means))
}
top_student <- find_top_student('student_homework.csv')
top_student
```

```
[1] 18
```

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

2nd HW

```
filepath = 'student_homework.csv'
df <- read.csv(filepath)
df[is.na(df)] <- 0
df_grades <- df[,-1]
toughest_hw <- which.min(colMeans(df_grades))
toughest_hw
```

```
hw2
  2
```