Class 6: R Functions

Aparajita Pranjal

4/21/23

In this class we will develop our own R functions to calculate average grades in a fictional class.

We will start with a simplified version of the problem, by calculating the average grade of one student.

Simplified version

```
# 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)
```

We are going to start by calculating the average score of the homework.

```
mean(student1)
```

```
[1] 98.75
```

To get the minimum score we can use `which.min` .

```
student1
```

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

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

```
[1] 8
```

The average of the first 7 homework scores:

```
mean(student1[1:7])
```

```
[1] 100
```

Another way to select the first 7 homework scores:

```
student1[1:7]
```

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

```
student1[-8]
```

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

Another way to drop the lowest score:

```
st1_lowest_score = student1[-which.min(student1)]
```

Then to find the mean of the scores after dropping the lowest:

```
mean(st1_lowest_score)
```

```
[1] 100
```

Now to calculate the average score for all the students we generalize the function. Starting with student 2:

```
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
st2_lowest_score = student2[-which.min(student2)]
st2_lowest_score
```

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

There is a way to calculate the mean dropping the missing values (or NA):

```r
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
mean(student2, na.rm = TRUE)
```

```
[1] 91
```

For student3:

```r
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
mean(student3, na.rm = TRUE)
```

```
[1] 90
```

We want to know the position of the NAs. So for student2 we can use the following:

```r
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
which(is.na(student2))
```

```
[1] 2
```

For student3:

```r
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
which(is.na(student3))
```

```
[1] 2 3 4 5 6 7 8
```

For considering the missing values, we can mask the NAs with zeros.

```r
student2[is.na(student2)] <- 0
student2[is.na(student2)]
```

```
numeric(0)
```

```r
student2
```

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

```
student3[is.na(student3)] <- 0
student3
```

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

```
mean(student3)
```

```
[1] 11.25
```

This is going to be our final working snippet of code for all students (with and without NA values).

```
student3[is.na(student3)] <- 0
student3_drop_lowest = student3[-which.min(student3)]
mean(student3_drop_lowest)
```

```
[1] 12.85714
```

Function grade()

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"

```
#' Calculate the average score for a vector of homework scores, dropping the lowest score,
#'
#' @param x A numeric vector of homework scores
#'
#' @return The average value of homework scores
#' @export
#'
#' @examples
grade <- function(x) {
  # Mask NA values with zero
  x[is.na(x)] <- 0
  # Drop lowest score
```

```
  x_drop_lowest = x[-which.min(x)]
  mean(x_drop_lowest)
}
```

Let's apply the function:

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Now applying it to the sample gradebook form this URL:

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

```
          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
```

Let's apply the function to the gradebook using `apply` and running it by **rows** using `MARGIN=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? **Student 18**

```
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? **HW2**

```
gradebook[is.na(gradebook)] <- 0
apply(gradebook, 2, mean)
```

```
  hw1   hw2   hw3   hw4   hw5
89.00 72.80 80.80 85.15 79.25
```

Instead of assigning NAs to zeros, if we remove the NAs then the toughest homework is **HW3** according to the mean. If we try using the median instead we get **HW2** as the hardest.

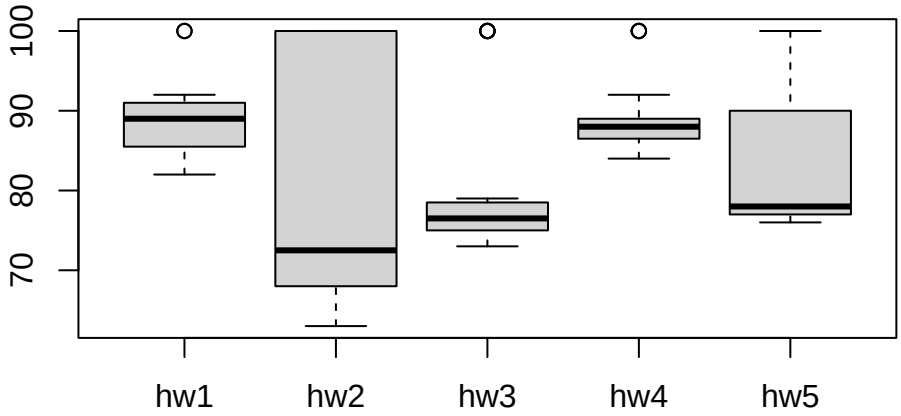
```
gradebook <- read.csv(URL, row.names = 1)
apply(gradebook, 2, mean, na.rm = TRUE)
```

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

```
apply(gradebook, 2, median, na.rm = TRUE)
```

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

```
boxplot(gradebook)
```



Q4. From your analysis of the gradebook, which homework was most predictive of overall score (i.e. highest correlation with average grade score)? **HW5**

```
overall_grades = apply(gradebook, 1, grade)
#For individual column
cor(gradebook$hw1,overall_grades)
```

```
[1] 0.4250204
```

```
#For all columns
gradebook[is.na(gradebook)] <- 0
correlation_values = apply(gradebook, 2, cor, y = overall_grades)
which.max(correlation_values)
```

```
hw5
  5
```