Class 06: R Functions

Jessica Diaz-Vigil

4/21/23

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

Simplified Version of Grade Averages

We will start with a simplified version of the problem: Calculating the average grade of one student

First we will bring in the values for the students:

```
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 for one student.

```
mean(student1)
```

```
[1] 98.75
```

We now need to find the lowest score in order to drop the minimum score.

`which.mean( )` can determine which score is the lowest by giving the position it is located.

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

```
[1] 8
```

Now we know which homework we will drop. Next, we need to take a new average.

```
student1[-8]
```

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

```
mean(student1[-8])
```

```
[1] 100
```

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

```
[1] 100
```

To simplify this, we need to find a better way to select all the homework scores except for the lowest.

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

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

We can get the mean of all the scores now with the exception of the lowest score.

```
mean(student1[-which.min(student1)])
```

```
[1] 100
```

A simpler way of writing this code is as follows (assigning a variable):

```
student1_drop_lowest = student1[-which.min(student1)]
student1_drop_lowest
```

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

```r
mean(student1_drop_lowest)
```

```
[1] 100
```

New variables can then be created for each student.

```r
student2_drop_lowest = student2[-which.min(student2)]
student3_drop_lowest = student3[-which.min(student3)]
student2_drop_lowest
```

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

```r
student3_drop_lowest
```

```
[1] NA NA NA NA NA NA NA
```

The problem with this is that NA is not being dropped, the lowest numerical score is being dropped.

To fix this, we can remove the NA.

```r
mean(student2, na.rm = TRUE)
```

```
[1] 91
```

This looks good for Student2 since there was only one NA, though Student3 has many more NA. To see which homework scores are NA, we can do the following:

```r
is.na(student3)
```

```
[1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```r
which(is.na(student3))
```

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

We now need to remove one NA score. To do this we can make a missing score equal to 0 so it will be removed when taking the mean.

```r
which(is.na(student3))
```

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

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

```
numeric(0)
```

```r
student3
```

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

Now we need to take the mean again, though we need to remember to remove the lowest value.

```r
mean(student3)
```

```
[1] 11.25
```

```r
student3_drop_lowest = student3[-which.min(student3)]
mean(student3_drop_lowest)
```

```
[1] 12.85714
```

We will do this for Student2 now, though the mean will not change since we removed the NA already earlier.

```r
student2[which(is.na(student2))] <- 0
student2_drop_lowest = student2[-which.min(student2)]
mean(student2_drop_lowest)
```

```
[1] 91
```

We have created the basis of a function though we need to apply them to all the students now instead of only one at a time. We can use `x` to represent each student.

```
# x[which(is.na(x))] <- 0
# x_drop_lowest = x[-which.min(x)]
# mean(x_drop_lowest)
```

Creating a Function

Now we need to write this as a function.

```
grade <- function(x)
  {
    x[which(is.na(x))] <- 0
    x_drop_lowest = x[-which.min(x)]
    mean(x_drop_lowest)
                        }
```

Now we can use this function.

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Application to a Gradebook

Lets apply our function to a grade book from this URL: ”https://tinyurl.com/gradeinput”

```
url <- 'https://tinyurl.com/gradeinput'
read.csv(url, row.names = 1)
```

```
           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
```

```
gradebook <- read.csv(url, row.names = 1)
```

Now we can apply the function to the grade book.

```
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 
```

We can write everything as a function now and document it!

```
#' Grade Calculator: Calculation of the average scores
#' for a vector of homework scores while dropping the
#' lowest score, considering NA values as 0
#'
```

```
#' @param x A numeric vector of scores
#'
#' @return The average value of homework scores
#' @export
#'
#' @examples
#' student <- c(100, 50, NA)
#' grade(student)
#'
grade <- function(x)
  {
    x[which(is.na(x))] <- 0
    x_drop_lowest = x[-which.min(x)]
    mean(x_drop_lowest)
                        }
```

Questions

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

```
max(apply(gradebook, 1, grade))
```

```
[1] 94.5
```

The maximum score is 94.5

```
which.max(apply(gradebook, 1, grade))
```

```
student-18
        18
```

The student with the highest score was student 18

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

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

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

The lowest average was from hw3 when NA values were removed.

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

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

The lowest average was from hw2 when the NA values were counted as 0.

Q4. From your analysis of the gradebook, which homework was most predictive of overall score (i.e. highest correlation with average grade score)? [**1pt**]

```r
url <- 'https://tinyurl.com/gradeinput'
read.csv(url, row.names = 1)
```

```
           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
```

```r
gradebook <- read.csv(url, row.names = 1)
overall_grades = apply(gradebook, 1, grade)
overall_grades
```

```
 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 
```

```r
cor(gradebook$hw1, overall_grades)
```

```
[1] 0.4250204
```

```r
cor(gradebook$hw2, overall_grades)
```

```
[1] NA
```

```r
cor(gradebook$hw3, overall_grades)
```

```
[1] 0.3042561
```

```r
cor(gradebook$hw4, overall_grades)
```

```
[1] NA
```

```r
cor(gradebook$hw5, overall_grades)
```

```
[1] NA
```

The highest correlation is the second homework, though this method took more effort.

To simplify we could do as follows:

```r
url <- 'https://tinyurl.com/gradeinput'
read.csv(url, row.names = 1)
```

```
           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
```

```r
gradebook <- read.csv(url, row.names = 1)
gradebook[is.na(gradebook)] <- 0
apply(gradebook, 2, cor, overall_grades)
```

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

```r
which.max(apply(gradebook, 2, cor, overall_grades))
```

```
hw5 
  5 
```

This method is much simpler and we can see that the highest correlation is still hw2.

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