Lab_06

Victor Yu

In this class we will work througgh the process of developing our own function for calculating average gradfes for fictional students in al fictional class.

We will start with a simplified version of the problem. Grade some vectors of student scores.

We want to drop the lowest score and get the average

```
# 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)
```

```
mean (student1)
```

```
[1] 98.75
```

We can find the smallest value with the `min()` function

```
min (student1)
```

```
[1] 90
```

```
#Returns the position of the lowest value in the vector
which.min(student1)
```

```
[1] 8
```

```
#Tells us the lowest value in the vector. However the student1[] tells us the position. Ne
student1[which.min(student1)]
```

```
[1] 90
```

```r
#Mean of vector of Student 1 dropping the lowest score!
mean(student1[-which.min(student1)])
```

```
[1] 100
```

```r
#Returns logical of every position in vector if NA is present. NA = true
is.na(student2)
```

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

```r
#First line sets NA value to 0.
student2[ is.na(student2) ] <- 0
student2[-which.min(student2)]
```

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

```r
mean(student2[-which.min(student2)])
```

```
[1] 91
```

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

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

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

```
[1] 12.85714
```

Q1: Write a function grade() to determine the overall grade from a vector of student assignments.

```r
#Our Function! It works for each individual student :0
grade <- function (x) {
  x[ is.na(x) ] <- 0
  x[-which.min(x)]
  mean(x[-which.min(x)])
}
grade(student1)
```

```
[1] 100
```

```r
grade(student2)
```

```
[1] 91
```

```r
grade(student3)
```

```
[1] 12.85714
```

We have our working snippet of code!

```r
#' Calculate the average score for a vector of scores, dropping the lowest score
#' N/A are treated 0 values
#' @param x A numeric vector of scores
#'
#' @return Average value returned
#' @export
#'
#' @examples
#' student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
#' grade(student 2)
grade <- function (x) {
  #NA values are treated as zero / = 0 now
  x[ is.na(x) ] <- 0
  #determines the position of the lowest value of the vector
  x[-which.min(x)]
  #finds the average of the vector excluding the lowest vector
  mean(x[-which.min(x)])
}
```

Explain of our function code ~~~. But also our answer to function #1

```
#Url stores the data set to varaible url. Gradebook now holds the dataset of interest for
#row.names = 1, removes
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
```

Our apply function! It finally returns the grade of each student in the gradebook.

```
# Apply function works apply (x, margin, fun)
# x= our dataset, whether it is a list, vector, or matrix
# margin = either row or column we are working with or both
# fun = function we want to apply to data set
 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?

We can use the sort function to organize the results we have been given. Bruh, we can also just use which.max / min function. Dr. Grant too smart :o

```
#sort (x, decreasing,)
#decreasing = FALSE (lowest to greatest) = TRUE (greatest to lowest)
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
```

Answer for Q2: The top scoring student is student-18 with an average of 94.50

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

```
#class example. Top 3 lines set N/A in data = 0. which allows us to truly find the hardest
mask <- gradebook
mask[is.na(mask)] <- 0
mask
```

```
           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   0  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   0  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   0
student-16  92 100  74  89  77
student-17  88  63 100  86  78
student-18  91   0 100  87 100
```

```
student-19  91  68  75  86  79
student-20  91  68  76  88  76
```

```
apply (mask,2, mean)
```

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

```
which.min (apply(mask, 2, mean))
```

```
hw2 
  2 
```

Answer to Q3: HW 2 was the toughest homework!

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]

Here we are going to look at the correlation of each Homework results (i.e. the columns in the gradebook) with the overall grade of students from the course (in the `results` object obtained from using our `grade()` function).

```
#Ok, so cor() functions need to applicable to both vectors. Either vector same or go throu
#NA are showing up again. we need to set NA zero and remove it.
naremoval <- gradebook
naremoval[is.na(naremoval)] <- 0
naremoval
```

```
           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   0  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   0  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   0
student-16  92 100  74  89  77
student-17  88  63 100  86  78
student-18  91   0 100  87 100
student-19  91  68  75  86  79
student-20  91  68  76  88  76
```

```
#Our X is naremoval since we are looking at the entire data set now with N/A as 0
#MARGIN = 2 since we want to look columns / homework to correlate the score
#Cor is our function since we are trying to correlate results and homework score
# Y = results because that is the vector in comparison. We want to retrieve values from ou
apply(naremoval, 2, cor, y = results )
```

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

```
which.max(apply (naremoval, 2, cor, y = results))
```

```
hw5 
  5 
```

Answer to Q4: The strongest correlation with homework to predictive score is homework 5!