R Functions Lab (Class 06)

Sofia Lanaspa, PID:A17105313

Background

Introducing *R functions* and how to write our own R function

Every function needs at least 3 things: - a name - input arguments - body (what it does)

```
add <- function(x,y) {
  x+y
}
```

Using add() function

```
add(2,4)
```

```
[1] 6
```

R allows functions to iterate over a vector:

```
add_vector <- c(10,1,1,10)
add(add_vector,1)
```

```
[1] 11  2  2 11
```

```
#Function will show error if there is only 1 argument, such as add(10), or more than 2, such
mean(c(10,10,NA),na.rm=T)
```

```
[1] 10
```

Lab Sheet Questions

Question 1

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]

Given vectors show the score of the students: student 1 - drop 90 student 2 - drop NA student 3 - has dropped

```
# 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)
```

Before writing a function write code that works

```
student1
```

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

```
#Avg student1
mean(student1)
```

```
[1] 98.75
```

```
#Avg student2 - gives NA because one NA present in vector
mean(student2)
```

```
[1] NA
```

```
#To ignore the NA and get a mean value do this:
mean(student2,na.rm=T)
```

```
[1] 91
```

```
#Avg student3 - not fair for their avg to be 90 despite dropping
mean(student3,na.rm=T)
```

```
[1] 90
```

Use more complex code

```
#Find the minimum score AND where it is on the vector
min(student3)
```

```
[1] NA
```

```
which.min(student3)
```

```
[1] 1
```

Code below explained: get the vector for student1, take the min value, index to subtract that value from the student1 vector, take mean of vector to get avg score

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

```
[1] 100
```

We still need to deal with NA values, we need to make them equal 0

```
#save student2 vector in x
x <- student2
#see which values are NA
is.na(x)
```

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

```
#make NA values equal 0
x[is.na(x)] <- 0
#take the mean with NA values as 0
x[ is.na(x) ] <- 0
mean(x[-(which.min(x))])
```

```
[1] 91
```

Now lets write the grade() function :)

```
grade <- function(x){
  #finds NAs and makes them 0
  x[ is.na(x) ] <- 0

  #drops min value and finds mean
  mean(x[-(which.min(x))])
}
```

Using the grade() function

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Get full gradebook

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

Using the apply function on the : apply(dataset, margin, function) To use apply() function on this gradebook dataset I need to decide whether I want to apply the grade() function over the rows (1) or columns (2) of the gradebook.

```
ans <- apply(gradebook, 1, grade)
ans
```

```
 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
```

Question 2

Q2: Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook? [3pts]

- Get the scores of the students
- Find the max score using which.max() function

```
# Which student got the highest score
which.max(ans)
```

```
student-18
        18
```

```
# What was the highest score itself
ans[which.max(ans)]
```

```
student-18
      94.5
```

Student 18 had the highest score of 94.5

Question 3

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

```
#Get the mean scores of each homework, change to 2 to get columns instead of rows, NA is true
avg.scores <- apply(gradebook, 2, mean, na.rm=T)
avg.scores
```

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

```
which.min(avg.scores)
```

```
hw3
  3
```

```
#use median instead of mean
med.scores <- apply(gradebook, 2, median, na.rm=T)
med.scores
```

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

```
which.min(med.scores)
```

```
hw2
  2
```

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

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

```
#we could also modify grade function to keep lowest score
grade2 <- function(x,drop.low=TRUE){
  #finds the NAs in 'x' and makes them 0
  x[ is.na(x) ] <- 0

  if(drop.low) {
    cat("Hello low")
    # Drops the lowest value and find mean
    mean(x[-(which.min(x))])
  }
  else {
    out <- mean(x)
    cat('No low')
  }
  return(out)
}

grade2(ans,drop.low=F)
```

```
No low
```

```
[1] 87.425
```

hw2 was the hardest since it had the lowest mean score of 72.5 (and lowest median score of 72.80)

Question 4

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]

```
#Are the final results (avg of each student) correlated with the scores of individual homewo

#Use cor() to calculate correlation
cor(ans,gradebook$hw1)
```

```
[1] 0.4250204
```

0.4250204 is not a very strong correlation

```
cor(ans,masked_gradebook$hw5)
```

```
[1] 0.6325982
```

Use apply() to the cor() function over the masked_gradebook and use the ans scores for the class *need to spell out extra ans argument for cors()

```
apply(masked_gradebook, 2, cor, y=ans)
```

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

hw5 had the greatest correlation with final grade

Question 5

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