Class06

Gregory Jordan

Table of contents

Question 1

```r
# 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)
```

I can use the mean() function to compute the average and min() function to find the smallest value

```r
#to get the average
mean(student1)
```

```
[1] 98.75
```

```r
#to find the minimum
min(student1)
```

```
[1] 90
```

I found the `which.min()` funciton, what does it do?

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

```
[1] 8
```

We can use the minus trick

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

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

Then we can take the average to get the average of all values except the minimum value

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

```
[1] 100
```

However, this will not work with student 2 (or student 3) because they each have NA values.

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

```
[1] NA
```

Cannot simply remove the NA values, because it will affect the calculations

```
#mean of student 3 is inaccurate if we just remove the NA values
student3
```

```
[1] 90 NA NA NA NA NA NA NA
```

```
mean(student3,na.rm = T)
```

```
[1] 90
```

However, we can convert NA values to 0. Which makes sense. If no assignment was given then the student gets 0 points

```
student1[is.na(student1)]<-0
student2[is.na(student2)]<-0
student3[is.na(student3)]<-0
#now our student lists have NA converted to a score of 0. Yay we can do math with them now
```

Write my snippet to combine my code chunks and make it applicable to all students

```
x<-student3
#make NA equal to 0
x[is.na(x)]<-0
#Get the mean
mean(x[-which.min(x)])
```

```
[1] 12.85714
```

Now I can turn this into a function!

```
#This function can take in any list of student scores and it will convert all NA values to
grade <- function(x){
  x[is.na(x)]<-0
  mean(x[-which.min(x)])
}
```

Question 2

```
#Now we will work with a gradebook of multiple students and use our previously made grade
#use read.csv to read in a csv file
#read in the csv gradebook file we will work with and store it in the gradebook variable
gradebook <- read.csv("https://tinyurl.com/gradeinput", 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
```

Now time to use the **apply()** function. We can apply the grade function we made prior to our gradebook now.

```
#apply applies a function to every element in an object.
# the second argument is 1 to apply over rows or we could have used 2 to apply over column
results<-apply(gradebook,1,grade)
results
```

```
 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
```

Now let's see which student scored the highest using **which.max()** and let's see what they scored

```
#use which.max to get position of which student scored highest
#position of highest scoring student was student 18 so then use 18 in brackets to get the
max_student<-which.max(apply(gradebook,1,grade))
max_student_position<-apply(gradebook,1,grade)[18]
#cat allows us to concatenate and print things together in a nice clean way
cat("The highest scoring student is student",max_student,"\nand they scored",max_student_p
```

```
The highest scoring student is student 18
and they scored 94.5
```

Question 3

```
#get the sum of all of the homeworks removing NA values
gradebook_colums<-apply(gradebook,2,sum,na.rm=TRUE)
#get which one of these homeworks has the lowest score
min_homework<-which.min(apply(gradebook,2,sum,na.rm=TRUE))
#print it out nice and neat using `cat()`
cat("Homework",min_homework,"is the hardest homework")
```

```
Homework 2 is the hardest homework
```

Question 4

```r
#let's do a correlation to figure out which HW is the most predictive of overall score. te
cor(gradebook$hw1,results)
```

```
[1] 0.4250204
```

let's make it easier though by using `apply()` of the `cor()` function over gradebook

```r
mask <- gradebook
mask[is.na(mask)] <- 0
#stored gradebook with NA switched to 0 as mask

#applied cor function over mask using 2 instead of 1 because we want to apply over columns
apply(mask,2,cor,y=results)
```

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

```r
cat("The most predictive homework of overall score was Homework",which.max(apply(mask,2,co
```

```
The most predictive homework of overall score was Homework 5
```