Class 6: R Functions Lab

Angela Bartolo (PID:A15932451)

Input vectors

```r
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)
```

> 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"

Average of student score:

```r
mean(student1)
```

```
[1] 98.75
```

Remove NA:

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

```
[1] 91
```

Set NA equal to zero:

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

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

This will change the student2 vector so a temporary variable should be used, in this case x.

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

```
[1] 11.25
```

Finally we want to drop the lowest score before calculating the mean. I can use the minus sign with which.min to exclude the min value in the vector

```
h <- student1
which.min(h)
```

```
[1] 8
```

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

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

Now I need to put this all back together to make our working snippet:

```
x<- student3
x
```

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

```
# Replace NA values with 0
x[is.na(x)] <-0
x
```

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

```r
# Exclude the lowest score and calculate the mean
mean(x[-which.min(x)])
```

```
[1] 12.85714
```

This is my working snippet that will be turned into a function called 'grade()'

All functions in R have at least 3 things: - **Name**, in our case "grade" - Input **arguments**, in our case the students - **Body**, this is our working snippet

```r
grade <- function(x){
# Replace NA values with 0
x[is.na(x)] <-0
# Exclude the lowest score and calculate the mean
mean(x[-which.min(x)])}
```

Can I use the function now?

```r
grade(student3)
```

```
[1] 12.85714
```

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

Read a gradebook from online:

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

```
           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
```

We can now use the ‘apply()’ function to grade all the students in this class with our new ‘grade()’ function.

The ‘apply()’ function allows us to run any function over with the rows or columns of a data.frame. Let’s see how it works

```
apply(hw, 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 
```

```
student_grades <- apply(hw, 1, grade)
student_grades[which.max(student_grades)]
```

```
student-18 
      94.5 
```

Student-18 had the highest score with a score of 94.5.

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

```
avg.scores <- apply(hw,2, mean, na.rm=T)
which.min( avg.scores )
```

```
hw3
  3
```

```
tot.scores <- apply(hw,2, sum, na.rm=T)
which.min( tot.scores )
```

```
hw2
  2
```

```
avg.scores
```

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

```
tot.scores
```

```
 hw1  hw2  hw3  hw4  hw5
1780 1456 1616 1703 1585
```

HW 2 was the toughest on students since the total score, the sum of the scores, was the lowest.

> 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)?

```
hw$hw1
```

```
 [1] 100  85  83  88  88  89  89  89  86  89  82 100  89  85  85  92  88  91  91
[20]  91
```

```
student_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
```

```
cor(hw$hw1, student_grades)
```

```
[1] 0.4250204
```

```
cor(hw$hw3, student_grades)
```

```
[1] 0.3042561
```

If I try on hw2, I get NA as there are missing homeworks (i.e NA values)

```
cor(hw$hw2, student_grades)
```

```
[1] NA
```

I will mask all NA values to zero

```
mask <- hw
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
```

```
cor(mask$hw2, student_grades)
```

```
[1] 0.176778
```

We can use the ‘appl()’ function here on the columns of hw (i.e. the individual homeworks) and pass it the overall scores for the class (in my ‘student_grades’ object as an extra argument).

```
apply(mask, 2, cor, y=student_grades)
```

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

Hw 5 was most predictive of overall score.