class06

Kaitlyn Madriaga, PID:A17217752

```
#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)
```

> 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

We can use the "mean()" function to calculate the average

```
mean(student1)
```

```
[1] 98.75
```

This doesn't work with student 2, because they have NA. To remove NA, we use na.rm.

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

```
[1] 91
```

For student 3, we can replace the missed assignment NA values with a score of 0.

We can do this by using the 'is.na()' function:

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

We create a temporary variable 'x' to store our data in case we mess up:

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

```
[1] 11.25
```

Finally, we want to drop the lowest score before calculating th mean. This is equivalent to allowing the student to drop their worst assignment score. We can do this by using 'which.min()' to find the lowest score and remove it from the list.

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

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

Now, we can put them all together to make our working snippet:

```
x <- student3

#Map/Replace NA values to zero
x[is.na(x)] <- 0
#Exclude the lowest score and calculate the mean
mean(x[-which.min(x)])
```

```
[1] 12.85714
```

This is our working snippet that we can turn into a function called 'grade()'

All functions in R have at least 3 things: - **Name**, in our case "grade" - Input **arguments**, student1, etc. - **Body**, this is our working snippet

```
grade <- function(x) {
  #Map/Replace NA values to zero
  x[is.na(x)] <- 0

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

Can I use the function now?

```
grade(student1)
```

```
[1] 100
```

Read a gradebook from online:

```
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
```

The function 'apply()' allows us to take any function and apply it to a dataframe.

apply(data, margin = 1 (rows) or 2 (columns), function)

We can use it to grade all students in the class with our 'grade()' function

to look at student grades:

```
ans <- apply(hw, 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
```

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

```
ans[which.max(ans)]
```

```
student-18
      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 = TRUE)
which.min(avg.scores)
```

```
hw3
  3
```

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

```
hw2
  2
```

```
tot.scores
```

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

```
avg.scores
```

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

hw2 seems to be the toughest on students

> 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
```

```
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
```

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

```
[1] 0.4250204
```

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

```
[1] 0.3042561
```

If I try hw2, it will return NA because there are missing homeworks/NA values in the dataset. We need to 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
```

We can now find correlation values for all hws:

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

```
[1] 0.176778
```

```
  cor(mask$hw4, ans)
```

```
[1] 0.3810884
```

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

```
[1] 0.6325982
```

to look at all correlation values for all hw:

```r
apply(mask, 2, cor, y = ans)
```

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

hw5 has the highest correlation value