Class06: R Functions

Isabella Franco

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).

> 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]

```
# 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)
```

We can use the `mean()` function the average for a given student vector.

```
mean(student1)
```

```
[1] 98.75
```

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

```
[1] 91
```

We can replace the missed assignment NA values wih a score of zero. How do I do this?

We can use the `is.na()` function to help?

```
student3
```

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

```
is.na(student3)
```

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

I can make these values be anything I want it is time to work with new temp object(that I will call `x` so I don't screw up my original objects

```
x<-student3
x
```

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

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

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

Now that we have assigned NA=0, we can get our mean

MEAN

```r
mean(x)
```

```
[1] 11.25
```

Finally, we want to drop the lowest score before calculating the mean. This is equivalent to allowing the student to drop their lowest assignment score. **Mean with dropped lowest score**

```r
x<-student1
x
```

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

```r
which.min(x)
```

```
[1] 8
```

```r
mean(x[-8])
```

```
[1] 100
```

Now i need to put this all back togehter to make oue working snippet:

```r
x<-student1
x
```

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

```r
# 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] 100
```

Cool! This is my working snippet that I 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 snipe above.

```
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 this 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
```

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

The `apply()` functions allows us to run any function over the rows or columns of a data.frame. let's see how it works: - apply(data, margin=1, Function)

```
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? [3pts]

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

```
student-18 
      94.5 
```

Student 18 was the top scoring student overall.

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

```
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
```

```
ave_score<-apply(hw,2,mean,na.rm=TRUE)
which.min(ave_score)
```

```
hw3
  3
```

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

```
hw2
  2
```

```
total_score
```

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

```r
ave_score
```

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

Homework 2 is the lowest scoring Homework and therefore was likely the toughest on students.

> 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]

```r
hw$hw1
```

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

```r
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 
```

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

```
[1] 0.4250204
```

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

```
[1] 0.3042561
```

If I try on HW2, I get NA because there is a missing homework assignment(s).

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

```
[1] NA
```

Because of this, 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$hw5, ans)
```

```
[1] 0.6325982
```

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

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

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

Homework 5 is the most predictive of the overall score.