Class 06: R Functions

Yufei (A16222438)

All about functions in R

Functions are the way we get stuff done in R. We call a function to read data, compute stuff, plot stuff, etc.

R makes writing functions accessable but we should always start by trying to get a working snippet of code before we write our function.

Today's lab

We will grade a whole class of student assignments. We will always try to start with a simplified version of the problem.

```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)
```

If we want to average, we can use the `mean()` function:

```r
mean(student1)
```

```
[1] 98.75
```

Let's be nice instructors and drop the lowest score so the answer here should be 100.

We can use `min()` function to find the lowest value

```r
min(student1)
```

```
[1] 90
```

We found the `which.min()` function that may be useful here. Let's try how it works:

```r
student1
```

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

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

```
[1] 8
```

We can find the lowest score by

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

```
[1] 90
```

And remove the lowest score element from the vector by adding a - syntax

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

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

Then calculate the new average after removing the lowest score

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

```
[1] 100
```

Check if it works for student2

```r
student2
```

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

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

```
[1] NA
```

Where is the problem

```r
mean(student2)
```

```
[1] NA
```

It is the `mean()` with NA input that failed to calculate the mean value. Remove the NA value by na.rm.

```r
student2
```

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

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

```
[1] 100  NA  90  90  90  90  97
```

```r
mean(student2[-which.min(student2)], na.rm = TRUE)
```

```
[1] 92.83333
```

Thisis still not what we want. We need to remove NA instead of 80.

Test on student3

```r
student3
```

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

```r
mean(student3, na.rm = TRUE)
```

```
[1] 90
```

Still, we don't want to remove all the NA. But we want to simplify vector names before further troubleshooting.

```r
x <- student2
```

We want to overwrite NA value with zero. Google and Claude suggests we can use `is.na()` function.

```r
x
```

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

```r
is.na(x)
```

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

```r
x[is.na(x)]
```

```
[1] NA
```

We can use logicals to index a vector

```r
y<- 1:5
y
```

```
[1] 1 2 3 4 5
```

```r
y>3
```

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

```r
# find the index of items that are greater than 3
y[y>3]
```

```
[1] 4 5
```

```
# give those items new value by <- sign
y[y>3] <- 100
y
```

```
[1]   1   2   3 100 100
```

We can do the same thing to x

```
# Give NA item value 0.
x[is.na(x)] <- 0
x
```

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

Calculate the score for student2

```
# Give NA item value 0.
x[is.na(x)] <- 0
x
```

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

```
# Drop the lowest value, and calculate mean
mean(x[-which.min(x)])
```

```
[1] 91
```

This is my working snippet of code that solves the problem for all my student inputs. Calculate the score for student3

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

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

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

```
[1] 12.85714
```

We can turn this into a function.

> **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]

```r
grade <- function (x) {
  # Mask NA value to 0
  x[is.na(x)] <- 0
  # Drop lowest score and get the mean
  mean(x[-which.min(x)])
}
```

Try to use this function

```r
grade(student1)
```

```
[1] 100
```

```r
grade(student2)
```

```
[1] 91
```

```r
grade(student3)
```

```
[1] 12.85714
```

We need to read from gradebook

```r
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
```

We can use `apply()` function to apply function to our data.

```
#apply(x, MARGIN, FUN...). Margin = 1 indicates to operate by row, 2 indicates column
overallGrade <- apply(gradebook, 1, grade)
overallGrade
```

```
 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?

```
which.max(overallGrade)
```

```
student-18 
        18 
```

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

```
mask = gradebook
mask[is.na(mask)] <- 0
#by taking mean, excluding NA values
hw.ave <- apply(gradebook, 2, mean, na.rm=T)
hw.ave
```

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

```
which.min(hw.ave)
```

```
hw3 
  3 
```

We can also take the sum using this

```
#We can also by taking sum
hw.sum <- apply(gradebook, 2, sum, na.rm = T)
hw.sum
```

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

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)?

```
#apply cor(x, y) function, x=mask, y=overallGrade
hw.cor = apply(mask, 2, cor, y=overallGrade)
hw.cor
```

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

```
which.max(hw.cor)
```

```
hw5 
  5 
```