Lab 6: R Functions

Nicholas Do (PID: A15053002)

10/14/2021

R functions

In today's class we are going to write a function together that grades some students' work.

> **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)
```

Let's start with student1 and find their average score.

```
mean(student1)
```

```
## [1] 98.75
```

But we want to drop the lowest score... We could try the **min()** function

```
min(student1)
```

```
## [1] 90
```

The **which.min()** function looks more useful:

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

```
## [1] 90
```

Then, we calculate the mean with the remaining values.

```
gradeVector <- student1[-minVal]
gradeVector
```

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

```
average <- mean(gradeVector)
average
```

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

#For student2

```
student2
```

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

However, it gives NA if a student has a missing homework

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

```
## [1] NA
```

We need to remove the NA elements of the vector

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

```
## [1] 91
```

```
#mean(student2[ -which.min(student2) ], na.rm = TRUE)
#Not what we want, drops 80 and not the NA
```

Let's look at student3

```
student3
```

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

One new idea/approach is we could replace the NA (missing homeworks) with zero.

Let's try with student2

```
student2
```

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

```
naIndex <- which(is.na(student2))
naIndex
```

```
## [1] 2
```

```
student2[naIndex] <- 0
student2
```

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

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

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

Lets try with student3:

```
student3
```

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

```
temp <- student3
naVector<- which(is.na(student3))
naVector
```

```
## [1] 2 3 4 5 6 7 8
```

```
temp[naVector] <- 0
temp
```

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

```
temp <- temp[-which.min(temp)]
temp
```

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

```
mean(temp)
```

```
## [1] 12.85714
```

```
grades <- function(x) {
  #Make sure our scores are all numbers
  currentGrades <- as.numeric(x)
  currentGrades[ which(is.na(currentGrades)) ] <- 0
  average <- mean(currentGrades[ -which.min(currentGrades) ])
  average


}
```

```
testVector <- c(100, 100, 100, 60, 40, 20)
grades(testVector)
```

```
## [1] 80
```

```
#student1
grades(student1)
```

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

```
#student2
grades(student2)
```

```
## [1] 92.83333
```

```
#student3
grades(student3)
```

```
## [1] 12.85714
```

Now read the full gradebook CSV file.

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

```
##            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
```

```
#grades(scores[1,])
finalScores <- apply(scores, 1, grades)
```

Q2 Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook? [3pts]

The top scoring student is student 18.

```
which.max(finalScores)
```

```
## student-18
##         18
```

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

Apply to the columns

```
apply(scores, 2, mean, na.rm=TRUE)
```

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

HW 3 was the worst