Writing_Functions_Class06

Andrew Sue PID: A13006809

2024-01-26

Writing Functions

Each function in R is defined by 3 things:

- A user selected *name*
- Comma separated *input of arguments* (none,one or more)
- A *function body* including an optional output *return value*

Setting a value within the function calls it as its default if an arguent is not given otherwise. Example: `function(x,y=1)`

A function to add two numbers

```r
sillyadd<-function(x,y=1){
  x + y
}
```

Let's call function

```r
sillyadd(3)
```

```
[1] 4
```

To learn what a function does or how it is structured, you can remove the () of the function and input it within the console to see the full structure.

You can also label chunks of code by typing a label name or identifier in the chunk as shown: {r *name*}.

#Lets do something more useful.

Write a function **grade()** to determine an overall grade from a vector of student homework assignment scores dropping the lowest single assignment score.

```
# 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)
gradebook <- data.frame(student1,student2,student3)
```

Start small when building function to then progress and modify it.

```
mean(student1)
```

```
[1] 98.75
```

```
min(student1)
```

```
[1] 90
```

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

```
[1] 8
```

Now that we've identified the minimum row/grade, now we want to exclude it.

```
student1[-8]
```

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

Now combine all the code to give you the value of what you want. Now the NA in the other student grades become a problem with mean() so how do we remove them?

```
x <- student2 #can change this anytime
ind <- which.min(x) #finds lowest value,
mean(x[-ind],na.rm = T) #excludes, and then takes mean
```

```
[1] 92.83333
```

This doesnt work for student3 given the amount of NAs.

```
x <- student3 #can change this anytime
mean(x[-which.min(x)],na.rm = T) #This put all together
```

```
[1] NaN
```

So how do we find and replace the NAs with another value?

```
x <- student3 #can change this anytime
x[is.na(x)]<- 0
mean(x[-which.min(x)]) #This put all together
```

```
[1] 12.85714
```

```
grade <- function(x){
  x[is.na(x)] <-0
  mean(x[-which.min(x)])
}
grade(student1)
```

```
[1] 100
```

Read a class gradebook CSV file

```
url <- "https://tinyurl.com/gradeinput"
gradebook <- read.csv(url, row.names = 1)
#row.names takes the first column and sets it as the row names
head(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
```

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 is vectorized, so unlike having to make loops to go through a dataframe and apply your function, we can just use the `apply()` function.

Apply function requires 3 arguments:

- x (your array or dataframe)
- how the function is applied by either row or column (1 or 2)
- the function that is going to be applied

```r
results<- apply(gradebook,1,grade)
results
```

```
 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]

```r
which.max(results)
```

```
student-18 
        18 
```

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

```r
apply(gradebook, 2, mean, na.rm = T)
```

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

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]

```
mask <- gradebook
mask[is.na(mask)]<-0
apply(mask,2, cor,results)
```

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