Class_06: R Functions

Toheeb-Balogun

2022-10-14

Table of contents

Function basics

All functions in R have at least 3 things:

- A **name** (we pick this),
- Input **arguments** (there can be loads comma seperated),
- A **body** (the R code that does the work)

```
# 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)
```

I can use the `mean`function to get the average

```
mean(student1)
```

```
[1] 98.75
```

I can find the lowest value with `min()` function

```
min(student1)
```

```
[1] 90
```

I found the `which.min()` function, what does it do?

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

```
[1] 8
```

```
student1[1:7]
```

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

Can we use the minus index trick?

```
student1[3]
```

```
[1] 100
```

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

```
[1] 100
```

```
student2
```

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

#Try for student 2

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

```
[1] NA
```

```
  mean(student3, na.rm=T)
```

```
[1] 90
```

We need another way

Can I replace NA values to zero

```
  is.na(student2)
```

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

```
  student2[ is.na(student2) ] <- 0
  student2
```

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

```
  #This codes turn NA to 0
  positions <-  is.na(student3)
  student3[ positions ] <- 0
  student3
```

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

```
  x <- c(1:4)
  x
```

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

```
  x[2] <- 100
  x
```

```
[1]   1 100   3   4
```

Let's put these two things back together and get my working snippet

```
student2[ is.na(student2) ] <- 0
mean( student2[ -which.min(student2)])
```

```
[1] 91
```

Re-write my snippet to be more simple

```
x <- student3
#Make NA zeros
x[ is.na(x) ] <- 0
mean( x[ -which.min(x)])
```

```
[1] 12.85714
```

Now I can make my function

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

Now use that to grade student1 etc

```
grade(student1)
```

```
[1] 100
```

#Extract function

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

Q2. Grade a class

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

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

Now I want to introduce the `apply()` function

```
  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
```

I can use `which.max` to find where the largest/max value is this results vector

```
  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?

We can use `apply()` but this time over the columns, i.e MARGIN=2

```
  apply(gradebook, 2, sum, na.rm=TRUE)
```

```
 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)?

```
mask <-  gradebook
mask [ is.na(mask) ] <- 0

cor(mask$hw5, results)
```

```
[1] 0.6325982
```

```
#cor(gradebook$hw2, results)
```

Can I apply the `cor()` function over the masked gradebook? Sure!

```
apply(mask, 2, cor, results)
```

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

Can we render this document