Class 6: R functions

Ani (A16647613)

2024-01-25

R Functions

Functions are how we get stuff done. We call functions to do everything useful in R.

One cool thing about R is that it makes writing your own functions comparatively easy.

All functions in R have at least three things:

- A **name** (we get to pick this)
- One or more **input arguments** (the input to our function)
- The **body** (lines of code that do the work)

```
funname <- function(input1, input2) {
  # The body with R code
}
```

Let's write a silly first function to add two numbers:

```
x <- 5
y <- 1
x + y
```

```
[1] 6
```

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

```
addme(100,100)
```

```
[1] 200
```

```
addme(100)
```

```
[1] 101
```

Lab for today

```
# Example input vectors as a start

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)
```

Question 1

First finding the mean.

```
mean(student1)
```

```
[1] 98.75
```

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

```
[1] 91
```

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

```
[1] 90
```

This isn't fair- there is no way student3 should have a mean of 90.

Come back to this NA problem. But things worked for `student1`.

We want to drop the lowest score before getting the `mean()`.

How do I find the lowest (minimum) score?

```
  student1
```

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

```
  min(student1)
```

```
[1] 90
```

I found the `which.min()` function. Maybe this is more useful?

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

```
[1] 8
```

Cool - it is the eighth element of the vector that has the lowest score. Can I remove this one?

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

```
[1] 90
```

Student1 grades with the lowest score dropped.

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

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

Mean of student1's scores with the lowest score dropped.

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

```
[1] 100
```

A more explicit way (as shown in class):

```
# Find the lowest score
ind <- which.min(student1)
# Remove lowest score and find the mean
mean(student1[-ind])
```

```
[1] 100
```

Use a common shortcut and use `x` as my input

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

```
[1] 100
```

We still have the problem of missing values.

One idea is to replace NA values with zero.

```
y <- 1:5
y==3
```

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

```
y[y==3]
```

```
[1] 3
```

```
y <- c(1,2,NA,4,5)
y == NA
```

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

```
y
```

```
[1]  1  2 NA  4  5
```

```
is.na(y)
```

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

How can I remove the NA elements from the vector? I need to flip the TRUE element

```
!c(F,F,F)
```

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

```
#y[is.na(y)]
```

```
y[!is.na(y)]
```

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

```
y[is.na(y)] <- 10000
y
```

```
[1]     1     2 10000     4     5
```

Okay, let's solve this:

Last step now that I have my working code snipped is to make my `grade()` function

```
grade <- function(student1, student2, student3) {
 x <- student1

# Change NA values to 0
x[is.na(x)] <- 0
mean(x[-which.min(x)])
}
```

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Question 2: Who was the top scoring student?

Now read the online gradebook (CSV file)

```
url <- "https://tinyurl.com/gradeinput"
gradebook <- read.csv(url, 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
```

```
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
```

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

```
student-18
        18
```

Question 3: Which homework was the toughest on students?

```
  hwresults <- apply(gradebook, 2, mean, na.rm = TRUE)
  hwresults
```

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

```
  which.min(hwresults)
```

```
hw3
  3
```

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

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

Question 4: Which homework was the most predictive (ie.highest correlation with average score)?

```
  # Make all NA to zero
  mask <- gradebook
  mask[is.na(mask)] <- 0
  #mask
```

We can use `cor()`function for correlation analysis.

```
  cor(gradebook$hw1, results)
```

```
[1] 0.4250204
```

This hw1 correlation indicates a relatively high correlation.

```r
cor(mask$hw5, results)
```

```
[1] 0.6325982
```

```r
cor(mask$hw3, results)
```

```
[1] 0.3042561
```

apply(mask, 2, cor)

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

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