Class 6: R Functions

James Garza (PID: A16300772

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 ir 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** (line 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
```

```r
addme(10)
```

```
[1] 11
```

Lab for today

```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)
```

```r
grade_func <- function(x) {mean(x, na.rm=TRUE)}
```

```r
grade_func(student1)
```

```
[1] 98.75
```

```r
grade_func(student2)
```

```
[1] 91
```

```r
grade_func(student3)
```

```
[1] 90
```

```r
df <- data.frame(student1, student2, student3)
```

```r
grade_func <- function(x) {(sum(x, na.rm=TRUE)-min(x, na.rm = TRUE))/(which.min(x)-1)}

grade_func(student1)
```

```
[1] 100
```

```r
grade_func(student2)
```

```
[1] 79.57143
```

```r
grade_func(student3)
```

```
[1] NaN
```

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

```
[1] 90
```

```r
grade_2 <-function(x) {mean(x[-which.min(x)], na.rm=TRUE)}

grade_2(student1)
```

```
[1] 100
```

```r
grade_2(student2)
```

```
[1] 92.83333
```

```r
grade_2(student3)
```

```
[1] NaN
```

```r
ind <- which.min(x)
# Find the lowest score
mean(student1[-which.min(student1)])
```

```
[1] 100
```

```
# remove tlowest score and find the mean
```

Use a common shortcut and usex as my input

```
x1 <- student1
x2 <- student2
x3 <- student3
mean(x1[-which.min(x1)])
```

```
[1] 100
```

```
is.na(x) <- 0

student3
```

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

```
grade_func <- function(x) {(sum(x, na.rm=TRUE)-min(x, na.rm = TRUE))/(which.min(x)-1)}

grade_func(student1)
```

```
[1] 100
```

```
grade_func(student2)
```

```
[1] 79.57143
```

```
grade_func(student3)
```

```
[1] NaN
```

Replace NA values with zeroes.

```
y <- 1:5
y[y == 3] <- 10000
y
```

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

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

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

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

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

How can I remove the NA elements from the vector? I first need to flip the true elements

```
!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)] <- 0
y
```

```
[1] 1 2 0 4 5
```

Testing that we can turn the NA values to be equivalent to the value 0

```
is.na(x3) <- 0
x3 <- ifelse(is.na(x3), 0, x3)
x3
```

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

```r
is.na(x2) <- 0
x2 <- ifelse(is.na(x2), 0, x2)
x2
```

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

```r
is.na(x1) <- 0
x1 <- ifelse(is.na(x1), 0, x1)
x1
```

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

```r
no_na <- function(x) {is.na(x) <- 0}

no_na(x2)
x2
```

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

We still have the problem of missing values.

Okay let's put Humpty Dumpty back together NA values have been changed to 0

Last step, working code snippet with the grade function

```r
grade <-function(x) {
  x[is.na(x)] <-0
  mean(x[-which.min(x)], na.rm=TRUE)}

grade(x1)
```

```
[1] 100
```

```r
grade(x2)
```

```
[1] 91
```

```
grade(x3)
```

```
[1] 12.85714
```

Q1

grade function code: **grade <-function(x) {x[is.na(x)] <-0 mean(x[-which.min(x)], na.rm=TRUE)}**

```
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
```

Function: APPLY() ; it takes multiple arguments and applies it over to a data set apply(input = gradebook, Margin, fun = grade) what is the margin argument? indicates the rows or the columns 1 indicates rows and 2 indicates columns

```
all_grades <- apply(gradebook,1,grade)
all_grades
```

```
 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

Top Scoring Students:

```r
which.max(all_grades)
```

```
student-18
        18
```

Q3

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

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

```r
which.min(all_h)
```

```
hw3
  3
```

```r
all_hw <- apply(gradebook, 2, sum, na.rm=TRUE)
all_hw
```

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

```r
which.min(all_hw)
```

```
hw2
  2
```

Q4

Correlation between Homework Scores and Grade Scores

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

```
           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   0  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   0  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   0
student-16  92 100  74  89  77
student-17  88  63 100  86  78
student-18  91   0 100  87 100
student-19  91  68  75  86  79
student-20  91  68  76  88  76
```

We can use the `cor()` function for correlational analysis.

```
cor(mask$hw1, all_grades)
```

```
[1] 0.4250204
```

```
cor(mask$hw2, all_grades)
```

```
[1] 0.176778
```

```
cor(mask$hw3, all_grades)
```

```
[1] 0.3042561
```

```r
cor(mask$hw4, all_grades)
```

```
[1] 0.3810884
```

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

```
[1] 0.6325982
```

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

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