Class06: R Functions

Xiaoxuan Teng

2024-01-26

Our first simple silly function

All functions in R have 3 parts. They have:

- a name
- input arguments (none, one or more)
- a body

A function to add two numbers

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

Let me try out this function.

```
sillyadd(10)
```

```
[1] 11
```

Let's do sth more useful

```
# Example 1
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)
```

```
mean(student1)
```

```
[1] 98.75
```

```
min(student1)
```

```
[1] 90
```

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

```
[1] 8
```

```
#which.min(student2)
#which.min(student3)
```

```
student1[-8]
```

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

```
x <- student2
# Find lowest value
ind <- which.min(x)
# Exclude lowest value and find mean
mean(x[-ind], na.rm = T)
```

```
[1] 92.83333
```

```
student3
```

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

```
x <- student3
# Find lowest value
ind <- which.min(x)
# Exclude lowest value and find mean
mean(x[-ind], na.rm = T)
```

```
[1] NaN
```

Find and replace the NA values with zero

```r
x <- 1:5
x
```

```
[1] 1 2 3 4 5
```

```r
x[x == 3] <- 10000
x
```

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

```r
x <- student2
x
```

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

```r
x[is.na(x)] <- 0
x
```

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

```r
x <- student3
x[is.na(x)] <- 0
mean(x[-which.min(x)])
```

```
[1] 12.85714
```

```r
x <- student3
x[is.na(x)] <- 0
ind <- which.min(x)
mean(x[-ind])
```

```
[1] 12.85714
```

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

```
grade_dropmin(student3)
```

```
[1] 12.85714
```

Read a class grade book CSV file from here: "https://tinyurl.com/gradeinput"

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

Now use our `grade()` function to grade the whole class...

We can "apply" our new `grade()` function over either the rows or the coulmns of the gradebook. with MARGIN = 1 or MARGIN = 2

```
results <- apply(gradebook, 1, FUN = grade_dropmin)
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]

```
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]

```
  toughest <- apply(gradebook, 2, mean, na.rm = T)
  toughest
```

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

```
  which.min(toughest)
```

```
hw3
  3
```

```
  grade <- function(x, drop.lowest=TRUE) {
    x[is.na(x)] <- 0
    if(drop.lowest){
    x <- x[-which.min(x)]
    ans <- mean(x)
    }
    else{
      ans <- mean(x)
    }
    ans
  }
```

```
  toughest <- apply(gradebook, 2, grade, drop.lowest = F)
  toughest
```

```
  hw1   hw2   hw3   hw4   hw5
89.00 72.80 80.80 85.15 79.25
```

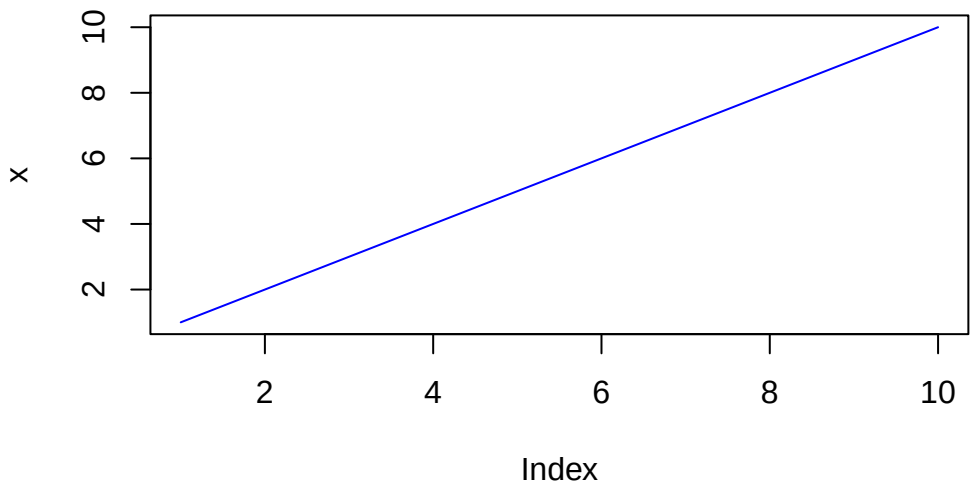
```
  which.min(toughest)
```

```
hw2
  2
```

```
  plotme <- function(x, ...){
    plot(x, ...)
```

```
}

plotme(1:10, col="blue", typ="l")
```



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

cor(mask$hw5, results)
```

```
[1] 0.6325982
```

```
cor(mask$hw4, results)
```

```
[1] 0.3810884
```

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

```
[1] 0.3042561
```

```r
cor(mask$hw2, results)
```

```
[1] 0.176778
```

```r
cor(mask$hw1, results)
```

```
[1] 0.4250204
```

```r
correlation <- apply(mask, 2, cor, y = results)
correlation
```

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

```r
which.max(correlation)
```

```
hw5
  5
```