class-06

Iulia Rusu

2024-01-26

all functions have 3 items:

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

a function to add two numbers

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

try out function

```
sillyadd(10)
```

```
[1] 11
```

```
sillyadd <- function (x, y= 1) { #If no y given do not specify y, it will be 1
 x + y
}

sillyadd(10)
```

```
[1] 11
```

```
sillyadd(10 + 3)
```

```
[1] 14
```

```
#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)
```

```
mean(student1)
```

```
[1] 98.75
```

```
min(student1)
```

```
[1] 90
```

```
# Find lowest value
x <- student1
lowest_index <- which.min(x)
#exclude lowest value and find mean
mean(x[-lowest_index])
```

```
[1] 100
```

```
# Find lowest value
student2
```

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

```
x <- student2
lowest_index <- which.min(x)
lowest_index
```

```
[1] 8
```

```
#exclude lowest value and find mean
mean(x[-lowest_index], na.rm=T)
```

```
[1] 92.83333
```

```
# Find lowest value
student3
```

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

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

```
[1] 2
```

```
#exclude lowest value and find mean
mean(x[-lowest_index], na.rm=T)
```

```
[1] 12.85714
```

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

#exclude lowest value and find mean
  mean(x[-lowest_index], na.rm=T)

}
```

```
grade(student1)
```

```
[1] 100
```

Read class gradebook csv file from here: "https://tinyurl.com/gradeinput"

```
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
```

We can apply our new 'grade' function over either the rows or the columns of the gradebook. with MARGIN=1, or MARGIN=2

```
apply(gradebook,1, grade)
```

```
 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.Usingyourgrade()functionandthesuppliedgradebook,Whoisthetopscoringstudent overall inthegradebook? [3pts]

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

which.max (averages)
```

```
student-18
        18
```

Q3.Fromyouranalysisofthegradebook,whichhomeworkwastoughestonstudents(i.e.obtained thelowestscoresoverall? [2pts]

HW3

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

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

```
grade <- function(x, drop.lowest= TRUE) {
  x[is.na(x)] <- 0

  if(drop.lowest){
    ans <- mean(x[-which.min(x)])
  }
  else {

    ans <- mean(x)
  }
    ans

}
```

Q4. OptionalExtension: Fromyouranalysisof thegradebook,whichhomeworkwasmost predictiveofoverallscore(i.e.highestcorrelationwithaveragegradescore)? [1pt]

```
gradebook$hw5
```

```
 [1]  79  78  77  76  79  77 100 100  77  76 100 100  80  76  NA  77  78 100  79
[20]  76
```

```
averages
```

```
 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
```

```
mask <- gradebook
mask[is.na(mask)] <- 0
cor( mask$hw5, averages)
```

```
[1] 0.6325982
```

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

```
[1] 0.3042561
```

```r
apply(mask, 2, cor, y =averages)
```

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