class06: Functions

Aigerim (PID: 09919142)

2024-01-26

Our first simple silly function

All functions in R have 3 parts:

- a name
- inpur argument (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(100)
```

```
[1] 101
```

Let's do something more useful

```
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
```

```
student1
```

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

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

```
[1] 8
```

in [-8] we remove number in the position 8 by adding “-”

```
student1
```

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

```
mean(student1[ -8 ])
```

```
[1] 100
```

Collected form it

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

```
[1] 100
```

```
x <- student2
x
```

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

```
# Find lowest value
ind <- which.min(x)
ind
```

```
[1] 8
```

```
# 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 with zero

```
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
```

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

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

```
[1] 12.85714
```

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

```r
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

> **Q1** Write a function grade() to determine an overall grade from a vector of student homework assignment scores dropping the lowest single score. If a student misses a homework (i.e. has an NA value) this can be used as a score to be potentially dropped. Your final function should be adquately explained with code comments and be able to work on an example class gradebook such as this one in CSV format: "https://tinyurl.com/gradeinput"

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

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

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

We can "apply" our new `grade()` function over wither the row or the columns of the gradebook, with the MARGIN

```
results <- apply(gradebook, 1, grade)
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** Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook?

```
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)?

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

```
     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 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
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
```

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

```
[1] 0.6325982
```

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

```
[1] 0.3042561
```

Let’s use apply to do this for the whole course!

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

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