Class 6: R Functions

Catherine (Cathy) Diep

Table of contents

Function Basics

All functions in R consist of at least 3 things: - A **name** which must start with a character - Input **arguments** as comma-separated inputs - The **body** where work actually happens

Q1 Write a function grade() to determine an overall grade from a vector of student homework assignment scores dropping the lowest single score.

```
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
```

```
  student1[-8]
```

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

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

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

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

```
[1] 100
```

- Square brackets to select a point within a vector
- Minus sign in brackets to remove a specific point within a vector
- Mean of the vector without the lowest value
- This won't work for student 2 or student 3 because it will give NA
- Need to change mean's na.rm argument which will drop NA- default is na.rm = FALSE

```
student2 [x = NA] <- 0
mean(student2[-which.min(student2)], na.rm= TRUE)
```

```
[1] 92.83333
```

- na.rm doesn't work for student 3, which would make their average 90
- replace NA with 0

```r
student3
```

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

```r
is.na(student3)
```

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

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

```
[1] 12.85714
```

- Too much copy-and-paste, room for error

```r
x <- student1
x[is.na(x)] <- 0
mean(x[-which.min(x)])
```

```
[1] 100
```

- Work has been simplified
- with this, just need to change 1st line

-Now, turn this into a function

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

Q2. Who is the top scoring student in your gradebook?

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

```
?apply(array, margin, ...)
results <- apply(gradebook, 1, grade)
```

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

```
student-18
        18
```

```
results[which.max(results)]
```

```
student-18
      94.5
```

- Student 18 is the highest scoring student.

Q3. Which homework was toughest on students?

```
average <- function(x) {
  x[is.na(x)] <- 0
  mean(x)}
homework <- apply(gradebook, 2, average)
homework[which.min(homework)]
```

```
 hw2
72.8
```

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

```
hw2
  2
```

- Homework 2 was the most difficult for students.

Q4 Which homework was most predictive of overall score?

```
cor(gradebook [,5], results)
```

```
[1] NA
```

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

```
[1] 0.6325982
```

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

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

- Homework 5 was most predictive of overall score.