Lab6: R functions and R packages from CRAN and BioConductor

Duy An Le (PID: 16400411)

Functions are how we get work done in R. We call functions to do everything from reading data to doing analysis and outputting plots and results.

All functions in R have at least 3 things:

- a name (user-defined)
- input arguments (there can be only one or one hundred, based on your needs)
- the body (where the work gets defined, the code between curley brackets)

Example, basic Function

Let's write a function to add some numbers. We can call it `add()`

```
x <- 10
y <- 10

x + y
```

```
[1] 20
```

The function will add 10 to an input x.

```
add <- function(x) {
  y <- 10
  x + y
}
```

Let's test `add()`

```
add(1)
```

```
[1] 11
```

Let's make it more flexible:

```
add <- function(x, y=1) { ## default y = 1
  x + y
}

add(x=10, y=10) ## y = 10
```

```
[1] 20
```

```
add(10)
```

```
[1] 11
```

Lab 6: Make functions for a class gradebook

> 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" [3pts]

Test grade() on individual examples

Some initial grade vectors:

```
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)
```

Replace NA with 0:

```
student1[is.na(student1)] <- 0
student2[is.na(student2)] <- 0
student3[is.na(student3)] <- 0
```

Remove minimum grade:

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

```
[1] 100
```

```
mean(student2[-which.min(student2)])
```

```
[1] 91
```

```
mean(student3[-which.min(student3)])
```

```
[1] 12.85714
```

Now put together to write the function:

```
grade <- function(student) {
  # student is a vector containing student grades
  student[is.na(student)] <- 0
  # replace NAs with 0s
  mean(student[-which.min(student)]) # remove the minimum and calculate grade
}

grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Use grade() on a csv file containing grades

```
gradebook <- read.csv('/Users/duyle/Downloads/BIMM 143/Lab6/student_homework.csv', row.nam
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
```

`apply()` function in R will apply functions to an array or matrix

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

Q2. Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook? [3pts]

```
top_student = names(which.max(final_grades))
print(paste("The top student is", top_student, "."))
```

```
[1] "The top student is student-18 ."
```

Q3. From your analysis of the gradebook, which homework was toughest on students (i.e. obtained the lowest scores overall? [2pts]

```
hardest_hw = names(which.min(apply(gradebook, 2, mean, na.rm=T)))
print(paste("The hardest homework is", hardest_hw, "."))
```

```
[1] "The hardest homework is hw3 ."
```

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]

Make all NAs into 0:

```
copy <- gradebook
copy[is.na(copy)] <- 0
good_predictor <- names(which.max(apply(copy, 2, cor, y=final_grades)))
print(paste("The best predictor of overall grade is", good_predictor, "."))
```

```
[1] "The best predictor of overall grade is hw5 ."
```