

Class6:Rfunctions

RamolaBaviskar (PID A12228297)

2/3/2022

Today we will explore R functions.

We will start with calculating a grade for these example students.

```
# 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)
```

We can use the mean() function to get the average.

```
mean(student1)
```

```
## [1] 98.75
```

```
mean(student2, na.rm=TRUE)
```

```
## [1] 91
```

How does the is.na() function work? Let's try it on student 2.

```
student2
```

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

```
is.na(student2)
```

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

We can use this result to get at our NA values, the TRUE positions.

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

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

```
mean(student2)
```

```
## [1] 79.625
```

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

```
## [1] 90  0  0  0  0  0  0  0
```

```
mean(student3)
```

```
## [1] 11.25
```

Now we'll find the lowest score.

```
min(student1)
```

```
## [1] 90
```

```
student1
```

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

```
min(student1)
```

```
## [1] 90
```

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

```
## [1] 8
```

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

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

Now we can turn this into our first function, called 'grade()'. All R functions have 3 things: a name (grade), input arguments (scores), and body (does the work).

```
#first set NA values to zero
```

```
x[is.na(x)] <- 0  
#remove lowest score and calculate mean  
x <- x[-which.min(x)]  
mean(x)
```

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

```
#first set NA values to zero
grade <- function(x) {
  x[is.na(x)] <- 0
  #remove lowest score and calculate mean
  mean(x[-which.min(x)])
}
```

Let's try it

```
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 adequately 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]

```
gradebook <- read.csv("https://tinyurl.com/gradeinput", 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 use the ‘`apply()`’ function to grade the whole class.

```
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
```

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

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

```
which.max(scores)
```

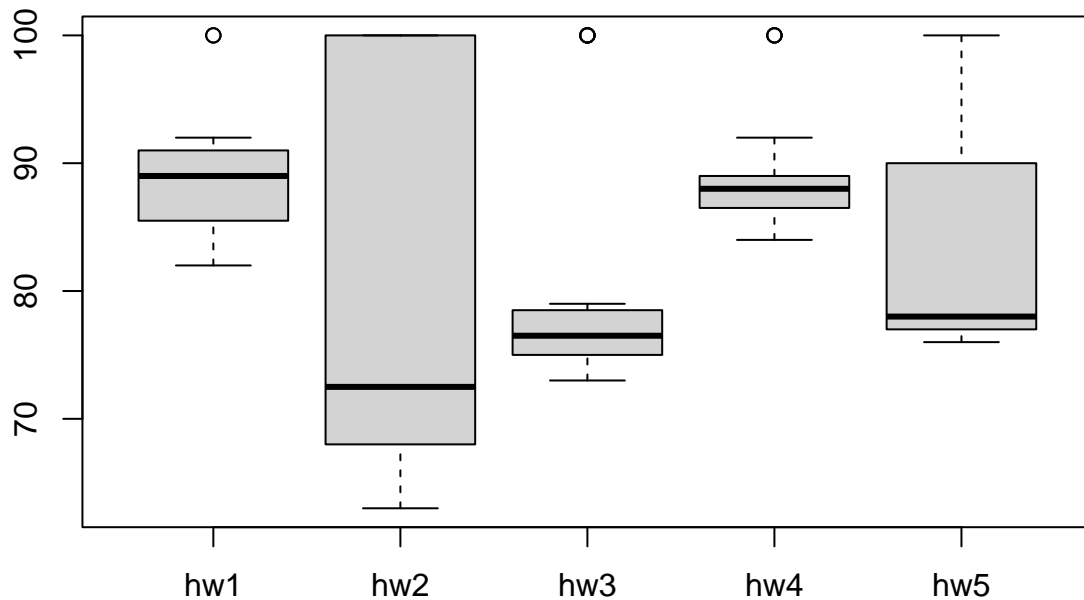
```
## student-18
##           18
```

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

```
hw.mean <- (apply(gradebook, 2, mean, na.rm = TRUE))  
which.min(hw.mean)
```

```
## hw3  
## 3
```

```
boxplot(gradebook)
```



According to the boxplot, HW2 has the lowest average but HW3 has the smallest spread and the lowest overall scores. Therefore, HW3 was the toughest on students.

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]