

Class 6: R Functions

Soobin (PID:A15201229)

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 could use the mean() function to calculate the average -> This is not fair since student 3 will only have one score to grade

```
mean(student1)
```

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

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

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

```
mean(student3, na.rm = TRUE)
```

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

How does this is.na() function work? Let's try it out on student2.

```
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 (i.e. the TRUE positions).

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

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

```
student3
```

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

Now we can find the average grade of students

```
mean(student2)
```

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

```
mean(student3)
```

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

Now we want to find the lowest score...

```
student1
```

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

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

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

The which.min() function tells us where the smallest value is. [-i]: print every thing except x[i]

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

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

Now let's simply this!

```
x <- student1
# First set NA values to zero
x[ is.na(x) ] <- 0
# Remove lowest score and calculate average
mean ( x[ -which.min(x) ] )
```

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

Now we can turn this into our first function. We will call this function 'grade()'.

All R functions have 3 things - a name (grade) - input arguments (student scores) - body (does the work!)

```

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

```

Let's try it out

```
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 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>”.

```

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

```

```
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
```

We can use the ‘apply()’ function to grade the whole class

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

Q2. Using your grade() function and the supplied gradebook, who is the top scoring student overall in the gradebook?

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

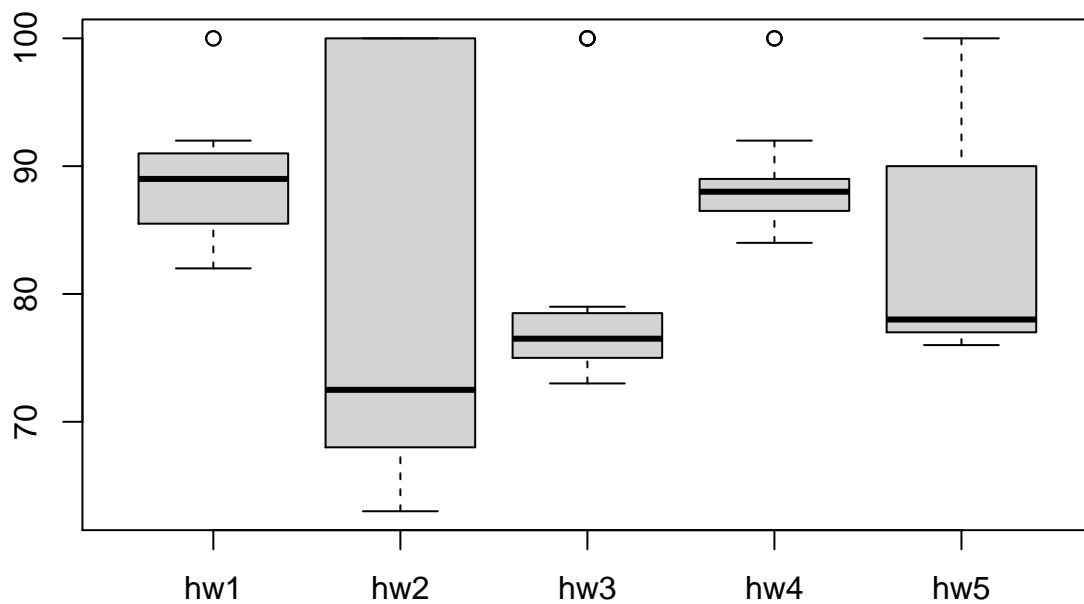
```
## student-18
##          18
```

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

```
# Since there are outliers, it is better use median rather than mean
hw_mean <- apply(gradebook, 2, mean, na.rm=TRUE)
which.min( hw_mean )
```

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

```
# HW2 has the lowest median
boxplot(gradebook)
```



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

```
## hw2
## 2
```

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)

```
gradebook2 <- gradebook
gradebook2[ is.na(gradebook2) ] <- 0
correlation <- apply(gradebook2, 2, cor, scores)
which.max(correlation)
```

```
## hw5
## 5
```