class06

Delaney (PID: A15567985)

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 an average.

```
mean(student1)
```

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

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

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

How does the 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 FALSE
```

We can use this result to get our NA values (i.e. the TRUE positions)

```
student2[is.na(student2)]
```

```
## [1] NA
```

We can set the NA value to 0 and find the average grade.

```
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 want to find the lowest score…

```
student1
```

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

```
min(student1)
```

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

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

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

The which.min() function tell us where the smallest valie is.

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

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

Average grade minus lowest score.

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

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

We are close to our working code snippet that will be the body of our first function.

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

```
## [1] 92.83333
```

Now we can turn this into our first function. We can 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(student2)
```

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

> 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"

```
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.

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

```
##  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(scores)
```

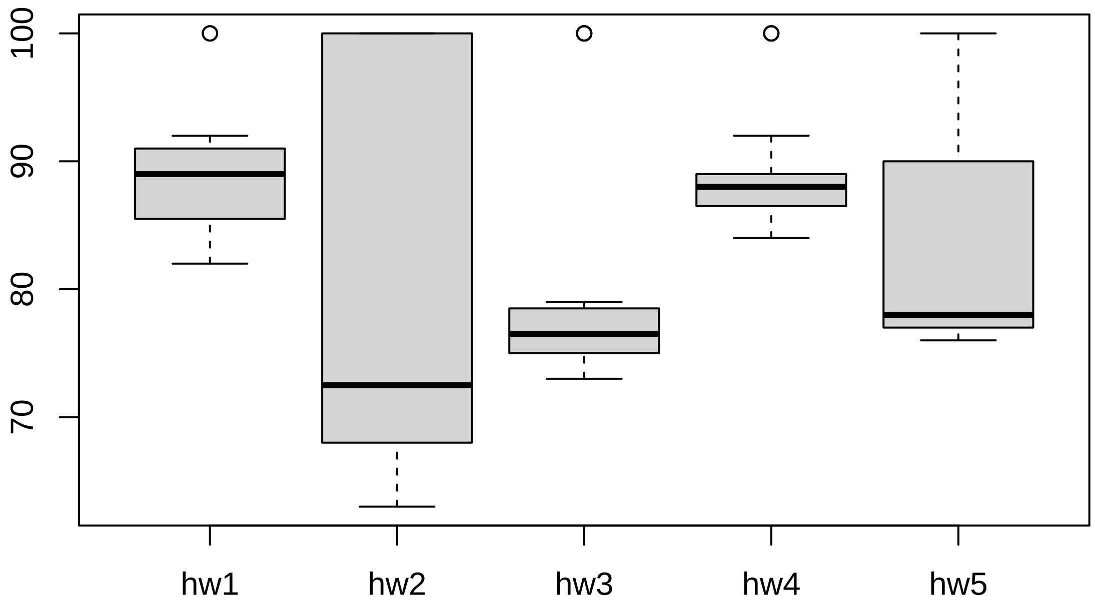
```
## student-18
##         18
```

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

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

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

```
boxplot(gradebook)
```



Q4. From your analysis of the gradebook, which homework was most predictive of overall score (i.e. highest correlation with average grade score)?

```
cor(scores, gradebook$hw1)
```

```
## [1] 0.4250204
```

Now apply over all columns (i.e. homeworks)

```
apply(gradebook, 2, cor, scores)
```

```
##       hw1       hw2       hw3       hw4       hw5 
## 0.4250204        NA 0.3042561        NA        NA
```

Change all NA to 0

```
mask <- gradebook
mask[ is.na(mask)] <- 0
```

Use mask instead of gradeboo

```
apply(mask, 2, cor, scores)
```

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

> Q5. Make sure you save your Rmarkdown document and can click the "Knit" button to generate a PDF foramt report without errors. Finally, submit your PDF to gradescope.