Week 3, R Functions

Charlie

2023-04-22

This week we're intorducing R functions and how to write our own.

> **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.

```
# 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)
```

Guidelines from class - write a working code snippet to solve a simple problem

```
#Straight forward  mean
student1 <- c(100, 100, 100, 100, 100, 100, 100, 90)
mean(student1)
```

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

However, we need to identify and drop the lowest score for each student.

```
#Which element of the student1 vector is the lowest?
which.min(student1)
```

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

If I want to drop (exclude) the lowest score from my mean calculation:

```
#This will return everything BUT the 8th element of my vector
student1[-8]
```

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

Now we can use the answer from which.min to find the answer to our question.

```
#Here is our first snippet of code.
mean(student1[-which.min(student1)])
```

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

For Students 2 and 3 we will have to replace all NAs with 0

Student 2:

```
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
x <- student2
is.na(x)
```

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

```
which( is.na(x))
```

```
## [1] 2
```

We've now identified the NA values we want to mask and replace NA with 0.

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

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

```
mean(x)
```

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

Now we just drop the lowest score.

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

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

Student 3:

```
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
x <- student3
x[is.na(x)] <- 0
mean(x[-which.min(x)])
```

```
## [1] 12.85714
```

Now we're nearly there with our snippet!

##Now take the snippet and turn it into a function

Every function has 3 parts: - A name (in our case `grade()`) - Input arguments, a vector of student scores - The body – our snippet

I will do this by selecting `Code then Extract Function` in Rstudio

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

```
grades(student1)
```

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

```
grades(student2)
```

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

```
grades(student3)
```

```
## [1] 12.85714
```

The code works! We just now need to add our comments to the function so that we can come back to this and understand it.

```
#' Calculate the average scroe for a vector of student scores, dropping the lowest value.
#' Missing values will be treated as 0
#'
#' @param x A numeric vector of student scores.
#'
#' @return Average score
#' @export
#'
#' @examples
#' student <- c(100, NA, 90, 97)
#' grade(student)
#'
#'
grades <- function(x) {
  #treat missing values as 0
  x[is.na(x)] <- 0
  #exclude lowest value from mean
  mean(x[-which.min(x)])}
```

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

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

```
apply(gradebook, 1, grades)
```

```
##  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
```

Can calculate highest perfomrer using the apply() function

```
results <- apply(gradebook, 1, grades)
results
```

```
##  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
```

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

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

The highest performing student was student-18

> **Q3.** Which homeowrk was the toughest??

```
avg.scores <- apply(gradebook, 2, mean, na.rm=TRUE)
avg.scores
```

```
##      hw1      hw2      hw3      hw4      hw5
## 89.00000 80.88889 80.80000 89.63158 83.42105
```

```
which.min(avg.scores)
```

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

```
med.scores <- apply(gradebook, 2, median, na.rm=TRUE)
med.scores
```

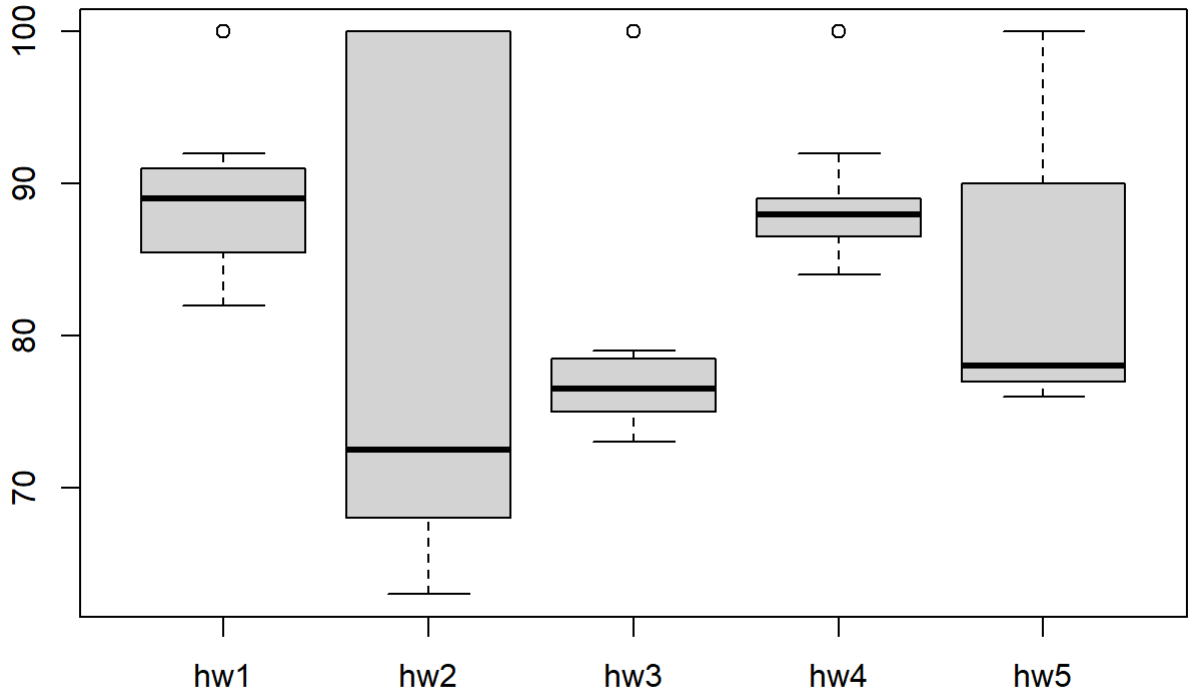
```
##  hw1  hw2  hw3  hw4  hw5
## 89.0 72.5 76.5 88.0 78.0
```

```
which.min(med.scores)
```

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

Which is better?

```
boxplot(gradebook)
```



Due to variation in the data, the median is a more reliable metric to use when answering this question.

The toughest homeowrk assignment was HW2

> **Q4.** From your analysis of the gradebook, which homework was most predictive of overall score?

Are the final results (avg score distribtuion) correlated with the results for individual homeworks?

```
masked.gradebook <- gradebook
masked.gradebook[is.na(masked.gradebook)] <- 0
masked.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   0  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   0  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   0
## student-16  92 100  74  89  77
## student-17  88  63 100  86  78
## student-18  91   0 100  87 100
## student-19  91  68  75  86  79
## student-20  91  68  76  88  76
```

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

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

```
apply(masked.gradebook, 2, cor, x=results)
```

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

HW5 is the best predictor of overall class grades

> **Q5.** Make sure you save your Quarto document and can click the “Render” (or Rmarkdown”Knit”) button to generate a PDF foramt report without errors. Finally, submit your PDF to gradescope.

Knit the document to produce a PDF *(or HTML that you’ll convert to PDF later if your computer is trash like mine)*