class6rmark

2023-05-01

We are looking at *R Functions* and how to write them.

> 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"

```
# 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)
```

Follow the guidelines from class - Write a working snippet of code that solves simple version of problem

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


mean(student1)
```

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

Must drop lowest score. Must identify lowest score in vector.

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

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

Time to drop lowest score from grade calculation

```
# return everything but the
# eighth element in the vector
student1[-8]
```

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

Now we use which.min() to return all other elements of vector minus the smallest one

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

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

```
# average of student 1 with lowest score dropped
# first working snippet of code
mean( student1[-which.min(student1)] )
```

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

Now other students??

using na.rm=TRUE only would be an unfair grading approach

```
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
mean(student2, na.rm=TRUE)
```

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

```
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
mean(student3, na.rm=TRUE)
```

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

One approach cold be to replace all NA values with zero

First, need to be able to find NA values of vector.

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

#tells us if any values in vector is NA
is.na(x)
```

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

```
#tells us location of NA values in vector
which(is.na(x))
```

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

NA elements identified. Time to "mask" them with value of 0 for calculations.

```
# Changing all NA elements in vector "x" to 0
x[is.na(x)] <- 0
x
```

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

```
mean(x)
```

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

Now need to be able to drop lowest score before calculating mean

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

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

Our working snippet code

```
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
```

Function

Take the snippet and turn into function Every function has 3 parts

- A name, in this case 'grade()'
- Input arguments, in this case a vector of student numeric scores
- The body, aka the working snippet

Using RStudio, will select 'Code' > 'Extract Function'

```
#' Calculate average score from a vector of
#' student scores while dropping the lowest score.
#' Missing values are assigned value of 0.
#'
#' @param x A numeric vector of homework scores of a student
#'
#' @return Average score
#' @examples
#'  student <- c(100, NA, 90, 97
#'  grade(student))

grade <- function(x) {
  # mask NA with zero
  # Treat NA as zero in calculation
  x[is.na(x)] <- 0
  # Exclude lowest score from calculation
  mean( x[-which.min(x)] )
}
```

```
grade(student1)
```

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

```
grade(student2)
```

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

```
grade(student3)
```

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

It works!

Now we can use our function on class data in the CSV file: "https://tinyurl.com/gradeinput"

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

```
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
```

Q2. Who is the top scoring student overall in the gradebook?

```
# Apply calculations to value 'results'
results <- apply(gradebook, 1, grade)
# Sort the values by decreasing order
sort(results, decreasing = TRUE)
```

```
## student-18  student-7  student-8 student-13  student-1 student-12 student-16
##      94.50      94.00      93.75      92.25      91.75      91.75      89.50
##  student-6  student-5 student-17  student-9 student-14 student-11  student-3
##      89.00      88.25      88.00      87.75      87.75      86.00      84.25
##  student-4 student-19 student-20  student-2 student-10 student-15
##      84.25      82.75      82.75      82.50      79.00      78.75
```

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

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

Q3. Which homework was the toughest on the students?

```
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
```

```
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
```

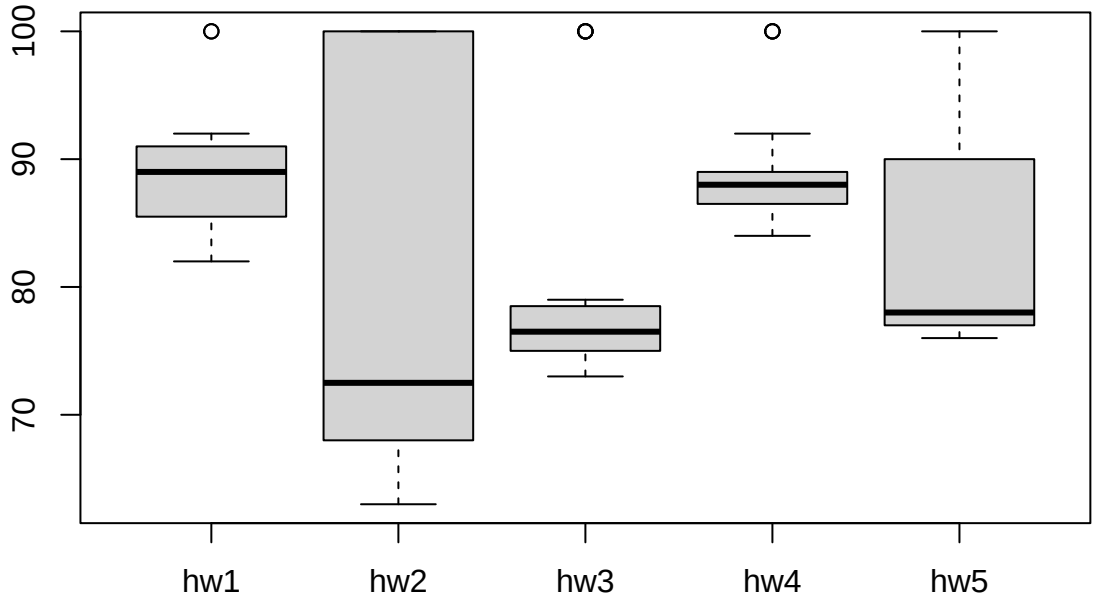
```
med.scores <- apply(gradebook, 2, median, na.rm = TRUE)
med.scores
```

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

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

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

```
boxplot(gradebook)
```


100
90
80
70
hw1
hw2
hw3
hw4
hw5