Class 06 R functions

Audrey Ting Zhu (A16898668)

2024-10-21

This week we are introducing R functions and how to write our own Questions to answer:

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.

```r
# 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)
```

```r
#finding the element showing lowest grade
which.min(student1)
```

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

I now need to exclude the lowest element

```r
student1[-8]
```

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

```r
#This excludes the lowest value manually. Replace 8 with which.min(student1)
student1[-which.min(student1)]
```

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

Now we can find mean:

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

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

However, this wouldn't work on the other vectors in students, as they contain NA. We need to replace NA values with 0. First, we need to find the NA elements.

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

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

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

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

Now we replace NA with 0

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

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

Put it all together

```r
#for student2
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
x<-student2
x[is.na(x)]<-0
mean(x[-which.min(x)])
```

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

```r
#for student3
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 make it a function. A function has 3 parts: 1. A name (ex.grade()) 2. input arguments(vector of student scores) 3. The body, working snippet of code

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

Check function

```r
grade(student1)
```

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

```r
grade(student2)
```

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

```r
grade(student3)
```

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

Adding comments so others can use the function

```r
#' Calclates the average score for a vector of student scores, lowest grade dropped, missing grades are
#'
#' @param x Numeric value of student scores
#'
#' @return Average score
#' @export
#'
#' @examples student<-c(90,97,90,NA)
#'               grade(student)

grade <- function(x) {
  #missing grades(NA)counted as zero because they are missing
  x[is.na(x)]<-0
  #lowest score is excluded before mean calculation
  mean(x[-which.min(x)])
}
```

Now we use our function on real whole class data. Class data is from this CSV format: “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. Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook?

We first save the grades calculated from the csv file into results.

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

Now I find the top scoring

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

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

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

```
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
```

Now we are looking at the averages of columns (the specific hw). Thus, the margin will be 2, instead of 1.

```
#find the average scores of HW, NA must be set to TRUE, in order for function to calculate.
ave.scores<-apply(gradebook,2,mean, na.rm=TRUE)
ave.scores
```

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

```
#Finding the worst HW by searching for the lowerst average score
which.min(ave.scores)
```

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

The worst HW seems to be 3.

However, sometimes mean is not the best indicator, as it is sensitive to outliers. We will use median.

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

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

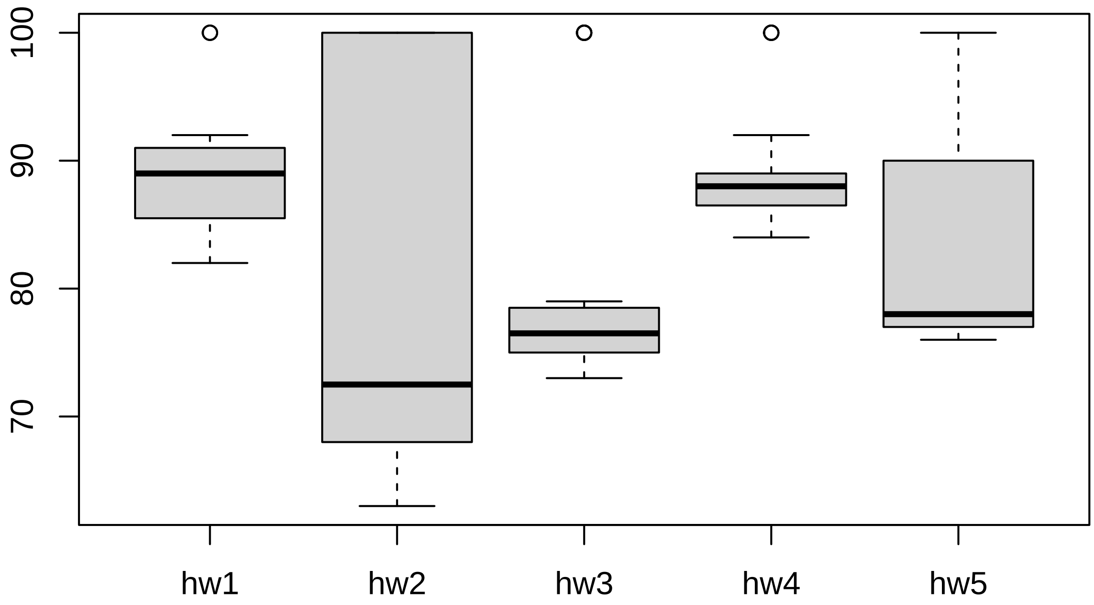
```
which.min(median.scores)
```

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

In this case, the worst HW is 2.

So which is the correct answer? We can use graphs to give a clearer answer by visualizing distribution.

```
boxplot(gradebook)
```



There is a really great distribution for HW2. Some did well. Some did horrible. HW3 does not have this wide distribution.

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)?

We hope that the students average score(grade) correlates to the scores they get on the hw (gradebook columns).

```
#mask the NA scores in the gradebook
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
```

Find correlation function

```
cor(results, masked.gradebook$hw5)
```

```
## [1] 0.6325982
```

Apply to all gradebook

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

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

It seems hw 5 has the highest correlation. Hw5 is most predictive.

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