

Grade Function Lab Week 5

Vera Sophia Beliaev

2/7/2022

Today we are creating **R functions** to process grades of a class, dropping the lowest score for each person.

Questions:

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> [3pts]

```
# 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)
```

Write code for a simple version of the problem

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

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

Lowest score needs to be identified then dropped

```
#Find which element of the vector contains the lowest score
which.min(student1)
```

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

Need to drop/exclude the lowest score from the mean()

```
#This returns everything except the 8th element
student1[-8]
```

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

Integrate `which.min` to find everything except that element

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

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

Calc the mean of all the elements of the vect excluding which.min elem

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

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

Looking to student 2 and 3

We try using the na.rm=TRUE arg but it's not fair

```
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
```

Replace all NA values with zero

Find the NA elements

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

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

```
#tell me which element contains NA value
which(is.na(x))
```

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

NA value is identified, now we need to switch it with a value of zero (mask them)

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

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

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

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

```
mean(x)
```

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

We need to drop the lowest score

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

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

Test of 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
```

The gold:

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

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

Now we make the function

snippet -> function

Function contains: a name, input args, the body

USing RStudio, select 'Code > Extract Function'

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

```
grade(student1)
```

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

```
grade(student2)
```

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

```
grade(student3)
```

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

```
## Calculate the average score for a vector of scores, dropping the lowest score. Missing values are given as NA
##
## @param x A numeric vector of homework scores
##
## @return Average score
## @export
##
## @examples
## student <- c(100, NA, 90, 97)
## grade(student)
##
grade <- function(x) {
  #Treat missing values as zero, mask NA with zero
  x[is.na(x)] <- 0
  #Exclude the lowest score from calculating the mean
  mean(x[-which.min(x)])
}
```

Now we apply the function to the whole class data 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? [3pts]

```
##Find the highest scoring student on average
```

```
##through sorting
sort((apply(gradebook, 1, grade)))
```

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

```
#highest average score
max(apply(gradebook, 1, grade))
```

```
## [1] 94.5
```

```
#which student scored highest
which.max(apply(gradebook, 1, grade))
```

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

Q3. From your analysis of the gradebook, which homework was toughest on students (i.e. obtained the lowest scores overall)? [2pts]

```
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
```

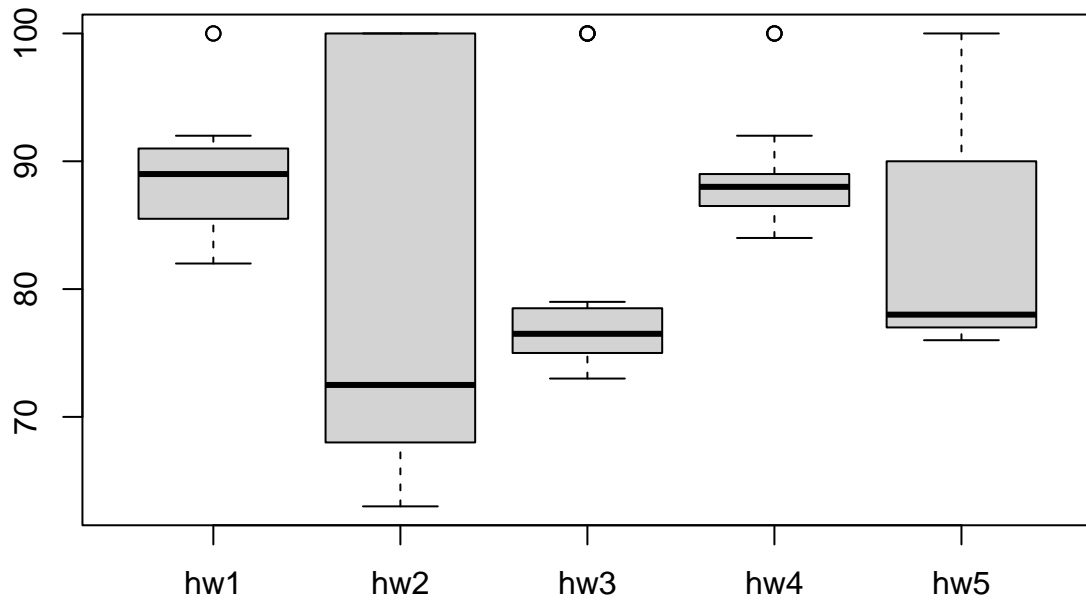
```
# Check stats of columns, check mean
which.min(apply(gradebook, 2, mean, na.rm=TRUE))
```

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

```
#Check stats of columns, check median
which.min(apply(gradebook, 2, median, na.rm=TRUE))
```

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

```
#make a boxplot
boxplot(gradebook)
```



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)? [1pt]

```
#correlation b/w x & y (between average overall class score and average score for each homework), apply
results <- apply(gradebook, 1, grade)
```

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

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

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

Knit the doc to make a PDF using the dropdown menu or switch output at top to pdf_document