---
title: 'calss 6 : R Functions'
author: "Tianyun Hua PID:A91041691"
date: "10/15/2021"
output: html_document
---

quick R Markdown intro

bold or *italic*

This is a level 1 heading

This is a level 3 heading

Here we will write a function to grade studetn homework. New code chunck: option + command + I (for R code)

We will start with input example–student homework scores

```
# example input vector
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)
```

> 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" [3pts]

First, find the lowest score using **min()**, and the **which.min()** function to find the position of the smallest value

```
student1
```

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

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

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

```
student1[8]
```

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

```
# to get everything but the minimum value
student1[-which.min(student1)]
```

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

Then we can take the mean

```
# this is my first solution
mean(student1[-which.min(student1)])
```

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

Is this a good idea?

```
student2
```

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

```
# remove the NA value
mean(student2, na.rm = TRUE)
```

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

```
mean(student3, na.rm = TRUE)
```

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

```
# not a good idea
```

This is a bad idea because this is unfair using the na.rm = TRUE argument.

Let's change NA values to 0 Use the 'is.na()' function

```
student2
```

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

```
is.na(student2)
```

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

```
x <- student2
x [is.na(x)] <- 0
x
```

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

```
mean(x)
```

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

Combine working snippets to find the average for student3

```
x3 <- student3
x3[is.na(x3)] <- 0
mean(x3[-which.min(x3)])
```

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

now we can make our function

use the "code"-"exrtact function"

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

Now use the function to grade 3 students

```
grade(student1)
```

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

```
grade(student2)
```

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

```
grade(student3)
```

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

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" Now **grade the whole class**

Document the newly defined function with "code"-"Roxygen Skeleton"

```
#' Calculate average scores for a vector of homework scores
#' dropping lowest single score. missing value will
#' be treated as zero score
#' @param x numeric vector of hw scores
#'
#' @return average score
#' @export
#'
#' @examples
#'  student <- c(100, NA, 90, 80)
#'  grade(student)
#'
grade <- function(x) {
  # map NA missing homework values to zero
```

```
  # missing homework is 0
  x[is.na(x)] <- 0
  # exclude lowest score hw
  mean(x[-which.min(x)])
}
```

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

We are going to use the super useful **apply()**

```
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?

```
results <- apply(gradebook, 1, grade)
# first approach
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
```

```
# second approach
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?

Here we want to calculate a summary stat for each column of the gradebook. Which stat should we use?

```
# Let's try average, by column (margin = 2)
apply(gradebook, 2, mean)
```

```
##  hw1  hw2  hw3  hw4  hw5
## 89.0   NA 80.8   NA   NA
```

I can ignore the NA missing values

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

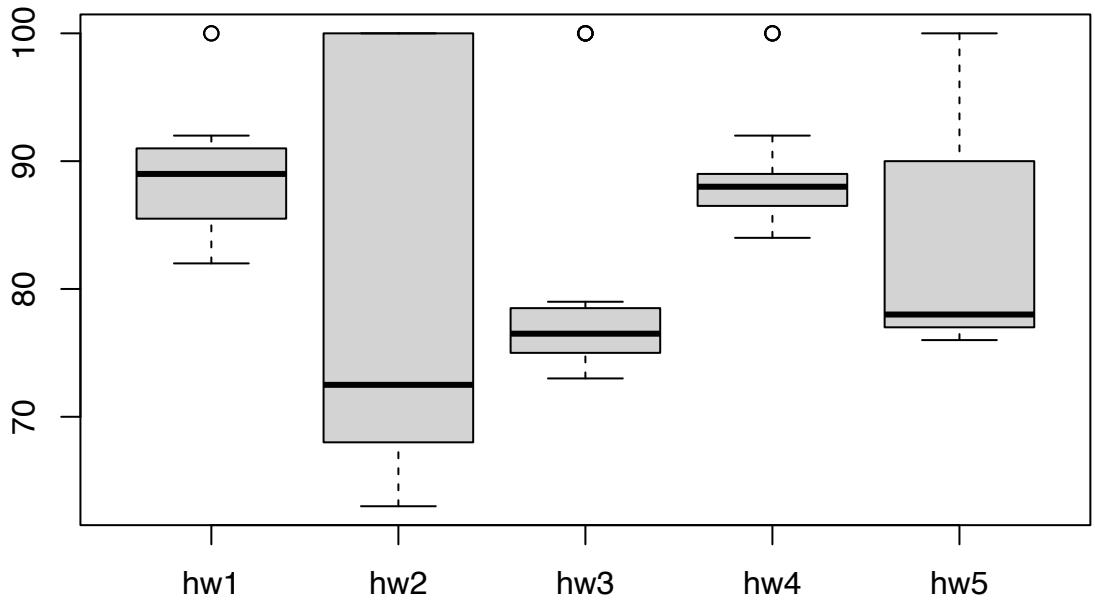
```
## hw3
##   3
```

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

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

Difference when using mean and median, plot the data and see

```
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)?

we will use **cor()** here

```
gradebook[is.na(gradebook)] <- 0
cor(results, gradebook$hw5)
```

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

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

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

Homework 5 is the most predicative of overall score