

# Class6Rfunctions

Josef Urrete

4/21/23

Developing our own R function to calculate average grades.

Starting with simple version of the problem #Simplified version

## Simplified problem

```
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)
```

Start by calculating average score of homeworks

```
mean(student1)
```

```
[1] 98.75
```

```
mean(student1, na.rm = T)
```

```
[1] 98.75
```

Getting minimum score via which.min

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

```
[1] 8
```

Return everything but the lowest score

```
student1[-8]
```

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

averaging first 7 homework scores excludes the 8th element minimum

```
mean(student1[1:7])
```

```
[1] 100
```

trying to generalize the function for any student that doesn't have the 8th element as the minimum score

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

```
[1] 100
```

```
student1_drop_min = student1[-which.min(student1)]
```

```
mean(student1_drop_min)
```

```
[1] 100
```

First working snippet above, try with student2

```
student2_drop_min <- student2[-which.min(student2)]  
student2_drop_min
```

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

```
mean(student2, na.rm=T)
```

```
[1] 91
```

```
mean(student2_drop_min, na.rm = TRUE)
```

```
[1] 92.83333
```

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

```
[1] 90
```

Problem is for student 3, it does not take into account that we only drop the lowest score  
try to find position of NA

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

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

```
which(is.na(student3))
```

```
[1] 2 3 4 5 6 7 8
```

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

```
[1] 2
```

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

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

where NA used to be, is now a 0

Take this snippet and use it for student3

```
student3[(is.na(student3))] <- 0  
student3
```

```
[1] 90 0 0 0 0 0 0 0
```

```
mean(student3)
```

```
[1] 11.25
```

This is our final working snippet of code for all students (with and without NA values)

```
student3_drop_min <- student3[-which.min(student3)]  
mean(student3_drop_min)
```

```
[1] 12.85714
```

## Generalize some more, build a function

```
x <- c(100,75,50,NA)  
x[ is.na(x)] <- 0  
x_drop_min <- x[-which.min(x)]  
mean(x_drop_min)
```

```
[1] 75
```

Write as a function

```
#' Title  
#'  
#' @param x A numeric vector of homework scores  
#'  
#' @return the average value of homework scores  
#' @export  
#'  
#' @examples  
#'  
#' student <- c('100', '50', NA)  
#' grade(student)  
#'  
grade <- function(x) {  
  # Mask NA values with zero  
  x[ is.na(x)] <- 0
```

```
# taking the mean after drop lowest score  
mean(x[-which.min(x)])  
}
```

Test

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

## Q1

make grade function see above

Purpose is to calculate the average score for a vector of student scores, with their lowest score dropped and missing values treated as zero

**applying function to a gradebook from URL:**  
**“<http://tinyurl.com/gradeinput>”**

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

```
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 top scoring student overall in gradebook

```
which.max(apply(gradebook,1,grade))
```

```
student-18
18
```

```
max(apply(gradebook,1,grade))
```

```
[1] 94.5
```

The max score is from student 18, which is 94.5

#Q3. Which homework was toughest on students (obtained lowest scores overall)

```
is.na(gradebook)
```

	hw1	hw2	hw3	hw4	hw5
student-1	FALSE	FALSE	FALSE	FALSE	FALSE
student-2	FALSE	FALSE	FALSE	FALSE	FALSE
student-3	FALSE	FALSE	FALSE	FALSE	FALSE

```

student-4 FALSE TRUE FALSE FALSE FALSE
student-5 FALSE FALSE FALSE FALSE FALSE
student-6 FALSE FALSE FALSE FALSE FALSE
student-7 FALSE FALSE FALSE FALSE FALSE
student-8 FALSE FALSE FALSE FALSE FALSE
student-9 FALSE FALSE FALSE FALSE FALSE
student-10 FALSE FALSE FALSE TRUE FALSE
student-11 FALSE FALSE FALSE FALSE FALSE
student-12 FALSE FALSE FALSE FALSE FALSE
student-13 FALSE FALSE FALSE FALSE FALSE
student-14 FALSE FALSE FALSE FALSE FALSE
student-15 FALSE FALSE FALSE FALSE TRUE
student-16 FALSE FALSE FALSE FALSE FALSE
student-17 FALSE FALSE FALSE FALSE FALSE
student-18 FALSE TRUE FALSE FALSE FALSE
student-19 FALSE FALSE FALSE FALSE FALSE
student-20 FALSE FALSE FALSE FALSE FALSE

```

First, mask NA values with zeroes

```

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

apply mean function to gradebook

```
apply(gradebook, 2, mean)
```

```
hw1  hw2  hw3  hw4  hw5
89.00 72.80 80.80 85.15 79.25
```

The toughest homework is HW2 considering the mean and considering missing homework as zero. It is possible that having missing homework as zero is too strict and is not good for representing homework difficulty overall.

remove missing values

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

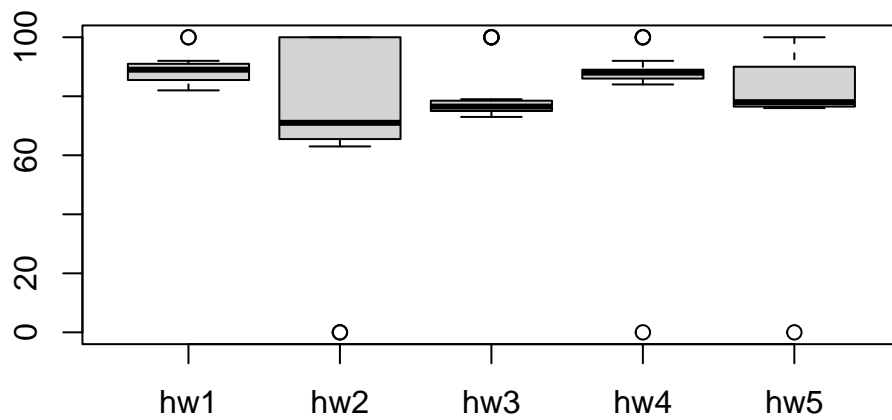
```
hw1  hw2  hw3  hw4  hw5
89.00 72.80 80.80 85.15 79.25
```

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

```
hw1  hw2  hw3  hw4  hw5
89.0 71.0 76.5 88.0 78.0
```

```
boxplot(gradebook)
```





**Q4 are final results correlated with results for individual homework assignments**

```
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

The maximum value is

```
which.max(apply(gradebook, 2, cor, y=masked.gradebook))
```

```
[1] 1
```

**Q5 knit to pdf**