

# class06

Jiachen Fan (A17662703)

## Example

Let's start with an example. Write a script to grade students' homework. We drop the lowest score of every student.

```
#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)
```

I can use 'min()' to find the lowest score.

```
min(student1)
```

```
[1] 90
```

```
min(student2)
```

```
[1] NA
```

```
min(student3)
```

```
[1] NA
```

We can use 'which.min()' to find the location of lowest score.

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

```
[1] 8
```

```
which.min(student2)
```

```
[1] 8
```

```
which.min(student3)
```

```
[1] 1
```

The result is correct for student1, but not for student2 and student3 because of 'NA'. So we have to change 'NA' to '0'.

```
student1[is.na(student1)] <- 0  
student2[is.na(student2)] <- 0  
student3[is.na(student3)] <- 0
```

We want to exclude the lowest score

```
final_student1 <- student1[-which.min(student1)]  
final_student1
```

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

```
final_student2 <- student2[-which.min(student2)]  
final_student2
```

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

```
final_student3 <- student3[-which.min(student3)]  
final_student3
```

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

Let's calculate the average using 'mean()'

```
mean(final_student1)
```

```
[1] 100
```

```
mean(final_student2)
```

```
[1] 91
```

```
mean(final_student3)
```

```
[1] 12.85714
```

**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>”**

Now we can build a function using the previous script.

```
grade <- function(scores){  
  scores[is.na(scores)] <- 0      # convert NA to '0'  
  mean(scores[-which.min(scores)]) # drop the lowest score and calculate the mean  
}  
  
# read the csv file from url  
scores <- read.csv("https://tinyurl.com/gradeinput", row.names = 1)  
scores
```

	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

```

A function is forced to use here which is 'apply()'. 'grade()' only works for a vector. 'apply(X, margin, FUN)' can help us grade all students

```

ans <- apply(scores, 1, grade)      # margin = 1 indicates rows
ans

```

```

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?**

```

# find the student with highest score using 'which.max()'
which.max(apply(scores,1,grade))

```

```

student-18
18

```

So student 18 is the top scoring student.

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

Get the average score over all students on the same homework to figure out which is toughest (means lowest average scores across different homework)

```
apply(scores, 2, grade) # margin = 2 indicates columns
```

```
      hw1      hw2      hw3      hw4      hw5  
89.36842 76.63158 81.21053 89.63158 83.42105
```

```
which.min(apply(scores, 2, grade))
```

```
hw2  
2
```

Homework 2 is the hardest.

### 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)?

To get the most predictive, I will use Pearson correlation between the average score of the student and the score in different homework.

```
scores[is.na(scores)] <- 0 # change NA to 0  
P <- numeric(5) # initialize 'P' as a numeric vector  
for (i in 1:5){ # for loop to go through every homework  
  P[i] <- cor(apply(scores, 1, grade), scores[,i]) # 'cor(x, y)' for Pearson correlation  
}  
P
```

```
[1] 0.4250204 0.1767780 0.3042561 0.3810884 0.6325982
```

```
which.max(P)
```

```
[1] 5
```

Based on Pearson correlation, homework 5 is the most predictive one.

```
# Here is a different way using 'apply()'
apply(scores,2,cor,y = ans)      # after function, we can add more arguments
```

hw1	hw2	hw3	hw4	hw5
0.4250204	0.1767780	0.3042561	0.3810884	0.6325982

```
which.max(apply(scores,2,cor,y = ans))
```

```
hw5
5
```

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