

Lab6

Qianqian Tao

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]

All functions in R have at least 3 things:

-**Name**, in our case “`grade`” -Input **argument**, `student1` etc. -**Body**, this is our working snippet above

```
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)

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

[1] 11.25

```
grade <- function(vector){
  #Map/Replace NA values to zero
  vector[is.na(vector)] <- 0
  #min_index <- which.min(vector)
  #Exclude the lowest score
  #vector <- vector[-which.min(vector)]
  #Exclude the lowest score and calculate the mean
  mean(vector[-which.min(vector)])
}
```

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

```
x
```

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

```
min(x)
```

```
[1] 0
```

```
which.min(x)
```

```
[1] 2
```

Q2. Using your `grade()` function and the supplied gradebook, Who is the top scoring student overall in the gradebook? [3pts]

```
#read.csv("https://tinyurl.com/gradeinput")  
class_data <- read.csv("https://tinyurl.com/gradeinput", row.names=1)  
class_average <- apply(class_data,1,grade)  
which.max(apply(class_data,1,grade))
```

```
student-18  
18
```

```
class_average[which.max(class_average)]
```

```
student-18  
94.5
```

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

```
new_mean <- function(vector){  
  vector[is.na(vector)] <- 0  
  mean(vector)  
}  
hw_average <- apply(class_data, 2, new_mean)  
hw_average[which.min(apply(class_data, 2, new_mean))]
```

```
hw2  
72.8
```

```
which.min(apply(class_data, 2, new_mean))
```

```
hw2  
2
```

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]

```
cor(class_data$hw1, class_average)
```

```
[1] 0.4250204
```

```
cor(class_data$hw2, class_average)
```

```
[1] NA
```

```
cor(class_data$hw3, class_average)
```

```
[1] 0.3042561
```

```
cor(class_data$hw4,class_average)
```

```
[1] NA
```

If I try on hw2, I get NA as there are missing homeworks. I will mask all NA values to Zero

```
mask <- class_data
mask[is.na(mask)] <- 0
mask
```

	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

```
cor(mask$hw1,class_average)
```

```
[1] 0.4250204
```

```
cor(mask$hw2,class_average)
```

```
[1] 0.176778
```

```
cor_homework <- apply(mask, 2, cor, y=class_average)
cor_homework[which.max(apply(mask, 2, cor, y=class_average))]
```

hw5
0.6325982