class06

Jenny Zhou

Write a function to determine an overall grade from a vector of student homework assignment scores dropping the lowest single alignment score

```
# student 1
student1 <- c(100,100,100,100,100,100,100,90)
# student 2
student2 <- c(100,NA,90,90,90,90,97,80)
#student 3
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
```

`mean()` to calculate average, `na.rm` can omit na values.

```
mean(student2)
```

```
[1] NA
```

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

```
[1] 91
```

We cannot do it on student3 because student3 has too much NA.

we can replace the missed assignment NA values with zero. `is.na()`function can help

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

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

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

```
[1] 2
```

```r
student2[is.na(student2)]
```

```
[1] NA
```

```r
student2[!is.na(student2)]
```

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

replace na to zero

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

but this one modified the original vector

it is time to work with new temp object (called`temp`) so I would not screw up my original objects

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

temp
```

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

Finally, we want to drop the lowest score before calculating the mean. This is equivalent to allowing the student to drop their worst assignment.

```r
min(temp)
```

```
[1] 0
```

```r
which.min(temp)
```

```
[1] 2
```

```r
#reverse selection
mean(temp[-which.min(temp)])
```

```
[1] 12.85714
```

Put this all back together make our working snippet:

```r
temp <- student3
#Replace NA values to zero
temp[is.na(temp)] <- 0
#Exclude the lowest score; Calculate the mean
mean(temp[-which.min(temp)])
```

```
[1] 12.85714
```

Turn into a function called `grade()` Required 3 things: - **Name** of the function, in our case "grade" - **input argument** - **body**

```r
grade <- function(x) {

  # Replace NA values to zero
  replaced_score <- replace(x,which(is.na(x)),0)

  #Exclude the lowest score; Calculate the mean
  mean(replaced_score[-which.min(replaced_score)])
}
```

```r
grade01 <- function(x) {

  # Replace NA values to zero
  x[is.na(x)] <- 0

  #Exclude the lowest score; Calculate the mean
  mean(x[-which.min(x)])
}
```

```r
grade(student2)
```

```
[1] 91
```

Read a gradebook form online:

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

```
           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 can use the `apply()` function to grade all the students in this class with our new `grade()` function.

The `apply()` function allows us to run any function over with rows or columns of a data.frame.

```r
ans <- apply(hw, 1, grade01)
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?

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

```
student-18 
      94.5 
```

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

```
hw_ans1 <- apply(hw, 2, mean, na.rm=TRUE)
hw_ans1
```

```
     hw1      hw2      hw3      hw4      hw5 
89.00000 80.88889 80.80000 89.63158 83.42105 
```

```
hw_ans1[which.min(hw_ans1)]
```

```
 hw3 
80.8 
```

```
hw_ans2 <- apply(hw, 2, sum, na.rm=TRUE)
hw_ans2
```

```
 hw1  hw2  hw3  hw4  hw5 
1780 1456 1616 1703 1585 
```

```
hw_ans2[which.min(hw_ans2)]
```

```
 hw2 
1456 
```

> 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)?

```r
cor(hw$hw1, ans)
```

```
[1] 0.4250204
```

```r
cor(hw$hw2, ans)
```

```
[1] NA
```

If I try on hw2, I get NA on missing homeworks. I want to mask all na values to zero.

```r
mask <- hw
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
```

try:

```
correlation <- cor(mask[1:5], ans)
correlation
```

```
         [,1]
hw1 0.4250204
hw2 0.1767780
hw3 0.3042561
hw4 0.3810884
hw5 0.6325982
```

```
which.max(correlation)
```

```
[1] 5
```

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

```
[1] 0.6325982
```

we can use `apply()` function here on the columns of hw

```
correlation2 <- apply(mask,2,cor, y=ans)
correlation2
```

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

```
correlation2[which.max(correlation2)]
```

```
      hw5 
0.6325982 
```