Class06

Loretta Cheng

Determining overall grades

I will be using this vector's in today's lab to determine overall scores between students.

```
# 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)
```

Using the grade() function

I will be using the grade() to determine the overall grade from a vector of student homework while dropping the lowest single score. Before I use the grade() I will be testing out the functions first.

We can use the `mean()` function to calculate the average for a given student vector. Here we are finding the mean for student 1:

```
mean(student1)
```

```
[1] 98.75
```

To subtract the lowest score from student one we would use the which.min() function to drop the lowest grade.

```
dropped_score <- function(x){
  x[is.na(x)] <- 0
  mean(x[-which.min(x)])
}
dropped_score(student1)
```

```
[1] 100
```

We know that by dropping the lowest score of student 1, their mean total grade will be 100 as 90 has been dropped.

We will now be calculating the total score with the lowest score being dropped with student 2 using the same idea but not counting the N/A.

```
dropped_score <- function(x){
  x[is.na(x)] <- 0
  mean(x[-which.min(x)])
}
dropped_score(student2)
```

```
[1] 91
```

We set N/A values to zero to count for either missing assignments or no show to class. Setting N/A to zero allows us to calculate the mean score with the lowest score dropped.

Here we find that with the lowest score dropped from the N/A = 0 in student two came out with the mean score of 91.

We will be applying the same function to student 3 to find the mean of scores after the lowest one have been dropped:

```
dropped_score <- function(x){
  x[is.na(x)] <- 0
  mean(x[-which.min(x)])
}
dropped_score(student3)
```

```
[1] 12.85714
```

Our average comes out to be 12.85714 once we replace all of our N/A with zeros and dropping lowest score.

```
mean(student2[-which.min(student2)])
```

```
[1] NA
```

I was playing around with the functions to see how to calculate the mean with the lowest score dropped.

I used this function:

```
dropped_score <- function(x){
  x[is.na(x)] <- 0
  mean(x[-which.min(x)])
}
dropped_score(student1)
```

```
[1] 100
```

This function makes it easier for me to calculate average mean with lowest grade dropped between all students. I set the function to grade() allowing me to recall it easier and more clear as well.

Q1

```
grade <-function(x){
  x[is.na(x)] <- 0
  mean(x[-which.min(x)])
}
grade(student1)
```

```
[1] 100
```

Class Notes & using dataset

However, we don't want NA to be calculated into the score. So we will be using student's 2.

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

```
[1] 91
```

What about student 3?

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

```
[1] 90
```

We can replace the homework assignments NA values with a score of zero. ** How do I do this?**

```r
student2
```

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

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

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

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

```
[1] NA
```

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

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

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

```
[1] 2
```

```r
dropped_score <- function(x){
  x[is.na(x)] <- 0
  mean(x[-which.min(x)])
}
dropped_score(student1)
```

```
[1] 100
```

```r
dropped_score(student2)
```

```
[1] 91
```

```
dropped_score(student3)
```

```
[1] 12.85714
```

read a gradebook from online:

```
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 students in this class using our grade() function

```
student_score <- apply(hw, 1, grade)
student_score
```

```
 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 grade() function and the supplied gradebook, who is the top scoring student overall in gradebook?

```
top_student <- student_score[which.max(student_score)]
top_student
```

```
student-18
      94.5
```

Q3: Which homework was hardest for students?

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

```
hw3
  3
```

```
total.score <- (apply(hw, 2, sum, na.rm=TRUE))
which.min(total.score)
```

```
hw2
  2
```

```
total.score
```

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

```
avg.score
```

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

Homework 2 was hardest for students because the total score for homework 2 between all students were 1456.

Q4: highest correlation with average score

```r
hw$hw1
```

```
 [1] 100  85  83  88  88  89  89  89  86  89  82 100  89  85  85  92  88  91  91
[20]  91
```

```r
ans <- apply(hw, 1, grade)
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 
```

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

```
[1] 0.4250204
```

If I try on hw2 I get NA as there are missing homeworks (i.e. NA values)

```r
hw$hw2
```

```
 [1]  73  64  69  NA 100  78 100 100 100  72  66  70 100 100  65 100  63  NA  68
[20]  68
```

I will mask all NA values to zero

```
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
```

```
cor(mask$hw5, ans)
```

```
[1] 0.6325982
```

```
apply(mask, 2, cor, ans)
```

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