

Lab 6 R functions

Anh Tran

10/13/2022

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.

```
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)
```

```
mean(student1)
```

```
[1] 98.75
```

```
min(student1)
```

```
[1] 90
```

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

```
[1] 8
```

Looking at the see also section of the `min()` help page I found out about `which.min()`. this finds the position of the lowest score.

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

```
[1] 8
```

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

```
[1] 100
```

is.na() helps assign a value to na

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

```
[1] 91
```

Does it matter to add square brackets here? Yes, it matters to use square bracket to access the elements because assigning 0 to the function does not make sense.

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

```
[1] 12.85714
```

Turning elements into x

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

```
[1] 100
```

Assigning function

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

```
url <- "https://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

Using `apply()`: `x` is an array (`gradebook`), `margin`: 1 for row, 2 for column (1), `fun` is function (`grade`)

```
results <- apply(gradebook, 1, grade)
results
```

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`? [3pts]

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

```
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]

```
which.min(apply(gradebook, 2, sum, na.rm=TRUE))
```

```
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)?

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

cor(mask$hw5, results)
```

```
[1] 0.6325982
```

Need to y=results to do correlation of x versus y. Advantage of using `apply()` is being able to indicate column/row

```
which.max(apply(mask, 2, cor, y=results))
```

```
hw5
5
```