Class06: R Functions

Hailey Wheeler (A13312713)

All about functions in R

Every function in R has at least 3 things: 1. A name (you pick it) 2. An argument (the input(s) to your function) 3. the body

Goal: Write a function to grade a class of student assignment scores (homework, etc)

Simplified vector where I know what the answer will be:

```
# 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)
```

Make sure it is displayed in the environment

```
mean(student1)
```

```
[1] 98.75
```

How do we drop the lowest score?

```
min(student1)
```

```
[1] 90
```

```
mean(student1) - min(student1)
```

```
[1] 8.75
```

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

```
[1] 8
```

```
  student1[-8]
```

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

```
  mean(student1, which.min(-8))
```

```
[1] 100
```

```
  student1 <- which.min(student1)
```

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

```
  grade(student1)
```

```
[1] NaN
```

```
  grade(student1)
```

```
[1] NaN
```

```
  grade(student2)
```

```
[1] 91
```

```
  grade(student3)
```

```
[1] 12.85714
```

Question #1: Write grade function to determine overall grade from a vector of student homework assignment scores droopping the lowest single score.

```
## Load Gradeboook
gradebook <- read.csv("https://tinyurl.com/gradeinput", row.names = 1)

## Write Function
grade <- function(x){
  ## Turn all NAs to 0
  x[is.na(x)] <- 0
  ## Drop the lowest score
  x_dropped <- x[-which.min(x)]
  ## Average student grade (excluding the lowest received score)
  mean(x_dropped)
}

## Confirm code works by using student2 example from before
grade(student2)
```

```
[1] 91
```

Question #2: Who is the top scoring student overall in the gradebook?

```
## Apply the grade function to the gradebook data frame to identify the average scores of
apply(gradebook, 1, grade)
```

```
 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
```

```
## Identify the top scoring student
which.max(apply(gradebook,1, grade))
```

```
student-18
        18
```

```
## Identify the top score
max(apply(gradebook, 1, grade))
```

```
[1] 94.5
```

Question #3: Which homework was toughest on students?

```
## Identify which homework (column) had the lowest scores.
which.min(apply(gradebook, 2, grade))
```

```
hw2
  2
```

```
## Note: the grade function drops the lowest score, so create new score without x_dropped

grade_nodrop <- function(x){
  x[is.na(x)] <- 0
  mean(x)
}

## Alternative way to mask NAs: create extra dataframe where the function 'mask' is used t
mask <- gradebook
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
```

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

```
hw2
  2
```

```
## Apply grade_nodrop function on gradebook database.
which.min(apply(gradebook, 2, grade_nodrop))
```

```
hw2
  2
```

Q4: Which homework(column) was most predictive of overall score? Do not remove zeros.

```
## Create a new data frame "Final Grade"
Final_Grade <- as.data.frame(apply(gradebook, 1, grade))
colnames(Final_Grade) = "Scores"

## Combine gradebook and Final Grade data frames
final_gradebook <- cbind(gradebook,Final_Grade)
final_gradebook
```

```
           hw1 hw2 hw3 hw4 hw5 Scores
student-1  100  73 100  88  79  91.75
student-2   85  64  78  89  78  82.50
student-3   83  69  77 100  77  84.25
student-4   88  NA  73 100  76  84.25
student-5   88 100  75  86  79  88.25
student-6   89  78 100  89  77  89.00
student-7   89 100  74  87 100  94.00
student-8   89 100  76  86 100  93.75
student-9   86 100  77  88  77  87.75
```

```
student-10  89  72  79  NA  76  79.00
student-11  82  66  78  84 100  86.00
student-12 100  70  75  92 100  91.75
student-13  89 100  76 100  80  92.25
student-14  85 100  77  89  76  87.75
student-15  85  65  76  89  NA  78.75
student-16  92 100  74  89  77  89.50
student-17  88  63 100  86  78  88.00
student-18  91  NA 100  87 100  94.50
student-19  91  68  75  86  79  82.75
student-20  91  68  76  88  76  82.75
```

```
## Mask NAs as 0s in the final gradebook dataframe
mask <- final_gradebook
mask[is.na(mask)] <-0
mask
```

```
           hw1 hw2 hw3 hw4 hw5 Scores
student-1  100  73 100  88  79  91.75
student-2   85  64  78  89  78  82.50
student-3   83  69  77 100  77  84.25
student-4   88   0  73 100  76  84.25
student-5   88 100  75  86  79  88.25
student-6   89  78 100  89  77  89.00
student-7   89 100  74  87 100  94.00
student-8   89 100  76  86 100  93.75
student-9   86 100  77  88  77  87.75
student-10  89  72  79   0  76  79.00
student-11  82  66  78  84 100  86.00
student-12 100  70  75  92 100  91.75
student-13  89 100  76 100  80  92.25
student-14  85 100  77  89  76  87.75
student-15  85  65  76  89   0  78.75
student-16  92 100  74  89  77  89.50
student-17  88  63 100  86  78  88.00
student-18  91   0 100  87 100  94.50
student-19  91  68  75  86  79  82.75
student-20  91  68  76  88  76  82.75
```

```
## Find the correlation between the final grade and the hw assignment.
cor <- cor(mask[,1:5], mask[,6])
```

```r
which.max(cor); max(cor)
```

```
[1] 5
```

```
[1] 0.6325982
```