Class06 Lab Functions

Sam Altshuler (PID: A59010373)

2/4/2022

In this class we are going to learn all about functions in R.

First we will write a function to grade some student scores. Use ctl+alt+I to insert code block.

```
# 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)
```

Calculate the scores of the student assignments for each student. Usually, we could just use a mean() function, but you cannot take a mean of a vector that contains NA values.

```
# Results in NA values
mean(student1)
```

```
## [1] 98.75
```

```
mean(student2)
```

```
## [1] NA
```

```
mean(student3)
```

```
## [1] NA
```

```
# Strip NA values using na.rm = TRUE
mean(student1, na.rm =T)
```

```
## [1] 98.75
```

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

```
## [1] 91
```

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

```
## [1] 90
```

mean(x, na.rm = T/F) will either remove or keep all NA values for a vector X. This isn't fair for student 3 since they only did one assignment and will still get a 90 for their overall homework grade. This could be fixed by setting NA to zero.

To find NA values (not available, AKA missing values), use `is.na()` function.

```
student2
```

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

```
# We know that only NA value for student 2 is in position 2
is.na(student2)
```

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

Quick review of logical vectors

```
x <- 1:5
x < 2
```

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

```
x == 3
```

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

Set all NA values to zero

```
# use a temp variable to not override the original data
n_student <- student3
n_student[is.na(student3)] <- 0
student3
```

```
## [1] 90 NA NA NA NA NA NA NA
```

```
n_student
```

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

We can use min() to find smallest value

```
# Return the smallest value
min(student1)
```

```
## [1] 90
```

```
# Return index (position) of the smallest value
which.min(student1)
```

```
## [1] 8
```

Put it all together into a function! This function should drop the single lowest score per student and then determine the overall score.

```
grade <- function(x){
  # Replace all NA values with zero
  x[is.na(x)] <- 0
  # Remove the smallest value
  y <- x[-which.min(x)]
  # Average out the score
  avg <- mean(y)
  # Return the average

  return(avg)
}
```

Test out the function.

```
grade(student1)
```

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

```
grade(student2)
```

```
## [1] 91
```

```
grade(student3)
```

```
## [1] 12.85714
```

Grade the Class

Input the dataset (it must be in the same workspace/directory) or use `read.csv()` with the file url as a string.

```
# use read.csv to insert the gradebook as a dataframe
gradebook <- read.csv("https://tinyurl.com/gradeinput", row.names = 1)
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
## 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
```

Q2. Use the grade() function to determine who is the top scoring student

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

```
which.max(score)
```

```
## student-18
##         18
```

Student 18 is the top scoring student according to this grade book with a score of 94.50

Q3. Which homework is the hardest for the students (which homework has the lowest overall score)?

```
# apply the mean function to every column
# use na.rm because NA values are outliers
# and should just be ignored
hw <- apply(gradebook, 2, mean, na.rm = TRUE)
hw
```

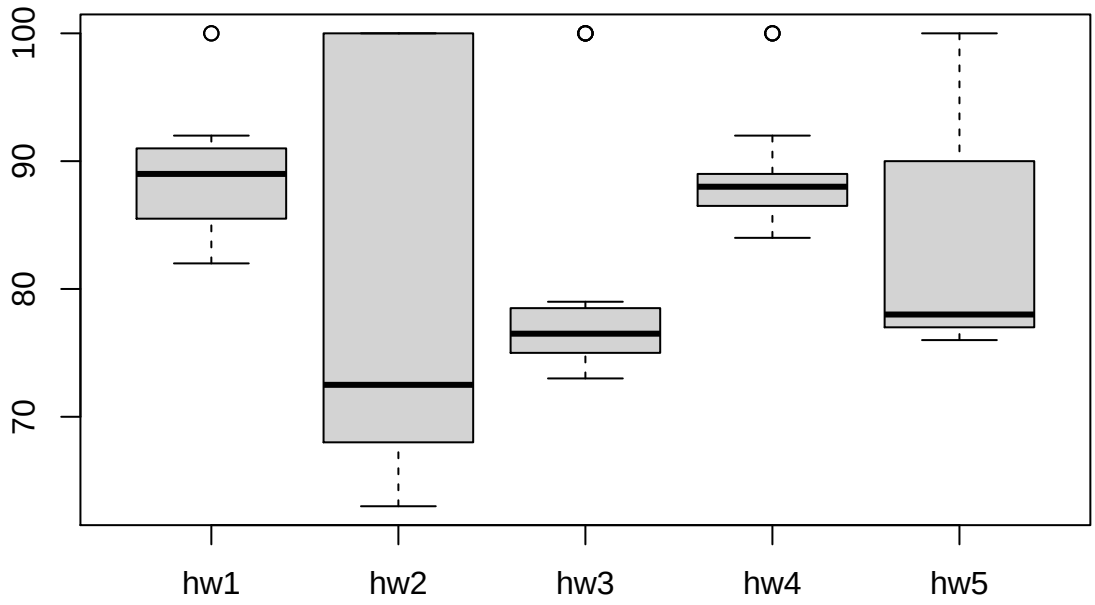
```
##      hw1      hw2      hw3      hw4      hw5
## 89.00000 80.88889 80.80000 89.63158 83.42105
```

```
which.min(hw)
```

```
## hw3
##   3
```

Homework 3 has the lowest mean score if NA's are removed. But... homework 2 has the lowest overall score as shown by the median in the boxplot below.

```
boxplot(gradebook)
```



Calculate the median to determine the homework that was the toughest as a better way to determine toughness than the mean (since outliers can affect the mean as shown by the boxplot above).

```
hw_new <- apply(gradebook, 2, median, na.rm = TRUE)
hw_new
```

```
##  hw1  hw2  hw3  hw4  hw5
## 89.0 72.5 76.5 88.0 78.0
```

```
which.min(hw_new)
```

```
## hw2
##   2
```

So homework 2 was the toughest with a median score of 72.5