feb04_2022 class

Rachel Diao

2/4/2022

This is an R Markdown header

Level 2 heading

Level 3 heading

This is just plain text.

We are now learning how to write functions.

First we will write a funciton to grade some student scores.

```
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
```

```
#For student 2, you'd have to drop the NA or else there'd be an error
mean(student2, na.rm = TRUE)
```

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

```
#For student 3... you're dropping NAs so you're not accounting for how little this person did
mean(student3, na.rm = TRUE)
```

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

```
#Convert all NAs in a vector to 0
student3_placehold <- student3
student3_placehold[is.na(student3)] <- 0
student3_placehold
```

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

```
mean(student3_placehold)
```

```
## [1] 11.25
```

Now we know how to get rid of NAs, we can write a function to calculate any student's average grade. This function takes in grade_list, a character vector containing numerical scores (or NA if any assignments were omitted).

Use if/else to approach this; if a student's grade list contains NA, then convert all the NAs to 0 before calculating the average and returning that as the output of the function. If there are no NAs, then proceed to normally calculate the mean.

```
calc_average <- function(grade_list){
  if (any(is.na(grade_list)) == TRUE) {
    grade_list[is.na(grade_list)] <- 0
    return(mean(grade_list))
  }
  else{
    return(mean(grade_list))
  }
}
```

Test out our function on each student

```
s1_avg <- calc_average(student1)
s2_avg <- calc_average(student2)
s3_avg <- calc_average(student3)
```

Now we can try altering this function to also calculate each student's grade after they drop their lowest score.

Function: grade Takes in a character vector containing numerical grades (and NAs for any omitted assignments

```
#Hint: which.min() finds out which position of a list contains a minimum (same with which.max())
which.min(student1)
```

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

```
student1[which.min(student1)] #prints out the lowest score after finding its position
```

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

```
which.min(student3_placehold)
```

```
## [1] 2
```

```
#In the case of student 3, there are multiple grades with the minimum value (0), but this function will

#Test out notation for dropping an element form a vector
student3_placehold_drop <- student3_placehold[-which.min(student3_placehold)]

grade <- function(gradelist){
  #First: if any NAs are present in the list, convert to 0
  if (any(is.na(gradelist)) == TRUE) {
    gradelist[is.na(gradelist)] <- 0
  }

  #Then we can find the minimum and drop it using vec_name[-element] notation
  dropped <- gradelist[-which.min(gradelist)]

  #Then return the average
  return(mean(dropped))
}
```

Test the 'grade' function on the students' grades.

```
grade(student1)
```

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

```
grade(student2)
```

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

```
grade(student3)
```

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

Question 1

Now apply this to dataframe 'student_homework.csv'. Import into R using read.csv. Each column is for a different homework assignment and each row is for a different student.

We want to use the function 'apply' to apply the 'grade' function to each row of this dataframe.

```
#Add argument row.names = 1 so that the first column (student names) is an index and no longer content
sh <- read.csv('student_homework.csv', row.names = 1)

#Apply function to the dataframe that has no student names
#Margin = shows how to apply the function --> i.e. 1 is rows, 2 is columns
apply(sh, MARGIN = 1, FUN = 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
```

Question 2

The top-scoring student in the gradebook is student 18, with an average of 94.50.

```
averages <- apply(sh, MARGIN = 1, FUN = grade)
which.max(averages)
```

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

Question 3

The homework that was toughest on the students was hw2, with a class average of 72.80.

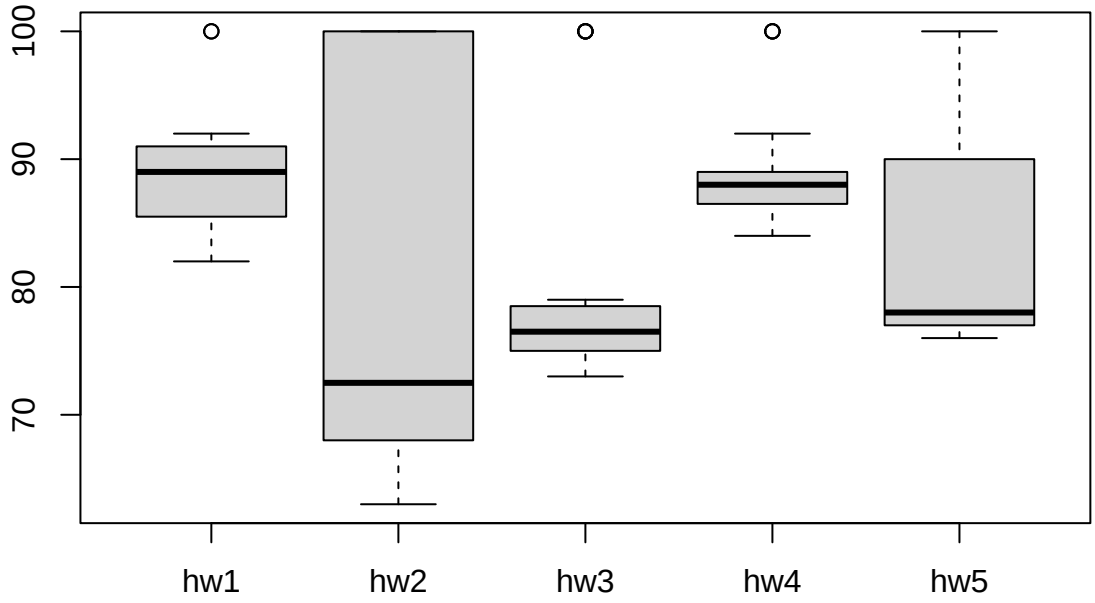
```
#Duplicate the dataframe and then convert all NAs to 0
dup_sh <- sh
dup_sh[is.na(dup_sh)] <- 0

#Find the average score for each column
colMeans(dup_sh)
```

```
##   hw1   hw2   hw3   hw4   hw5
## 89.00 72.80 80.80 85.15 79.25
```

```
#Alternative method:
apply(sh, 2, mean, na.rm=TRUE)
```

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

```
#BUT THE ANSWERS ARE DIFFERENT depending on whether you remove NAs or convert them to 0
#Visualize the data to figure out what's going on here
boxplot(sh)
```


100
90
80
70
hw1
hw2
hw3
hw4
hw5