Assignment 2: Coding Basics

Gretchen Barbera

OVERVIEW

This exercise accompanies the lessons/labs in Environmental Data Analytics on coding basics.

Directions

1. Rename this file `<FirstLast>_A02_CodingBasics.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.
6. After Knitting, submit the completed exercise (PDF file) to Canvas.

Basics, Part 1

1. Generate a sequence of numbers from one to 55, increasing by fives. Assign this sequence a name.
2. Compute the mean and median of this sequence.
3. Ask R to determine whether the mean is greater than the median.
4. Insert comments in your code to describe what you are doing.

```
#1
I_love_math <- seq(1,55,5) #I named my sequence I_love_math and it is showing up in my environment as t
seq(1,55,5)
```

```
##  [1]  1  6 11 16 21 26 31 36 41 46 51
```

```
#2
 mean(I_love_math) #this calculates the mean of my sequence that I named I_love_math
```

```
## [1] 26
```

```
median(I_love_math) #this calculates the median of my sequence
```

```
## [1] 26
```

```
#3
mean(I_love_math) > median(I_love_math) #to see if the mean is greater than the median, I used the grea
```

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

Basics, Part 2

5. Create three vectors, each with four components, consisting of (a) student names, (b) test scores, and (c) whether they are on scholarship or not (TRUE or FALSE).
6. Label each vector with a comment on what type of vector it is.
7. Combine each of the vectors into a data frame. Assign the data frame an informative name.
8. Label the columns of your data frame with informative titles.

```
a <- c("PJ", "Teddy", "Poppy", "Simon")
student_names <- c("PJ", "Teddy", "Poppy", "Simon") #I am creating the vector student_names for the nam

b <-c(100,45,78,86)
test_scores <- c(100,45,78,86) #test scores vector- numbers

c <- c(TRUE,TRUE, FALSE, TRUE)
scholarships <- c(TRUE,TRUE, FALSE, TRUE) #scholarships vector-logic

#7

student_achievers <- data.frame(student_names,test_scores,scholarships) #I have combined the vectors in

#8
#I renammed the rows first because it made better sense to me while tidying the data- I didn't want to
```

9. QUESTION: How is this data frame different from a matrix?

   Answer: This data frame has three different variables of data, numeric, character, and logic while a matrix needs to have all the same type of variables (all numeric, all characters, all logic- there can't be a mix).

10. Create a function with one input. In this function, use `if...else` to evaluate the value of the input: if it is greater than 50, print the word "Pass"; otherwise print the word "Fail".
11. Create a second function that does the exact same thing as the previous one but uses `ifelse()` instead if `if...else`.
12. Run both functions using the value 52.5 as the input
13. Run both functions using the **vector** of student test scores you created as the input. (Only one will work properly...)

```r
#10. Create a function using if...else
check_pass_fail <- function(score)
{if (score > 50) {print("Pass")} else {print("Fail")}  }

#11. Create a function using ifelse()
check_pass_fail_ifelse <- function(score) {
result <- ifelse(score>50, "Pass", "Fail") }

print(check_pass_fail_ifelse(25)) #practice attempt
```

```
## [1] "Fail"
```

```r
print(check_pass_fail_ifelse(52.5)) #practice attempt
```

```
## [1] "Pass"
```

```r
#12a. Run the first function with the value 52.5
check_pass_fail(52.5)
```

```
## [1] "Pass"
```

```r
#12b. Run the second function with the value 52.5

print(check_pass_fail_ifelse(52.5))
```

```
## [1] "Pass"
```

```r
#13a. Run the first function with the vector of test scores

results <- sapply(test_scores,check_pass_fail)
```

```
## [1] "Pass"
## [1] "Fail"
## [1] "Pass"
## [1] "Pass"
```

```r
#I used chatGPT to figure out how to apply the check_pass_fail for all the different test score vectors

#13b. Run the second function with the vector of test scores
#here I had to rename my results so I could get the results through the ifelse function- I had to look

restults <- ifelse(test_scores > 50, "Pass", "Fail")
results_ifelse <- ifelse(test_scores > 50, "Pass","Fail")
print(results_ifelse) #enter this code to see the answers
```

```
## [1] "Pass" "Fail" "Pass" "Pass"
```

14. QUESTION: Which option of `if...else` vs. `ifelse` worked? Why? (Hint: search the web for "R vectorization")

Answer: They both worked.. I needed to use the sapply for the if.. else to my results because of the multiple vectors the code had to account for. I think the if.. else worked more effiently for me

NOTE Before knitting, you'll need to comment out the call to the function in Q13 that does not work. (A document can't knit if the code it contains causes an error!)