Assignment 2: Coding Basics

Chrissie Pantoja

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

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

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

```
## [1] "Is the mean greater than the median? 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.

```
##   Student Name Test Score On Scholarship
## 1          Ana         85           TRUE
## 2         Zane         92          FALSE
## 3     Chrissie         78           TRUE
## 4        Paula         90          FALSE
```

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

   Answer: The matrix must contain the same data type for all elements, two-dimensional array with rows and columns, where all elements are of the same type, can be used for mathematical computations. Data Frame can contain different data types (numerica, character, etc) in different columns, can have a list of vectors of equal length, and columns can have names.

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".

```
#score already has the vector with all the scores ,
#score is just the name.
evaluate_score <- function(score) {
  if (score > 50) {
    print("Pass")
  } else {
    print("Fail")
  }
}
```

11. Create a second function that does the exact same thing as the previous one but uses `ifelse()` instead if `if...else`.

```
evaluate_score_ifelse <- function(score) {
  result <- ifelse(score > 50, "Pass", "Fail")
  print(result)
}
```

12. Run both functions using the value 52.5 as the input

```
#12a. Run the first function with the value 52.5

result_if_else <- evaluate_score(52.5)
```

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

```
print(result_if_else)
```

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

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

result_ifelse <- evaluate_score_ifelse(52.5)
```

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

```
print(result_ifelse)
```

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

13. Run both functions using the **vector** of student test scores you created as the input. (Only one will work properly...)

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

# Vector of student test scores
#result_if_else_vector <- evaluate_score(test_scores)
#print(result_if_else_vector)

#13b. Run the second function with the vector of test scores

result_ifelse_vector <- evaluate_score_ifelse(test_scores)
```

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

```
print(result_ifelse_vector)
```

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

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

    Answer: It worked ifelse (). When working with vectors, ifelse() checks the condition for each element in the vector and applies the TRUE or FALSE branches accordingly for each element. In the other hand, if...else is scaled-based, so it evaluates a single condition. It doesn't evaluate each element in the vector independently.

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!)