Assignment 2: Coding Basics

Nargis Taraki

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.

```
#Here I inserted the command to creat the sequence
#1.
sequence <- seq(1, 55, by = 5)
sequence
```

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

```
#Here I computed Mean and Median of the sequence.

#2.
Mean_Seq <- mean(sequence)
Median_Seq <- median(sequence)
Mean_Seq
```

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

```
Median_Seq
```

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

```
#Here I asked R if Mean sequence is greater than Median of Sequance:
#3.
Mean_Seq > Median_Seq
```

```
## [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.

```
# In Vector1 I recorded names of student.
Students_names <- c("Rachel", "Bob", "David", "John")

#In vector2 I assigned test scores.
Test_scores <- c(50, 75, 65, 80)

#Vector3 indicates whether each student has a scholarship.
Scholarship <- c(TRUE, FALSE, TRUE, FALSE)

#Here I made a dataframe.
Students_data <- data.frame(Name = Students_names, Score = Test_scores, Scholarship = Scholarship)
Students_data
```

```
##     Name Score Scholarship
## 1 Rachel    50        TRUE
## 2    Bob    75       FALSE
## 3  David    65        TRUE
## 4   John    80       FALSE
```

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

   Answer: A data frame is different from a matrix in that it can contain columns of different data types (such as, numeric, character, logical) and has named columns and rows for better organization. A matrix, on the other hand, only contains one type of data and does not inherently support different data types within the same structure.

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...)

```
#10. Create a function using if...else
Pass_fail_ifelse <- function(Score){ if(Score>50){ return("Pass")} else { return("Fail")}}

#11. Create a function using ifelse()
Pass_fail_ifelse_Vector <- function(Score) {return(ifelse(Score >50, "Pass", "Fail"))}

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

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

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

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

```
#13a. Run the first function with the vector of test scores
#Pass_fail_ifelse(Test_scores)

#this command shows an error and not working.

#13b. Run the second function with the vector of test scores
Pass_fail_ifelse_Vector(Test_scores)
```

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

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

    Answer: The ifelse() function worked because it is vectorized, which means it can operate over entire vectors at once, while if...else is not vectorized and only works with individual elements.

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!)