Assignment 2: Coding Basics

Lauren Ng

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.

```r
#1. Generate sequence from 1 to 55, increasing by 5. Store this as an object called sequence_five
seq(1,55,5)
```

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

```r
sequence_five <- seq(1,55,5)
sequence_five
```

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

```r
#2. Compute the mean and median of the stored object, sequence_five
mean(sequence_five)
```

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

```
median(sequence_five)
```

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

```
#3. Determine if it is true that the mean of sequence_five is greater than the median
mean(sequence_five)>median(sequence_five)
```

```
## [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.

```
# Create vector with student names
vector_studentnames <- c("Janet", "Ming", "Ariel", "Jose") #Vector type: Character
vector_testscores <- c(94,78,88,90) #Vector type: numeric
vector_scholarship <- c(TRUE,TRUE,FALSE,FALSE) #Vector type: logical
Student_Data <- data.frame("Name"=vector_studentnames,"Score"=vector_testscores, "Scholarship"=vector_s
Student_Data
```

```
##    Name Score Scholarship
## 1 Janet    94        TRUE
## 2  Ming    78        TRUE
## 3 Ariel    88       FALSE
## 4  Jose    90       FALSE
```

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

> Answer:Data frames can contain multiple classes of data, while matrices can only contain one class of data.A matrix has fixed number of rows and columns, while data frame has variable number of rows and columns.

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
Greater_50 <- function(x){
  if(x > 50) {
    print("PASS")
  }
  else {
    print ("FAIL")
  }
}

#11. Create a function using ifelse()
Greater_50_short <- function(x){
  ifelse(x>50,print("PASS"),print("FAIL"))
}

#12a. Run the first function with the value 52.5
Greater_50(52.5)
```

```
## [1] "PASS"
```

```r
#12b. Run the second function with the value 52.5
Greater_50_short(52.5)
```

```
## [1] "PASS"
```

```
## [1] "PASS"
```

```r
#13a. Run the first function with the vector of test scores
# Greater_50(vector_testscores) #this is the function that does not work

#13b. Run the second function with the vector of test scores
Greater_50_short(vector_testscores)
```

```
## [1] "PASS"
```

```
## [1] "PASS" "PASS" "PASS" "PASS"
```

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

    Answer:The ifelse worked. The ifelse gives an output vector equal to length of x, whereas if function gives output of length 1. This is because the if statement is not vectorized. For vectorized statements, we should use ifelse.

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!)