Assignment 2: Coding Basics

Prisha

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. Sequence of numbers from 1 to 55
multiple5 <- seq(1,55,5) #from, to, by
multiple5
```

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

```
#2. Mean and median of sequence
meanofseq5 <- mean(multiple5)
meanofseq5
```

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

```
medianofseq5 <- median(multiple5)
medianofseq5
```

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

```
#3. Determining whether the mean is greater than the median
ifelse(meanofseq5 > medianofseq5, TRUE, FALSE)
```

```
## [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.

```
#Part 5
names <- c("Prisha", "Aishwarya", "Hiranmayi", "Prithveeraj")
test_scores <- c("9","8.5","10","8")
scholarship <- c(FALSE, FALSE, TRUE, TRUE)

#Part 6
class(names) #character
```

```
## [1] "character"
```

```
class(test_scores) #character
```

```
## [1] "character"
```

```
class(scholarship) #logical
```

```
## [1] "logical"
```

```
#Part 7
class_info <- data.frame(names, test_scores, scholarship)
class_info
```

```
##         names test_scores scholarship
## 1      Prisha           9       FALSE
## 2   Aishwarya         8.5       FALSE
## 3   Hiranmayi          10        TRUE
## 4 Prithveeraj           8        TRUE
```

```
#Part 8
colnames(class_info) <- c("Names","Test Scores", "Scholarship Status")
class_info
```

```
##          Names Test Scores Scholarship Status
## 1       Prisha           9              FALSE
## 2    Aishwarya         8.5              FALSE
## 3    Hiranmayi          10               TRUE
## 4 Prithveeraj           8               TRUE
```

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

   Answer: Dataframes can have multiple classes of data while a matrix can have only one class of data.

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".

```
pass_fail_check <- function(x) {
  if(x > 50) {
    print("Pass")
  } else {
    print("Fail")
  }
}

pass_fail_check(59)
```

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

```
pass_fail_check(30)
```

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

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

```
pass_fail <- function(x) {
  ifelse(x>50,"Pass","Fail")
}

pass_fail(39)
```

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

```
pass_fail(68)
```

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

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

```
pass_fail(52.5)
```

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

```
pass_fail_check(52.5)
```

```
## [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
pass_fail(test_scores)
```

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

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

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

    Answer: The ifelse option worked and the if + else option did not. This is because if + else is scalar and expects a single logical value and operates only on it. But ifelse is vectorized and is optimized to handle vectors and process each element 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!)