Assignment 2: Coding Basics

Tori Newton

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 1-55, increasing by 5
sequence1 <- seq(1,55,5)

#2. Mean of sequence1
mean(sequence1)
```

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

```
#2. Median of sequence1
median(sequence1)
```

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

```
#3. Is the mean greater than the median?
mean(sequence1)>median(sequence1)
```

```
## [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.

```
#5. Creating three vectors
vector1 <- c("Audrey","Emily","Shivani","Abby")
vector2 <- c(75,80,90,95)
vector3 <- c(TRUE, FALSE, TRUE, FALSE)

#6. What type is each vector?
class(vector1) #vector1 is character
```

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

```
class(vector2) #vector2 is numeric
```

```
## [1] "numeric"
```

```
class(vector3) #vector3 is logical
```

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

```
#7. Combine the vectors into a data frame with an informative name
student_data <- data.frame(vector1,vector2,vector3)

#8. Label columns of data frame with informative titles
colnames(student_data) <- c("Student Name", "Test Score", "On Scholarship")
```

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

   Answer: A data frame is more general than a matrix, columns can have different modes. In a matrix all elements must be of the same data type, but in a data frame each column can contain a different data type.

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
evaluate_score <- function(score) {
  if (score > 50) {
    print("Pass")
  } else {
   print("Fail")
  }
}

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

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

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

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

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

```r
#13a. Run the first function with the vector of test scores
#evaluate_score(vector2)

#13b. Run the second function with the vector of test scores
evaluate_score_ifelse(vector2)
```

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

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

    Answer: 'ifelse' is the option that worked. The ifelse function is vectorized, which means that it can apply the conditional logic to each element of a vector in a single call. The 'if...else' function is not vectorized and only works with a single value at a time, which is why it does not work.

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!)