Assignment 2: Coding Basics

Emily Yang

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.
data <- seq (1, 55, 5) # creating sequence
data # displaying sequence
```

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

```
#2.
mean_data <- mean(data) # finding mean
median_data <- median(data) # finding median
mean_data
```

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

```
median_data
```

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

```
#3.
mean_data > median_data # asking R to determine if the mean is greater than the median
```

```
## [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.
student_name <- c("Amy", "Brain", "Carol", "Danny") # character vector
test_score <- c(99, 98, 97, 96) # numeric vector
scholarship_status <- c(TRUE, TRUE, FALSE, FALSE) #logical vector

#7.
student_record <- data.frame(student_name, test_score, scholarship_status)

#8.
colnames(student_record) <-c("Student Name", "Test Score", "Scholarship Status")
```

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

   Answer: Matrix is a 2 dimensional structure that contains only same type of data. Dataframe can contain many different types of data in different 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...)

```
#10. Create a function using if...else
number1 <- function(x) {
  if(x > 50) {
    return("Pass")
  }
  else {
    return("Fail")
  }
}
```

```r
#11. Create a function using ifelse()
number2 <- function(x){
  ifelse(x>50, "Pass", "Fail")

}


#12a. Run the first function with the value 52.5
number1_52.5 <- number1(52.5)

#12b. Run the second function with the value 52.5
number2_52.5 <- number2 (52.5)

#13a. Run the first function with the vector of test scores
# number1_students <- number1(test_score) # got an  error

#13b. Run the second function with the vector of test scores
number2_students <- number2(test_score)
number2_students
```

```
## [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 second option, "ifelse" worked. According to internet search, ifelse is a vectorized function in R, and it runs each element in the vector. Instead, if...else is not vectorized and only processes one element at a time.

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!)