Assignment 2: Coding Basics

Sarah Sussman

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.

```
fiftyfivebyfive_sequence <- seq(1,55,5)
```

2. Compute the mean and median of this sequence.

```
mean(fiftyfivebyfive_sequence)
```

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

```
median(fiftyfivebyfive_sequence)
```

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

3. Ask R to determine whether the mean is greater than the median.

```
# Create object representing the mean of the sequence
fiftyfivebyfive_mean <- mean(fiftyfivebyfive_sequence)
# Create an object representing the median of the sequence
fiftyfivebyfive_median <- median(fiftyfivebyfive_sequence)
# Determine if mean object is greater than the median object
fiftyfivebyfive_mean > fiftyfivebyfive_median
```

```
## [1] FALSE
```

4. Insert comments in your code to describe what you are doing.

```
#1. Generate a sequence of numbers from 1-55, increasing by fives called
#"fiftyfivebyfive_sequence".

#2. Compute the mean and median of "fiftyfivebyfive_sequence".

#3. Is the mean of "fiftyfivebyfive_sequence" greater than the median of
#"fiftyfivebyfive_sequence"?
```

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).

```
studentnames <- c("Sarah", "Jake", "Megan", "Natalie")
testscores <- c(88, 94, 97, 85)
scholarship <- c(TRUE, FALSE, TRUE, FALSE)
```

6. Label each vector with a comment on what type of vector it is.

```
studentnames # Character
```

```
## [1] "Sarah"   "Jake"    "Megan"   "Natalie"
```

```
testscores # Numeric
```

```
## [1] 88 94 97 85
```

```
scholarship # Logic
```

```
## [1]  TRUE FALSE  TRUE FALSE
```

7. Combine each of the vectors into a data frame. Assign the data frame an informative name.

```
student_info_df <- data.frame("Student name" = studentnames,
                              "Test score" = testscores,
                              "Scholarship" = scholarship)
student_info_df
```

```
##   Student.name Test.score Scholarship
## 1        Sarah         88        TRUE
## 2         Jake         94       FALSE
## 3        Megan         97        TRUE
## 4      Natalie         85       FALSE
```

8. Label the columns of your data frame with informative titles.

```
# I did this in step 7.
```

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

   Answer: A matrix would have labeled rows and columns, the data frame created above just has 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".

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

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

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

```
## function(x) {
##   ifelse(x>50, "Pass", "Fail")
## }
```

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

```
passfail(52.5)
```

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

```
passfail_ifelse(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...)

```
#passfail(testscores)
passfail_ifelse(testscores)
```

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

```
#10. Create a function using if...else

#11. Create a function using ifelse()

#12a. Run the first function with the value 52.5

#12b. Run the second function with the value 52.5

#13a. Run the first function with the vector of test scores

#13b. Run the second function with the vector of 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' statement did not. This is because the 'ifelse' statement is vectorized, meaning it can evaluate each condition in the vector. The 'if'...'else'... statement is not vectorized, so it can only evaluate one condition 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!)