Assignment 2: Coding Basics

Samantha Jensen

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. creating a sequence of numbers from 1 to 55 counting by 5, inclusive.
seq1 <- seq(1,55,5)
seq1
```

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

```
#2. Calculating the mean and median of seq1
mean(seq1)
```

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

```
median(seq1)
```

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

```r
#3. Stacked if statements to compare the mean and median of the vector sequence.
compare <- function() { if (mean(seq1) < median(seq1)) {
  print("Median is greater.")
} else if (mean(seq1) > median(seq1)) {
  print("Mean is greater.")
} else {
  print("They are equal")
}
  }
compare()
```

```
## [1] "They are equal"
```

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.

```r
names <- c("Joe", "Sally", "Leo", "Polly") #Character
scores <- c(85,95, 75, 90) # Numeric
scholarship <- c(TRUE, FALSE, FALSE, TRUE) #Logical

# Creating a data frame with the 3 vectors
student_info <- data.frame(names, scores, scholarship)
# Naming Columns
names(student_info) <- c("Name","Test Score","Scholarship")
## View(student_info)
```

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

   Answer: This data fram is different from a matrix because it contains multiple different data types.

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

function1 <- function() {if (x > 50) {
  print("Pass")
} else {
  print("Fail")
}}

#11. Create a function using ifelse()
function2 <- function() { ifelse(x>50,print("Pass"), print("Fail"))
}

#12a. Run the first function with the value 52.5
x <- 52.5
function1()
```

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

```r
#12b. Run the second function with the value 52.5
function2()
```

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

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

```r
#13a. Run the first function with the vector of test scores
x <- scores
## function1()
#13b. Run the second function with the vector of test scores
function2()
```

```
## [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' function worked. This one worked because the function was able to operate on all of the elements in the vector at once without needing to loop through. the 'if'...'else' function tries to act on each element one at a time and the code doesn't run because the condition it is trying to check has a length greater than 1.

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!)