Assignment 2: Coding Basics

Brock Keller

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_by_five <- seq(1,55,5) #defining a sequence of numbers from 1, to 55, by 5, and giving it a na
sequence_by_five #calling that sequence
```

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

```
#2.
mean <- mean(sequence_by_five) #defining mean of my sequence
median <- median(sequence_by_five) #defining median of my sequence
mean #calling on mean
```

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

```
median #calling on median
```

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

```
#3.
mean>median #simple true or false; is the mean 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.

```
student_names <- c("Eenie","Meenie","Minie","Moe") #character
test_scores <- c(60,95,40,80) #numeric
scholarship <- c(TRUE,TRUE,FALSE,TRUE) #logical

class(student_names)
```

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

```
class(test_scores)
```

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

```
class(scholarship)
```

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

```
Student_Results <- data.frame(Name=student_names, Score=test_scores, Scholarship_Status=scholarship)
Student_Results
```

```
##     Name Score Scholarship_Status
## 1  Eenie    60               TRUE
## 2 Meenie    95               TRUE
## 3  Minie    40              FALSE
## 4    Moe    80               TRUE
```

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

   Answer: The data frame is different from a matrix becuase it includes multiple types of variables. My data frame has character, numeric, and logical data in the same frame whereas a matrix would be limited to only data of the same 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...)

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

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

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

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

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

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

```
#13a. Run the first function with the vector of test scores
#first_function_vector <- test_results(test_scores)

#first_function_vector

#13b. Run the second function with the vector of test scores
second_function_vector <- test_results_modified(test_scores)

second_function_vector
```

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

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

    Answer: ifelse is the only one that worked. if...else gives back the error code "condition has length >1". This is because if...else only evaluates a single element while ifelse() evaluates and returns something for every element in a vector. ifelse() returns a vector of its own results.

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!)