Assignment 2: Coding Basics

Maeve Gualtieri-Reed

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.
ByFives <- seq(1, 55, 5) #1 is the start, 55 is the end, 5 is the increment. Assigns to the vector "ByF

#2.
mean(ByFives) #returns 26 as the mean of the vector
```

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

```
median(ByFives) #returns 26 as the median of the vector
```

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

```
#3.
mean(ByFives) > median(ByFives) #returns false because the mean is not 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.

```
StudentNames <- c("Maeve", "Eric", "Liz", "Tom") #Vector tyep: string
TestScores <- c(79, 89, 85, 92 ) #vector type: numeric
Scholarship <- c(TRUE, FALSE, FALSE, TRUE) #vector type: boolean
NamesDf<- as.data.frame(StudentNames) #converts the student names to data frame
StudentScores <-cbind(NamesDf,TestScores,Scholarship) #adds test scores and scholarship as columns to t
colnames(StudentScores) <- c("Names", "Test scores", "On scholarship?") #labels columns
```

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

   Answer: a matrix must have all the same type of data (all strings or all numbers for example). A data frame can have different types of data. StudentScores data frame created above has strings, numbers, and booleans.

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
PassOver50<-function(input){
   if (input >50){
     print("Pass")
   }else{
    print("Fail")
   }
  }

#11. Create a function using ifelse()
PassIfElse <- function(input){
  ifelse(input>50, "Pass", "Fail")
}
#12a. Run the first function with the value 52.5
PassOver50(52.5) #Pass
```

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

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

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

```r
#13a. Run the first function with the vector of test scores
#PassOver50(TestScores) #Error in if (input > 50) { : the condition has length > 1
#13b. Run the second function with the vector of test scores
PassIfElse(TestScores) #[1] "Pass" "Pass" "Pass" "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:ifelse worked because the function could operate on the whole vector and return values for each individual element in the vector. if can only take one value, so it returned an error since the input was a vector with multiple values.

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!)