Assignment 2: Coding Basics

Irina Roybal

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.
seq(1,55,5) #creating a sequence with values from 1 to 55, counting by 5
```

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

```
count.by.fives <- seq(1,55,5) #naming the sequence
count.by.fives #checking result
```

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

```
#2.
mean(count.by.fives) #determining the mean of the numeric vector
```

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

```r
median(count.by.fives) #determining the median of the numeric vector
```

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

```r
#3.
mean(count.by.fives)>median(count.by.fives)
```

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

```r
#determining if mean is greater than median, output is FALSE in console
```

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
student.names <- c("Gabby", "Adam", "Edward", "Lionel")
#column with student names, string vector type
test.scores <- c(87, 72, 98, 86)
#column with test scores, numeric vector type
scholarship <- c(FALSE, TRUE, TRUE, FALSE)
#column with scholarship status, logical vector type

student_statuses <- data.frame("Student Name"= student.names,
                        "Test Scores" = test.scores,
                        "Scholarship Status" = scholarship)
student_statuses
```

```
##   Student.Name Test.Scores Scholarship.Status
## 1        Gabby          87              FALSE
## 2         Adam          72               TRUE
## 3       Edward          98               TRUE
## 4       Lionel          86              FALSE
```

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

   Answer: Data frames contain multiple types of data (logical, numeric, and string). Matrices can only support data that is of the same class.

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
function1 <- function (x){
  if (x>50){
    print("Pass")
  }
  else{
   print("Fail")
  }
}
#11. Create a function using ifelse()
function2 <- function(x){
  ifelse(x>50,"Pass", "Fail")
}
#12a. Run the first function with the value 52.5
function1(52.5)
```

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

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

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

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

```
## [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' is the only function that worked because 'if'...'else' only works for single values. The 'ifelse' function is vectorized, so it supports multiple inputs within a vector.

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!)