

Assignment 2: Coding Basics

Andrew Barfield

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_1 <- seq(1,55,5)  
# assigning sequence_1 the values of our sequence from 1 to 55, increasing by 5  
sequence_1
```

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

```
#2.  
median(sequence_1) # assessing the median of sequence_1
```

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

```
mean(sequence_1) # assessing the mean, or average, of sequence_1
```

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

```
#3.  
mean(sequence_1) > median(sequence_1)
```

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

```
# asking R whether the mean of sequence_1 is greater than the median of sequence_1
```

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.

```
#5.  
a <- c("Adam", "Tyler", "John", "Maggie") # Vector type: Character  
b <- c(86, 76, 94, 91) # Vector type: Numeric  
c <- c(TRUE,FALSE,TRUE,FALSE) # Vector type: Logical
```

```
#7.  
student_dataframe <- data.frame(a,b,c)
```

```
#8  
names(student_dataframe) <- c("Students","Scores","Scholarship")
```

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

Answer: This data frame is different from a matrix because it is more general in the sense that it can store multiple different classes or types of data (students, test scores, and scholarships), while a matrix can only contain one class or type of data.

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 of `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  
Passing1 <- function(x)  
  if(x > 50){  
    print("Pass")  
  } else {  
    print("Fail")  
  }
```

```

}
#11. Create a function using ifelse()
Passing2 <- function(x){
  ifelse(x > 50,"Pass","Fail")
}
#12a. Run the first function with the value 52.5
Passing1(52.5)

```

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

```

#12b. Run the second function with the value 52.5
Passing2(52.5)

```

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

```

#13a. Run the first function with the vector of test scores
# Passing1(b)
#13b. Run the second function with the vector of test scores
Passing2(b)

```

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

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

Answer: Only the option “ifelse” worked. The “if ... else” option failed and gave the error message “the condition has length > 1” when referring to the “if” part of the function. The problem with the “if ... else” function is that it only takes in one element, and so only that element can be used. Considering we had a length of 4, or 4 elements, the first function was unable to work. The second function, however, was able to work because it is vectorized and able to act on all elements of 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!)