

Assignment 2: Coding Basics

Aidan Power

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) #seq stands for sequence, first number in the parentheses is the first in the sequence, the
```

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

```
seq(1,55,5) -> Seq1 #Designating a name for this sequence in order that it may be easily used in comput  
#2.  
mean(Seq1) #Using the name I gave the sequence in the mean computation
```

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

```
median(Seq1) #Using the name I gave the sequence in the median computation
```

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

#3.

```
mean(Seq1) > median(Seq1) #The mean being greater than the median is something R can answer with either
```

```
## [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("Aidan", "Lizzy", "Vig", "Schuyler") #Vector Type: Character
test_scores <- c(100, 100, 0, 100) #Vector Type: Numeric
scholarship <- c(TRUE, TRUE, FALSE, TRUE) #Vector Type: Logical

Student_Academics <- data.frame(student_names, test_scores, scholarship)
Student_Academics
```

```
##   student_names test_scores scholarship
## 1      Aidan         100          TRUE
## 2      Lizzy         100          TRUE
## 3        Vig          0          FALSE
## 4    Schuyler         100          TRUE
```

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

Answer: While a matrix can only contain one type of vector for all columns, a dataset can have multiple kinds of vectors.

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

```
#11. Create a function using ifelse()  
if.else2 <- function(input) {ifelse(input > 50, "Pass", "Fail")}
```

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

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

```
#12b. Run the second function with the value 52.5  
if.else2(52.5)
```

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

```
#13a. Run the first function with the vector of test scores  
#if.else(test_scores)
```

```
#13b. Run the second function with the vector of test scores  
if.else2(test_scores)
```

```
## [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` worked because of it being a vectorized version of `if...else`. `ifelse` is a vector function based on how a simple set of values can be turned into an output. Since the `if...else` function does not work for vectors, an error message was read.

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!)