

Assignment 2: Coding Basics

Kayla Emerson

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.
basic_sequence <- seq(1, 55, 5) #creating a new object called basic sequence that is a list of numbers
#2.
mean(basic_sequence) #calculating the mean of the sequence
```

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

```
median(basic_sequence) #calculating the median of the sequence
```

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

```
#3.
mean(basic_sequence) > median(basic_sequence) #logical expression is mean is greater than median; R wil
```

```
## [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("Kayla", "Sam", "Nicole", "Victoria") #character vector
test_scores <- c(90, 95, 100, 85) #numerical vector
scholarship <- c(FALSE, TRUE, FALSE, TRUE) #logical vector

class_dataframe <- data.frame(student_names, test_scores, scholarship)
names(class_dataframe) <- c("Student Name", "Test Score", "On Scholarship?")
```

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

Answer: This data frame is different from a matrix because the data frame has different data types, character, numerical, and logical, whereas a matrix only contains one 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

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

#11. Create a function using ifelse()

```
function_two <- function(x){
  ifelse(x>50, print("Pass"), print("Fail"))
}
```

#12a. Run the first function with the value 52.5

```
function_one(52.5)
```

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

#12b. Run the second function with the value 52.5

```
function_two(52.5)
```

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

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

#13a. Run the first function with the vector of test scores

```
#function_one(test_scores)
```

#13b. Run the second function with the vector of test scores

```
function_two(test_scores)
```

```
## [1] "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: The second function, which uses `ifelse`, worked with the vector of test scores because `ifelse` is a vectorized function in R meaning it can work with vectors and singular elements.

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!)