

Assignment 2: Coding Basics

Natalie Holsclaw

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. Creating a sequence from 1 to 55 increasing by 5s  
seq1 <- seq(1, 55, 5)
```

```
#2. Mean and median of seq1  
mean(seq1) # The mean is 26
```

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

```
median(seq1) # The median is 26
```

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

```
#3. Is the mean greater than the median?  
mean(seq1) > median(seq1)
```

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

```
#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.

```
# Vector of student names
student_names <- c("Maria", "Emma", "Brad", "Charley") # character vector

# Vector of student test scores
test_scores <- c(100,75,82,98) # numeric vector

# Vector identifying whether or not the student has a scholarship
scholarship <- c(TRUE,FALSE,FALSE,TRUE) # logical vector

# Data frame combining all student information
student_information <- cbind(student_names, test_scores, scholarship)
```

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

Answer: Matrices can only contain one data structure while data frames can contain multiple types.

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
```

```
func1 <- function(x) {
  if(x > 50) {
    print("Pass")
  }
  else {
    print("Fail")
  }
}
```

```
#11. Create a function using ifelse()
```

```
func2 <- function(x) {  
  ifelse(x > 50, "Pass", "Fail")  
}
```

```
#12a. Run the first function with the value 52.5  
func1(52.5)
```

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

```
#12b. Run the second function with the value 52.5  
func2(52.5)
```

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

```
#13a. Run the first function with the vector of test scores  
# func1(test_scores)  
# Error in if (x > 50) { : the condition has length > 1
```

```
#13b. Run the second function with the vector of test scores  
func2(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: The ‘`ifelse`’ function worked while the ‘`if ... else`’ did not. The ‘`if ... else`’ statement only takes one value and is not vectorized while the ‘`ifelse`’ function is.

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!)