# Assignment 2: Coding Basics

Cal Oakley

## OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on coding basics.

## Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your first and last name into the file name (e.g., "FirstLast_A02_CodingBasics.Rmd") prior to submission.

## Basics Day 1

1. Generate a sequence of numbers from one to 100, increasing by fours. 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.
seq100by4 <- seq(0,100,4) #4. creating sequencing beginning at 0, ending at 100,
#4. with a step of 4 (every 4th value between 0 & 100 is included)

#2.
seq_mean <- mean(seq100by4) #4. calculating mean and assigning it a name for
#4. later
seq_median <- median(seq100by4) #4. calculating median and assigning it a name
#4. for later
#3.
seq_mean > seq_median #4. testing whether the mean of my sequence is greater
```

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

```
#4. than the median, answer should be FALSE since they are both '50'
```

## Basics Day 2

5. Create a series of vectors, each with four components, consisting of (a) names of students, (b) test scores out of a total 100 points, and (c) whether or not they have passed the test (TRUE or FALSE) with a passing grade of 50.
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("Emma","Rae","Cal","Sam","Aidan","Liam") #6. chr
b <- runif(6, 0, 100) #6. num
c <- b > 50 #6. logi

#7.+ #8.
score_df <- data.frame("Name" = a, "Score" = b, "Pass?" = c)
```

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

   Answer: Data frames differ from matrices in that they can contain multiple data types (like characters, numbers, and boolean values in this case) while matrices can only contain a single data type. Matrices also have a fixed number of rows and columns while data frames do not.

10. Create a function with an if/else statement. Your function should determine whether a test score is a passing grade of 50 or above (TRUE or FALSE). You will need to choose either the `if` and `else` statements or the `ifelse` statement. Hint: Use `print`, not `return`. The name of your function should be informative.

11. Apply your function to the vector with test scores that you created in number 5.

```
#10.
# grade_tests <- function(x){
#   if(x > 50) {
#     print("Pass")
#   }
#   else {
#     print("Fail")
#   }
# }
#^first attempt at #10. but it only ran the function on the first value in the
#Score field
grade_tests <- function(x){
  ifelse(x > 50, print("Pass"), print("Fail"))
}
#^when applied, this one ran the 'test' on all six entries in the Score field
#11.
grade_tests(score_df$Score)
```

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

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

12. QUESTION: Which option of `if` and `else` vs. `ifelse` worked? Why?

    Answer: 'ifelse' worked the way I wanted while 'if' followed by 'else' didn't. More details can be found in my comments above. One other odd thing happened too though. 'ifelse' also printed each of the different outputs that the function produced as individual outputs prior to printing the outputs of the entire Score field. I couldn't find a way to prevent this.