# Assignment 2: Coding Basics

Sierra Kindley

## 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 last name into the file name (e.g., "Salk_A02_CodingBasics.Rmd") prior to submission.

The completed exercise is due on Tuesday, January 21 at 1:00 pm.

## 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. sequence1 <- seq(1, 100, by = 4) #create a sequence from 1 to 100 (beginning with 1) increasing by

#2. mean(sequence1); median(sequence1) #compute the mean of the sequence 'sequence1'; compute the media

#3. mean(sequence1) > median(sequence1) #ask R if the mean of 'sequence1' is greater than the median of
```

## 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.

```
students <- c("Anna", "John", "Katie", "Zach") #create a four-component character vector of names of st
scores <- c(84, 79, 90, 48) #create a four-component numeric vector of student test scores and assign t
passed <- c(TRUE, TRUE, TRUE, FALSE) #create a four-component logical vector denoting whether or not ea
testresults <- data.frame(students, scores, passed) #combine the vectors 'students', 'scores', and 'pas
names(testresults) <- c("Student Name", "Test Score", "Passed") #label the columns of the data frame 't
testresults #generate an output of the data frame 'testresults' in the console
```

```
##   Student Name Test Score Passed
## 1         Anna         84   TRUE
## 2         John         79   TRUE
## 3        Katie         90   TRUE
## 4         Zach         48  FALSE
```

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

   Answer: This data frame is different from a matrix in that it is comprised of columns of different data types (character, numeric, logical). In contrast, a matrix contains elements that are the same type of data.

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.

```
passtest <- function(x) {
  result <- ifelse(x >= 50, "Pass", "Fail")
  print(result)
} #create a function with the name 'passtest' that determines whether a test score is a passing grade o
passtest(scores) #apply the function 'passtest' to the vector 'scores' and print the results of the fun
```

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

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

    Answer: The option of writing a function using the 'ifelse' statement worked. I believe this is because using the 'ifelse' statement allows you to evaluate the entire vector based on the given conditions and print a result for each of the elements in the vector at once. If 'if' and 'else' statements were used alone, R would be unable to evaluate and generate an output for each of the elements in the vector all at once. When I tried to use 'if' and 'else' statements to write my function, I received a warning message in the console saying the condition had a length > 1 and that only the first element would be used. As they were written, the 'if' and 'else' statements would only be able to evaluate and print a result for one of the elements in the vector at a time (and only the first element in the vector would be used) without additonal code accompanying them.