

# Assignment 2: Coding Basics

Ayden Schirmacher

## 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.
seq1<-seq(1,55,5) #sequence of numbers 1-55 counting by 5s, assigned name seq1

#2.
mean<-mean(seq1) #find mean of seq1, value named mean
median<-median(seq1) #find median of seq1, value named median

#3.
mean>median #see whether mean is greater than median
```

```
## [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.

```
vectorA<-c("Riley", "Erin", "Max", "Ryan") #vector type: character; student names
vectorB<-c(98, 94, 87, 32) #vector type: numeric; test scores
vectorC<-c(TRUE, FALSE, TRUE, FALSE) #vector type: logical; scholarship status

data.frame.vectors<-data.frame(vectorA, vectorB, vectorC) #creating the data frame
names(data.frame.vectors)<-c("Student Name", "Test Score", "Scholarship Status") #labeling the names of
data.frame.vectors #view data frame
```

```
##   Student Name Test Score Scholarship Status
## 1      Riley      98      TRUE
## 2      Erin      94     FALSE
## 3       Max      87      TRUE
## 4      Ryan      32     FALSE
```

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

Answer: A data frame is different from a matrix because data frames contain elements of the same type while data frames can contain elements of different types (i.e. numeric vs. logical). However, both are 2 dimensional structures.

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
function1<-function(x) {
  if (x/1.5>50) {
    "PASS"}
  else {"FAIL"}
}

#11. Create a function using ifelse()
function2<-function(x){
  ifelse(x/1.5>50, "PASS", "FAIL")}

#12a. Run the first function with the value 52.5
function1(52.5)
```

```
## [1] "FAIL"
```

```
#12b. Run the second function with the value 52.5  
function2(52.5)
```

```
## [1] "FAIL"
```

```
#13a. Run the first function with the vector of test scores  
#--did not work--
```

```
#13b. Run the second function with the vector of test scores  
function2(vectorB)
```

```
## [1] "PASS" "PASS" "PASS" "FAIL"
```

14. QUESTION: Which option of `if...else` vs. `ifelse` worked? Why? (Hint: search the web for “R vectorization”)

Answer: In part 13, only the `ifelse` command worked, not the written our “`if... else...`”. This is because the `ifelse()` function is a shorthand vectorized alternative to the standard “`if...else...`” statement.

**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!)