

# Assignment 2: Coding Basics

Julia Weinberg

## 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.  
seq(1,100,4) #from, to, by  
  
## [1] 1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
```

```
#2.  
mean(seq(1,100,4)) #average number of the sequence  
  
## [1] 49  
median(seq(1,100,4)) #middle number in the sequence
```

```
## [1] 49  
  
#3.  
mean(seq(1,100,4))>median(seq(1,100,4))  
  
## [1] FALSE  
#returns true if mean is greater than median and false in other conditions
```

## 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
vector1 <- c("Julia","John","Max","Sarah") #character vector

vector2 <- c(95,83,48,72) #numeric vector

vector3 <- c(TRUE,TRUE,FALSE,TRUE) #logical vector

#7

passing.scores.dataframe <-data.frame(vector1,vector2,vector3)
#combing vectors into data frame

#8

names(passing.scores.dataframe) <-c("Name","Score","Passed"); View(passing.scores.dataframe) #label col

passing.scores.dataframe

##      Name Score Passed
## 1 Julia     95    TRUE
## 2 John      83    TRUE
## 3 Max       48   FALSE
## 4 Sarah     72    TRUE
```

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

Answer: A matrix only contains data of one type, such as all numeric, all character, or all logical data. However, a data frame can combine different kinds of data into function.

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
student_pass<-function(x){ifelse(x>=50, print(TRUE), print(FALSE))} #function to determine if student h

#11
test_score<-student_pass(passing.scores.dataframe$Score)

## [1] TRUE
## [1] FALSE
test_score

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

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

Answer: 'ifelse' worked and 'if' and 'else' did not work. This is because the 'if' and 'else' statements would only check if there is an occurrence where the statement can be true or false. Therefore based on my data set, when I used 'if' and 'else' it returned TRUE and FALSE only one time

each since both occurred in the dataframe. However, when I used 'ifelse' it returned with TRUE and FALSE an appropriate number of times for how often the condition of  $x \geq 50$  was met.