

# Assignment 2: Coding Basics

Savannah Artusi

## 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.
Assignment1_Question1 <- seq(1, 100, 4) #Sequence from 1 to 100, by 4
Assignment1_Question1

## [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.
Assignment1_Question2a <- mean(Assignment1_Question1) #Mean of Sequence from 1 to 100, by 4
Assignment1_Question2a

## [1] 49

Assignment1_Question2b <- median(Assignment1_Question1) #Median of Sequence from 1 to 100, by 4
Assignment1_Question2b

## [1] 49

#3.
Assignment1_Question2a > Assignment1_Question2b

## [1] FALSE
#True or False: The Mean of the Sequence is greater than the median
```

## 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.

```
ComponentA_NamesOfStudents <- c("Sushito", "Harpa", "Haukea", "Bob") #Create list of names  
ComponentA_NamesOfStudents #Vector Type: Character
```

```
## [1] "Sushito" "Harpa" "Haukea" "Bob"
```

```
ComponentB_TestScores <- floor(runif(4, min=0, max=101)) #Generate Scores  
ComponentB_TestScores #Vector Type: Numeric
```

```
## [1] 45 63 70 68
```

```
ComponentC_GradingScheme <- function(x){  
  ifelse(x<50, "FALSE", "TRUE")
```

```
} #Create function to see if scores are passing
```

```
ComponentC_IsItAPassingGrade <- ComponentC_GradingScheme(ComponentB_TestScores)  
#Run scores through function
```

```
ComponentC_IsItAPassingGrade #Vector Type: Logical
```

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

```
StudentInfo_NamesScoresGrades <- data.frame("Name" = c(ComponentA_NamesOfStudents),  
                                             "Score" = c(ComponentB_TestScores), "Passed" =  
                                             c(ComponentC_IsItAPassingGrade))
```

```
#Create the data frame with column names
```

```
StudentInfo_NamesScoresGrades #Data Frame
```

```
##      Name Score Passed  
## 1 Sushito   45  FALSE  
## 2  Harpa   63   TRUE  
## 3 Haukea   70   TRUE  
## 4   Bob   68   TRUE
```

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

Answer: This data frame is different from a matrix because although the vectors do have the same length (which is required both for a matrix and a data frame), they have different modes.

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.

```
#Same function as before--sorry, I did it this way before reading on!
```

```
ComponentC_GradingScheme <- function(x){  
  ifelse(x<50, "FALSE", "TRUE")
```

```
}
```

```
ComponentC_IsItAPassingGrade <- ComponentC_GradingScheme(ComponentB_TestScores)
ComponentC_IsItAPassingGrade
```

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

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

Answer: I chose “`ifelse`” because it is simpler.