

# Assignment 2: Coding Basics

Jessica Citrola

## 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. Creating a sequence of numbers, first number is "from", second number is "to", last number is "by".  
sequence <- seq(1, 100, 4)
```

```
#2. Calculated the mean and median for the above sequence and assigned names for each.  
m<- mean(sequence)  
m
```

```
## [1] 49
```

```
md <- median(sequence)  
md
```

```
## [1] 49
```

```
#3. Used the names designated in #2 to ask R if mean is greater than median.  
m > md
```

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

## 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, #6
Names <- c("Jessica", "Lucas", "Kristen", "Christina", "Emma", "Tyler", "Connor")
Scores <- c(95, 90, 84, 49, 60, 78, 74)
Pass <- c(TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, TRUE)

#7, #8
StudentGrades <- data.frame(Names, Scores, Pass)
StudentGrades
```

```
##      Names Scores Pass
## 1  Jessica     95  TRUE
## 2   Lucas     90  TRUE
## 3  Kristen     84  TRUE
## 4 Christina     49 FALSE
## 5    Emma     60  TRUE
## 6   Tyler     78  TRUE
## 7  Connor     74  TRUE
```

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

Answer:

A data frame is used for storing data tables and can list a vector of equal length. The columns can have multiple data types, such as characters, numeric, and integers (like a table on an excel sheet). On the other hand, a matrix is a collection of data sets arranged in a two dimensional rectangular format. The data in the matrix has to be the same type of data and the matrix contains a fixed number of rows and columns.

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.

```
PassingGrade <- function(x){
  ifelse(x>50,"TRUE","FALSE")
}
```

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

```
PassingGrade(Scores)
```

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

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

Answer:

When using ‘`if`’ and ‘`else`’, I received an error message that the condition has length  $> 1$  and only the first element will be used. The ‘`if`’ function works on things that are a length of 1. In this case, I have a vector with a length of 7 (7 scores). The function ‘`ifelse`’ works with vector lengths that are greater than 1. In this case, ‘`ifelse`’ has three main arguments, the expression (Scores), what to do if true (write TRUE), what to do if false (write FALSE).