

# Assignment 2: Coding Basics

Iman Byndloss

## 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.

```
one_fiftyfive <- seq(1, 55, 5)
```

2. Compute the mean and median of this sequence.

```
mean(one_fiftyfive)
```

```
## [1] 26
```

```
median(one_fiftyfive)
```

```
## [1] 26
```

3. Ask R to determine whether the mean is greater than the median.

```
mean(one_fiftyfive) > median(one_fiftyfive)
```

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

4. Insert comments in your code to describe what you are doing.

*#1. Creating and naming a sequence which is structured (first #, last #, by #)*

*#2. Using mean() to find the mean, i.e., sum divided by amount of numbers in the sequence, and median()*

*#3. Using conditional statement >, which indicates greater than, to determine whether the mean is greater*

## 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).

```
student <- c("Jimmy","Deshawn","Clementine","Jean") # Named and created a vector with student names, wh
score <- c(70, 95, 82, 99) # Named and created a vector with test scores
scholarship <- c(FALSE,TRUE,FALSE,TRUE) # Named and created a vector with scholarship information, stat
```

6. Label each vector with a comment on what type of vector it is.

```
# class() tells you what type of vector it is
class(student)
```

```
## [1] "character"
```

```
class(score)
```

```
## [1] "numeric"
```

```
class(scholarship)
```

```
## [1] "logical"
```

7. Combine each of the vectors into a data frame. Assign the data frame an informative name.

```
dfSchool <- data.frame(student, score, scholarship) # Combined the vectors into a named data frame, usi
```

8. Label the columns of your data frame with informative titles.

```
names(dfSchool) <- c("Student Name", "Test Score", "Scholarship") # Labeled the columns of dfSchool, us
```

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

Answer: A matrix lists columns of the same length and mode. A data frame is more general than a matrix, listing vectors of equal length but different modes.

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*

```
f1 <- function(x){
  if(x>50) {
    print("Pass")
  }
  else {
    print("Fail")
  }
}
```

*# Used if() to explain what to do if expression is TRUE*

*# Used else() to explain what to do if expression is FALSE*

*#11. Create a function using ifelse()*

```
f2 <- function(x){
  result <- ifelse(x>50, "Pass", "Fail")
  print(result)
}
```

*# Used ifelse() which is structured (expression, if TRUE, if FALSE)*

*# An error occurs when print is included in the if TRUE and if FALSE statement, so it had to be removed*

*#12a. Run the first function with the value 52.5*

```
f1(52.5)
```

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

*#12b. Run the second function with the value 52.5*

```
f2(52.5)
```

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

*#13a. Run the first function with the vector of test scores*

```
# f1(score)
```

*#13b. Run the second function with the vector of test scores*

```
f2(score)
```

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

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

Answer: `if...else()` did not work for the vector of test scores. While `ifelse()`, being capable of running multiple times for different values, can handle vectorized operations, `if...else()` cannot handle vectorized operations due to the fact that it can only run once for a singular value.

**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!)