# Assignment 2: Coding Basics

Damon L. Daniels

## OVERVIEW

This exercise accompanies the lessons 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 Sakai.

### Basics, Part 1

1. Generate a sequence of numbers from one to 30, increasing by threes. Assign this sequence a name.

```
sequence_1_to_30_by_3s <- seq(1,30,3) #sequence from 1 to 30, by 3s
print(sequence_1_to_30_by_3s)
```

```
##  [1]  1  4  7 10 13 16 19 22 25 28
```

2. Compute the mean and median of this sequence.

```
mean (sequence_1_to_30_by_3s) # find mean of sequence = 14.5
```

```
## [1] 14.5
```

```
median (sequence_1_to_30_by_3s) # find median of sequence = 14.5
```

```
## [1] 14.5
```

```
# We are writing codes that give us sequences, assigning new names for codes,
# determining means and medians, and determining conditional statements.
```

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

```
mean(sequence_1_to_30_by_3s) != median(sequence_1_to_30_by_3s) # are the mean
```

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

```
# and median equal (TRUE or FALSE)?
mean(sequence_1_to_30_by_3s) == median(sequence_1_to_30_by_3s) # are the mean
```

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

```
# and median equal (TRUE or FALSE)?
mean(sequence_1_to_30_by_3s) < median(sequence_1_to_30_by_3s) # is the mean less
```

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

```
# than the median (TRUE or FALSE)?
mean(sequence_1_to_30_by_3s) > median(sequence_1_to_30_by_3s) # is the mean
```

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

```
# greater than the median (TRUE or FALSE)?
```

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

```
# We are writing codes that give us sequences, assigning new names for codes,
# determining means and medians, and determining conditional statements to compare the mean and median
```

```
#1.

sequence_1_to_30_by_3s <- seq(1,30,3)

#2.

mean (sequence_1_to_30_by_3s)
```

```
## [1] 14.5
```

```
median (sequence_1_to_30_by_3s)
```

```
## [1] 14.5
```

```
#3.

mean(sequence_1_to_30_by_3s) != median(sequence_1_to_30_by_3s)
```

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

```
mean(sequence_1_to_30_by_3s) == median(sequence_1_to_30_by_3s)
```

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

```
mean(sequence_1_to_30_by_3s) < median(sequence_1_to_30_by_3s)
```

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

```
mean(sequence_1_to_30_by_3s) > median(sequence_1_to_30_by_3s)
```

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

## Basics, Part 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.

```
test_scores <- c(75, 40, 97, 86, 34) #numeric operator (test scores)
student_name <- c("Damon Daniels",
                  "Cornell Jordan",
                  "Aarushi Tripathi",
                  "Maria Mohyuddin",
                  "Angeline Yin-Chia") #character operator (name of student)
test_passed <- test_scores >= 50 #logical operator (test score threshold)
```

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

```
# Please see above.
```

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

```
df_test_outcome <- data.frame(
  "NAME" = student_name,
  "SCORE" = test_scores,
  "STUDENT_PASSED" = test_passed
)

print(df_test_outcome)
```

```
##                NAME SCORE STUDENT_PASSED
## 1     Damon Daniels    75           TRUE
## 2    Cornell Jordan    40          FALSE
## 3  Aarushi Tripathi    97           TRUE
## 4   Maria Mohyuddin    86           TRUE
## 5 Angeline Yin-Chia    34          FALSE
```

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

```
# Please see above.
```

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

   Answer: This data frame contains multiple types/categories of data, whereas a matrix can only contain one.

10. Create a function with an if/else statement. Your function should take a **vector** of test scores and print (not return) whether a given 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.

```
PASSorFAIL <- function(x){
  thegrade <- ifelse(x>=50,TRUE,FALSE)
  print(thegrade)
}
```

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

```
PASSorFAIL(df_test_outcome$SCORE)
```

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

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

    Answer: “if” and “else” did not work b/c there were more than one test score.