# Assignment 2: Coding Basics

GuruBandaa Khalsa

## 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 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. I used the sequence function to generate a sequence of numbers from one to 100, increasing by fours
seq(1, 100, 4)
```

```
##  [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
```

```
four_sequence <- seq(1, 100, 4)

#2. I used the mean and median functions to compute the mean and median of the sequence created above,
mean(four_sequence)
```

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

```
median(four_sequence)
```

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

```
seqmean <- mean(four_sequence)
seqmedian <- median(four_sequence)

#3. I used a conditional statement to ask R to determine whether the mean is greater than the median or
if (seqmean > seqmedian) {
  print("The mean is greater than the median.")
} else if (seqmean < seqmedian) {
  print("The median is greater than the mean.")
} else {
  print("The mean and median are equal.")
}
```

```
## [1] "The mean and median are equal."
```

```
#4. I inserted comments in my code to describe what I am doing.
```

## 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. I created three vectors, each with four components, titled "students," "scores," and "passing."
students <- c("Jimmy", "Johnny", "Timmy", "Victor")  # The vector "students" is a character vector.
scores <- c(99, 76, 45, 95)  # The vector "scores" is a double vector.
passing <- c(TRUE, TRUE, FALSE, TRUE)  # The vector "passing" is a logical vector.

#6. I used the typeof function to determine the type of each vector and labeled each vector with a comm
typeof(students)  # The vector "students" is a character vector.
```

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

```
typeof(scores)  # The vector "scores" is a double vector.
```

```
## [1] "double"
```

```
typeof(passing)  # The vector "passing" is a logical vector.
```

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

```
#7. I combined each of the vectors into a data frame and assigned the data frame the informative name "
test_results <- data.frame(students, scores, passing)
test_results
```

```
##   students scores passing
## 1    Jimmy     99    TRUE
## 2   Johnny     76    TRUE
## 3    Timmy     45   FALSE
## 4   Victor     95    TRUE
```

```
#8. I labeled the columns of my data frame with the informative titles: "Students," "Scores," and "Pass
colnames(test_results) <- c("Students", "Scores", "Passing")
test_results
```

```
##   Students Scores Passing
## 1    Jimmy     99    TRUE
## 2   Johnny     76    TRUE
## 3    Timmy     45   FALSE
## 4   Victor     95    TRUE
```

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

   Answer: This data frame is different from a matrix because this data frame contains three classes of data (character, double, and logical vectors). Matrices can only manage a single class of data.

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.

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

```
#10. I created a function with an if/else statement that takes a vector of test scores and prints wheth
tester <- function(x) {
  tester2 <- ifelse(x > 50, "TRUE", "FALSE")
  print(tester2)
}

#11. I applied my function to the vector with test scores that I created in number 5.
tester(scores)
```

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

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

    Answer: "If" and "else" did not work and "ifelse" did work. This is because ifelse statements are a vector equivalent form of the "if" "else" statement. This was necessary in this scenario since we are working with a vector.