Assignment 2: Coding Basics

Laura Martinez

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. Create a sequence using the seq(from, to, by) by inputting the requirements and name the sequence t

sequence <- seq(1, 100, 4)

#2. Calculate the mean and median of the sequence by inputting into the mean and median function.

mean <- mean(sequence)

median <- median(sequence)

#3. Determine whether the mean is greater than the median. The output will generate if the statement is

mean > median
```

```
## [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. Create 3 vectors with 4 components for each vector. Give vectors a name.

names <- c("Omar", "Jacob", "Kiera", "Dave") # character data
scores <- c(100, 47, 88, 77) # number data
passed <- c(TRUE, FALSE, TRUE, TRUE) #logical data

#7. Combine the 3 vectors into a data frame using the data.frame function and name to save the data fra

student_scores <- data.frame(names, scores, passed)

#8. Label the columns of the data frames using the colnames(dataframe) function.

colnames(student_scores) <- c("Student Name", "Test Score", "Passed")
```

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

   Answer: Data frames can contain different types of data, such as the data frame student_scores, which contains character data, numbers, and logical data. Matrices can only contain one type 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.

```
x <- scores
testScores <- function(x) {
  pass <- ifelse(x>50, "TRUE", "FALSE")
  print(pass)
}
testScores(x)
```

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

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

    Answer: The if and else statement works for specified conditions, whereas, the ifelse function worked because it tested each of the elements in the scores vector.