Assignment 2: Coding Basics

Justin Maynard

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.

```r
sequence_1_to_30 <- seq(1,30,3) #Generated a sequence from 1 to 30 counting by 3
```

2. Compute the mean and median of this sequence.

```r
mean_seq <- mean(sequence_1_to_30)
median_seq <- median(sequence_1_to_30) #Computed mean and meadian values of above sequence
```

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

```r
mean_seq > median_seq #Determined if mean is greater than the median
```

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

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

```r
#1.
#Generated a sequence from 1 to 30 counting by 3
#2.
#Computed mean and median values of above sequence
#3.
#Determined if mean is greater than the median
```

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.

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

```
student_names <- c("a", "b", "c", "e")
test_scores <- c(85, 92, 71, 30)
passed <- c(TRUE, TRUE, TRUE, FALSE) #Created vectors for student names, test scores, and if they passe
```

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

```
student_test_data1 <- as.data.frame(cbind(student_names, test_scores, passed) )
```

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

```
colnames(student_test_data1) = c("Student Name", "Test Score", "Passed?")
```

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

   Answer: A data frame can contain any types of data in any given column, a matrix takes only one type of data. Matrixes do not have column and row names and the number of rows/columns are fixed. Data frames have row and column labels and values can be manipulated within it.

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.

```
function1 <- function(x) {
  for(i in x) {
    if(i >= 50) {
    print(TRUE)
  }
  else {
    print(FALSE)
  }
  }
}

function2 <- function(x) {
  for(i in x) {
    ifelse(i >= 50, print(TRUE), print(FALSE))
  }
}
```

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

```r
function1(test_scores)
```

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

```r
function2(test_scores)
```

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

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

    Answer: Both using the "if" and "else" or "if" and "ifelse" worked for me. They both work as both functions do the same thing, which is categorizing the value in the vector as true or false based on if the value is >= to 50. the Ifelse function just takes up less lines of code and looks 'cleaner'.