Assignment 2: Coding Basics

Quinn Bankson

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.

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. Sequence name: by_threes
by_threes <- seq(1, 30, 3)
by_threes
```

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

```
#2. Mean: 14.5 Median: 14.5
mean(by_threes)
```

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

```
median(by_threes)
```

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

```
#3. FALSE
#Greater than symbol to determine if the mean of by_threes is greater then the median of by_threes. Arr
greater <- mean(by_threes) > median(by_threes)
#viewing 'greater' to reveal its boolean value. A value of "FALSE" for the object greater indicates tha
greater
```

```
## [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.

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.

```
#creating student_name vector with 4 names, text data type
student_name <- c("John", "Josephine", "Enoch", "Hannah")
#creating test_score vector with 4 numbers, num data type
test_score <- c(42, 89, 71, 50)
#creating passed vector with 4 boolean values, logi data type
passed <- c(FALSE, TRUE, TRUE, TRUE)
#creating data frame with name test_restults_df, labeling calumns with informative titles
test_results_df <- data.frame(Names = student_name, Score = test_score, Passes = passed)
print(test_results_df)
```

```
##       Names Score Passes
## 1      John    42  FALSE
## 2 Josephine    89   TRUE
## 3     Enoch    71   TRUE
## 4    Hannah    50   TRUE
```

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

   Answer: This data frame contains mixed data types. num, text, and boolean data types are involved in the data frame. The data frame also has column names that are specified and not set by default.

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.

```
passing_score <- function(test_score){
  passing <- ifelse(test_score >= 50, TRUE, FALSE)
  print(passing)
}
```

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

```
#printing logi data types without returning them with "passing_score" function
passing_score(test_score)
```

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

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

    Answer: ifelse worked best. ifelse allowed the function to work on a vector with a smaller amount of code. ifelse did not need a for or while loop to function for this task.