Assignment 2: Coding Basics

Katie Tack

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.
seq(1, 30, 3) #starting with 1 ending in 30, increasing by 3
```

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

```
thirty_sequence <- seq(1, 30, 3) #naming the sequence
thirty_sequence #same result as line 35, can be done without doing line 35 first
```

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

```
#2.
mean(thirty_sequence) #find mean, doesn't need to be done for finding mean > median
```

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

```
median(thirty_sequence) #find median
```

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

```
thirty_mean <- mean(thirty_sequence) #assign name to mean so it can be used later
thirty_median <- median(thirty_sequence) #assign name to median so it can be used later
#3.
thirty_mean > thirty_median #ask R if the mean is greater than the median
```

```
## [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.

```
student_names <- c('Amanda','Brian','Jeff','Ana','Debbie','Ethan') #character
test_scores <- c(79,42,95,51,88,32) #numeric
if_passed <- test_scores >= 50 #logical
df_student_info <- data.frame(StudentName = student_names, StudentScores = test_scores, Passed = if_pass
colnames <- c('Student Names', 'Test Scores', 'Passed') #renamed columns
```

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

   Answer: This data frame is stored in a table with multiple different data types (character, numerical, and logical) in many columns, whereas a matrix is two-dimensional, can only be one type of data, and is homogenous.

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.

```
passed <- ifelse(test_scores >= 50, 'TRUE', 'FALSE')
print(passed)
```

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

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

Answer: I used 'ifelse' because it is basically a more simplified version of 'if' and 'else' and thus takes up fewer lines of code. It worked as intented and prints out TRUE or FALSE based whether the test scores from the vector I created earlier pass with a score of 50 or higher. I named the function passed so that I could use print(passed) after determining whether the scores passed or not. When I included the print function within the ifelse statement, it would first print TRUE and FALSE and then give me what I was looking for, but making a separate print function solved that issue. I had some troubles with attempting to use 'if' and 'else' for this project, so that is ultimately why I decided to use 'ifelse'. I am still going to play around with the code a bit afterwards to see what else I can do with it and gain some more practice with 'if' and 'else'.