Assignment 2: Coding Basics

Andrew Brantley

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on coding basics.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your first and last name into the file name (e.g., "FirstLast_A02_CodingBasics.Rmd") prior to submission.

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. Creating sequence 1-100, increasing by fours

seq_1to100_by4s <- seq(1,100,4)

#2. Mean and median of seq_1to100_by4s

seq_mean <- mean(seq_1to100_by4s)
seq_median <- median(seq_1to100_by4s)

#3. Determining if mean of seq_1to100_by4s is greater than the median

seq_mean > seq_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.

```
#creating name vector (character vector)
Names <- c("Andrew", "Matthew", "Carter", "Cheryl")

#creating test score vector (number vector)
Test_Scores <- c(78, 49, 56, 91)

#creating passing (true/false) vector (logical vector)
Passing_Score <- c(TRUE,FALSE, TRUE, TRUE)

#creating data frame from these vectors
Student_Test_Results <- data.frame(Names, Test_Scores, Passing_Score)
```

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

   Answer: This data frame is different from a matrix becuase it has vectors/columns in it that are of different types (character, number, logical) while a matrix can only have data of the same mode.

10. Create a function with an if/else statement. Your function should determine whether a 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. Hint: Use `print`, not `return`. The name of your function should be informative.

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

```
#Determining if score is passing or not with function using if, else statement
PassingScore_if_else <- function(x) {
  if(x >= 50) {
      print(TRUE)
    }
  else{
      print(FALSE)
  }
}

#Testing function on Test_Scores vector
PassingScore_if_else(Test_Scores)
```

```
## Warning in if (x >= 50) {: the condition has length > 1 and only the first
## element will be used
```

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

```
#Determining if score is passing or not with function using ifelse statement
PassingScore_TorF_ifelse <- function(x) {
  ifelse(x>=50, TRUE, FALSE)
}

#Testing function on Test_Scores vector
PassingScore_TorF_ifelse(Test_Scores)
```

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

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

    Answer: The `ifelse` option worked because `if` and `else` can only operate on vectors with a length of 1, so only one test score at a time. The `ifelse` option can operate on vectors with

lengths greater than 1 so it can report T/F response when the function is ran on the Test_Scores vector.