Assignment 2: Coding Basics

Maia Griffith

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)  #first created a seq of numbers using seq (start, end, interval)
```

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

```
Thirty_by_3 <- seq(1,30,3) #assigning the seq to a descriptive name

#2.
mean_seq <- mean(Thirty_by_3) #creating a variable for the mean and another for the median
med_seq <- median(Thirty_by_3)

#3.
mean_seq > med_seq #then ask R to compare the two; TRUE (if mean is greater) or FALSE
```

```
## [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.

```
#a)
student_names <- c("Mary", "Carol", "Ryan", "Steve") #making a CHARACTER vector with 4 student names

#b) I used a random number generator to get 4 numbers and made a vector to contain them.
test_score <- c(88, 37, 49, 99) #NUMERIC vector

#c)
result<- (test_score > 50) #LOGICAL vector; returns TRUE/FALSE results

#7.
student_test_results <- data.frame(student_names, test_score, result) #created data frame

#8.changed column names in data frame
colnames(student_test_results) <- c("Student_Name", "Test_Score", "Pass_or_Fail")
```

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

   Answer: Data frames can have multiple classes of data while a matrix can only use one kind.

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.

```
#10. creating my function
pass_or_fail_fctn <- function(test_score) {
  ifelse (test_score > 50, TRUE, FALSE)
}

#11. testing my function
pass_or_fail_fctn(test_score)
```

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

```
#now trying 'if' and 'else' instead, function is able to be created BUT does not end up working
# if_else_fctn_test <- function(test_score) {
#   if (test_score > pass) {
#     print("True")
#   }
#
```

```
#   else {
#     print("False")
#   }
#
# }
#
# if_else_fctn_test(test_score)

#Result gets an error that says the condition has length > 1 (test_score is a vector of length 4)
```

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

    Answer: The `ifelse` worked because it can be used for vectors, while if the `if` and `else` can only be used for a single variable and NOT for vectors.