Assignment 2: Coding Basics

Julia Plasynski

1/21/22

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.

```r
#1.

sequence1 <- seq(1,100,4)
sequence1
```

```
##  [1]  1  5  9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
```

```r
#I named the sequence as "sequence1" and set the range from 1 to 100, increasing by 4.

#2.

mean1 <- mean(sequence1)
median1 <- median(sequence1)

mean1
```

```
## [1] 49
```

```r
median1
```

```
## [1] 49
```

```r
#I named the mean and median as "mean1" and "median1." Then using the sequence from above (sequence1),

#3.
```

```
mean1 > median1
```

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

```
#I typed mean1 > median1 to ask R if the mean is greater than the median.
```

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.

```
names <- c("Julia", "Rachel", "Bob","Sam") #character
scores<- c(100, 89, 70, 33) #double
pass<- c(TRUE,TRUE,TRUE,FALSE) #logical

class_testscores.df <- data.frame("Student_Names"= names, "Test_Scores" = scores, "Did_They_Pass"= pass

class_testscores.df
```

```
##   Student_Names Test_Scores Did_They_Pass
## 1         Julia         100          TRUE
## 2        Rachel          89          TRUE
## 3           Bob          70          TRUE
## 4           Sam          33         FALSE
```

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

   Answer: Data frames can have different types of modes (characters, numbers, dates, etc.), while a matrix cannot. A matrix is a collection with all of the same type of data.

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.

```
passing_grade_testscores_function <-ifelse(scores >= 50, "TRUE", "FALSE")
print(passing_grade_testscores_function)
```

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

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

    Answer: I used the ifelse code since I only needed two options for the output (TRUE vs. FALSE). First the code checked to see if the score was greather than or equal to 50. Depending on the result, it either outputted TRUE or FALSE. Using if and else would have worked too, but this code is easier for this situation.