Assignment 2: Coding Basics

Ashton Cloer

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) #creating sequence of numbers 1-30 by threes
```

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

```
threes_sequence <- seq(1, 30, 3) #naming sequence
threes_sequence #calling out the sequence
```

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

```
#2.

mean(threes_sequence) #calculating sequence mean
```

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

```
median(threes_sequence) #calculating sequence median
```

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

```
#3.

mean(threes_sequence) > median(threes_sequence) #asking R if mean of sequence is greater than the media
```

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

```
median(threes_sequence) > mean(threes_sequence) #asking R if median of sequence is greater than mean
```

```
## [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("Ashton", "Beth", "Caroline", "Deanna") #character
student_names
```

```
## [1] "Ashton"   "Beth"     "Caroline" "Deanna"
```

```
test_scores <- c(100, 80, 75, 40) #numerical
test_scores
```

```
## [1] 100  80  75  40
```

```
pass_or_fail <- c(TRUE, TRUE, TRUE, FALSE) #logical

df_student_test_scores <- as.data.frame(student_names)

df_scores <- as.data.frame(test_scores)

df_logical <- as.data.frame(pass_or_fail)

df_student_test_scores_values <- cbind(df_student_test_scores, df_scores, df_logical)

colnames(df_student_test_scores_values)[1] ="Names"
colnames(df_student_test_scores_values)[2] ="Score"
colnames(df_student_test_scores_values)[3] ="Passing"
```

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

   Answer: This data frame combines various types of data such as characters, numeric, and logical data. A matrix can only hold a single data type.

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.

```
check_passing_grades <- ifelse(test_scores >=50, "Passing", "Failing")

print(check_passing_grades)
```

```
## [1] "Passing" "Passing" "Passing" "Failing"
```

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

    Answer: Ifelse worked for the function. I believe it is because ifelse works best with the entire string of data in the vector test_scores whereas if and else seemed to only work on individual values in the vector.