Assignment 2: Coding Basics

Taylor Coleman

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 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. Creation of a sequence function that starts with 1 and ends at 100, increasing by fours, with the a
sequence_by_fours <- seq(1, 100, 4)

#2. Determining the mean and the median of "sequence_by_fours" using the "mean" and "median" functions:
mean(sequence_by_fours)
```

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

```r
median(sequence_by_fours)
```

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

```r
#3. 'Asking' R if the mean is greater than the median by using a conditional statement:
mean(sequence_by_fours) > median(sequence_by_fours)
```

```
## [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.

```
# 5. (a) Names of students, featuring a vector consisting of characters:
student_names <- c("Ana", "Bob", "Cameron", "David")

# 5. (b) Test scores of students, featuring a vector that consists of numbers:
test_scores <- c(86, 97, 85, 65)

# 5. (c) Determining whether or not students have passed their test, with a passing grade of 50 or abov
pass_fail <- c("Ana"= TRUE, "Bob"= TRUE, "Cameron"= TRUE, "David"= TRUE)

# 7. Combining each of the three vectors into one data frame. Please note that I left my column names a
smart_class <- data.frame(student_names, test_scores, pass_fail)
row.names(smart_class) <- NULL
print(smart_class)
```

```
##   student_names test_scores pass_fail
## 1           Ana          86      TRUE
## 2           Bob          97      TRUE
## 3       Cameron          85      TRUE
## 4         David          65      TRUE
```

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

   Answer: This data frame differs from a matrix because it contains data elements of different modes, combining numbers with characters and logical variables from the three sets of vectors.

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 a function with an if/else statement, using the 'ifelse' option.
passing_score <- function(test_scores){
A <- ifelse(test_scores >= 50, TRUE, FALSE)
return(A)
}

# Attempting the 'if' and 'else' statements:
passing_score_2 <- function(test_scores){
B <-  if(test_scores >= 50) {
    print("TRUE")
```

```
  }
  else {
    print("FALSE")
  }
return(B)
}

# 11. Applying the two 'if/else' function options to my vector for test scores.
application_A <- passing_score(test_scores); application_A
```

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

```
application_B <- passing_score_2(test_scores = 60); application_B
```

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

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

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

    Answer: The 'ifelse' command worked because it can handle vectors whereas the 'if' and 'else' command resulted in an error because it can only handle a single value (I set the test score value equal to an arbitrary value of 60 for "application_2" to test this, and only under this defined value did it work)!