Assignment 2: Coding Basics

Yeeun Kim

OVERVIEW

This exercise accompanies the lessons/labs 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 Canvas.

Basics, Part 1

1. Generate a sequence of numbers from one to 55, increasing by fives. 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.
onetofiftyfive <- seq(1,55,5) # Using assignment statement, I assigned a sequence a name of "onetofifty

#2.
mean_data <- mean(onetofiftyfive) # I assigned the name to the mean data.
median_data <- median(onetofiftyfive) # I assigned the name to the median data.
mean_data # The mean is 26.
```

```
## [1] 26
```

```
median_data # The median is 26.
```

```
## [1] 26
```

```
#3.
mean_data > median_data # Since the mean and the median is the same, this result is FALSE.
```

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

Basics, Part 2

5. Create three vectors, each with four components, consisting of (a) student names, (b) test scores, and (c) whether they are on scholarship or not (TRUE or FALSE).
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.
vector1 <- c("Amy", "David", "Yeni", "Kevin")
vector2 <- c(10,20,30,40)
vector3 <- c(FALSE, FALSE, TRUE, TRUE)

#6.
class(vector1)
```

```
## [1] "character"
```

```
vector1 # Vector type: character
```

```
## [1] "Amy"   "David" "Yeni"  "Kevin"
```

```
class(vector2)
```

```
## [1] "numeric"
```

```
vector2 # Vector type: numeric
```

```
## [1] 10 20 30 40
```

```
class(vector3)
```

```
## [1] "logical"
```

```
vector3 # Vector type: logical
```

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

```
#7.
df_name_of_student <- data.frame(vector1)
df_test_score <-data.frame(vector2)
df_scholarship <- data.frame(vector3)


#8.
colnames(df_name_of_student) <- c("Student_Name")
colnames(df_test_score) <- c("Test_Score")
colnames(df_scholarship) <- c("Scholarship")

df_name_of_student
```

```
##   Student_Name
## 1          Amy
## 2        David
## 3         Yeni
## 4        Kevin
```

```
df_test_score
```

```
##   Test_Score
## 1         10
## 2         20
## 3         30
## 4         40
```

```
df_scholarship
```

```
##   Scholarship
## 1       FALSE
## 2       FALSE
## 3        TRUE
## 4        TRUE
```

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

   Answer:Data frame can have different data types, such as numeric, character, and logical. However, a matrix only can have columns that have same mode and same length.

10. Create a function with one input. In this function, use `if...else` to evaluate the value of the input: if it is greater than 50, print the word "Pass"; otherwise print the word "Fail".
11. Create a second function that does the exact same thing as the previous one but uses `ifelse()` instead if `if...else`.
12. Run both functions using the value 52.5 as the input
13. Run both functions using the **vector** of student test scores you created as the input. (Only one will work properly...)

```
#10. Create a function using if...else
assignment_1 <- function(x){
  if(x>50){
   return ("Pass")
  }
  else{
    return ("Fail")
  }
}

#11. Create a function using ifelse()
assignment_2 <- function(x){
  ifelse(x>50, "Pass", "Fail")
}

#12a. Run the first function with the value 52.5
assignment_1(52.5)
```

```
## [1] "Pass"
```

```
#12b. Run the second function with the value 52.5
assignment_2(52.5)
```

```
## [1] "Pass"
```

```
#13a. Run the first function with the vector of test scores
#assignment_1(df_test_score)

#13b. Run the second function with the vector of test scores
assignment_2(df_test_score)
```

```
##      Test_Score
## [1,] "Fail"
## [2,] "Fail"
## [3,] "Fail"
## [4,] "Fail"
```

14. QUESTION: Which option of `if...else` vs. `ifelse` worked? Why? (Hint: search the web for "R vectorization")

    Answer: Only the function using ifelse worked. This is because ifelse is vectorized and if—else is not vectorized. Vectorization in R allows to apply a fumnction to an entire vector at once.

NOTE Before knitting, you'll need to comment out the call to the function in Q13 that does not work. (A document can't knit if the code it contains causes an error!)