Assignment 2: Coding Basics

Xueying Feng

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 last name into the file name (e.g., "Salk_A02_CodingBasics.Rmd") prior to submission.

The completed exercise is due on Tuesday, January 21 at 1:00 pm.

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.

```
#1.
Fours_1_100<-seq(1, 100, 4) # from 1 to 100, by 4
Fours_1_100 # renamed the sequence
```

```
##  [1]  1  5  9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89
## [24] 93 97
```

```
#2.
mean(Fours_1_100)
```

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

```
median(Fours_1_100)
```

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

```
#3.
mean(Fours_1_100) > median(Fours_1_100)
```

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

```
mean(Fours_1_100) == median(Fours_1_100)
```

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

```
# According to the results, mean is not greater than the median. They are equal.
```

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.

```
#a
Names<-c("Lily","Bill","Kiki","Elsa") # character vector
Names
```

```
## [1] "Lily" "Bill" "Kiki" "Elsa"
```

```
Scores<-c(80,70,95,45) # numeric vector
Scores
```

```
## [1] 80 70 95 45
```

```
Pass <- c("True", "True", "True", "False") # character vector
Pass
```

```
## [1] "True"  "True"  "True"  "False"
```

```
Name_Score <- data.frame(Names,Scores,Pass)
Name_Score
```

```
##   Names Scores  Pass
## 1  Lily     80  True
## 2  Bill     70  True
## 3  Kiki     95  True
## 4  Elsa     45 False
```

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

   Answer: In a data frame, differnet columns can contain different types of data, but in a matrix, they are 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.

```
Scores
```

```
## [1] 80 70 95 45
```

```
Pass_Or_Not<-c(ifelse(Scores>50, print("TRUE"),print("FALSE")))
```

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

```
Pass_Or_Not
```

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

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

    Answer: `ifelse` worked. If I want to work with vectors that are length > 1, I need to use ifelse. Also, the ifelse function performs the check on each component and returns a vector.