Assignment 2: Coding Basics

Yuxiang Ren

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.

```
#1.
seq(1, 100, 4) #generate 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 93 97
```

```
sequence100 <- seq(1, 100, 4)  #assign a name

#2.
resultmean <- mean(sequence100); resultmean  #compute mean, assign the result a name and show the resul
```

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

```
resultmedian <- median(sequence100); resultmedian  #compute median, assign the result a name and show t
```

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

```
#3.
greater <- resultmean > resultmedian; greater #comparison, assign the result a name, and show the resul
```

```
## [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
Name <- c("Alex" ,"Reino", "Ellen", "Marcia") #vector type: character
Score <- c(78,90,32,66) #vector type: numeric
Pass <- c(TRUE, TRUE ,FALSE, TRUE)#vector type: logical

#7
df_name_score_pass <- data.frame("Name"= Name, "Score"= Score, "Pass" = Pass)
```

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

   Answer: Matrix only contains a single class of data. Similarly, each column in the data frame contains a single class. However, the different columns could be of different classes of data.

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

passing <- function(x){
  ifelse(x < 50, "FALSE", "TRUE")} #not added print in ifelse function, to avoid function print additio

#11
passing(Score)
```

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

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

    Answer: 'ifelse' worked. When using 'if' and 'else' to operate the vector 'Score' with the condition (x<50), there is an error message: "the condition has length > 1". The reason is that in the 'if' and 'else' functions, the input x should be the only integer or object. Multiple values in the vector exceed the requirements of the condition. While in 'ifelse' function, it checks the condition for every element of a vector. Therefore, 'ifelse' worked.