Assignment 2: Coding Basics

Lily Zhang

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. I use sequence to generate numbers. The first two numbers in parentheses mean the range of the sequ
A <- seq(1,30,3)
#2. I compute the mean and median of sequence A and assign mean as mean_A, median as median_A.
mean_A <- mean(A)
median_A <- median(A)
#3. I test if mean_A is greater than median_A and the result is FALSE, which indicates mean is not grea
mean_A > median_A
```

```
## [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.

```
#I assigned a collection of five student names of my friends.
student_name <- c("Lily", "Hope", "Mia", "Christal", "Zhuowen")
#I used this function to randomly generated five integer within the range of 0 to 100, corresponding to
```

```r
test_score <- sample.int(100,5)
#I test if the number generated above is larger or equal to 50. If so, it returns as 'TRUE', if not, it
pass_notpass <- test_score >=50
#I constructed a data frame titled as test_result, encompassing the previous information.
test_result <- data.frame (student_name, test_score, pass_notpass)
```

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

   Answer: Data frame can consist of different data types in each column but matrix can only contain numerical 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.

```r
#I apply an 'ifelse' function that tests whether if the given value equals or exceeds 50 (the standard
test <- function (x) {
  result <- ifelse(x >= 50,"pass","fail")
  print(result)
}
#I assigned the function test with value of test_score generated in section 2.
test(test_score)
```

```
## [1] "pass" "pass" "fail" "fail" "fail"
```

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

    Answer: 'ifelse' works for the previous question because it works well with vectors and data frames while 'if' and 'else' are used strictly for scalar.