Assignment 2: Coding Basics

Aye Nyein Thu

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.

```r
#1. Generate a sequence named "Age". Elements ranges from 1 to 50, increased by 5.
Age <- seq(1,55,5)

#2. Get the mean and the median of "Age" sequence
mean_Age<-mean(Age)
median_Age<-median(Age)

#3. Check if the mean of Age is greater than the median.
mean_Age > median_Age
```

```
## [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.

```r
#5 Vectors
#(a) Student Names
Student_Names <- c("Annie","Bobby","Carl","Dean")

#(b) Test Scores
Test_Score <- c(60,70,80,90)

#(c) Scholarship
Scholarship <- c(TRUE,FALSE,FALSE,TRUE)

#6 Labeling
#(a) The vector of Student Names is the character type.
#(b) The vector of Test Scores is the numeric type.
#(c) The vector of Scholarship is the logical type.

#7 Converting to Data Frame
Student_Database <- data.frame(Student_Names,Test_Score,Scholarship)

#8 Labeling the Columns of Data Frame
names(Student_Database) <- c("Student","Score","Scholarship")
```

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

   Answer: While a matrix includes the two dimensional data structure with elements of the same type, a data frame is more general and may comprise elements of the different mode.The "Student Database" is a data frame type as it includes more than one data type. While student names is the character type, the test score is numeric and the scholarship is logic. Due to its variation in modes, it is suited more with the title of data frame than a matrix.

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...)

```r
#10. Create a function using if...else
Exam_Score1 <- function(x) {if(x>50) {"Pass"} else {"Fail"}}

#11. Create a function using ifelse()
Exam_Score2 <- function(x) {ifelse(x>50,"Pass","Fail")}

#12a. Run the first function with the value 52.5
Exam_Score1(52.5)
```

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

```r
#12b. Run the second function with the value 52.5
Exam_Score2(52.5)
```

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

```r
#13a. Run the first function with the vector of test scores
#Exam_Score1(Test_Score) - Comment out as it did not work in knitting.

#13b. Run the second function with the vector of test scores
Exam_Score2(Test_Score)
```

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

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

    Answer: Only the ‘ifelse’ option worked in Question 13 and the ‘if … else’ option did not work. It is because ‘ifelse’ option is the vectorized fuction which works for all values in the entire vector while the ‘if…else’ works for a single value only.

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!)