Assignment 2: Coding Basics

Ye Khaung Oo

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.

```r
#1. Generating sequence of numbers from 1 to 30. Variable name is num
num <- seq(1,30,3)

#2. Computing Mean and median of the sequence
mean.num <- mean(num)
median.num <- median(num)
summary(num)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    7.75   14.50   14.50   21.25   28.00
```

```r
#3. Finding out whether mean is greater than medium
mean.num > median.num
```

```
## [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.

```
student_name <- c("Luffy","Zoro","Nami","Robin") #Character Vector
test_score <- c(35,45,90,75)  #Numeric Vector
passing_grade <- test_score >= 50 & test_score <=100 #Logical Vector
class(student_name)
```

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

```
class(test_score)
```

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

```
class(passing_grade)
```

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

```
df.student <- data.frame(student_name,test_score,passing_grade) #Creating Data Frame for students
class(df.student)
```

```
## [1] "data.frame"
```

```
names(df.student) <- c("Student Names","Test Scores","Pass?") #Naming the columns
print(df.student)
```

```
##   Student Names Test Scores Pass?
## 1         Luffy          35 FALSE
## 2          Zoro          45 FALSE
## 3          Nami          90  TRUE
## 4         Robin          75  TRUE
```

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

   Answer: Data frame can contain different formats of data while a matrix can contain only one class of data with fixed number of rows and columns. In this data frame, we have different vector formats with varying rows and columns.

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.

```
Pass_or_Fail <- function(test_score) {
    if (test_score >= 50) {
      print(": Pass")
    } else {
      print(": Fail")
    }
}
Pass_or_Fail(35)
```

```
## [1] ": Fail"
```

```
Pass_or_Fail(45)
```

```
## [1] ": Fail"
```

```
Pass_or_Fail(90)
```

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

```
Pass_or_Fail(75)
```

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

```
Pass_status <- function(test_scores) {
  result <- ifelse(test_scores >= 50, TRUE, FALSE)
  print(result)
}

Pass_status(35)
```

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

```
Pass_status(45)
```

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

```
Pass_status(90)
```

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

```
Pass_status(75)
```

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

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

    Answer: Both worked for me in my test.