Assignment 2: Coding Basics

Kaichun Yang

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
### Use seq() to generate a sequence of numbers and assign the sequence as kc1
kc1 <- seq(1,100,4)
print(kc1)
```

```
##  [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
```

```
#2
### Calculate mean and median of the sequence kc1 and assign them as kc2 and kc3
kc2 <- mean(kc1)
kc3 <- median(kc1)
print(kc2)
```

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

```
print(kc3)
```

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

```r
#3 & #4
### Use if/else if/else to compare kc2, the mean of the sequence kc1 and kc3,
### the median of the sequence kc1
### The function checks if the kc2 is bigger, if the kc3 is bigger, if kc2 and
### kc3 are equal in order and when one of them is checked, the solution will be printed.
if (kc2>kc3) {
print('The mean is bigger')
} else if (kc2<kc3){
  print('The median is bigger')
} else {
  print('The mean and median are equal')
}
```

```
## [1] "The mean and median 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.

```r
#5 & #6
### v1's type is character
### v2's type is double
### v3's type is character
v1 <- c('Alex', 'Brandon', 'Clara', 'David')
typeof(v1)
```

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

```r
v2 <- c(33, 55, 77, 99)
typeof(v2)
```

```
## [1] "double"
```

```r
v3 <- c('Flase', 'True', 'True', 'True')
typeof(v3)
```

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

```r
#7 & #8
### Combine v1, v2, v3 together and assign titles
grades <- data.frame('Name' = v1, 'Scores' = v2, 'Pass' = v3)
class(grades)
```

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

```
print(grades)
```

```
##      Name Scores  Pass
## 1    Alex     33 Flase
## 2 Brandon     55  True
## 3   Clara     77  True
## 4   David     99  True
```

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

   Answer: Dataframe is able to contain different types of data, matrix has similar data type in its array. In other word, dataframe is a heterogeneous while matrix is homogeneous. Data frame is a generalized form of matrix and can have column and row names.

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.

```
#10
### The function 'grader' determine if the students pass or not, the function 'result'
###conduct grader and then print its result
result <- function(x){
grader <- function(x){
  ifelse(x < 50, 'FAIL', 'PASS')
}
print(grader(x))
}
```

11. Apply your function to the vector with test scores that you created in number 5.

```
#11
### Use the created function for v2, the vector contains scores
result(v2)
```

```
## [1] "FAIL" "PASS" "PASS" "PASS"
```

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

    Answer: 'ifelse' works since it can process vectors. Other conditonal statements including if, else, if/else, and one line if…else are not vector operations but dealing with single value.