Assignment 2: Coding Basics

Samriddha Ghosh

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. Creating a sequence of numbers from one to 100, increasing by fours.
seq_of_hundred<-seq(1,100,4)    # Object is created which has the sequence of numbers desired
seq_of_hundred
```

```
##  [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. Mean and median of the sequence
mean(seq_of_hundred)
```

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

```
median(seq_of_hundred)
```

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

```
#3. Checking if mean is greater than median for the sequence named sequence_of_hundred
mean(seq_of_hundred) > median(seq_of_hundred)
```

```
## [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.Created 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.
student_name <- c('sam','abhi','jai','dev') #data type is strings inputting names
student_name
```

```
## [1] "sam"  "abhi" "jai"  "dev"
```

```
test_score <- c('77','40','90','55')  #data type is integers inputting test scores out of 100
test_score
```

```
## [1] "77" "40" "90" "55"
```

```
test_results<- ifelse(test_score>50,'Pass','Fail') #data type returning to test_results is string
print(test_results)
```

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

```
#6, Combining vectors into a data frame
student_chart<-data.frame(student_name,test_score,test_results)
student_chart
```

```
##   student_name test_score test_results
## 1          sam         77         Pass
## 2         abhi         40         Fail
## 3          jai         90         Pass
## 4          dev         55         Pass
```

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

   Answer: Compared to the dataframe obtained in the Q. 8, matrices are the types of data structures that must have all columns with the same mode and length. Here, the dataframe has same length but every column has different modes including string and integers.

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
#10. A function is created with 'ifelse' statement, to check whether a given test
#score is a passing grade of 50 or above.
#11. Function applied to the vector with test scores that were created in Q5.
student_results<-function(x)
{
  ifelse(x>50,'TRUE','FALSE')
}
student_results(test_score)
```

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

Comparison of the two methods of if and else

```r
#whole code is commented to avoid issues in knitting. But when executed an error is thrown as:-
#Error in if (x > 50) { : the condition has length > 1

#student_results<-function(x)
#{
# if (x>50)
#    {
#   print('TRUE')
#   }
# else
#   {
#   print('FALSE')
#   }
# }
# student_results(test_score)
```

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

    Answer: ‘ifelse’ worked instead of ‘if’ and ‘else’. This is because ‘ifelse’ is a vectorized version of an if … else control structure every programming language has in one way or the other. However, if and else statement can only work with variables. If we need the same answer for the function student_results, we gotta use the for loop to work on each of the values in a vector.