Assignment 2: Coding Basics

Shubhangi Gupta

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.
Qn1.1 <- seq(1,30,3)  #Forming a sequence of numbers from 1 to 30 that increase
#by 3. Assigning it to a name called Qn1.1
Qn1.1
```

```
##  [1]  1  4  7 10 13 16 19 22 25 28
```

```
#2.
MeanQn1.1 <- mean(Qn1.1)  #Calculating the mean of this sequence and assigning
#it a name
MeanQn1.1
```

```
## [1] 14.5
```

```
MedianQn1.1 <- median(Qn1.1) #Calculating the medium of this sequence and
#assigning it a name
MedianQn1.1
```

```
## [1] 14.5
```

```
#3.
MeanQn1.1 > MedianQn1.1 #Asking R if the mean is greater than the median.
```

```
## [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.

```
StudentNames <- c("Joy", "Ari", "Shu", "Max") #character vector (Qn5&6)
TestScores <- c(45, 80, 72, 50) #numerical vector (Qn5&6)
TestResults <- TestScores >= 50 #logical vector (Qn5&6)
df_TestResults <- cbind(StudentNames, TestScores, TestResults) #Qn7
colnames(df_TestResults)<-c("Name", "Score", "Result") #Qn8
df_TestResults
```

```
##      Name  Score Result
## [1,] "Joy" "45"  "FALSE"
## [2,] "Ari" "80"  "TRUE"
## [3,] "Shu" "72"  "TRUE"
## [4,] "Max" "50"  "TRUE"
```

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

   Answer: Both a dataframe and matrix are rectangular datasets that contain the same number of objects in each column. The main difference is that all the data points/ objects in a matrix need to be of the same type (integers/ characters/ logical/ etc) whereas a dataframe can contain objects/ data of different types.

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.

```
#Using ifelse (Qn10)
TestResults_ifelse <- function(x){
 Results <- ifelse(x >= 50, "TRUE", "FALSE")
    print(Results)
}
TestResults_ifelse(TestScores) #Qn11
```

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

```
#Using if and else - putting the entire code in comments as the document cannot
#knit with code that doesn't work.
#TestResults_ifelse2 <- function(z){
# Results2 <- if (z >= 50 ){
# "TRUE"
#}
#else {
# "FALSE"
#}
#print(Results2)
#}
#TestResults_ifelse2(TestScores)
```

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

    Answer: Only "ifelse" worked. "If and else" gave me an error saying that the condition has a length > 1