Assignment 2: Coding Basics

Candela Cerpa

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.

```
OneToThirthy <- seq(1,30, by=3)
print(OneToThirthy)
```

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

2. Compute the mean and median of this sequence.

```
MeanOTT <- mean(OneToThirthy)
print(MeanOTT)
```

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

```
MedianOTT <- median(OneToThirthy)
print(MedianOTT)
```

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

3. Ask R to determine whether the mean is greater than the median.

```
Greater <- MeanOTT>MedianOTT
print(Greater)
```

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

4. Insert comments in your code to describe what you are doing.

```
#1. I assigned the sequence a name and applied the sequence function,
#noting the value by which to increase
#Print shows me the function's result

#2. I assigned each measure a name and applied the corresponding function
#Print shows me the function's result

#3. The function returns FALSE, so mean is not greater than the median
#Print shows me the function's result
```

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.

```
Student <- c("Candela", "Samantha", "Elizabeth", "Renata")
TestScore <- c(40, 80, 90, 100)
PassedTest <- TestScore > 50
```

6. Label each vector with a comment on what type of vector it is.

```
typeof(Student)
```

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

```
#The Student vector is a character type
typeof(TestScore)
```

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

```
#The TestScore vector is a double type
typeof(PassedTest)
```

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

```
#The PassedTest vector is a logical type
```

7. Combine each of the vectors into a data frame. Assign the data frame an informative name.

```
CourseName <- data.frame(Student, TestScore, PassedTest)
```

8. Label the columns of your data frame with informative titles.

```
colnames(CourseName)<-c("Name","Score","Passed")
```

9. QUESTION: How is this data frame different from a matrix? > Answer: This dataframe has multiple data classes in its various columns (character, numeric, logical), while a matrix can only have a single type of data class.
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.

```
TestCalc1 <- function(x){
      ifelse(x >= 50, "TRUE", "FALSE")
  }

TestCalc2 <- function(x){
     if (x >= 50) {print("TRUE") } else {print("FALSE")}
  }
```

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

```
TestCalc1(TestScore)
```

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

```
TestCalc2(TestScore)
```

```
## Error in if (x >= 50) {: the condition has length > 1
```

12. QUESTION: Which option of `if` and `else` vs. `ifelse` worked? Why? > Answer: if…else doesn't work because it returns values the same shape as the test, which as a boolean only allows for a length of one. Ifelse works because it's the vectorized alternative to if..else, designed to test a vector of booleans and returning a vector of the same length. However, a for loop applied to the if…else function could work. #