Assignment 2: Coding Basics

David Robinson

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.
#Note that this sequence stops at 28 instead of 30 because adding 3 to 28 would result in 31.
seq(1,30,3)
```

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

```
sequence_1To30 <- seq(1,30,3)

#2.

#Assigning the mean of the sequence to a variable name.
mean_sequence1To30 <- mean(sequence_1To30)

#Assigning the median of the sequence to a variable name.
median_sequence1To30 <- median(sequence_1To30)

#Showing the variables in the console to check outputs.
mean_sequence1To30
```

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

```
median_sequence1To30
```

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

```
#3.
#Asking R to determine whether the mean is greater than the median via Boolean logic.
mean_sequence1To30>median_sequence1To30
```

```
## [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.

```
#5 and #6
#This vector is a list of string (character) values.
StudentNames <- c("Leonardo", "Raphael","Michelangelo", "Donatello")

#This vector is a list of integer values.
TestScores <- c(100, 75, 50, 25)

#This vector is a list of Boolean (logical) values.
TestScoresPassing <- c(TRUE, TRUE, TRUE, FALSE)

#7 and #8

df_StudentNamesAndTestScores <- data.frame(StudentNames,TestScores,TestScoresPassing)
df_StudentNamesAndTestScores
```

```
##   StudentNames TestScores TestScoresPassing
## 1     Leonardo        100              TRUE
## 2      Raphael         75              TRUE
## 3 Michelangelo         50              TRUE
## 4    Donatello         25             FALSE
```

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

   Answer: #9 Matrices can only contain a single class of data, while data frames can consist of many different classes of data (Source: https://carpentries-incubator.github.io/lc-litsearchr/07-data-frames-matrices-and-lists/index.html#:~:text=The%20main%20difference%20is%20that,grid%20of%20rows%20and%2

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.

```
evaluateScores <- function(x){
  evaluationX <- ifelse(x>=50,TRUE,FALSE)
  print(evaluationX)
}

#For reviewer / TA -- note that the code below is just reference for Question #12.
# 'evaluateScores2 <- function(y){
 # 'evaluationY <- if(y>50) TRUE else FALSE
  # 'print(evaluationY)
# '}
```

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

```
#11

evaluateScoresFrom5 <- evaluateScores(TestScores)
```

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

```
evaluateScoresFrom5
```

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

```
#For reviewer / TA -- note that the code below is just reference for Question #12.
# 'evaluateScores2(TestScores)
```

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

    Answer: The option 'ifelse' worked. Per demonstration code above for function 'evaluateScores2', the following code is produced: 'Error in if (y > 50) TRUE else FALSE : the condition has length > 1'. It appears that 'if' and 'else' cannot handle vectors as they have length > 1 object.