Assignment 2: Coding Basics

Wanchen Xiong

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics (ENV872L) on coding basics in R.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Use the lesson as a guide. It contains code that can be modified to complete the assignment.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file. You will need to have the correct software installed to do this (see Software Installation Guide) Press the `Knit` button in the RStudio scripting panel. This will save the PDF output in your Assignments folder.
6. After Knitting, please submit the completed exercise (PDF file) to the dropbox in Sakai. Please add your last name into the file name (e.g., "Salk_A02_CodingBasics.pdf") prior to submission.

The completed exercise is due on Thursday, 24 January, 2019 before class begins.

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.

```
# produce a sequence of numbers from 1 to 100 increasing by 4, and assign this sequence as sequence1
sequence1 <- seq(1, 100, 4)
# calculate the mean of sequence1
mean(sequence1)
```

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

```
# calculate the median of sequence1
median(sequence1)
```

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

```
# determine whether the mean of the sequence is greater than its median
mean(sequence1) > median(sequence1)
```

```
## [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.

```
#Create a character vector with four student names
name <- c("Adam", "Brandon", "Charles", "Daniel")
#Create a numerical vector with scores of each student
score <- c(89, 88, 50, 49)
#Create a logical vecror
pass <- score > 50
#Create a dataframe that shows all the student and their score information
studentinfo <- data.frame(name, score, pass)
```

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

   ANSWER: Matrix contains only one type of data like numbers, whereas data frame can contain different types of data.

10. Create a function with an if/else statement. Your function should determine whether a 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. The name of your function should be informative.
11. Apply your function to the vector with test scores that you created in number 5.

```
#create a function using ifelse statement, if the score is equal to or more than 50, then returns true,
check_score <- function(x) {
  ifelse(x >= 50, TRUE, FALSE)
}

#Apply this function to the numerical vector and check the results
pass_or_not <- check_score(score) ; pass_or_not
```

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

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

    ANSWER: “if” and “else” did not work out because it applies to one element; however, the vector had more than one elements, thus only the first element was used. “ifelse” worked because it returned value based on the elements in the condition; it worked with multiple elements.