Assignment 2: Coding Basics

Walker Grimshaw

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.

```
# 1
by_four <- seq(1, 100, by = 4) # assign the sequence from 1 to 100, by four, to the variable "by_four"
# 2
# mean of sequence by_four assigned to variable and computed
mean_by_four <- mean(by_four); mean_by_four
```

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

```
# median of sequence by_four assigned to variable and computed
median_by_four <- median(by_four); median_by_four
```

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

```
# 3
mean_by_four > median_by_four # is the mean greater than the median?
```

```
## [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
# a vector of student names of type character
student_names <- c("Alex", "Babs", "Charles", "Darren")
# b vector of test scores of type numeric
test_scores <- c(96, 55, 67, 80)
# c vector of passing of type logical
student_pass <- test_scores > 50
# 7 create data frame student_scores
student_scores <- data.frame(student_names, test_scores, student_pass)
# 8
names(student_scores) <- c("Name", "Score", "Pass")
```

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

   ANSWER: This data frame is different than a matrix because each of the three columns has data of a different type, and a matrix must have data all of the same type.

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.

```
# 10
pass <- function(x) {
  y <- ifelse(x >= 50, TRUE, FALSE)
  return(y)
}

# 11
pass(test_scores)
```

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

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

    ANSWER: ifelse worked for the function because it returns a value of the same shape as the input. In this case, the input was a vector and the output was a vector.