Assignment 2: Coding Basics

Vicky Jia

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on coding basics.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Salk_A02_CodingBasics.Rmd") prior to submission.

The completed exercise is due on Tuesday, January 21 at 1:00 pm.

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. Define "seq1" as a sequence of number from 1 to 100 increasing by 4
seq1 <- seq(1, 100, 4)
seq1
```

```
##  [1]  1  5  9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89
## [24] 93 97
```

```
#2. Calculate the mean and median of "seq1"
mean(seq1)
```

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

```
median(seq1)
```

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

```
#3. Directly compare mean and median of "seq1" in R.
#If mean is not greater than median, R will return "FALSE"
mean(seq1) > median(seq1)
```

```
## [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.

```
name <- c("Ann", "Bill", "Cindy", "Dan")  #Character vector
score <- c(79, 43, 29, 61)  #Numeric vector
test <- c(TRUE, FALSE, FALSE, TRUE)  #Logical vector

test_result <- data.frame(name,score,test)
names(test_result) <- c("Name", "Score", "Pass")
```

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

   Answer: This data frame can store different types of data. Columns will maintain the original types of the vector. For example, if input vectors are logical vectors, the vector type in the data frame will still be logical. The Environmental tab indicates the vector type of each column in the data frame, which can be checked seperately. Also, in a data frame, each column requires a column name. However, in a matrix, all elements are the same type of data, and the column name is not required.

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. Hint: Use `print`, not `return`. The name of your function should be informative.
11. Apply your function to the vector with test scores that you created in number 5.

```
#pass_score <- function(x) {
#  if(x>= 50) {
#    TRUE
#  }
#  else {
#    FALSE
#  }
#}

pass_score <- function(x) {
  ifelse(x<50, FALSE, TRUE)
  }

check_test <- pass_score(score)
check_test
```

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

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

    Answer: 'ifelse' worked. The list of score has several elements. The 'if' and 'else' statement can only read the first element in the list because the 'if' statement is not vectorized. The 'ifelse' statement operates on the whole logical vector, so it can go through each element.