Assignment 2: Coding Basics

Pierre Mishra

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.

```r
#1. Using the function "seq" to generate numbers from 1-100 and stepping the values by 4
four_seq <- seq(1,100,4)

#2. Calculating mean and median of the sequence and assigning them to "mean" and "median"
mean <- mean(four_seq)
median <- median(four_seq)

#3. Comparing mean and median. It returns FALSE meaning mean is not greater than median
mean > 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.

```
name <- c("John Cena", "Michael Scott", "Tom Haverford") # It is a character vector
score <- c(89, 45, 96) # It is an integer vector
result <- score > 49 # It is a logical vector
stud_info <- data.frame(name, score, result)
names(stud_info) <- c("Student Name", "Exame Score", "Passed?")
stud_info
```

```
##    Student Name Exame Score Passed?
## 1     John Cena          89    TRUE
## 2 Michael Scott          45   FALSE
## 3 Tom Haverford          96    TRUE
```

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

   Answer: A dataframe can contain more than one kind of data types while a matrix can have only one type 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. 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.

```
# Creating functions using both 'if' and 'else', and 'ifelse' methods
pass_nopass <- function(x) {
  if (x<50) {
    print (FALSE)
  }
  else {
    print (TRUE)
  }
}

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

pass_nopass(score) #### This does not work...
```

```
## Warning in if (x < 50) {: the condition has length > 1 and only the first
## element will be used
```

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

```
p_np(score)  #### This works!!!
```

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

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

    Answer: The `ifeslse` method worked. This is because in the `if` and `else` method, the two conditions are separately defined. As soon as the 1st value matces anyone of the conditions, the code ends. On the other hand, `ifelse` is a vectorized form of indexing values. The arguments for`ifelse` are in a single step which allows for multiple values to pass through `ifelse`.