Assignment 2: Coding Basics

Ardath Dixon

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 first and last name into the file name (e.g., "FirstLast_A02_CodingBasics.Rmd") prior to submission.

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.
ListByFourTo100 <- seq(1,100,4) #creates sequence and assign name

#2.
ListByFour_mean <- mean(ListByFourTo100) #assigns name to calculated mean
ListByFour_median <- median(ListByFourTo100) #assigns name to calculated median

ListByFour_mean #shows mean in console
```

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

```
ListByFour_median #shows median in console
```

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

```
#3.
ListByFour_mean > ListByFour_median #test if mean > median. FALSE output since both are 49
```

```
## [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. Create series of vectors
Names <- c("Abe", "Bobbi", "Carly", "Dan") #character vector
Scores <- c(85, 93, 82, 45) #numeric vector
Passing <- c(TRUE, TRUE, TRUE, FALSE) #logical vector

#6. see comments above for vector types, found with the class() function

#7.
df_StudentScores <- data.frame(Names,Scores,Passing)

#8. see labels above in #5
```

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

   Answer: This data frame lists vectors of different modes (character, numeric, & logical), while a matrix requires all columns to have the same mode.

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.

```
# Process with if and else separately
PassingScore_if_else <- function(x){
  if(x<50){print(FALSE)}
  else{print(TRUE)}}

# Process with ifelse together
PassingScore_ifelse <- function(x){
  ifelse(x<50,FALSE,TRUE)}
```

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

```
PassingScore_ifelse(Scores)
```

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

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

    Answer: The 'ifelse' option worked with the test scores created above. When 'if' and 'else' were separated into two commands, it stated that only the first element could be used. However, 'ifelse' was able to apply the process to every term in the vector.