Assignment 2: Coding Basics

Elise Harrigan

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.

```
seq <- seq(1,100,4)
```

2. Compute the mean and median of this sequence.

```
mean_seq <- mean(seq)
median_seq <- median(seq)
#naming the sequence of mean and median
```

3. Ask R to determine whether the mean is greater than the median.

```
mean_seq > median_seq #statement to get a TRUE or FALSE to see if mean is greater than the median.
```

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

4. Insert comments in your code to describe what you are doing.

```
#1. Created a sequence called seq. To get a sequence from 1 to 100 counting by 4's you but (1, 100, 4)

#2. Named mean and median by using mean_seq and median_seq

#3. To find out from R if the mean is greater than the median you are looking for a TRUE/FALSE so I use
```

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.

```
names <- c("tom", "sarah", "jen", "laura")
  #names is a character vector
typeof(names)
```

```
## [1] "character"
```

```
 scores <- c(61, 41, 80, 92)
 #scores are double vector
 typeof(scores)
```

```
## [1] "double"
```

```
  passed <- (scores > 50)
  #passed are logical vector
  typeof(passed)
```

```
## [1] "logical"
```

```
student_df <- data.frame("names"=names, "scores"=scores, "passed"= passed)
#created data frame for the 3 vectors above and labeled the coloumns with informative names
```

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

   Answer: A data frame is a list of vectors with equal lengths and is heterogenous. This is different from a matrix because a matrix homogenous collection of datasets.

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.

```
grades_func <- function(x){
  ifelse(x>50, TRUE, FALSE)
}

# apply scores vector to function
 Passing_grades <- grades_func(scores)

 #print passing_grades
 print(Passing_grades)
```

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

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

    Answer: ifelse worked because the if/else combination gave me an error since the condition was greater than 1 and then only the first element would have run with the function. ifelse runs through each of the 4 components in the score vector when put into the function.