Assignment 2: Coding Basics

Sam Saltman

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.
sequence <- seq(1, 100, 4) #1 = lower range, 100 = upper range, 4 = increasing factor
sequence
```

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

```
#2.
mean(sequence) #Determine mean
```

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

```
median(sequence) #Determine median
```

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

```
summary(sequence) #Finds statistics including mean and median
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1      25      49      49      73      97
```

```
#3.
mean(sequence) > median(sequence) #False as 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.

```
Names.of.students <- c("Sam", "Michael", "Tom", "Jackson")
class(Names.of.students) #Character vector
```

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

```
Test.Scores <- c(100, 80, 60, 40)
class(Test.Scores) #Numeric vector
```

```
## [1] "numeric"
```

```
PassingScore <- c("TRUE", "TRUE", "TRUE", "FALSE")
class(PassingScore) #Character vector
```

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

```
Class.Test.Scores <- data.frame("Name"=Names.of.students,"Score"=Test.Scores,"Pass"=PassingScore)
```

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

   Answer: A data frame can store multiple data types. The data frame created above has both numeric and charactor columns. A matrix is different because it can only store one type of data between columns.

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.

```
Pass.Fail <- function(x){
  ifelse(x>=50, print(TRUE), print(FALSE))
} #Ifelse method - this works when applied to test scores
```

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

```
Pass.Fail(Test.Scores) #Ifelse method produces desired results
```

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

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

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

    Answer: The 'ifelse' works, and the 'if' and 'else' method does not work. The 'if' and 'else' method produces a length greater than 1 and cannot interpret both true and false as a result. The ifelse method does not encounter this issue and can compute both true and false.