Assignment 2: Coding Basics

Israel Golden

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.

```r
#1.
hundred_by_4 <- seq(1,100,4)
hundred_by_4
```

```
##  [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
```

```r
# here in problem number 1 I've created an object with the seq() function that counts
# from 1 to 100 by increments of 4 (1,5,9...) and named that object hundred_by_4

#2.
mean(hundred_by_4)
```

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

```r
median(hundred_by_4)
```

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

```
# Here I’ve used the mean() and median() functions to take the mean and median
# of the hundred_by_4 object I created in the previous exercise.

#3.
mean(hundred_by_4)>median(hundred_by_4)
```

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

```
# Finally, I’ve applied a logical statement to test whether the mean
# of my object (hundred_by_4) is greater than the median of this same object.
# Both the mean and median are 49 so the answer is 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.
student_names <- c(’john’, ’paul’, ’george’, ’ringo’)
student_grades <- c(100, 90, 80, 12)
pass_or_fail <- c(TRUE, TRUE, TRUE, FALSE)

#6.
class(student_names) # this is a character vector because it has text objects in it
```

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

```
class(student_grades) # this is a numeric vector because it is composed of numbers
```

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

```
class(pass_or_fail) # this is a logical vector because it is full of boolean statements
```

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

```
#7 & #8.
student_records_df <- data.frame("student"=student_names,"grade"=student_grades,"pass.fail"=pass_or_fail

student_records_df
```

```
##   student grade pass.fail
## 1    john   100      TRUE
## 2    paul    90      TRUE
## 3  george    80      TRUE
## 4   ringo    12     FALSE
```

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

   Answer: This dataframe is different from a matrix because it has different data types like character, numeric, and logical statements. By contrast a matrix could only hold one type of data (e.g., character OR numeric OR logical statements).

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.

```
auto_grader <- function(x){
  ifelse(x<50, 'FAIL', 'PASS')
}
```

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

```
auto_grader(student_records_df$grade)
```

```
## [1] "PASS" "PASS" "PASS" "FAIL"
```

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

    Answer: In my auto_grader function I chose the 'ifelse' statement to determine whether my four pupils (John, Paul, George, and Ringo [no relation]) passed their test. Here, I will explain why the 'ifelse' statement works for this application. The first part of the 'ifelse' statement is the logical statement, in my case I wanted to know if a given input (x) was less than 50. IF this statement was true, I elected to print FAIL because anything less than 50 was a failing grade in this class. ELSE, meaning the number was not less than 50, would be a passing grade so I elected to print PASS under these conditions. Importantly, 50 is itself a passing grade.