Assignment 2: Coding Basics

Jonathan Gilman

OVERVIEW

This exercise accompanies the lessons/labs in Environmental Data Analytics on coding basics.

Directions

1. Rename this file `<FirstLast>_A02_CodingBasics.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.
6. After Knitting, submit the completed exercise (PDF file) to Canvas.

Basics, Part 1

1. Generate a sequence of numbers from one to 55, increasing by fives. 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.
# create a sequence of numbers from 1 to 55 that increases by 5s
# in this case I will have 1:11 values, and each one is multiplied by 5
seq5 <- c(1:(55/5)*5)

#2.
# mean of the sequence created above
mean_seq5 <- mean(seq5)
# median of the sequence
median_seq5 <- median(seq5)

#3.
# asking if the mean is greater than the median
mean_seq5 > median_seq5
```

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

```
# output = FALSE
```

Basics, Part 2

5. Create three vectors, each with four components, consisting of (a) student names, (b) test scores, and (c) whether they are on scholarship or not (TRUE or FALSE).
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.

```
# create vector for student names
names <- c("Jill", "Jack", "Jane", "John")
class(names) # type of vector = character
```

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

```
# create vector for student test scores
scores <- c(90, 92, 94, 96)
class(scores) # type of vector = numeric
```

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

```
# create vector for if they are on scholarship or not
scholarship <- c(TRUE, FALSE, TRUE, FALSE)
class(scholarship) # type of vector = logical
```

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

```
# create data frame with all three vectors
df_student_info <- data.frame(names, scores, scholarship)

#label columns of data frame
names(df_student_info) <- c("Name","Test Score","Scholarship T/F")
```

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

   Answer: This data frame is different from a matrix because it contains more than one class of data (character, numerical, logical). A matrix can only contain one class of data.

10. Create a function with one input. In this function, use `if...else` to evaluate the value of the input: if it is greater than 50, print the word “Pass”; otherwise print the word “Fail”.
11. Create a second function that does the exact same thing as the previous one but uses `ifelse()` instead if `if...else`.
12. Run both functions using the value 52.5 as the input
13. Run both functions using the **vector** of student test scores you created as the input. (Only one will work properly...)

```
#10. Create a function using if...else

greater_than50 <- function(x) {
  if(x > 50) {
    print("Pass")
  }
  else {
    print("Fail")
  }
}


#11. Create a function using ifelse()

greater_than50_concise <- function(x){
  ifelse(x>50, print("Pass"), "Fail") #log_exp, if TRUE, if FALSE

}

#12a. Run the first function with the value 52.5
greater_than50(52.5)
```

```
## [1] "Pass"
```

```
#12b. Run the second function with the value 52.5
greater_than50_concise(52.5)
```

```
## [1] "Pass"
```

```
## [1] "Pass"
```

```
#13a. Run the first function with the vector of test scores
#greater_than50(scores)

#13b. Run the second function with the vector of test scores
greater_than50_concise(scores)
```

```
## [1] "Pass"
```

```
## [1] "Pass" "Pass" "Pass" "Pass"
```

14. QUESTION: Which option of `if`...`else` vs. `ifelse` worked? Why? (Hint: search the web for "R vectorization")

    Answer: When using the vector of test scores, the `ifelse` option worked but the `if`...`else` option did not. This is because the `if`...`else` statement is not vectorized, so it can only handle single values. The `ifelse` option, on the other hand, is vectorized and are designed to handle vectors.

NOTE Before knitting, you'll need to comment out the call to the function in Q13 that does not work. (A document can't knit if the code it contains causes an error!)