Assignment 2: Coding Basics

Vicky Fong

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.

```
#1. Create vector from one through 55
seq_one <- seq(1,55,5)
seq_one
```

```
##  [1]  1  6 11 16 21 26 31 36 41 46 51
```

```
#2. Find mean and median of vector
mean_one <- mean(seq_one)
mean_one
```

```
## [1] 26
```

```
med_one <- median(seq_one)
med_one
```

```
## [1] 26
```

```
#3. Is mean greater than median?
mean_one > med_one
```

```
## [1] 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.

```
#5 and 6. Create vectors
names <- c("Darcy","Tara","Charlie","Nick")
names
```

```
## [1] "Darcy"   "Tara"    "Charlie" "Nick"
```

```
class(names) #character vector
```

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

```
scores <- c(99,58,93,87)
scores
```

```
## [1] 99 58 93 87
```

```
class(scores) #numeric vector
```

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

```
sch <- c(FALSE,TRUE,FALSE,FALSE)
sch
```

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

```
class(sch) #logical vector
```

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

```
#7. Create dataframe
scores_scholarship <- data.frame(names,scores,sch)
scores_scholarship
```

```
##     names scores   sch
## 1   Darcy     99 FALSE
## 2    Tara     58  TRUE
## 3 Charlie     93 FALSE
## 4    Nick     87 FALSE
```

```
#8. Add column names
names(scores_scholarship) <- c("Name","Test Score","Scholarship")
scores_scholarship
```

```
##      Name Test Score Scholarship
## 1   Darcy         99       FALSE
## 2    Tara         58        TRUE
## 3 Charlie         93       FALSE
## 4    Nick         87       FALSE
```

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

   Answer: This data frame contains different types of vectors (character, numeric, logical), but a matrix can only contain one type of vector.

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
f1 <- function(x) {
  if(x > 50) {
    "Pass"
  }
  else {
    "False"
  }
}

#11. Create a function using ifelse()
f2 <- function(x) {
  ifelse(x>50,"Pass","False")
}

#12a. Run the first function with the value 52.5
f1(52.5)
```

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

```r
#12b. Run the second function with the value 52.5
f2(52.5)
```

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

```r
#13a. Run the first function with the vector of test scores
#f1(scores)

#13b. Run the second function with the vector of test scores
f2(scores)
```

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

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

    Answer: 'ifelse' worked because this function is designed to work with vectors and return TRUE/FALSE values of the same vector length.

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!)