Assignment 2: Coding Basics

Zoe Wong

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.
by4 <- seq(1, 100, 4)  # creating sequence 1-100, jumping by 4

#2.
mean(by4)  # finding mean of sequence
```

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

```
median(by4)  # finding median of sequence
```

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

```
#3.
mean(by4) > median(by4)  # determining if mean is greater than median; if so, R will return TRUE
```

```
## [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.

```
students <- c("Zoe", "Caitlin", "Joyce", "Grace")
scores <- c(97, 94, 49, 78)
pass <- c(TRUE, TRUE, FALSE, TRUE)
testresults <- data.frame("students" = students, "scores" = scores, "pass" = pass)
```

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

   Answer: It has multiple different data types (numerical, logical, and characters).

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.

```
passorfail <- function(score){
  if (score >= 50){
    print(score >= 50)
  }
  else {
    print(score>50)
  }
}

result <- passorfail(scores)
```

```
## Warning in if (score >= 50) {: the condition has length > 1 and only the first
## element will be used

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

```
result
```

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

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

    Answer: If and else worked because there are only two options in this case - either the student got a score of 50 or above and passed, or they got a score below 50 and failed. Ifelse would be useful if there were three possible outcomes or if the conditions for passing were more complex.