Assignment 2: Coding Basics

Hannah Nelson

OVERVIEW

This exercise accompanies the lessons 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 Sakai.

Basics, Part 1

1. Generate a sequence of numbers from one to 30, increasing by threes. 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.
#creating a sequence from 1-30 that counts by 3
sequence <- seq(1, 30, by=3)
sequence
```

```
##  [1]  1  4  7 10 13 16 19 22 25 28
```

```
#2.
#finding the mean and median values of the sequence and naming those objects
mean <- mean(sequence)
mean
```

```
## [1] 14.5
```

```
#the mean is 14.5
median <- median(sequence)
median
```

```
## [1] 14.5
```

```
#the median is 14.5

#3.
#using an if/ if else function to generate text indicating if the mean is greater, less than, or equal
#this tells me that the mean and median are equal

if (mean > median) {
  cat("The mean is greater")
} else if (mean < median) {
  cat("The mean is less")
} else if (mean==median) {
  cat("They are equal")
}
```

```
## They are equal
```

Basics, Part 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. & #6.
#names vector
names <- c("Bella", "Emily", "Alex", "Rachel")
#scores vector
scores <- c(80, 90, 70, 40)
#true/false passing score vector
pass <- c("TRUE", "TRUE", "TRUE", "FALSE")

#7. & #8.
#creating data frame by combining the three vectors and naming the columns
testing <- data.frame("names"=names, "scores"=scores, "passing_score"=pass)
testing
```

```
##    names scores passing_score
## 1  Bella     80          TRUE
## 2  Emily     90          TRUE
## 3   Alex     70          TRUE
## 4 Rachel     40         FALSE
```

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

   Answer: This data frame contains multiple types of data, while a matrix can only contain one type.

10. Create a function with an if/else statement. Your function should take a **vector** of test scores and print (not return) whether a given 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.

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

```r
#scores vetor
scores <- c(80, 90, 70, 40)

#10.
#function that prints "true" if score if 50 or higher, and "false" if it is lower than 50
passfail <- function(scores) {
  passfail <- ifelse(scores >= 50, "TRUE", "FALSE")
  print(passfail)
}


#11.
#run function on vector to produce true/false list indicating if each score in the vector was passing
passfail(scores)
```

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

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

    Answer: the “ifelse” function worked becuase it allows you to apply a function to an entire vector, while “if” and “else” do not.