Assignment 2: Coding Basics

Maeve Arthur

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 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.
by_fours<-seq(1,100,4) #creating a object for a sequence that goes from 1-100 by 4s
by_fours #calling the by_fours object
```

```
##  [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
```

```
#2.
mean_fours<-mean(by_fours) #creating an object for the mean of the sequence
median_fours<-median(by_fours) #creating an object for the median of the sequence

#3.
isTRUE(mean_fours>median_fours) #asking R if the mean of the sequence is greater than the median of the
```

```
## [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.

```
student_names<-c("maeve","aileen","laura","ally") #character vector
test_scores<-c(99,98,97,49) #numeric vector
passed<-c("true","true","true","false") #character vector

student_stats<-cbind(student_names,test_scores,passed)
```

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

   Answer: This data frame consists of different classes of vector data, while a maxtrix may only consist of one type of data.

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.

```
#using 'ifelse'
passed_or_not<-ifelse(test_scores>=50,TRUE,FALSE)
print(passed_or_not)
```

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

```
#using 'if else'
passed_function<-function(test_scores) {
  if(test_scores >= 50) {
    (TRUE)
  }
  else {
    (FALSE)
  }
}
print(passed_function)
```

```
## function(test_scores) {
##   if(test_scores >= 50) {
##     (TRUE)
##   }
##   else {
##     (FALSE)
##   }
## }
```

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

    Answer: The 'ifelse' option worked here. This is because ifelse works better with vectors that are longer while if and else works with vectors that are a length of 1.