Assignment 2: Coding Basics

Jasmine Papas

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. Generating Sequence and Naming it
seq(1,100,4)
```

```
##  [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
```

```
sequence <- seq(1,100,4)
#2. Mean and Median Functions
mean(sequence)
```

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

```
median(sequence)
```

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

```
#3. Mean > Median?
mean(sequence)>median(sequence)
```

```
## [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.

```
#Vector Components
name_of_students<- c("Jasmine", "Everett","Aileen","Ana","Laura") #character vector
testscore<- c(98, 75, 46, 100, 84) #numerical vector
passfail<-c("passed", "passed", "failed","passed","passed") #character vector
who_passed_the_exam<-cbind(name_of_students, testscore, passfail)
```

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

   Answer: A matrix requires all inputs to be the same mode and length, but this data frame has a mix of modes.

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.

```
exam_results<- function(x){
  if(x > 50){
    "passed"
  }
  else {
    "failed"
  }
}

exam_results_<- function(x){
  ifelse(x>50, "passed", "failed")

}
```

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

    Answer:Both functions work, however, 'ifelse' allows for a cleaner code and less possible errors while 'if' 'else' requires more syntax and therefore more room for error.