Assignment 2: Coding Basics

Austin Guimond

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.
seq(1, 100, 4)#create sequence from 1 - 100 in increments of 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
```

```
FourSequence<-seq(1,100,4) #name sequence
#2.
mean(FourSequence) #Compute mean of sequence
```

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

```
median(FourSequence) # Compute median of sequence
```

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

```r
#3.
#check to see of values are equal or not
median(FourSequence)>mean(FourSequence)
```

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

```r
median(FourSequence)<mean(FourSequence)
```

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

```r
median(FourSequence)==mean(FourSequence)
```

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

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.

```r
StudentName <- c("John","Jill","Jack","Jess","Liz") ###vector1 Names
#Vector 2 Test Scores
TestScore<-c(99,88,94,50,74)
#Vector 3 Pass or fail (True or False)
PassingGrade<-c(TRUE,TRUE,TRUE,FALSE,TRUE)
#Data frame
d <- c("John","Jill","Jack","Jess","Liz")#Vector1
e <- TestScore<-c(99,88,94,50,74)#Vector2
f <- PassingGrade<-c(TRUE,TRUE,TRUE,FALSE,TRUE)#Vector3
TestGrades <- data.frame(d,e,f)#create data fram
names(TestGrades) <- c("Name","Grade","Passing Grade")#rename columns
#View(TestGrades)#view test scores data frame
```

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

   Answer: A matrix and a data frame can both be used to store two dimensional data in rows and columns. The difference between the two is a dataframe can store muliple data types while a mtrix can only contain a single class of data. For example, a mtrix could not contain both numeric and character 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.

```r
#ifelse statement, TRUE=Pass FALSE=Fail
ifelse(TestScore>50,TRUE,FALSE)
```

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

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

    Answer:The ifelse function worked to determine which test scores were passes and which were fails while I could not get the if and else option to work. The primary purpose of the ifelse function is to simplify R code to a single line and print the return. The if and else function can work when determining if individual values are true or false. For example I could make a function for X and and if and else function saying if x>50 pass, else fail. But for this function I would have to type the numbers individually into the code. With the ifelse function, I can input the vector TestScore and have R read out the True/False statements using a single line of code.