Assignment 2: Coding Basics

Langston Alexander

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.

sequencebyfour <- seq(1,100,4) #creating the sequence
sequencebyfour #printing that sequence
```

```
##  [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_sequencebyfour <- mean(sequencebyfour) #pulling the mean of the sequence and assigning it to a var
median_sequencebyfour <- median(sequencebyfour) #pulling the median of the sequence and assigning it to

mean_sequencebyfour #printing the mean and median variables
```

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

```
median_sequencebyfour
```

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

```
#3.
mean_sequencebyfour>median_sequencebyfour #Checking if the mean is greater than the median. Returns fal
```

```
## [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('Lorelai', 'Rory', 'Lane', 'Luke') #character vector
scores <- c(84, 98, 51, 48) # double vector
pass<- c(TRUE, TRUE, TRUE, FALSE) #logical vector

testresults <-data.frame(students, scores, pass) #combining vectors into a df

colnames(testresults) <- c("Student Name", "Test Score", "Did They Pass?") #renaming the columns in the
```

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

   Answer: Matrices can only hold a single data type, while a data frame can hold multiple data types. Matrices also have a set number of row and columns while we can add rows and columns to the data frame.

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.

```
pass_trueorfalse<- function(x) {
  ifelse(x>50, print(TRUE), print(FALSE)) #if x(a students score), is 50 or over this function will pri

}

test <- pass_trueorfalse(51) #testing if the function works, this should return "TRUE"
```

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

```
studentpass_trueorfalse <- pass_trueorfalse(testresults$`Test Score`) #Passing the students test scores
```

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

```
studentpass_trueorfalse #printing the results,
```

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

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

   Answer: I used `ifelse` to divide scores into two categories, those 50 and greater and those less than 50. If the score is 50 and greater the function prints "TRUE". For any other input the function prints "FALSE".