Assignment 2: Coding Basics

Jacob Freedman

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.

```r
#1. Made a sequence from 1 to 100 counting by 4 and named it SequenceHundred.
SequenceHundred<-seq(1,100,4)

#2. Computed mean and median of SequenceHundred and stored as mean_seq_hundred and median_seq_hundred.
mean_seq_hundred<-mean(SequenceHundred)
median_seq_hundred<-median(SequenceHundred)

#3. Asked r if mean of the sequence was greater than median. Output of TRUE or FALSE is the answer.
mean_seq_hundred>median_seq_hundred
```

```
## [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.

```
#5. Made 3 vectors each with data for 4 students
Student_name<-c("John","Jacob","Jingleheimer","Schmidt")
  #character vector
Test_score<-c(45,55,65,75)
  #This is a double vector which is a type of numeric vector
Did_they_pass<-50<=Test_score
  #logical vector

#6. See above

#7. Student_grades contains all 3 vectors from #5. Used data.frame function.
Student_grades<-data.frame(Student_name, Test_score, Did_they_pass)

#8. Renamed the columns with more simple, informative names.
colnames(Student_grades)<-c("Name", "Score", "Pass")
```

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

   Answer: A matrix requires that all data stored must be the same type (e.g. all data is numeric or all data is character etc). This data frame contains numeric data (the test scores), character data (the student names), and logical data (did they pass).

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.

```
#10. I stored the results of my ifelse function in a vector called Grade_results. This shows whether th
Grade_results <- function(Test_score) {
  Pass<-ifelse(Test_score>=50, TRUE, FALSE)
  print(Pass)

}

#11. I used my function Grade_results to determine whether the scores in the Student_grades data frame
Grade_results(Student_grades$Score)
```

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

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

    Answer: ifelse worked because it is meant to input vectors (Test_score is a vector) rather than scalars. I could use if and else to see if a single student passed the test.