Assignment 2: Coding Basics

Tasha Griffiths

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.
hundredbyfour <- seq(1, 100, 4)
#creating a value named 'hundredbyfour' that captures a sequence of numbers from 1 to 100 by 4

#2.
mean(hundredbyfour)
```

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

```
median(hundredbyfour)
```

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

```
#using a function to pull the mean and median of the pre-set named value 'hundredbyfour'

#3.
mean(hundredbyfour) > median(hundredbyfour)
```

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

```
#using a function to pull the mean of 'hundredbyfour' and then using basic math symbols to see if the m
```

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_First_Name <- c('Tasha', 'Erick', 'Izzy', 'Meilin')
#Vector-Class is Character

Test_Scores <- c(46, 71, 92, 100)
#Vector-Class is Numeric

Student_Passed <- c(FALSE, TRUE, TRUE, TRUE)
#Vector-CLass is logic

df_Student_Test1_Results <- data.frame(Student_First_Name,Test_Scores, Student_Passed)
df_Student_Test1_Results
```

```
##   Student_First_Name Test_Scores Student_Passed
## 1              Tasha          46          FALSE
## 2              Erick          71           TRUE
## 3               Izzy          92           TRUE
## 4             Meilin         100           TRUE
```

```
#creating dataframe from series of three vectors to make columns and then printing to visulize data

row.names(df_Student_Test1_Results)
```

```
## [1] "1" "2" "3" "4"
```

```
is.data.frame(df_Student_Test1_Results)
```

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

```
#confriming that dataframe creation worked
```

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

   Answer: Dataframes pull together different types or classes of vectors/values (characters, numbers, logic), however a matrix can only combine a single vecto-class.

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.

```
Is_Student_Passing <- function(x) {
ifelse(x>50, 'Pass', 'Fail')
}
#setting up a fuction named 'is student passing' where if any value of x is greater than 50, the studen

Is_Student_Passing(df_Student_Test1_Results$Test_Scores)
```

```
## [1] "Fail" "Pass" "Pass" "Pass"
```

```
#running the fuction with the x value replaced by the test scores column in the student test results da
```

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

    Answer: The ifelse worked because both the 'if' situation and the 'else' situation logic statements were incorporated. So 'if' the first statment is correct the fuction code would give out one result, if the code did not meet the logic statement (an 'else') a second result would be shown.