Assignment 2: Coding Basics

Sara Sayed

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.

```r
#1. I am creating a sequence of numbers that goes
#from 1 to 100 in increments of 4 then I create a
#variable with the sequence called Number_Seq

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
```

```r
Number_Seq<- seq(1,100,4)

#2. I want to see the summary statistics of this
#sequence are so I use the mean and median function
#to calculate the mean, median and create a variable
#for each called Mean & Median

Mean <- mean(Number_Seq)
Mean
```

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

```r
Median <-median(Number_Seq)
Median
```

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

```
#3. I want to see if the mean is greater than
#the median so I write a code asking if the
#mean > median, the console provides the response

Mean > Median
```

```
## [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.
Vector_A <- c("Nora", "Jack", "Peter", "Kara", "Nate", "Jenna")
Vector_A
```

```
## [1] "Nora"  "Jack"  "Peter" "Kara"  "Nate"  "Jenna"
```

```
#Vector A is a list of students
Vector_B <- c(90,40,77,85, 30, 50)
Vector_B
```

```
## [1] 90 40 77 85 30 50
```

```
#Vector B is the test score of each student
Vector_C <- c("True", "False", "True", "True","False", "False")
Vector_C
```

```
## [1] "True"  "False" "True"  "True"  "False" "False"
```

```
#Vector C states whether each student has passed or not

#7.
student_summary <- data.frame("Student Name" = Vector_A, "Student Grade"= Vector_B,
                              "Passed?" = Vector_C)
student_summary
```

```
##   Student.Name Student.Grade Passed.
## 1         Nora            90    True
## 2         Jack            40   False
## 3        Peter            77    True
## 4         Kara            85    True
## 5         Nate            30   False
## 6        Jenna            50   False
```

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

   Answer: This is a data frame because it includes more than one type of data. A matrix can only contain one type of data.

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.

```
#10. Created an ifelse statement to determine if a student
#passed or failed based on their test results. I used the
#string statement to determine the name of column I wanted
#in the data frame, then I wrote the ifelse statement with
#the conditions that if the student grade is more than 50,
#the student passes, but if not they fail

str(student_summary)
```

```
## 'data.frame':    6 obs. of  3 variables:
##  $ Student.Name : chr  "Nora" "Jack" "Peter" "Kara" ...
##  $ Student.Grade: num  90 40 77 85 30 50
##  $ Passed.      : chr  "True" "False" "True" "True" ...
```

```
x <- student_summary
x$Result<- ifelse (x$"Student.Grade">50, "Pass", "Fail")

#11 I applied the function to my data frame student_summary
#Then printed the results to show it is now appended to my dataframe

student_summary$Result<- ifelse (student_summary$"Student.Grade">50, "Pass", "Fail")
print(student_summary)
```

```
##   Student.Name Student.Grade Passed. Result
## 1         Nora            90    True   Pass
## 2         Jack            40   False   Fail
## 3        Peter            77    True   Pass
## 4         Kara            85    True   Pass
## 5         Nate            30   False   Fail
## 6        Jenna            50   False   Fail
```

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

    Answer: I think the ifelse statement worked the because we were looking at a simple type of data that could be combined into one statement. If the data was more complicated I would created individual if and else statements.