Assignment 2: Coding Basics

Tani Valdez Rivas

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, Part 1

1. Generate a sequence of numbers from one to 30, increasing by threes. 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.

# Establishing that the sequence is starting at one, increasing by 3s until it reaches 30.
seq(1, 30, 3)
```

```
##  [1]  1  4  7 10 13 16 19 22 25 28
```

```
seq_to_thirty <- seq(1, 30, 3)
seq_to_thirty
```

```
##  [1]  1  4  7 10 13 16 19 22 25 28
```

```
#2.

# Summary statistics can help calculate the mean and median of a data set
mean_seq <- mean(seq_to_thirty)
median_seq <- median(seq_to_thirty)
mean_seq
```

```
## [1] 14.5
```

```
median_seq
```

```
## [1] 14.5
```

```
#3.

# Asking if the mean is greater than the median
mean_seq > median_seq
```

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

Basics, Part 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.

# The c function can be used to create a vector
Students <- c('Siobhan', 'Kendall', 'Logan', 'Tom')
Students
```

```
## [1] "Siobhan" "Kendall" "Logan"   "Tom"
```

```
# The c function can also be used for numerical values
Scores <- c(85, 72, 91, 10)
Scores
```

```
## [1] 85 72 91 10
```

```
Scores > 50 # Which scores are above 50
```

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

```
#7.

# Assessing if each vector is a data frame
is.data.frame(Students)
```

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

```r
is.data.frame(Scores)
```

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

```r
# Converting the objects to a data frame
df_Students <- as.data.frame(Students)
df_Students
```

```
##   Students
## 1  Siobhan
## 2  Kendall
## 3    Logan
## 4      Tom
```

```r
df_Scores <- as.data.frame(Scores)
df_Scores
```

```
##   Scores
## 1     85
## 2     72
## 3     91
## 4     10
```

```r
# Binding the data together
df_Students_scores <- cbind(df_Students,df_Scores, Scores > 50)
class(df_Students_scores) # Assessing if the object is a data frame
```

```
## [1] "data.frame"
```

```r
df_Students_scores
```

```
##   Students Scores Scores > 50
## 1  Siobhan     85        TRUE
## 2  Kendall     72        TRUE
## 3    Logan     91        TRUE
## 4      Tom     10       FALSE
```

```r
# Labeling the column
colnames(df_Students_scores)
```

```
## [1] "Students"    "Scores"      "Scores > 50"
```

```r
Label_Columns_for_Student_Scores <- data.frame("Student"=Students, "Scored"=Scores,
                                               "Passed?" = Scores > 50)
Label_Columns_for_Student_Scores
```

```
##   Student Scored Passed.
## 1 Siobhan     85    TRUE
## 2 Kendall     72    TRUE
## 3   Logan     91    TRUE
## 4     Tom     10   FALSE
```

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

   Answer: A matrix is a data structure that is two-dimesnioal as it only contains rows and columns. Matrices possesses the same data type and has a fixed number of rows and columns. A data frame is another data structure that can contain various types of data and is one-dimensional

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.

```
Student_Scores <- function(Scores){ # Writing the function
  ifelse(Scores > 50, print("True"), print("False")) # Setting the return value of the condition
}

Student_Passed <- Student_Scores(Scores) # Determines if students received a passing grade
```

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

```
Student_Passed
```

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

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

    Answer: Using the 'ifelse' statement worked best to determine if a student passed with a score of 50 and above. Using the 'if' and 'else' statement required a lot more input from the user. In comparison, the 'ifelse' uses one line of code and it streamlines the process more efficiently.