Assignment 2: Coding Basics

Wynona Curaming

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.
sequencefrom1to30<-seq(1:30)
sequencefrom1to30
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30
```

```
# I have assigned the sequence to an object.

#2.
meanseq <-mean(sequencefrom1to30)
medianseq <- median(sequencefrom1to30)
meanseq
```

```
## [1] 15.5
```

```
medianseq
```

```
## [1] 15.5
```

```
#3.
mean(sequencefrom1to30)>median(sequencefrom1to30)
```

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

```
# I am calculating the mean of a sequence from 1 to 30 by creating a function
# Then I'm comparing it with the median of the same sequence by creating a function.
# When running the code, we find that the mean is not larger than the median.
# In this case, the mean and median are equal.
```

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.

```
student_names <- c("Ariane","Nikhil","Anais", "Wynona") #character vector
test_score <- c(90, 86, 99, 49) #numeric vector
pass_status <- test_score>=50 #logical vector

# alternative method: class_performance<- cbind(student_names, test_score, pass_status)
class_performance <- data.frame (
  "Name" = student_names,
  "Score" = test_score,
  "Passed" = pass_status
)
class_performance
```

```
##     Name Score Passed
## 1 Ariane    90   TRUE
## 2 Nikhil    86   TRUE
## 3  Anais    99   TRUE
## 4 Wynona    49  FALSE
```

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

   Answer: A matrix would have the same data type across each column and no column names. Data frame could have columns with different data types and columns would have names.

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_names <- c("Ariane","Nikhil","Anais", "Wynona") #character vector
test_score <- c(90, 86, 99, 49) #numeric vector
passing_status_2 <- ifelse(test_score >= 50, TRUE, FALSE)

class_performance3 <- data.frame (
  "Name" = student_names,
  "Score" = test_score,
  "Passed" = passing_status_2
)
return(class_performance3)
```

```
##     Name Score Passed
## 1 Ariane    90   TRUE
## 2 Nikhil    86   TRUE
## 3  Anais    99   TRUE
## 4 Wynona    49  FALSE
```

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

    Answer: 'ifelse' worked because it generates a vector that can be combined with other vectors to create a data frame. 'if' and 'else' generate only functions based on conditional arguments.