Assignment 2: Coding Basics

Sammy DiLoreto

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.

```
#1.
sequence.q1 <- seq(1, 100, 4)
  #creating a sequence from one to 100, increasing by 4 (named sequence.q1)
sequence.q1 #returning sequence as output
```

```
##  [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
```

```
#2.
mean.q1 <- mean(sequence.q1) #computing the mean of sequence.q1
mean.q1 #returning mean as output
```

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

```
median.q1 <- median(sequence.q1) #computing the median of sequence.q1
median.q1 #returning median as output
```

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

```
#3.
mean.q1 > median.q1 #determining if mean.q1 is greater than median.q1
```

```
## [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.
student.names <- c("Jenny", "Grace", "Delaney", "Natalie")
  #vector containing student names
test.scores <- c(49, 72, 97, 81)
  #vector containing student's test scores (out of 100)
passed <- c(FALSE, TRUE, TRUE, TRUE)
  #vector containing whether or no the students passed (passing grade of 50)

#6.
class(student.names) #character vector of student names
```

```
## [1] "character"
```

```
class(test.scores) #numeric vector of test scores
```

```
## [1] "numeric"
```

```
class(passed) #logical vector of whether or not they passed
```

```
## [1] "logical"
```

```
#7. and 8.
students.scores_df <- data.frame("StudentNames"=student.names,
                                 "TestScores"=test.scores, "Passed"=passed)
students.scores_df
```

```
##   StudentNames TestScores Passed
## 1        Jenny         49  FALSE
## 2        Grace         72   TRUE
## 3      Delaney         97   TRUE
## 4      Natalie         81   TRUE
```

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

Answer: A data frame is a list of vectors with the same length but difference mode (character, numeric, logical) whereas a matrix has the same length and same mode. A matrix would have to consist of all numeric or all character, etc.

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.

    Possible Function1: passing.grade <- function(test.scores){ output <- ifelse(test.scores>=50, TRUE, FALSE) print(output)} passing.grade(test.scores)

    Possible Function2: passing.grade2 <- function(test.scores){ output <- if(test.scores>=50) {TRUE} else {FALSE} print(output)} passing.grade2(test.scores)

11. Apply your function to the vector with test scores that you created in number 5.

```
passing.grade <- function(test.scores){
  output <- ifelse(test.scores>=50, TRUE, FALSE)
  print(output)}
passing.grade(test.scores)
```

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

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

    Answer: “ifelse” worked but “if” and “else” does not because “if” cannot run all values in a vector, it can only run a single value. A “for” loop would be needed to run all values in the vector.