Assignment 2: Coding Basics

Kelly Davidson

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. generating a sequence from 1 to 100, increasing by 4 and naming it "kelly_sequence"

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
```

```
kelly_sequence <- seq(1, 100, 4)

#2. computing mean and median

mean(kelly_sequence)
```

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

```
median(kelly_sequence)
```

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

```
#3. is mean > median?

isTRUE(mean(kelly_sequence) > median(kelly_sequence))
```

```
## [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.

```r
# 5 and 6. creating and labeling 3 vectors: student_names, test_scores, pass_fail

student_names <- c("Ashley" , "Brad" , "Claire" , "Dave")  #character vector
test_scores <- c(98, 72 , 49 , 61)  #numeric vector
pass_fail <- c(TRUE , TRUE , FALSE , TRUE)  #logical vector

#7. combining 3 vectors above into a data frame

fall_2022_grades_data_frame <- data.frame(student_names, test_scores, pass_fail)

#8. assigning informative titles to columns wihtin the data frame

colnames(fall_2022_grades_data_frame) <- c("Student Name", "Test Score", "Passing Grade")
```

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

   Answer: A data frame can have mixed data types (numeric, character, logical) whereas matrices can only include 1 data type.

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.

```r
#10. creating 'ifelse' function to determine whether a test score is a passing grade of 50 or above

grading_function <- function (test_scores) {
  ifelse(test_scores > 50, TRUE, FALSE)
  }

#11. applying 'grading_function' to 'test_scores' vector from #5

grading_function(test_scores)
```

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

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

    Answer: Only the ‘ifelse’ statement works in this scenario because the ‘if’ and ‘else’ statement has a condition limit (return item in curly brackets {}) of 1 character.