Assignment 2: Coding Basics

Stephanie Kinser

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.

```
#1.
four_sequence <- seq(1,100,4) #creating sequence from 1 to 100 by 4
four_sequence #printing sequence
```

```
##  [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_seq <- mean(four_sequence) #calculating mean of four_sequence
mean_seq
```

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

```
median_seq <- median(four_sequence) #calculating median of four_sequence
median_seq
```

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

```
#3.
mean_seq > median_seq #checking whether mean is greater than 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.

```
#a
students <- c("Stephanie", "Asher", "Alex", "Nicole") #creates vector of student names
typeof(students)
```

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

```
#students = character vector

#b
test_scores <- c(100, 99, 48, 87) #creates vector of test scores
typeof(test_scores)
```

```
## [1] "double"
```

```
#test_scores = double vector

#c
passing_grade <- c(TRUE, TRUE, FALSE, TRUE) #creates vector of grades that are passing = TRUE or failed
typeof(passing_grade)
```

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

```
#passing_grade = logical vector

student_tests_df <- data.frame("Student" = students, "Test_Score" = test_scores, "Passing_Grade" = passi
student_tests_df
```

```
##     Student Test_Score Passing_Grade
## 1 Stephanie        100          TRUE
## 2     Asher         99          TRUE
## 3      Alex         48         FALSE
## 4    Nicole         87          TRUE
```

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

Answer: A data frame can include different types of data, such as character, logical, or numeric vectors. However, a matrix can only include a single type of data. This data frame contains three types of vectors: character, double, and logical.

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.

```
grade_pass <- function(x){
  ifelse(x>=50, print(TRUE), print(FALSE))
}

grade_pass(test_scores)
```

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

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

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

Answer: Ifelse worked because it applies the function to all elements in a vector, whereas if and else only apply the function to the first element.