Assignment 2: Coding Basics

Sophie Moyer

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.

```
seq(1,100, by = 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
```

```
test_numbers <- seq(1,100, by = 4)

## I am using the sequence command to create a sequence of numbers from 1 to 100
## then I saved the sequence to "test_numbers"

mean(test_numbers)
```

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

```
median(test_numbers)
```

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

```r
mean <- mean(test_numbers)
median <- median(test_numbers)

## here I just used the mean and median functions built into R to calculate the value for the sequence
## then I saved the mean and median values to make the next part of code easier

mean > median
```

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

```r
## the above code will only print a TRUE value is the left value (the mean) is greater than the right v
## since it printed a FALSE value, we know the mean is not greater than the median
```

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
student <- c("Sarah", "Max", "George", "Emily") ## character vector
grade <- c(92,56,87,42) ## numeric vector
final <- c(TRUE, TRUE, TRUE, FALSE) ## logical vector

class2023 <- data.frame(student, grade, final)

colnames(class2023)[1] = "Student Name"
colnames(class2023)[2] = "Class Grade"
colnames(class2023)[3] = "Passed Final"

print(class2023)
```

```
##   Student Name Class Grade Passed Final
## 1        Sarah          92         TRUE
## 2          Max          56         TRUE
## 3       George          87         TRUE
## 4        Emily          42        FALSE
```

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

   Answer: This is different than a matrix because it contains different classes of data, while a matrix could only contain one class.

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.

```
ifelse(grade > 50, "True", "False")
```

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

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

    Answer: 'ifelse' worked because we were testing to see if a value was above '50' - if it was, it was then assigned a "TRUE" but if it was not above a '50' then it was assigned a "FALSE". So the function had to look for an 'if' and then an 'else' if that 'if' was not true.