Assignment 2: Coding Basics

Jessalyn Chuang

OVERVIEW

This exercise accompanies the lessons/labs 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 Canvas.

Basics, Part 1

1. Generate a sequence of numbers from one to 55, increasing by fives. 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.

```r
#1.
#Generating a sequence of numbers from 1 to 55, increasing by 5s, using the seq() function.
seq1_55 = seq(1,55,5)

#2.

#variables are created to store the mean and median of seq1_55. The mean()
#and median() functions were used.
mean1_55 = mean(seq1_55)
median1_55 = median(seq1_55)

#print the value of each variable
mean1_55
```

```
## [1] 26
```

```r
median1_55
```

```
## [1] 26
```

```
#3.

#Returns TRUE/FALSE after checking if mean1_55 is greater than median1_55
mean1_55 > median1_55
```

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

Basics, Part 2

5. Create three vectors, each with four components, consisting of (a) student names, (b) test scores, and (c) whether they are on scholarship or not (TRUE or FALSE).
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.

```
#Character vector that contains names of students
student_names <- c("Hermione", "Harry", "Ron", "Malfoy")

#Numeric vector that contains test scores
test_scores <- c(100, 85, 75, 45)

#Logical vector that contains information on whether they are on scholarship or not
scholarship_status <- c(TRUE, TRUE, TRUE, FALSE)

#combine the three vectors into a dataframe using the data.frame() function
student_df <- data.frame("Name"=student_names,"Score"=test_scores,
                         "Scholarship Status" = scholarship_status)

#show student_df
student_df
```

```
##       Name Score Scholarship.Status
## 1 Hermione   100               TRUE
## 2    Harry    85               TRUE
## 3      Ron    75               TRUE
## 4   Malfoy    45              FALSE
```

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

   Answer: This data frame is different from a matrix because it contains heterogeneous data and is a list of vectors of equal length. Meanwhile, matrices can only contain homogenous data. Data frames are better for visualizing data tables.

10. Create a function with one input. In this function, use `if...else` to evaluate the value of the input: if it is greater than 50, print the word "Pass"; otherwise print the word "Fail".
11. Create a second function that does the exact same thing as the previous one but uses `ifelse()` instead if `if...else`.

12. Run both functions using the value 52.5 as the input
13. Run both functions using the **vector** of student test scores you created as the input. (Only one will work properly…)

```
#10. Create a function using if...else
greater_than_50 <- function(x){
  if (x>50){
    print("Pass")
  }
  else{
    print("Fail")
  }
}

#11. Create a function using ifelse()
greater_than_50_v2 <- function(x){
  ifelse(x>50, "Pass", "Fail")
}

#12a. Run the first function with the value 52.5
greater_than_50(52.5)
```

```
## [1] "Pass"
```

```
#12b. Run the second function with the value 52.5
greater_than_50_v2(52.5)
```

```
## [1] "Pass"
```

```
#13a. Run the first function with the vector of test scores
#greater_than_50(test_scores)

#13b. Run the second function with the vector of test scores
greater_than_50_v2(test_scores)
```

```
## [1] "Pass" "Pass" "Pass" "Fail"
```

14. QUESTION: Which option of `if…else` vs. `ifelse` worked? Why? (Hint: search the web for "R vectorization")

    Answer: Ifelse worked and if…else did not. Ifelse is vectorized, which means that the function is capable of evaluating each element of a vector and returning a vectorized output ("R ifelse() Function," n.d.). If…else is not vectorized, and therefore is expecting a single value and will return an error.

    Citation: R ifelse() Function. (n.d.). Retrieved September 16, 2024, from https://www.programiz.com/r/ifelse-function

NOTE Before knitting, you'll need to comment out the call to the function in Q13 that does not work. (A document can't knit if the code it contains causes an error!)