Gottumukkula_Sriya_A02_CodingBasics.Rmd

Sriya Gottumukkula

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.

```
data1 <- seq(1,55,5) #creating a sequence
```

2. Compute the mean and median of this sequence.

```
meandata1 <- mean(data1) #calculating the mean

mediandata1 <-median(data1) # calculating the median
```

3. Ask R to determine whether the mean is greater than the median

```
comparisondata1 <-(meandata1>mediandata1) # seeing if mean is greater than median
```

4. Insert comments in your code to describe what you are doing.

```
#1. creating a sequence

#2. calculating mean and median

#3. seeing if mean is greater than median
```

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).

```
names <- c("sriya","priya", "niya", "giya") #character vector
scores <- c(10,20,30,40) #numerical vector
scholarship <- c("yes","yes","no","no") #character vector
```

6. Label each vector with a comment on what type of vector it is.

```
#1 Character Vector
#2 Numerical Vector
#3 Character Vector
```

7. Combine each of the vectors into a data frame. Assign the data frame an informative name.

```
dataframe <- data.frame("names","scores","scholarship")
```

8. Label the columns of your data frame with informative titles.

```
colnames(dataframe) <- c("names","scores","scholarship")
```

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

   Answer: this data frame is different from a matrix because a data frame can hold multiple data types (characters, numbers) in different columns, while a matrix can only contain a single data type across all its elements.

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".

```
code1 <- function(x) {
  if(x > 50) {
   (x <- "pass")
  }
  else {
    (x <- "fail")
  }
}
```

11. Create a second function that does the exact same thing as the previous one but uses `ifelse()` instead if `if...else`.

```
code2 <- function(x){
  ifelse(x>50, "pass", "fail")

}
```

12. Run both functions using the value 52.5 as the input

```r
output1 <- code1(52.5)
output2 <- code2(52.5)
```

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

```r
#output3 <- code1(scores)
output4 <- code2(scores)
```

```r
#10. Create a function using if...else

code3 <- function(x) {
  if(x < 4) {
    x*3
  }
  else {
    x*2
  }
}

#11. Create a function using ifelse()

code4 <- function(x){
  ifelse(x<4, x*3, x*2)

}

#12a. Run the first function with the value 52.5

output5 <- code3(52.5)

#12b. Run the second function with the value 52.5

output6 <- code4(52.5)

#13a. Run the first function with the vector of test scores

#output7 <- code3(scores)

#13b. Run the second function with the vector of test scores

output8 <- code4(scores)
```

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

    Answer: only ifelse worked as vectors only work with ifelse() and not if and else because if statements in R are designed to handle a single logical condition, not multiple values. When a vector is used in an if statement, R only checks the first element and ignores the rest, leading to incorrect results or warnings. In contrast, ifelse() is vectorized, meaning it applies the condition element-wise to each value in the vector, returning a result for each one

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!)