

Assignment 2: Coding Basics

Hanbin Lyu

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.

```
#1.  
A = c(1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 51)  
#set a sequence that from 1 to 55 that adjacent numbers different by 5
```

```
#2.  
mean_of_A = mean(A)  
#calculate the mean of the sequence A  
mean_of_A
```

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

```
median_of_A = median(A)  
#compute the median of the sequence A  
median_of_A
```

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

```
#3.
which_one_greater = mean_of_A > median_of_A
#ask R if mean is greater than median
which_one_greater
```

```
## [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.

```
#5.
student_names = c("A","B","C","D")
#set a vector for students' name, type:value
test_scores = c(78,99,81,62)
#set a vector for students' test scores, type:numeric
scholarship = c(FALSE,TRUE,TRUE,FALSE)
#set a vector to clarify if they are on scholarship or not, type:logic

#7.
student_information = data.frame(student_names,test_scores,scholarship)
#combine three vectors together, be a data frame

#8.
names(student_information) = c("Student Names","Test Scores","Scholarship")
#rename columns
student_information
```

```
##   Student Names Test Scores Scholarship
## 1           A           78         FALSE
## 2           B           99          TRUE
## 3           C           81          TRUE
## 4           D           62         FALSE
```

```
#check the data frame
```

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

Answer: in the matrix, all the elements should be same data type, while the data type in data frame could be different.

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 of `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
function1 = function(input) {
  if (input > 50) { print("Pass")}
  else {print("Fail")}}

#11. Create a function using ifelse()
function2 = function(input) {
  result = ifelse(input > 50, "Pass", "Fail")
  print(result)
}

#12a. Run the first function with the value 52.5
function1(52.5)
```

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

```
#12b. Run the second function with the value 52.5
function2(52.5)
```

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

```
#13a. Run the first function with the vector of test scores
function1
```

```
## function(input) {
##   if (input > 50) { print("Pass")}
##   else {print("Fail")}}
```

```
#error occurred, the function does not work
```

```
#13b. Run the second function with the vector of test scores
function2(test_scores)
```

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

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

Answer: the function with ‘ifelse’ worked. ‘if’...‘else’ only work with single number while ‘ifelse’ can work with vector because it is vectorized, “meaning that the function will operate on all elements of a vector without needing to loop through and act on each element one at a time.” “R for Novices: Vectorization.” Accessed September 16, 2024. <https://docs.ycrc.yale.edu/r-novice-gapminder/09-vectorization/>.

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!)