

Assignment 2: Coding Basics

Weilin Wang

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.  
seq1<- seq(1, 55, by = 5)  
seq1
```

```
## [1] 1 6 11 16 21 26 31 36 41 46 51
```

```
#2.  
mean_val <- mean(seq1)  
mean_val
```

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

```
#3.  
# Calculate the median of the sequence  
median_val <- median(seq1)  
# Compare if the mean is greater than the median (returns TRUE or FALSE)  
greater_ornot <- mean_val > median_val  
# Print the result of the comparison  
greater_ornot
```

```
## [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.
# (a) Character vector
student_names <- c("Alex", "Weilin", "Mia", "Bobby")
# (b) Numeric vector
test_scores <- c(70, 92, 88, 82)
# (c) logical vector
on_scholarship <- c(FALSE, TRUE, TRUE, FALSE)
#7.
student_info <- data.frame(Name = student_names,
                           Score = test_scores,
                           Scholarship = on_scholarship)
student_info
```

```
##      Name Score Scholarship
## 1   Alex    70         FALSE
## 2 Weilin    92          TRUE
## 3    Mia    88          TRUE
## 4  Bobby    82         FALSE
```

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

Answer: Matrices can only contain a single type of data, while data frames can consist of many different types of data.

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”.

```
evaluate_val <- function(score) {
  if (score > 50) {
    print("Pass")
  } else {
    print("Fail")
  }
}
```

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

```
evaluate_val_diff <- function(score) {
  result <- ifelse(score > 50, "Pass", "Fail")
  print(result)
}
```

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

```
evaluate_val(52.5)
```

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

```
evaluate_val_diff(52.5)
```

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

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

```
# evaluate_val(test_scores) Error,
evaluate_val_diff(test_scores)
```

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

```
#10.
saving_money1 <- function(money){
  if (money>100){
    print("Sucess!")
  }
  else{
    print("no Budget.")
  }
}

#11. Create a function using ifelse()
saving_money2 <- function(money){
  result <- ifelse(money>100, "Sucess!", "no Budget.")
  print(result)
}
```

```
#12a. Run the first function with the value 52.5
saving_money1(52.5)
```

```
## [1] "no Budget."
```

```
#12b. Run the second function with the value 52.5
saving_money2(52.5)
```

```
## [1] "no Budget."
```

```
#13a. Run the first function with the vector of test scores  
# saving_money1(test_scores)  
#13b. Run the second function with the vector of test scores  
saving_money2(test_scores)
```

```
## [1] "no Budget." "no Budget." "no Budget." "no Budget."
```

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

Answer: The `ifelse()` function worked because it is vectorized, meaning it automatically applies the condition to each element in the vector. In contrast, `if...else` is not vectorized, so it only works for single values unless a loop is used to handle each element. Vectorized functions like `ifelse()` are more efficient and ideal for operating on vectors in R.

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!)