

Assignment 2: Coding Basics

Ana Andino

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.
sequence1 <- seq(1,55,5)
sequence1 #In task one I gave a name to the sequence and then created the sequence from 1 to 55 but even
```

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

```
#2.
mean_seq <- mean(sequence1)
median_seq<- median(sequence1) #I computed the mean and median of the sequence.

#3.
mean_greater <- mean_seq > median_seq #I asked R to tell me if the mean is greater than the median which
mean_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 and #6 Created three different vectors with 4 components/items each
vector_names <- c("Ana Andino", "Chrissie Pantoja", "Juana Perez", "Chris Smith") #Vector Type: Character
vector_scores <- c(100, 99, 98, 97) #Vector Type: Numeric
vector_scholarship <- c("TRUE", "FALSE", "FALSE", "TRUE") #Vector Type: Logic/character

class(vector_scholarship) #just writing it as a reminder for future-self
```

```
## [1] "character"
```

```
#7 Combining vectors into dataframe
df_student_profiles <- data.frame(vector_names, vector_scores, vector_scholarship)

#8 Labeling columns
names(df_student_profiles) <- c("Student Name", "Test Schores", "Has Scholarship?")
```

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

Answer: Even though they are both two-dimensional, matrixes contain elements only of the same type but in data frames columns can have different modes (or data types) and we can also name columns.

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
score1 <- function(x) {
  if (x>50) {
    print("Pass")
  }
  else{
    print("Fail")
  }
}

#11. Create a function using ifelse()
```

```
score2<- function(x)
  ifelse(x>50, "Pass", "Fail") #log_exp, if TRUE, if FALSE

#12a. Run the first function with the value 52.5
student_result1 <- score1(52.5)
```

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

```
#12b. Run the second function with the value 52.5
student_result2 <- score2(52.5)
```

```
#13a. Run the first function with the vector of test scores
#student_result1 <- score1(vector_scores)
student_result1 <- sapply(vector_scores, score1)
```

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

```
#13b. Run the second function with the vector of test scores
student_result2 <- score2(vector_scores)
print(student_result2)
```

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

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

Answer: Using `if...else` does not work because it is not designed to handle vector but just single numbers. Thus, you must use `sapply`.

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!)