

Assignment 2 Coding Basics

Rosie Wu

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.name for the sequence is fifty_five_seq, inside the parenthesis follows the number of "from, to, by"
fifty_five_seq <- seq(1,55,5)
#2. assign average_seq being the mean of the sequence named above, while the med_seq being the median
average_seq <- mean(fifty_five_seq)
med_seq <- median(fifty_five_seq)
#3. returns a boolean of whether average_seq is greater than med_seq
average_seq > med_seq
```

```
## [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.

```
# vector1 is student names, which vector type is strings/ character
vector1 <- c("Sally", "Ann", "John", "Bob")
# vector2 is test scores, which the vector type is numeric
vector2 <- c(66,99,77,88)
# vector3 is whether they are on scholarship, which type is factor
vector3 <- c(FALSE, TRUE, FALSE, TRUE)
# combine all vectors into one dataframe, and the name is assigned as vector_all
vector_all <- data.frame(vector1, vector2, vector3)
# label what's being referred to each column as its title
names(vector_all) <- c("Name","Scores","Scholarship")
```

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

Answer: Matrices can only contain a single class of data, while data frames can consist of different classes or types of data. Dataframe is a list of vectors of equal length. It is a generalized form of a matrix. The name of rows are unique with no empty columns. The data can be in numeric, character or factor type.

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
f1 <- function(x) {
  if(x > 50) {
    "Pass"
  }
  else {
    "Fail"
  }
}

#11. Create a function using ifelse() # log_exp, if TRUE, if FALSE
f2 <- function(x){
  f2 <- ifelse(x>50, "Pass","Fail") #log_exp, if TRUE, if FALSE
}

#12a. Run the first function with the value 52.5
x = 52.5
run1 <- f1(52.5)
run1
```

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

```
#12b. Run the second function with the value 52.5
x = 52.5
run2 <- f2(52.5)
run2
```

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

```
#13a. Run the first function with the vector of test scores
# run_v1 <- f1(vector2)
# run_v1
#13b. Run the second function with the vector of test scores
run_v2 <- f2(vector2)
run_v2
```

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

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

Answer: `ifelse` worked while “`if...else`” didn’t. The `ifelse()` function is a shorthand vectorized alternative to the standard `if...else` statement. The `ifelse` statement is counting the number of individual elements that are greater than 55 rather than calculating the sum of all elements.

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!)