

Assignment 2: Coding Basics

Tallulah Bowden

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.  
fiftyfivesequence <- seq(1, 55, 5) #creating a sequence from 1 to 55 that increases by fives  
fiftyfivesequence #print out the sequence I made
```

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

```
#2.  
mean <- mean(fiftyfivesequence) #compute the mean of the sequence I made  
mean #print this mean
```

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

```
median <- median(fiftyfivesequence) #compute the median of the sequence I made  
median #print this median
```

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

```
#3.
mean > median #ask if the mean is greater than the median
```

```
## [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.

```
studentNames <- c("Gina", "Fafa", "Sloane", "Meg", "Maddy") #made a vector of student names.
#this is a character vector
testScores <- c(87, 91, 90, 88, 95) #made a vector of test scores.
#this is a numerical vector
scholarship <- c(FALSE, TRUE, TRUE, FALSE, FALSE) #made a vector showing whether they are on scholarship.
#this is a logical vector

dataframe2 <- data.frame(studentNames, testScores, scholarship)
names(dataframe2) <- c("Student Name", "Test Score", "Scholarship Status")
dataframe2
```

```
##   Student Name Test Score Scholarship Status
## 1      Gina      87      FALSE
## 2      Fafa      91       TRUE
## 3     Sloane      90       TRUE
## 4       Meg      88      FALSE
## 5      Maddy      95      FALSE
```

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

Answer: Data frames are different from matrices as they can have different modes. Matrices have to have elements of the same type. In this data frame, there are elements that are numeric, characters, and logical. A matrix could only contain one of these types.

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

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

#11. Create a function using ifelse()

```
passOrFail2 <- function(x) {  
  ifelse(x > 50, "Pass", "Fail")  
}
```

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

```
passOrFail(52.5)
```

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

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

```
passOrFail2(52.5)
```

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

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

```
#passOrFail(testScores)
```

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

```
passOrFail2(testScores)
```

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

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

Answer: The `ifelse` option worked when using the test score vector. The standard `if` statement is not vectorized. This means that the `if` statement will only work with one conditional (logic clause). When a vector is used, there are the same number of conditionals as there are items in the vector. This gives us an error. The `ifelse` option is vectorized which means that it is repeated as many times as needed to end the loop. It will repeat until all items in the vector have been assessed.

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!)