

Assignment 2: Coding Basics

Kendall Barton

OVERVIEW

This exercise accompanies the lessons 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 Sakai.

Basics, Part 1

1. Generate a sequence of numbers from one to 30, increasing by threes. 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.  
sequence3 <- seq(1,30,3) #number sequence 1-30 by 3s  
#2.  
#find mean and median and assign names  
s3mean <- mean(sequence3)  
s3median <- median(sequence3)  
#3.  
s3mean > s3median #use names to compare mean and median
```

```
## [1] FALSE
```

Basics, Part 2

5. Create a series of vectors, each with four components, consisting of (a) names of students, (b) test scores out of a total 100 points, and (c) whether or not they have passed the test (TRUE or FALSE) with a passing grade of 50.
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 & 6
students <- c("Matt","Steve","Kevin","Priscilla") #string vector
test_scores <- sample(0:100,4) #numerical vector
passing <- test_scores >= 50 #boolean/logical vector
#7 & 8
student_test_scores_df <- data.frame("Student Name" = students, "Test Score" = test_scores, "Test Passed" = passing)
```

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

Answer: Data frames can hold different types of data in the same data frame (for example, strings, integers and boolean values), while all data in a matrix has to be the same kind.

10. Create a function with an if/else statement. Your function should take a **vector** of test scores and print (not return) whether a given test score is a passing grade of 50 or above (TRUE or FALSE). You will need to choose either the if and else statements or the ifelse statement.
11. Apply your function to the vector with test scores that you created in number 5.

```
#10
pass_function <- function(scores){
  print(ifelse(scores>=50, TRUE, FALSE)) #scores>=50, if yes:TRUE, if no:FALSE
}
#11
print(test_scores) #I will also print the test scores so you can confirm my function works!
```

```
## [1] 92 3 76 54
```

```
pass_function(test_scores)
```

```
## [1] TRUE FALSE TRUE TRUE
```

12. QUESTION: Which option of if and else vs. ifelse worked? Why?

Answer: Both options can work, but ifelse is better. ifelse returns a vector of TRUE and FALSE in just one step. In order to get something similar using if and else, you would have to use a for loop to iterate through each item in the vector, choose an outcome of TRUE or FALSE and assign those values to a vector. Using ifelse is less code and I assume it's more efficient.