

# Assignment 2: Coding Basics

Anne Harshbarger

## OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on coding basics.

## Directions

1. Change “Student Name” on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your first and last name into the file name (e.g., “FirstLast\_A02\_CodingBasics.Rmd”) prior to submission.

## Basics Day 1

1. Generate a sequence of numbers from one to 100, increasing by fours. 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.  
seq100_by4 <- seq(1,100,4); seq100_by4 #create sequence from 1-100 by 4 called "seq100_by4,"
```

```
## [1] 1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
```

```
#then call sequence to view it
```

```
#2.  
mean(seq100_by4) #calculate mean of seq100_by4
```

```
## [1] 49
```

```
median(seq100_by4) #calculate median of seq100_by4
```

```
## [1] 49
```

```
#3.
mean(seq100_by4) > median(seq100_by4) #evaluate statement that the mean of the sequence is
```

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

```
#greater than the median and return TRUE or FALSE
```

## Basics Day 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 and 6.
names_v <- as.character(c("Harper", "Olivia", "Lance", "Sarah")) #Create student name vector
#of character data
scores_v <- as.numeric(c(49, 100, 80, 77)) #Create test score vector of numeric data
pass_v <- as.logical( c("FALSE", "TRUE", "TRUE", "TRUE")) #Create pass/fail vector of logical data
```

```
#7.
student_test_results_df <- as.data.frame(cbind(names_v,scores_v,pass_v)) #Bind vectors into dataframe
```

```
#8.
names(student_test_results_df) <- c("student_names", "test_scores", "pass_fail") #Rename columns
names(student_test_results_df) #View column names to verify renaming worked
```

```
## [1] "student_names" "test_scores"   "pass_fail"
```

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

Answer: The data frame is able to contain different types of data. I created this data frame by binding three vectors of equal length, each containing a different type of data: character, numeric, and logical. If I had created a matrix from three vectors, all three would have needed to contain the same type of data (e.g. all numeric or all character).

10. Create a function with an if/else statement. Your function should determine whether a 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. Hint: Use `print`, not `return`. The name of your function should be informative.
11. Apply your function to the vector with test scores that you created in number 5.

```

#10.
passing_grade <- function(score){

  x=score

  # if(x>=50) {
  #   print=TRUE
  # }
  # else {
  #   print=FALSE
  # }

  ifelse(x>=50, TRUE, FALSE)

}

#11.
results <- passing_grade(score=scores_v); results #create vector containing outputs of

## [1] FALSE TRUE TRUE TRUE

#"passing_grade" function, where input is vector "scores_v"

```

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

Answer: “ifelse” worked, but “if” and “else” did not; “if” and “else” (code visible as comments) was only able to run one element at a time, so the function written with this combination only determined whether the first score in the vector was a passing grade. However, “ifelse” was able to run each element on the list and return the results as another vector with length 4.

**Duke Community Standard affirmation:** I have adhered to the Duke Community Standard in completing this assignment. - Anne Harshbarger