

Assignment 2: Coding Basics

Rebecca Murphy

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. Creating a sequence of numbers from 1 - 100 by intervals of 4; sequence name is sequence1  
seq(1,100,4)
```

```
## [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
```

```
sequence1 <- seq(1,100,4)
```

```
#2. Computing the mean and median of sequence1  
mean(sequence1)
```

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

```
median(sequence1)
```

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

```
#3. Determining if the mean > median  
mean(sequence1) > median(sequence1)
```

```
## [1] 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.

```
student_names <- c("Sarah", "Chris", "Mark", "Rachel") #character vector
test_scores <- c(100L, 45L, 75L, 88L) #Integer vector
pass_or_fail <- c(TRUE, FALSE, TRUE, TRUE) #Logical vector

grading_df <- data.frame(student_names, test_scores, pass_or_fail)
names(grading_df) <- c("Names", "Test Scores", "Pass?")
View(grading_df)
```

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

Answer: Although both are 2D structures, by having character, numeric, and logical vectors, the data frame differs from a matrix in the fact that it allows different modes/elements. Matrices would only be able to have one of those vector types.

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: Function that gives "pass" if the grade is above 50 and "fail" if it is below 50 -
#Note - I didn't need to use print function to get the desired output.*

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

#11: applying this function against the test scores from #5
grade_test(test_scores)

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

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

Answer: The 'ifelse' statement worked (instead of if and else) because there was more than one component for the test scores vector. If there had only been one test score, I could have used 'if' and 'else'.