

Assignment 2: Coding Basics

Kim Myers

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 last name into the file name (e.g., “Salk_A02_CodingBasics.Rmd”) prior to submission.

The completed exercise is due on Tuesday, January 21 at 1:00 pm.

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.
s <- seq(1, 100, 4) # sequence from 1 to 100 by 4

#2.
mean(s) # mean and median of the sequence

## [1] 49
median(s)

## [1] 49

#3.
mean(s) > median(s) # if mean is greater than median, return true

## [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.

```
names <- c('Abby', "Bobby", "Connor", "Diana") # character
scores <- c(92, 88, 49, 95) # numeric
pass <- c(TRUE, TRUE, FALSE, TRUE) #logical

exam_results <- data.frame(names, scores, pass)
names(exam_results) <- c('StudentNames', 'ExamScores', 'Pass')
```

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

Answer: Matrices hold only numeric data. Dataframes can hold multiple data 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.

```
passgrade <- function(x) {
  if(x < 50) {
    print('Fail')
  }
  else {
    print('Pass')
  }
}

passgrade(scores) # if and else statement only runs first item
```

```
## Warning in if (x < 50) {: the condition has length > 1 and only the first
## element will be used
```

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

```
ifelse(scores<50, 'Fail', 'Pass') # ifelse statement runs correctly
```

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

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

Answer: The `ifelse` statement worked while the `if` and `else` statement did not because the former runs through each component of a vector while the latter only looks at the first component.