

Assignment 2: Coding Basics

Cristiana Falvo

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. sequence 1-100 (by 4's)
scores <- seq(from = 1, to = 100, by = 4)
scores

## [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

#2. meand and median of my sequence (scores)
score_mean <- mean(scores)
score_med <- median(scores)

score_mean

## [1] 49
score_med

## [1] 49

#3. determine if mean is greater than median (false, they are equal)
score_mean > score_med

## [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("Tom", "Brenda", "Sasha", "Charlie") #character vector
scores <- c(45, 66, 75, 88) #numeric vector
passing <- c(FALSE, TRUE, TRUE, TRUE) #logic vector

chem_class <- data.frame(names, scores, passing)
names(chem_class) <- c("Student Names", "Test Scores", "Pass/Fail"); View(chem_class)
```

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

Answer: while both store data as a grid, a matrix only stores numeric data while a data frame can store different 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.

```
grade <- function(x){
  if(x >= 50){
    return(TRUE)}
  else{
    return(FALSE)
  }
}

grade(45)
```

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

my function works when I put in a specific number but I want to be able to have the function go through the whole vector

```
grade(scores[1])
```

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

```
grade(scores[2])
```

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

```
grade(scores[3])
```

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

```
grade(scores[4])
```

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

```
# would be more efficient to be able to cycle through all 4 without calling each separately  
# maybe I need to change the (..arguments?) of my function (tried for loop, didn't work)
```

```
grades <- ifelse(scores >= 50, TRUE, FALSE)  
grades
```

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

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

Answer: ah, `ifelse` is powerful because you can put a vector into the ‘test’ argument, thus enabling me to iterate through each number in the vector and run the same test on it. Nice to be able to do this in a single line of code!