

Assignment 2: Coding Basics

Lydie Costes

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. Create a sequence from 1 to 100 by 4s and assign a name  
seq100 <- seq(1,100,4)
```

```
#2. Calculate the mean and median  
mean(seq100)
```

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

```
median(seq100)
```

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

```
#3. Check whether the mean is greater than the median  
mean(seq100) > median(seq100)
```

```
## [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.

```
#5. Create vectors of data and 6. label them
stud_names <- c("John", "Jim", "Jamie", "Jules") #character/string
test_scores <- c(95, 80, 45, 100) #double/numerical
pass_scores <- c(T, T, F, T) #logical/boolean

#7. Combine vectors into a dataframe
df_stud_tests <- data.frame(stud_names, test_scores, pass_scores)

#8. Label dataframe columns
colnames(df_stud_tests) <- c("StudentName", "TestScore", "Passing")
```

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

Answer: A dataframe can contain different types of data like strings and numeric, whereas a matrix has to be all uniform.

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. Create function to determine whether test score is passing
pass_function <- function(x){
  if(x >= 50){
    print("You passed!")
  }
  else{
    print("You did not pass.")
  }
}

pass_function2 <- function(x){
  ifelse(x>=50, "You passed", "You did not pass")
}

# 11. Apply function to test score vector
pass_function(test_scores)
```

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

```
## [1] "You passed!"
```

```
pass_function2(test_scores)
```

```
## [1] "You passed"      "You passed"      "You did not pass" "You passed"
```

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

Answer: I tried both. The ‘if’/‘else’ method only generates one output at a time, so you have to call the vector index you want or add something like a for loop. The ‘ifelse’ command does appear capable of iterating through a vector.