

Assignment 2: Coding Basics

Tommy Hancock

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.
num_list <- seq(1, 100, 4) # Generate a sequence of numbers from 1 to 100 by 4s
#2.
mean(num_list) # Calculate the mean

## [1] 49

median(num_list) # Calculate the median

## [1] 49

#3.
mean(num_list) > median(num_list) # Return TRUE if the mean is greater than the median

## [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("Scotty", "Lily", "Samantha", "Ronald") # String vector of names
scores <- c(76, 98, 86, 82) # Numeric vector with test scores
pass <- scores >= 50 # Logical vector determining whether the student passed

test_results <- data.frame(names, scores, pass) # Create a dataframe from the vectors
names(test_results) <- c("Name", "Score", "Pass")

test_results
```

```
##      Name Score Pass
## 1  Scotty    76 TRUE
## 2   Lily    98 TRUE
## 3 Samantha   86 TRUE
## 4  Ronald   82 TRUE
```

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

Answer: The different columns of the dataframe have different types of data in them (string, numeric, and logical), whereas a matrix requires all the data to be of the same type.

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.

```
pass_func1 <- function(x) { # Function using if/else to determine if a score passes
  if (x >= 50) {
    print("Pass")
  }
  else {
    print("Fail")
  }
}

pass_func2 <- function(x) { # Function using ifelse to determine if a score passes
  ifelse(x >= 50, "Pass", "Fail")
}

pass_func1(scores)
```

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

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

```
pass_func2(scores)
```

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

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

Answer: The ifelse function worked when applied to the “scores” vector, whereas the if/else function did not work for the vector. (Both worked for individual scores, though.) I got these results because the if/else statement requires a single value to evaluate, whereas the ifelse function is applied sequentially to all values within the vector.