

Assignment 2: Coding Basics

Answer Key

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.
sequence_hundy <- seq(1,100,4) #this code writes sequence from 1 to 100 with increment of 4.
sequence_hundy

## [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.
mean_seq <- mean(sequence_hundy) #this code calculates mean of sequence
mean_seq

## [1] 49

median_seq <- median(sequence_hundy) #this code calculates median of sequence
median_seq

## [1] 49

#3.
mean_seq > median_seq #this code determines whether mean is greater than 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.

```
name_v <- c("Sarah", "Luana", "Jon", "Abhishek") #character vector
test_scores_v <- c(90, 30, 20, 95) #numeric vector
pass_v <- c(test_scores_v >= 50) #logical vector

test_df <- data.frame("Names" = name_v, "Score" = test_scores_v, "Passing" = pass_v)

test_df
```

```
##      Names Score Passing
## 1    Sarah    90     TRUE
## 2    Luana    30    FALSE
## 3      Jon    20    FALSE
## 4 Abhishek    95     TRUE
```

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

Answer: A matrix is only capable of containing a single type of data, such as numbers or strings. However, a data frame is able to contain multiple elements of different data types. Like this one which contains string, integer, and logical values.

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.

```
grading <- function(x){ifelse(x >= 50, TRUE, FALSE)}
print(grading(test_scores_v))
```

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

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

Answer: `ifelse()` works for vectors whereas `if()` `else()` does not.