

Assignment 2: Coding Basics

Max Hermanson

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.
sequence1 <- seq(1,100,4)
sequence1

## [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_sequence1 <- mean(sequence1)
mean_sequence1

## [1] 49

#3.
median_sequence1 <- median(sequence1)
median_sequence1

## [1] 49

mean_vs_median <- function(x,y){      #function with two inputs
  if(x == y){                          # conditional statement
    print("true")}                    #outputs
  else {print("false")}
}
output_mean_vs_median<- mean_vs_median(mean_sequence1, median_sequence1) # created a function in which

## [1] "true"
```

```
output_mean_vs_median
```

```
## [1] "true"
```

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 <- c("Max", "Maria", "Joe", "Timor") #character
Test_Score <- c(40, 70, 90, 59) #numeric
Passing<- c(FALSE, TRUE, TRUE, TRUE) #logical

student_ID_Grade_df<- data.frame(Name, Test_Score, Passing)
student_ID_Grade_df
```

```
##      Name Test_Score Passing
## 1    Max          40   FALSE
## 2  Maria          70    TRUE
## 3    Joe          90    TRUE
## 4  Timor          59    TRUE
```

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

Answer: not all of the vectors have the same mode. In a matrix, all of the columns would be the same (ex: all are numeric)

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

```
Name <- c("Max", "Maria", "Joe", "Timor") #character
Test_Score <- c(40, 70, 90, 59)

pass_fail <- function(x){
  ifelse(x>50, print("True"), print("False"))
}
tester <- pass_fail(Test_Score)
```

```
## [1] "True"
## [1] "False"
```

```
print("did the students pass their exam?")
```

```
## [1] "did the students pass their exam?"
tester
```

```
## [1] "False" "True"  "True"  "True"
```

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

Answer: `ifelse()`. I initially tried using `if` and `else` individually, but when I tried inputting the test scores vector, I received an error saying that only a single element from the vector would be accepted. Thus it seems that `if/else` only accepts single elements, whereas `ifelse()` accepts entire vectors.