

Assignment 2: Coding Basics

Kendra Sultzer

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.  
seq1 <- seq(1,100,4) #creating sequence from 1-100, increasing by 4 and assigning it as seq1  
seq1 #displaying sequence
```

```
## [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(seq1) #computing mean of seq1
```

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

```
median(seq1) #computing median of seq1
```

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

```
#3.  
mean(seq1) > median(seq1) #is mean of seq1 greater than the median of seq1?
```

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

```
#no since they're the same
```

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. and #6.
```

```
Name <- c("Bobby", "Gerald", "Martha", "Luna") #this is a character vector  
class(Name)
```

```
## [1] "character"
```

```
Test_Score <- c(57,42,99,86) #this is a numeric vector  
class(Test_Score)
```

```
## [1] "numeric"
```

```
Pass <- c(TRUE,FALSE, TRUE, TRUE) #this is a logical vector  
class(Pass)
```

```
## [1] "logical"
```

```
#7.
```

```
class_grades <- cbind(Name,Test_Score,Pass) #combining vectors  
class_grades <-as.data.frame(class_grades) #transforming into data frame  
class_grades #viewing data frame
```

```
##      Name Test_Score  Pass  
## 1 Bobby           57  TRUE  
## 2 Gerald          42 FALSE  
## 3 Martha          99  TRUE  
## 4  Luna           86  TRUE
```

```
#8.
```

```
#I had already made my vectors titles informative, so there was no need to change the column names.
```

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

Answer: This data frame has different modes or data types, including characters, numeric, AND logical. A matrix can only have one type of mode or data type (ie numeric, logical, character, etc.).

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.
Passing_grade <- function(x){
  ifelse(x>=50,"TRUE", "FALSE")
} #creating function with if/else statement

#11.
Passing_grade(Test_Score) #applying function to vector with test scores

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

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

Answer: “Ifelse” worked well for me because this was a simple question of whether or not the answer was 50 or greater or less than 50. I think using the “if” and “else” would have required more notation and would have been a bit confusing.