

# Assignment 2: Coding Basics

Ana Bishop

## OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on coding basics.

## Directions

1. Rename this file `<FirstLast>_A02_CodingBasics.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change “Student Name” on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.
6. After Knitting, submit the completed exercise (PDF file) to Sakai.

## 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 <- seq(1,100,4) # creating a sequence that starts at 1, ends at 100, and
sequence # calling the sequence I created
```

```
## [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 <- mean(sequence) #function that finds the mean of a variable
median <- median(sequence) #function that finds the median of a variable

mean
```

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

```
median
```

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

```
#3.  
mean>median
```

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

```
# testing whether the mean is greater than the median. If it's not,  
#the output will be "FALSE" (which it was)
```

## 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.

```
students <- c("Anna", "Brad", "Charlie", "Susie") #character vector  
scores <- c(95, 60, 80, 40) #numeric vector  
pass <- c(TRUE, TRUE, TRUE, FALSE) #logical vector  
  
class(students) #shows what type of vector each vector is
```

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

```
class(scores)
```

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

```
class(pass)
```

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

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.

```
test_information <- data.frame(students, scores, pass)  
test_information
```

```
##   students scores pass  
## 1    Anna     95  TRUE  
## 2    Brad     60  TRUE  
## 3  Charlie     80  TRUE  
## 4    Susie     40 FALSE
```

8. Label the columns of your data frame with informative titles.

```
colnames(test_information) <- c("student_names", "test_scores", "pass_fail")  
test_information
```

```
## student_names test_scores pass_fail
## 1 Anna 95 TRUE
## 2 Brad 60 TRUE
## 3 Charlie 80 TRUE
## 4 Susie 40 FALSE
```

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

Answer: Dataframes can hold multiple classes of data, while matrices can only hold one.

10. Create a function with an if/else statement. Your function should take a **vector** of test scores and print (not return) whether a given 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.

```
## if/else
test_function <- function(x){
  if(x>=50) {
    TRUE
  }
  else{
    FALSE
  }
}

grades <- c(70, 40, 65, 20, 50, 90)
## test_function(grades)
# I had to make this^ a comment in order for the document to knit without errors
```

```
## ifelse
passing_grade <- function(x){
  ifelse(x>=50, TRUE, FALSE)
}

passing_grade(grades)
```

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

11. Apply your function to the vector with test scores that you created in number 5.

```
passing_grade(scores)
```

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

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

Answer: ifelse worked, while if/else did not. The if/else function returned an error stating that “the condition has length > 1”. This means that the if/else function could not work with a vector that contained multiple grades, while the ifelse function could. The ifelse function knew to apply the conditional statement individually to each component of the vector.