

Assignment 2: Coding Basics

Laura Brockington

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.  
seq100x4 <- seq(1,100,4) #creating a sequence from 1 to 100 by 4; sequence is named seq100x4  
seq100x4 #printing the output of the sequence code
```

```
## [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.  
x <- mean(seq100x4) #calculating the mean of the sequence and naming it x  
x #printing the mean
```

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

```
y <- median(seq100x4) #calculating the median of the sequence and naming it y  
y #printing the median
```

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

```
#3.  
x > y #asking R 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.

```
student_names <- c("Jill", "Terry", "Fred", "Kate") #creating a vector of student names  
class(student_names) #determining the class of the vector
```

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

```
#student_names is a vector of type character  
test_scores <- c(100, 48, 97, 80) #creating a vector of test scores  
class(test_scores) #determining the class of the vector
```

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

```
#test_scores is a vector of type numeric  
pass_test <- c(TRUE, FALSE, TRUE, TRUE) #creating a vector of whether the student passed the test or no  
class(pass_test) #determining the class of the vector
```

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

```
#pass_test is a vector of type logical  
  
df_student_grades <- as.data.frame(student_names) #creating a data frame with the student names  
cbind(df_student_grades, test_scores, pass_test) #adding the test scores and if the student passed the
```

```
##   student_names test_scores pass_test  
## 1         Jill         100      TRUE  
## 2         Terry          48     FALSE  
## 3          Fred          97      TRUE  
## 4          Kate          80      TRUE
```

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

Answer: This data frame is different from a matrix because it contains multiple types of data (character, numeric, and logical). Matrices can only have one type of data.

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

```
pass_grade <- function(x) {  
  if(x >= 50) {print(TRUE)}  
  else {print(FALSE)}  
} #creating a function to determine if the student passed the test with if and else  
pass_grade2 <- function(x) {  
  ifelse(x >= 50, TRUE, FALSE)  
} #creating a function to determine if the student passed the test with ifelse  
  
pass_grade2(test_scores) #seeing if the function works on a vector
```

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

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

Answer: Only 'ifelse' worked for my 'test_scores' vector because the if and else function is only designed to work with things that are a length of one. You have to use the ifelse function if you have something with more than one component, such as our vector.