

# Assignment 2: Coding Basics

Yingchi Cheung

## 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.
hundred_sequence <- seq(1, 100, 4) #Naming and creating a sequence from 1 to 100, increasing by 4.

#2.
mean(hundred_sequence) #Finding the mean of this sequence.

## [1] 49

median(hundred_sequence) #Finding the median of this sequence.

## [1] 49

#3.
mean(hundred_sequence) > median(hundred_sequence) #Determine is the mean 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.

```
Names <- c("Liam","Olivia","Grace","Noah","Oliver","Emma","Lucy","Drew") #character vector
Test_Scores <- c(60, 53, 96, 20, 64, 73, 45, 81) #numeric vector
Passed <- c(TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, FALSE, TRUE) #logical vector

d <- c("Liam","Olivia","Grace","Noah","Oliver","Emma","Lucy","Drew")
e <- c(60, 53, 96, 20, 64, 73, 45, 81)
f <- c(TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, FALSE, TRUE)
Student_Exam <- data.frame(d,e,f)
names(Student_Exam) <- c("Names","Test_Scores","Passed"); Student_Exam

##      Names Test_Scores Passed
## 1    Liam          60    TRUE
## 2 Olivia          53    TRUE
## 3  Grace          96    TRUE
## 4   Noah          20   FALSE
## 5 Oliver          64    TRUE
## 6   Emma          73    TRUE
## 7   Lucy          45   FALSE
## 8   Drew          81    TRUE
```

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

Answer: Data frame is more general than a matrix. Its columns contain different types of data (numeric, character, factor, etc.). But in a matrix, all the elements are the same 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.

```
passed <- function(Test_Scores){
  if (Test_Scores < 50) {print(FALSE)}
  else {print(TRUE)}
}
```

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

```
Exam <- passed(60)
```

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

```
Exam <- passed(53)
```

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

```
Exam <- passed(96)
```

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

```
Exam <- passed(20)
```

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

```
Exam <- passed(64)
```

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

```
Exam <- passed(73)
```

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

```
Exam <- passed(45)
```

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

```
Exam <- passed(81)
```

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

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

Answer: The 'if' and 'else' worked. Because the value can't print if using 'ifelse'.