

# Assignment 2: Coding Basics

Vanshika Mittal

## 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, Part 1

1. Generate a sequence of numbers from one to 30, increasing by threes. Assign this sequence a name.

```
seq (from = 1, to = 30, by = 3) #generating a sequence
```

```
## [1] 1 4 7 10 13 16 19 22 25 28
```

2. Compute the mean and median of this sequence.

```
seq1 <- seq(from = 1, to = 30, by = 3) #assigning a sequence name  
mean(seq1) #calculating its mean and media
```

```
## [1] 14.5
```

```
median(seq1)
```

```
## [1] 14.5
```

3. Ask R to determine whether the mean is greater than the median.

```
#assigning the mean and median of a sequence to variables  
mean.seq1 <- mean(seq1)  
median.seq1 <- median(seq1)  
mean.seq1 > median.seq1 #determining whether mean is greater than median
```

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

4. Insert comments in your code to describe what you are doing.

```
#Comments have been inserted in codes above.
```

## Basics, Part 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.

```
student_name <- c("Ally", "Ben", "Charlie", "Dina") #creating a vector with student names  
student_score <- c(88, 45, 67, 93) #creating a vector with random scores of choice  
passing_score <- c(TRUE, FALSE, TRUE, TRUE) #creating a vector based on the passing grade of 50
```

6. Label each vector with a comment on what type of vector it is.

```
#finding what class do the vectors belong to  
class(student_name)
```

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

```
class(student_score)
```

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

```
class(passing_score)
```

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

7. Combine each of the vectors into a data frame. Assign the data frame an informative name.

```
report_card <- cbind(student_name, student_score, passing_score) #combining the above vectors  
is.data.frame(report_card) #checking whether this is a data frame
```

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

```
report_card_df <- as.data.frame(report_card) #converting the vector into a data frame  
is.data.frame(report_card_df) #confirming whether its been converted
```

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

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

```
colnames(report_card_df) <- c("Name", "Score", "Passing Grade") #renaming columns
report_card_df #viewing our df to confirm
```

```
##      Name Score Passing Grade
## 1    Ally   88      TRUE
## 2     Ben   45     FALSE
## 3 Charlie   67      TRUE
## 4    Dina   93      TRUE
```

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

Answer: All values in a matrix belong to the same class however this dataframe contains three different classes of vectors. The “Name” column is a character vector, “Score” column is numeric vector and “Passing Grade” column is a logical vector

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.

```
#generating a loop of ifelse conditions for the entire "Score" column in our df
for (x in report_card_df$Score){
  Passing_Status <- ifelse (x > 50, print(TRUE), print(FALSE))
}
```

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

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

Answer: Both option ‘if’ and ‘else’ and ‘ifelse’ should since they imply the same working inputs. I have chosen to use the ifelse statement as it is easier to consolidate a larger argument for a vector in it.