

# Assignment 2: Coding Basics

Andi Mujollari

## 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.
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.

```
# My sequence will be from 1 to 30, added by 3
my_sequence <- seq(1,30, 3)
my_sequence
```

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

```
#With this code we tell R to find the mean value of the sequence we just created
mean_value <- mean(my_sequence)
#With this code we tell R to find the median value of our sequence
median_value <- median(my_sequence)
#Here we ask R if mean is grater than the median
mean_grater_than_median <- mean_value > median_value
cat("Mean:", mean_value, "\n")
```

```
## Mean: 14.5
```

```
cat("Median:",median_value, "\n")
```

```
## Median: 14.5
```

```
cat("Is mean grater than the median?", mean_grater_than_median, "\n")
```

```
## Is mean grater than the median? FALSE
```

```
#1.
```

```
#2.
```

```
#3.
```

## 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.
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.

```
# In the first Vector i have assigned the names of the students
student_names <- c("Shila","Migel","Jona", "Uarda")
#In the second Vector i have assigned the test scores of each of them
student_scores <- c(98, 67, 76, 34)
#In the third Vector i see if they passed the test (TRUE or FALSE) with a passing grade of 50
passed_status <- student_scores >=50
# In the end i collect the vectors into a data frame
student_table <- data.frame(Name = student_names, Score = student_scores, Passed = passed_status)
student_table
```

```
##      Name Score Passed
## 1 Shila     98   TRUE
## 2 Migel     67   TRUE
## 3 Jona      76   TRUE
## 4 Uarda     34  FALSE
```

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

Answer: Data frames allows me to use different data types combined together. On the other hand, if i had to use a matrix instead, i would have to convert all the data in one type.

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.

```
#Here i will check if the students passed or failed the test by using if & else
check_passing_score <- function(score) {
  if (score >= 50) {
    return(TRUE)
  } else {
    return(FALSE) }
}

for (score in student_scores) {
  cat("Score:", score, "Passing:", check_passing_score(score), "\n")
}
```

```
## Score: 98 Passing: TRUE
## Score: 67 Passing: TRUE
## Score: 76 Passing: TRUE
## Score: 34 Passing: FALSE
```

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

```
#To compare the options first i have to try also the ifelse command
check_passing_score <- function(score) {
  result <- ifelse(score >=50, "TRUE", "FALSE")
  cat(result, "\n")
}

for (score in student_scores) {
  cat("Score:", score, "Passing:", check_passing_score(score), "\n")
}
```

```
## TRUE
## Score: 98 Passing:
## TRUE
## Score: 67 Passing:
## TRUE
## Score: 76 Passing:
## FALSE
## Score: 34 Passing:
```

Answer: Both of the options worked!