

Assignment 2: Coding Basics

Brian Mulu Mutua

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.

```
#1. Generating the set of numbers from 1 to 30 using the seq function  
ten_number_sequence <- seq(1,30,3)
```

```
#2. Computing the mean and median of the sequence  
ten_number_sequence_mean <- mean(ten_number_sequence)  
ten_number_sequence_median <- median(ten_number_sequence)
```

```
#3. Checking if mean of the sequence is greater than the median  
print("Is the mean greater than the median?:")
```

```
## [1] "Is the mean greater than the median?:"
```

```
ten_number_sequence_mean > ten_number_sequence_median
```

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

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.

#5. Creating a series of vectors of student names, test scores, and pass/fail status

```
student_names <- c("George Costanza", "Elaine Benes", "Jerry Seinfeld","Cosmo Kramer")
↪ #Vector of type character
student_test_scores <- c(45,70,30,60) #Vector of type integer
student_pass_status <- c(FALSE,TRUE,FALSE,TRUE) #Vector of type logical
```

#7. Combining each of the vectors in a data frame

```
df_students_and_test_scores <-
↪ data.frame(student_names,student_test_scores,student_pass_status)
```

#8. Labeling the columns of the data frame with informative titles

```
colnames(df_students_and_test_scores) <-
↪ c("StudentName","StudentTestScore","HasStudentPassed")
print(df_students_and_test_scores) #Printing out resulting data frame
```

```
##      StudentName StudentTestScore HasStudentPassed
## 1 George Costanza             45             FALSE
## 2   Elaine Benes             70              TRUE
## 3  Jerry Seinfeld             30             FALSE
## 4   Cosmo Kramer             60              TRUE
```

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

Answer: In a matrix, all columns must have the same mode and length. A data frame is more general than a matrix and consists of a list of vectors of equal length, which could have different modes.

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.

#10. Creating a function with an if/else statement to check the vector of test scores and
↪ check if a given test score is a passing grade or not

```
has_student_passed_test <- function(test_scores){
  pass_status <- c(ifelse(test_scores >= 50,TRUE,FALSE)) #Checking if test scores are
↪ sufficient for a passing grade and storing the results in a vector
  print(data.frame(test_scores,pass_status)) #Printing the results
}
```

#11. Applying function to vector containing test scores

```
has_student_passed_test(student_test_scores)
```

```
## test_scores pass_status
## 1          45        FALSE
## 2          70         TRUE
## 3          30        FALSE
## 4          60         TRUE
```

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

Answer: The `'ifelse'` option worked but the `'if'` and `'else'` option did not. This is because the `'if'` and `'else'` option can only accept a condition that has a length of 1. Hence, attempting to pass a vector for the condition returned the error “*Error in if (test_scores >= 50) { : the condition has length > 1*”. In contrast, the `'ifelse'` option can accept a condition with a length > 1 , hence passing a vector in the condition worked.