

Assignment 2: Coding Basics

Isabel Zungailia

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. # Generated a sequence of numbers from 1 to 100, increasing by fours. Assigned sequence the name "s"  
seq(1, 100, by = 4)
```

```
## [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
```

```
s <- seq(1, 100, by = 4)
```

```
#2. Calculated the mean of "s". Gave the mean value the name of "mean.result".  
mean(s)
```

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

```
mean.result <- mean(s)
```

```
# Calculated the median of "s". Gave the median value the name of "median.result".  
median(s)
```

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

```
median.result <- median(s)
```

```
#3. Asked R to determine if the mean is greater than the median.  
is_mean_greater <- mean.result > median.result  
print(is_mean_greater)
```

```
## [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.

```
#5. Created three vectors, each with four components. Titled the vectors "names", "scores", and "pass_or_fail".
names <- c("Isabel", "Libby", "Jenna", "Jaimie")
```

```
scores <- c(98, 95, 95, 90)
```

```
pass_or_fail <- c(TRUE, TRUE, TRUE, TRUE)
```

```
#6. Labeled each vector with a comment on what type of vector it is.
names # character vector
```

```
## [1] "Isabel" "Libby" "Jenna" "Jaimie"
```

```
scores # numerical vector
```

```
## [1] 98 95 95 90
```

```
pass_or_fail # logical vector
```

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

```
#7. Combined the three vectors into a data frame. Named the data frame "student_scores".
student_scores <- data.frame(names=names, scores=scores, pass_or_fail=pass_or_fail)
student_scores
```

```
##   names scores pass_or_fail
## 1 Isabel     98         TRUE
## 2 Libby      95         TRUE
## 3 Jenna      95         TRUE
## 4 Jaimie      90         TRUE
```

```
#8. Labeled the data frame columns with the following names: "Student Name", "Test Score", and "Pass?".
colnames(student_scores) <- c("Student Name", "Test Score", "Pass?")
student_scores
```

```
##   Student Name Test Score Pass?
## 1 Isabel      98      TRUE
## 2 Libby       95      TRUE
## 3 Jenna       95      TRUE
## 4 Jaimie      90      TRUE
```

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

Answer: # A data frame is a two-dimensional object that has rows and columns, essentially a data table. A matrix is similar to a dataframe, except all values in a matrix are numeric.

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. Used an 'ifelse' statement to determine whether a given test score is a passing grade of 50 or above  
ifelse(student_scores$`Test Score` > 50, print("Passed"), print("Failed"))
```

```
## [1] "Passed"
```

```
## [1] "Passed" "Passed" "Passed" "Passed"
```

```
grade_function <- function (scores) {  
  ifelse(student_scores$`Test Score` > 50, print("Passed"), print("Failed"))  
}
```

```
#11. Applied 'grade_function' to 'scores' vector from #5.  
grade_function(scores)
```

```
## [1] "Passed"
```

```
## [1] "Passed" "Passed" "Passed" "Passed"
```

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

Answer: # When I tried to create an `if` and `else` statement, I got an error because the condition length is greater than 1. `if` and `else` is used when testing a single condition. The `ifelse` statement worked because it is able to evaluate all the elements in the vector without producing an error.