

Assignment 2: Coding Basics

Angela Edwards

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. Naming a sequence of 1 to 100 with increments of 4, with the name hundred_sequence:  
hundred_sequence <- seq(1, 100, 4)
```

```
#2. Finding the mean and median of hundred_ sequence:  
mean(hundred_sequence)
```

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

```
median(hundred_sequence)
```

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

```
#3. Asking R to determine whether the mean of hundred_sequence is greater than its mean:  
mean(hundred_sequence) > median(hundred_sequence)
```

```
## [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.

```
#5a & 6) Creating vector with four student names; this is a character vector.
student_name <- c("Kai", "Casey", "Pat", "Jessie")

#5b & 6) Creating vector with four test scores; this is an integer vector.
test_score <- c(88, 92, 49, 75)

#5c & 6) Creating vector for having passed the test; this is a logical vector.
passed_test <- c(TRUE, TRUE, FALSE, TRUE)

#7 & 8) Combining vectors into a data frame with an assigned name and named columns:
student_test_scores <- cbind(student_name, test_score, passed_test)
```

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

Answer: In a matrix, all columns must have the same mode and the same length. In this data frame, the columns do have the same length, but they have different modes, specifically: character, integer, and logical.

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.

```
# Using the if and else statements:
passing_score_1 <- function(x){
  passed <- x>50
  failed <- x<50
  print(list(passed = passed,
             failed = failed))
}

passed_1 <- passing_score_1(test_score)

## $passed
## [1] TRUE TRUE FALSE TRUE
##
## $failed
## [1] FALSE FALSE TRUE FALSE

passed_1

## $passed
## [1] TRUE TRUE FALSE TRUE
##
## $failed
## [1] FALSE FALSE TRUE FALSE

# Using the ifelse statement:
passing_score_2 <- function(x){
  ifelse(x>50, print(TRUE), print(FALSE))
}

passed_2 <- passing_score_2(test_score)

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

```

passed_2

## [1] TRUE TRUE FALSE TRUE
# Alternate print command using the ifelse statement:
passing_score_3 <- function(x){
  ifelse(x<50, print("Failed"), print("Passed"))
}

passed_3 <- passing_score_3(test_score)

## [1] "Failed"
## [1] "Passed"

passed_3

## [1] "Passed" "Passed" "Failed" "Passed"

```

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

Answer: I'm not sure if I did this exercise correctly, but it looks like they both worked, at least when using the integer vector of `test_score`. The different versions showed the information a bit differently; the first example displays output as two lists (`$passed` and `$failed`), whereas the two `if/else` examples show one line of output.