

Assignment 2: Coding Basics

Sebastian Bognar

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics (ENV872L) on coding basics in R.

Directions

1. Change “Student Name” on line 3 (above) with your name.
2. Use the lesson as a guide. It contains code that can be modified to complete the assignment.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document. Space for your answers is provided in this document and is indicated by the “>” character. If you need a second paragraph be sure to start the first line with “>”. You should notice that the answer is highlighted in green by RStudio.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file. You will need to have the correct software installed to do this (see Software Installation Guide) Press the **Knit** button in the RStudio scripting panel. This will save the PDF output in your Assignments folder.
6. After Knitting, please submit the completed exercise (PDF file) to the dropbox in Sakai. Please add your last name into the file name (e.g., “Salk_A02_CodingBasics.pdf”) prior to submission.

The completed exercise is due on Thursday, 24 January, 2019 before class begins.

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.

```
# create a sequence from 1-100 increasing by fours
```

```
Seq_4s <- seq(1,100, by = 4)
print(Seq_4s)
```

```
## [1] 1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89
## [24] 93 97
```

```
# find the mean of the sequence
```

```
mean <- mean(Seq_4s)
print(mean)
```

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

```
#find the median of the sequence
```

```
median <- median(Seq_4s)
print(median)
```

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

```
# figure out if mean is greater than the median
```

```
isTRUE(mean > median)
```

```
## [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.

```
# create a vector with student names
```

```
Student_names <- c("shannon", "clark", "colette", "mark") # character vector
```

```
# create a vector with student names
```

```
Test_scores <- c(92, 78, 83, 49) # numeric vector
```

```
# create a vector with t/f to see if they passed the test
```

```
IF_Passed_Test <- c(TRUE, TRUE, TRUE, FALSE) # logic vector
```

```
# create a dataframe
```

```
Student_Progress<- data.frame(Student_names, Test_scores, IF_Passed_Test)
```

```
names(Student_Progress) <- c("Name", "Score", "Passed")
```

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

ANSWER: This dataframe is composed of more than one data type (numeric, character, logical) whereas a matrix consists of one data type. A dataframe combines features of matrices and lists.

10. Create a function with an if/else statement. Your function should determine whether a 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. The name of your function should be informative.
11. Apply your function to the vector with test scores that you created in number 5.

```
# creating function with if and else statement
```

```
passed_test_if <- function(Test_score) {  
  Passed <- if (Test_score > 50) {  
    print("Passed")  
  }  
  else {print("Failed")  
  }  
}
```

```
passed_test_if(Test_scores) # the function with else and if statement does not work because you are inp
```

```
## Warning in if (Test_score > 50) {: the condition has length > 1 and only  
## the first element will be used
```

```
## [1] "Passed"
# creating the function to tell if you passed or failed the test using ifelse

Passed_the_Test <- function(Test_score) {
  passed <- (ifelse(Test_score > 50, "Passed", "Failed"))
  return(passed)
}

# testing the function
Passed_the_Test(Test_scores)

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

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

ANSWER: The ifelse statement created the function that worked whereas the if and else statement function did not work because it required a for loop to cycle through the test score inputs. The if and else function only considers the first value in the vector and provides a warning. The ifelse function performs a check on each component and then returns a vector.