

Assignment 2: Coding Basics

Njeri Kara

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.

```
#1. Generating a sequence of numbers form 1 to 100 increasing by 4s  
One.to.100.seq <- seq(1, 100, 4); One.to.100.seq # returning the variable
```

```
## [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
```

```
#2. Computing the mean and median of the sequence  
mean(One.to.100.seq)
```

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

```
median(One.to.100.seq)
```

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

```
#3. Determing whether the mean is greater than the median  
mean(One.to.100.seq)>median(One.to.100.seq)
```

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

```
#False. The mean is not greater than the median.
```

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. and 6.

```
##creating the charcter vector of student names; returning the created vector
student.names <- c("Mary","Paul","Jack","Alice"); student.names #Character vector
```

```
## [1] "Mary" "Paul" "Jack" "Alice"
```

```
#creating the number vector for test scores; returning the created vector
test.scores <- c(40,60,98,65);test.scores #number vector
```

```
## [1] 40 60 98 65
```

```
#creating the logic vector for passed tests,
##by determinging if each test.score element is greater than 50;
#returning the created vector
passed.test <- test.scores>=50; passed.test #logic vector
```

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

#7.

```
##Creating a data frame of the combined vectors
student.test.results <- data.frame(student.names,test.scores,passed.test)
```

```
#Assigning names to each of the collumns of the created data frame
names(student.test.results) <- c("Names","Scores","Passed")
View(student.test.results) #viewing the created data frame
```

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

ANSWER: A data frame can contain or hold elements that are of different data types for example characters, integers and logic elements. A matrix on the other hand can only hold elements of the same data type.

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 an ifelse function to determine that the test grade is a passing grade
passed.test.func <- function(x) {
  ifelse(x>=50,"True","False")
}
```

```
#applying the created function to test scores
passed.test.func(test.scores)
```

```
## [1] "False" "True" "True" "True"
```

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

ANSWER: ifelse worked because the input variable to the function was a vector. If and else can only be used for a function with an input variable that is one element.