

# Assignment 2: Coding Basics

Qianyi Xia

## 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.

```
hundred_sequence <- seq(1,100,4)
#assign a sequence name called hundred_sequence, from 1 to 100 increase everytime by 4.
mean(hundred_sequence) #Compute the mean of the sequence

## [1] 49

median(hundred_sequence) #Compute the median of the sequence

## [1] 49

mean(hundred_sequence)>median(hundred_sequence)

## [1] FALSE
#conditional statements to ask R if mean is greater than median for this hundred_sequence.
```

## 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.

```
vector_Names <-c("Lucy", "Jack", "Nina", "Tom") #name of students: character vector
vector_Scores <-c(100, 47, 30, 76) #test scores: numeric vector
vector_Pass <-c(TRUE, FALSE, FALSE, TRUE) # logical vector

dataframe_TestInfo <- data.frame(vector_Names, vector_Scores, vector_Pass)
names(dataframe_TestInfo) <- c("Name", "Score", "Passed the test");
```

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

ANSWER: This data frame is more general than a matrix, it contains three types of elements in three columns including character, numeric, and logical vector; While in a matrix, there can only be one type of element. A matrix is a collection of data elements arranged in a two-dimensional rectangular layout. This dataframe is used for student test data tables.

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.

```
ScorePass <-function(x){
  if((x<50)&(x>=0)){
    return(FALSE)
  }
  else if
    ((x>=50)&(x<=100)) {
    return(TRUE)
  }
  else {
    return(NA)
  }
}

ScorePass2 <-function(x){
  ifelse(x>=50, TRUE, FALSE)
}
ScorePass2(vector_Scores)
```

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

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

ANSWER: In this situation 'ifelse' worked when we input a vector which contains numbers of elements. The 'if...else' statement works when there is only 1 element in the vector, or our input is a specific score. Otherwise, it will only automatically condition the first element in the vector. Therefore, when we try to transfer data, or determine a vector of strings or numbers, it may be better to use the 'ifelse' statement.