

# Assignment 2: Coding Basics

Xuening Tang

## 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. First list the whole sequence from 1 to 100, then name a sequence called "hundred_sequence" and def
seq(1, 100)
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100
```

```
hundred_sequence <- seq(1, 100,4)
hundred_sequence
```

```
##   [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
```

```
#2. Request R to calculate the mean and median value of hundred_sequence we set in question 1.
mean(hundred_sequence)
```

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

```
median(hundred_sequence)
```

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

```
#3. Command in R to see the TRUE or FALSE that mean value of hundred_sequence is greater than the median  
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.

```
names<-c("Abby", "Bob", "Carol", "Dave")  
scores<-c(78,48,85,91)  
results<-c(TRUE,FALSE,TRUE,TRUE)  
  
class(names) #character
```

```
## [1] "character"
```

```
class(scores) #numeric
```

```
## [1] "numeric"
```

```
class(results) #logical
```

```
## [1] "logical"
```

```
grade<- cbind(names,scores,results)  
df_grade<- as.data.frame(grade)  
df_grade
```

```
##   names scores results  
## 1 Abby     78     TRUE  
## 2 Bob      48    FALSE  
## 3 Carol    85     TRUE  
## 4 Dave     91     TRUE
```

```
df_grade_report <- data.frame("Student_Name"=names,"Test_Score"=scores,"Passing_Result"=results)
df_grade_report
```

```
##   Student_Name Test_Score Passing_Result
## 1      Abby      78      TRUE
## 2       Bob      48     FALSE
## 3     Carol      85      TRUE
## 4      Dave      91      TRUE
```

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

Answer: A matrix can only contain a single class of data, while the data frame we created can consist of many different classes of data.

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.

```
#I chose 'if' and 'else' function to print answers
grades_judge<-function(x){
  if (x<50){
    print(FALSE)}
  else {
    print(TRUE)}
}

isgreaterthan<-lapply(scores, grades_judge)
```

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

```
#The answers in the function chunk are consistent with the one we judged before.
```

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

Answer: In this question, the function of 'if' and 'else' works. Because after trying the 'ifelse' function, it failed to get results. At the same time, the 'ifelse' function is used to assign one object or another depending on whether the first argument, test, is TRUE or FALSE, which could not be used under situation to simply judge what outcomes the vector data print.