

# Assignment 2: Coding Basics

April Qian

## OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on coding basics.

## Directions

1. Change “Student Name” on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your first and last name into the file name (e.g., “FirstLast\_A02\_CodingBasics.Rmd”) prior to submission.

## 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. generate the sequence from 1 to 100, increasing by 4  
seq(1, 100, 4)
```

```
## [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
```

```
sequence100_by4 <- seq(1, 100, 4)  
sequence100_by4
```

```
## [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. compute the mean and median of the above sequence  
meanseq <- mean(sequence100_by4)  
medianseq <- median(sequence100_by4)  
meanseq
```

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

```
medianseq
```

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

```
#3. whether the above mean is greater than the above median  
meanseq > medianseq
```

```
## [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.

```
# 5 and 6
```

```
# 5-a creat 10 student names
```

```
StudentNameVect <- c ("Jack","Peter", "April", "John", "Rebacca","Julia","Max", "Lisa", "Grace", "Pari")
StudentNameVect
```

```
## [1] "Jack" "Peter" "April" "John" "Rebacca" "Julia" "Max"
## [8] "Lisa" "Grace" "Pari"
```

```
class(StudentNameVect) #6 the type of the vector of StudentNameVect
```

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

```
# 5-b create the vector of student test scores
```

```
StudentScoreVect <- ceiling(runif(10, min=0, max=100)) # get 10 random number from 0 to 100 and round up
StudentScoreVect
```

```
## [1] 42 77 77 97 40 6 81 75 56 32
```

```
class(StudentScoreVect) #6 the type of the vector of StudentScoreVect
```

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

```
# 5-c whether they have passed the test
```

```
IfPassVect <- StudentScoreVect > 50
IfPassVect
```

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

```
class>IfPassVect) #6 the type of the vector of IfPassVect
```

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

```
#7 get a data frame! #8 get beautiful names for columns!
```

```
#creat a dataframe
```

```
StudentTest <- data.frame ("StudentName" = StudentNameVect, "StudentScore" = StudentScoreVect, "IfPass" = IfPassVect)
StudentTest
```

```
##      StudentName StudentScore IfPass
## 1         Jack          42 FALSE
## 2         Peter          77  TRUE
## 3         April          77  TRUE
## 4          John          97  TRUE
## 5        Rebacca          40 FALSE
## 6          Julia           6 FALSE
## 7           Max          81  TRUE
## 8          Lisa          75  TRUE
## 9         Grace          56  TRUE
## 10         Pari          32 FALSE
```

```
is.data.frame(StudentTest) # to verify!
```

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

```
View(StudentTest) # to have a look!
```

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

Answer: A data frame can contain different vector types, for example, a data frame could be a combination with columns with type of numeric, logical, or character, while a matrix can only contain one type of the vector (only numeric or logical or other type at the same time)

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. Hint: Use `print`, not `return`. The name of your function should be informative.

11. Apply your function to the vector with test scores that you created in number 5.

```
IfPass_func <- function(x){  
  ifelse(x>50, print("pass"), print("fail"))  
} # 10 the function  
Passorfail <- IfPass_func(StudentTest$StudentScore)
```

```
## [1] "pass"
```

```
## [1] "fail"
```

```
Passorfail # 11 Vector generation
```

```
## [1] "fail" "pass" "pass" "pass" "fail" "fail" "pass" "pass" "pass" "fail"
```

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

Answer: They all worked! We can choose either to enter “If `x>50` {`print ("pass")`}, `else`{`print("fail")`}, or just use `ifelse` as I’ve used above.

Comments for the assignment: It is too much to finish all past videos and assignments in two days for a poor beginner (I just registered for the course before the drop/off ddl TvT), life is so hard!