

Assignment 2: Coding Basics

Yao Yao

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.  
#A sequence of numbers from one to 100, increasing by fours  
sequence <- seq(1, 100, by = 4)  
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.  
#Mean and median of "sequence"  
mean<-mean(sequence)  
median<-median(sequence)  
mean
```

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

```
median
```

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

```
#3.  
#Determine whether mean is bigger than median  
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.

```
names <- c("Lily", "Mandy", "Jack", "Liam")  
names
```

```
## [1] "Lily" "Mandy" "Jack" "Liam"
```

```
class(names)
```

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

```
#names are character
```

```
test_scores <- c(90, 89, 71, 40)  
test_scores
```

```
## [1] 90 89 71 40
```

```
class(test_scores)
```

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

```
#test_scores are numeric
```

```
pass <- test_scores>50  
pass
```

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

```
class(pass)
```

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

```

#pass is logical

#Converting vectors to dataframes
df_names <- as.data.frame(names)
df_test_scores <- as.data.frame(test_scores)
df_pass <- as.data.frame(pass)

#Combining the dataframes and give names to the columns
student_test_performance <- data.frame("Names"=names, "Test Scores"=pass, "Pass or Not"=pass)
class(student_test_performance)

```

```
## [1] "data.frame"
```

```
student_test_performance
```

```
##   Names Test.Scores Pass.or.Not
## 1  Lily         TRUE         TRUE
## 2 Mandy         TRUE         TRUE
## 3  Jack         TRUE         TRUE
## 4  Liam        FALSE        FALSE
```

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

Answer: I think the main difference is that matrices only contain a single class of data, while data frames 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.

```

pass_test <- function(x) {
  ifelse(x>=50, TRUE, FALSE)
}

```

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

```

results <- pass_test(df_test_scores)
results

```

```
##      test_scores
## [1,]         TRUE
## [2,]         TRUE
## [3,]         TRUE
## [4,]        FALSE
```

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

Answer: In my R studio, ifelse works at last, since can only check one element in a vector at one time, but using this code we attempted to check every element in the vector at once.