

Assignment 2: Coding Basics

Changxin Yu

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.  
seq0 <- seq(1,100,4) # generate a sequence  
  
#2.  
mean(seq0) # compute the mean of the sequence
```

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

```
median(seq0) # compute the median of the sequence
```

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

```
#3.  
mean(seq0) > median(seq0) # determine if 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.

```
#5.  
name <- c("Jack", "Anne", "Chris", "Sam") # character vector  
score <- c(90, 67, 51, 46) # numeric vector  
pass_or_not <- c(T,T,T,F) # logical vector
```

```
#6.  
mode(name)
```

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

```
mode(score)
```

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

```
mode(pass_or_not)
```

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

```
#7.  
students_grade <- as.data.frame(cbind(name,score,pass_or_not))
```

```
#8.  
colnames(students_grade) # it already had informative column names
```

```
## [1] "name"          "score"         "pass_or_not"
```

```
students_grade
```

```
##   name score pass_or_not  
## 1 Jack    90         TRUE  
## 2 Anne    67         TRUE  
## 3 Chris   51         TRUE  
## 4 Sam     46        FALSE
```

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

Answer: For a matrix, all columns must have same mode and same length, but for a data frame, they should not be.

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_fun <- function(score){  
  result <- ifelse(score<50, F, T)  
  print(result)  
}
```

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

```
pass_fun(score)
```

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

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

Answer: When we just do a simple logical judgement and one line of code for each case, **ifelse** works well. Otherwise, we should use **if** and **else** to fit more complicated cases.