

Assignment 2: Coding Basics

Joanna Huertas

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. seq(1, 10)
fours_sequence <- seq(1, 100, 4) # from, to, by
fours_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<-mean(fours_sequence) #finding the mean, assigning a name
median<-median(fours_sequence) #finding the median, assigning a name
mean
```

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

```
median
```

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

```
#3.  
mean>median #asking R for True or False
```

```
## [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.

```
#creating a series of vectors  
studentnames <- c('Anna','Tom','Lucas','Lucy') #character  
tests <- 90:93 #sequence  
# Which scores are >= 50?  
results <- tests >= 50 #logical  
results
```

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

```
students_data <- data.frame(studentnames, tests, results) # Apply data.frame function  
students_data
```

```
##   studentnames tests results  
## 1      Anna     90     TRUE  
## 2       Tom     91     TRUE  
## 3     Lucas     92     TRUE  
## 4      Lucy     93     TRUE
```

```
colnames(students_data) <- c('Student Names','Test Scores','Pass')  
students_data
```

```
##   Student Names Test Scores Pass  
## 1      Anna     90     TRUE  
## 2       Tom     91     TRUE  
## 3     Lucas     92     TRUE  
## 4      Lucy     93     TRUE
```

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

Answer: In a data frame the columns contain different types of data, but in a matrix all the elements are the same type of data. A matrix in R is like a mathematical matrix, containing all the same type of thing (usually numbers).

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.

```
#trying with 'if' and 'else'
```

```
pass <- function(x) {  
  if(x >= 50) {  
    'TRUE'  
  }  
  else {  
    'FALSE'  
  }  
}  
  
pass_or_not <- pass(tests); pass_or_not
```

```
## Error in if (x >= 50) {: the condition has length > 1
```

```
## Error in eval(expr, envir, enclos): objeto 'pass_or_not' no encontrado
```

```
pass_or_not
```

```
## Error in eval(expr, envir, enclos): objeto 'pass_or_not' no encontrado
```

```
#trying with 'ifelse'
```

```
pass1 <- function(x){  
  ifelse(x >= 50, 'TRUE', 'FALSE') #log_exp, if TRUE, if FALSE  
}  
  
pass_or_not1 <- pass1(tests); pass_or_not1
```

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

```
pass_or_not1
```

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

```
#this works
```

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

Answer:ifelse worked because the 'if' function is designed to work with things that are length 1, like a single name, which is why we got an error using it with 4 elements; while if we want to work with vectors that are length > 1, like in this case, we use 'ifelse'.