

Assignment 2: Coding Basics

Rosalind Hu

OVERVIEW

This exercise accompanies the lessons/labs 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 Canvas.

Basics, Part 1

1. Generate a sequence of numbers from one to 55, increasing by fives. Assign this sequence a name.

```
vector1 <- seq(1, 55, 5) # 2. Compute the mean and median of this sequence. mean1 <- mean(vector1)
mean1 median1 <- median(vector1) median1 # 3. Ask R to determine whether the mean is greater than
the median. mean1 <- 26 median1 <- 26 if (mean1 > median1){ result <- “True” } else{ result <- “False”
} # 4. Insert comments in your code to describe what you are doing.
```

```
#1. vector1 <- seq(1, 55, 5) which means that a sequence of numbers from 1 to 55, increasing by 5. And
#2. First we have to assign a name for mean and median of vector1, and use the function mean and functi
#3. First I assign two items I want to compare into number then use if function that whether the mean i
```

Basics, Part 2

5. Create three vectors, each with four components, consisting of (a) student names, (b) test scores, and (c) whether they are on scholarship or not (TRUE or FALSE).

```
studentnames <- c(“David”, “Amy”, “Ruby”, “Lucy”) studentnames testscores <- c(82, 91, 88, 86) testscores
scholarship <- c(“TRUE”, “TRUE”, “FALSE”, “FALSE”) scholarship # 6. Label each vector with a comment
on what type of vector it is. class(studentnames) class(testscores) class(scholarship) # 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.

```
studentnames <- c("David", "Amy", "Ruby", "Lucy")
studentnames
```

```
## [1] "David" "Amy"   "Ruby"  "Lucy"
```

```
class(studentnames)
```

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

```
testscores <- c(82, 91, 88, 86)
testscores
```

```
## [1] 82 91 88 86
```

```
class(testscores)
```

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

```
scholarship <- c(TRUE, TRUE, FALSE, FALSE)
scholarship
```

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

```
class(scholarship)
```

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

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

Answer: So for the vectors, they only have 1 dimensional structure that contains elements, while matrices have 2 dimensional structure, which contains rows and columns

10. Create a function with one input. In this function, use `if...else` to evaluate the value of the input: if it is greater than 50, print the word “Pass”; otherwise print the word “Fail”. `> input1 <- 52`

```
if (input1 > 50){ result <- "Pass" } else{ result <- "Fail" }
```

result 11. Create a second function that does the exact same thing as the previous one but uses `ifelse()` instead of `if...else`. `input3 <- 47 ifelse(input3 > 50, "Pass", "Fail")` result 12. Run both functions using the value 52.5 as the input

13. Run both functions using the **vector** of student test scores you created as the input. (Only one will work properly...)

```
#10. Create a function using if...else  
input1 <- 52
```

```
if (input1 > 50){  
  result <- "Pass"  
} else{  
  result <- "Fail"  
}
```

```
result
```

```
## [1] "Pass"
```

```
#11. Create a function using ifelse()  
input2 <- 47
```

```
ifelse(input2 > 50, "Pass", "Fail")
```

```
## [1] "Fail"
```

```
result
```

```
## [1] "Pass"
```

```
#12a. Run the first function with the value 52.5  
input3 <- 52.5
```

```
if (input3 > 50){  
  result <- "Pass"  
} else{  
  result <- "Fail"  
}
```

```
result
```

```
## [1] "Pass"
```

```
#12b. Run the second function with the value 52.5  
input3 <- 52.5
```

```
ifelse(input3 > 50, "Pass", "Fail")
```

```
## [1] "Pass"
```

```
#13a. Run the first function with the vector of test scores
```

```
testscores_comparison <- function(testscores) {  
  if (testscores > 50) {  
    result <- "True"  
  } else {
```

```

    result <- "False"
  }
  return(result)
}

# Example usage
print(testscores_comparison(50))

```

```
## [1] "False"
```

#13b. Run the second function with the vector of test scores

```

result <- ifelse(testscores > 50, "Pass", "Fail")
result

```

```
## [1] "Pass" "Pass" "Pass" "Pass"
```

14. QUESTION: Which option of `if...else` vs. `ifelse` worked? Why? (Hint: search the web for “R vectorization”)

Answer: From the previous questions to operate `if...else` function and `ifelse`, I would say that `ifelse` would be more convenient. First the logic to put into the function(`x,y,z`), `x` which can set the conditions and then if it meets the condition then it will show “y”, if not meeting the condition, then will show “z”

NOTE Before knitting, you’ll need to comment out the call to the function in Q13 that does not work. (A document can’t knit if the code it contains causes an error!)