

Assignment 2: Coding Basics

Leyi Ruan

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. I'm building a sequence from 1 to 100, increased by 4.
```

```
#Then assign the sequence just built to seq1.
```

```
seq1 <- seq(1, 100, 4)
```

```
seq1
```

```
## [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. Use mean command to compute the mean of the sequence,
```

```
#and use median command to calculate the median of the sequence.
```

```
mean1 <- mean(seq1)
```

```
mean1
```

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

```
median1 <- median(seq1)
```

```
median1
```

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

```
#3. Use conditional statements to check if mean is greater than median.
```

```
#The result turns out to be false. Thus, mean is not greater than median.
```

```
mean1 > median1
```

```
## [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 & 6. Create 3 vectors: names, scores, and results
names <- c("Lily", "Mark", "James", "Bob") #Character Vector
scores <- c(99, 68, 35, 51) #Integer vector
results <- c(TRUE, TRUE, FALSE, TRUE) #Logical vector

# 7. Combine vector into a dataframe
df_names <- as.data.frame(names)
df_scores <- as.data.frame(scores)
df_results <- as.data.frame(results)
test <- cbind(df_names, df_scores, df_results)
test

##   names scores results
## 1  Lily     99     TRUE
## 2  Mark     68     TRUE
## 3 James     35    FALSE
## 4   Bob     51     TRUE

# 8. Label Columns with Informative Titles
test <- data.frame("Name"=names, "Score"=scores, "Result"=results)
test

##   Name Score Result
## 1  Lily     99     TRUE
## 2  Mark     68     TRUE
## 3 James     35    FALSE
## 4   Bob     51     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.

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.

```
# 10. Create Function to examine Pass/Fail
Pass_Fail <- function(x){
  if(x>=50) {
    Result_2 <- "TRUE"
  }
  else{
    Result_2 <- "FALSE"
  }
}
```

```

    return(Result_2)
}

# 11. Apply the function to the vector with test scores
# To distinguish the result derived from using function with the original #result vector, I name the new
Result_2 <- lapply(scores, Pass_Fail)
Result_2 #Result_2 is a list

## [[1]]
## [1] "TRUE"
##
## [[2]]
## [1] "TRUE"
##
## [[3]]
## [1] "FALSE"
##
## [[4]]
## [1] "TRUE"

```

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

Answer: I think theoretically both should work, but I'm not sure how to return values from `ifelse` statement. Thus, `if else` statement works better under this scenario.