

Assignment 2: Coding Basics

Tay Holliday

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.

```
hundred_by_four <- seq(1,100,4) # creates a sequence of numbers from 1 to 100 by 4. Assigns sequence th
hundred_by_four #prints the 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_of_seq <- mean(hundred_by_four) # finds mean of sequence hundred_by_four. Assigns name of mean_of_
median_of_seq <- median(hundred_by_four) # find median of sequence hundred_by_four. Assigns name of med
mean_of_seq # prints mean of sequence
```

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

```
median_of_seq # prints median of sequence
```

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

#3.

```
mean_of_seq > median_of_seq # determines if mean is greater than median. If true, will return true. If .
```

```
## [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("Ann","Bobby","Tay","Fred") # Character vector
scores <- c(35,72,100,51) # numeric vector
passed <- c(FALSE,TRUE,TRUE,TRUE) #Logical vector
test_data_frame <- data.frame(names, scores, passed)
names(test_data_frame) <- c("Name","Score","Passed?"); View(test_data_frame)
print(test_data_frame)
```

```
##      Name Score Passed?
## 1   Ann     35   FALSE
## 2 Bobby     72    TRUE
## 3   Tay    100    TRUE
## 4  Fred     51    TRUE
```

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

Answer: This data frame uses vectors. A vector is one dimensional, meaning that each column in this case is a separate vector. To make the data frame, we combine the separate vectors. A matrix has rows and columns already assigned.

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.

```
is_grade_passing <- function(x){
  if (x > 50){
    TRUE
  }
  else if (x == 50){
    TRUE
  }
  else {
    FALSE
  }
}
test_one <- is_grade_passing(scores)
```

```
## Warning in if (x > 50) {: the condition has length > 1 and only the first
## element will be used
```

```
## Warning in if (x == 50) {: the condition has length > 1 and only the first
## element will be used
```

```
test_one
```

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

```
is_grade_passing_ifelse <- function(x){
  ifelse(x > 50, 'TRUE', 'FALSE')
}
test_two <- is_grade_passing_ifelse(scores)
test_two
```

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

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

Answer: `ifelse` worked because it returned the entire vector whereas `if, else` only return the first value in the vector.