

Assignment 2: Coding Basics

Allison_Barbaro

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. #creating sequence with the format seq (from, to, by)  
seq(1, 100, 4)
```

```
## [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
```

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

```
#2. #finding the mean and median of my sequence, which I named hundred_seq  
mean(hundred_seq)
```

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

```
median(hundred_seq)
```

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

```
#3. #determining whether the mean of this sequence is greater than the median
mean(hundred_seq) > median(hundred_seq)
```

```
## [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 #created vectors of student names, scores, and whether they passed or failed
student_names <- c("Ally", "Maeve", "Laura", "Aileen")
#this is a character vector
student_scores <- c(48, 75, 99, 92)
#this is a double vector
pass_fail <- c(FALSE, TRUE, TRUE, TRUE)
#this is a logical vector

df <- data.frame(student_names, student_scores, pass_fail)
gradebook <- df
gradebook
```

```
##   student_names student_scores pass_fail
## 1      Ally      48      FALSE
## 2     Maeve      75       TRUE
## 3     Laura      99       TRUE
## 4    Aileen      92       TRUE
```

```
colnames(gradebook) <- c('Names', 'Scores', 'P/F')
gradebook
```

```
##   Names Scores  P/F
## 1  Ally    48 FALSE
## 2 Maeve    75  TRUE
## 3 Laura    99  TRUE
## 4 Aileen   92  TRUE
```

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

Answer: A matrix can only hold a single dataset, while data frames can hold multiple.

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.

```
scores_2 <- function(x) {  
  ifelse(x > 50, "TRUE", "FALSE")  
}  
  
pass_fail <- scores_2(student_scores); pass_fail
```

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

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

Answer: 'ifelse' worked, while 'if' and 'else' did not. 'if' and 'else' functions can only check one element in a vector at a time, while 'ifelse' can check every element in a vector. Our vector contains multiple elements.