

Assignment 2: Coding Basics

Rachel Gordon

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. Use the seq command to get a sequence of numbers from 1 to 100 at intervals of 4  
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
```

```
#Assigning sequence above the name sequence.a  
sequence.a<-seq(1,100,4)
```

```
#2. Compute the mean and median of this sequence  
mean(sequence.a)
```

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

```
median(sequence.a)
```

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

```
#3. Determining if the mean of sequence.a is greater than the median  
mean(sequence.a)>median(sequence.a)
```

```
## [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.

```
# create a series of vectors
a <- c("Rachel", "Sara", "John", "Michael") #character vector
b <- c(87,43,90,82) #numeric vector
c <- c(TRUE,FALSE,TRUE,TRUE) #logical vector

# Combining vectors into a data frame, assigning the data frame a name
student.scores <-data.frame(a,b,c) #combining vectors into a data frame and naming data frame

#label columns of data frame
names(student.scores) <-c("Name","Score","Passed"); View(student.scores)

print(student.scores)
```

```
##      Name Score Passed
## 1  Rachel    87    TRUE
## 2   Sara    43   FALSE
## 3   John    90    TRUE
## 4 Michael    82    TRUE
```

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

Answer: This data frame is different from a matrix because in this data frame, there is data of different types in each column. In a matrix, however, all the data would be of the same data type.

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.

```
# Create a function with an if/else statement to indicate True/False for a passing grade more than 50

passing.test.score <- function(x) {
  ifelse(x >= 50,"TRUE","FALSE")
}

# Apply function to the test score vector
passing.test.score(b)
```

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

```
# Create a function with 'if' and 'else' statement to indicate True/False for a passing grade more than 50
passing.test.score2 <- function(x) {
  if(x >= 50) {
    TRUE
  }
  else {
    FALSE
  }
}
```

```
# Apply function to the test score vector  
passing.test.score2(b)
```

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

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

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

Answer: `'ifelse'` worked. This is because `'ifelse'` is used to check every element of the vector, whereas using just `'if'` and `'else'` will only check the first element of the vector. As shown above, using `'if'` and `'else'` creates a warning message that only the first element will be used and then only returns the data for the first element of the vector when the function is applied.