

Assignment 2: Coding Basics

Ellen Nirenblatt

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, Part 1

1. Generate a sequence of numbers from one to 30, increasing by threes. 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.  
thirty_seq <- seq(1, 30, 3) #from 1 to 30, by 3  
thirty_seq # runs the function
```

```
## [1] 1 4 7 10 13 16 19 22 25 28
```

```
#2.  
mean(thirty_seq)
```

```
## [1] 14.5
```

```
median(thirty_seq)
```

```
## [1] 14.5
```

```
#3.

#make mean and median into objects
x <- mean(thirty_seq)
y <- median(thirty_seq)

#name function and compare values of mean(x) and median(y)
sequenceFunction <- function() {
  if(x > y) {
    print(FALSE)
  }
  else if (x < y) {
    print(FALSE)
  }
  else {
    print(TRUE)
  }
}

sequenceFunction()
```

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

Basics, Part 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.

students <- c("caroline","charles","ellen","bella") #vector type is a string of characters
scores <- c(82, 84, 93, 97) #vector type is an integer
passingscore <- (scores > 50) #vector type is Boolean

is.vector(students)
```

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

```
is.vector(scores)
```

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

```
testresults <- data.frame(
  Names <-students,
  testscores <-scores,
  Result <-passingscore
)

print(testresults)
```

```
##   Names....students testscores....scores Result....passingscore
## 1      caroline           82           TRUE
## 2      charles           84           TRUE
## 3        ellen           93           TRUE
## 4        bella           97           TRUE
```

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

Answer:

#This data frame is different from a matrix because a matrix consists of the same data type. For example, a matrix would consist only of integers or characters. This data frame has three types of data.

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.

```
#passinggrade <- function(x){
#  if (scores > 50) {
#    print("True")
#  }
#  else if (scores < 50) {
#    print ("false")
#  }
#  else {
#    print ("50")
#  }
#}
#passinggrade()

passedexam <- function() {
  ifelse(scores>50,TRUE,FALSE)
}
passedexam()
```

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

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

Answer: ifelse worked and not the “if and else”. I found that the reason was because the first “if, else” function only works when there is one variable in a list/it will only work on the first variable in a list. By using the ifelse statement, it iterates through each variable in the list, which is why it returned multiple values.