

Assignment 2: Coding Basics

Rachel Schoenecker

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.  
# Creating sequence from 1 to 100 by 4's and naming it "seq.4"  
seq.4 <- seq(1, 100, 4)  
seq.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  
  
#2.  
# Calculating mean and median of seq.4  
mean(seq.4); median(seq.4)  
  
## [1] 49  
## [1] 49  
  
#3.  
# Logical statement asking whether the mean of seq.4 is greater than the median  
mean(seq.4) > median(seq.4)  
  
## [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.

```
#5a.
stud.names <- c("Blair", "Serena", "Dan", "Nate") # stud.names is a character vector
stud.names
```

```
## [1] "Blair" "Serena" "Dan" "Nate"
```

```
#5b.
stud.grades <- c(98, 74, 91, 48) # stud.grades is a numeric vector
stud.grades
```

```
## [1] 98 74 91 48
```

```
#5c.
stud.pass <- c(T, T, T, F) # stud.pass is a logical vector
stud.pass
```

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

```
#7.
gradebook <- data.frame(stud.names, stud.grades, stud.pass)
gradebook
```

```
##   stud.names stud.grades stud.pass
## 1    Blair         98      TRUE
## 2   Serena         74      TRUE
## 3     Dan          91      TRUE
## 4     Nate          48     FALSE
```

```
#8.
colnames(gradebook) <- c("Name", "Grade", "Pass")
gradebook
```

```
##   Name Grade Pass
## 1 Blair    98  TRUE
## 2 Serena   74  TRUE
## 3  Dan     91  TRUE
## 4  Nate    48 FALSE
```

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

Answer: Both are two-dimensional features containing data, but a data frame can include data of many different types (e.g., numeric, character, and logical) while a matrix can only contain one type of data (e.g., numeric or character).

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.
pass.test <- function(grade){
  outcome <- ifelse(grade >= 50, T, F)
  print(outcome)
}
```

#11.

```
pass.test(gradebook$Grade)
```

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

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

Answer: The 'ifelse' option worked because it can evaluate multiple values in a vector in one step while the if/else statement can only evaluate one value at a time. In order to use if/else for a vector of multiple values a 'for' loop would need to be employed as well.