

Assignment 2: Coding Basics

Jess Ozog

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.  
seq1<-seq(1,100,4) ; seq1
```

```
## [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.  
mean1<-mean(seq1) ; mean1
```

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

```
med1<-median(seq1) ; med1
```

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

```
#3.  
mean1 <= med1
```

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

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 vectors
stuNames <- c("Jess", "Laura", "Chris", "Matt") # characters
scores <- c(80, 95, 74, 49) # numeric
passfail <- c("TRUE", "TRUE", "TRUE", "FALSE") # characters

# create data frame with vectors
studentGrades <- data.frame(stuNames, scores, passfail)

# rename columns
colnames(studentGrades) <- c("Student Name", "Test Score", "Pass or Fail")
```

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

Answer: This data frame differs from a matrix because in a data frame, different columns can have different data types. In a matrix, each column must have the same data type. In this data matrix, student names and pass/fail are both character data types, while the test score is a numeric 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.

```
# "if" + "else" statement
gradeFunction <- function(x) {
  if (x >= 50) {
    print("TRUE")
  }
  else {
    print("FALSE")
  }
}
gradeFunction(scores)
```

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

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

```
# "ifelse" statement
gradeFunction2 <- function(x) {
  ifelse(x>=50, "TRUE", "FALSE")
}
gradeFunction2(scores)
```

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

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

Answer: The “ifelse” option worked. For the “if” + “else” option to have worked, a ‘for’ loop would have been needed to cycle through all the values in the vector. The “ifelse” statement looped through all the entire test score vector.