

Assignment 2: Coding Basics

Kallie Davis

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. The sequence (seq) function is being used to generate a sequence of values  
# going from 1 to 100, increasing by fours.  
sequence_1 <- seq(1, 100, 4)
```

```
#2. Calculated the mean and median using the respective functions.  
sequence_1_mean <- mean(sequence_1)  
sequence_1_median <- median(sequence_1)  
sequence_1_mean
```

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

```
sequence_1_median
```

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

```
#3. I created a logical statement to see if the sequence median and mean values  
# were the same.  
sequence_1_mean == sequence_1_median
```

```
## [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.

```
student_names <- c("Amelia", "Sophia", "Charlie", "Sam")      #character
test_scores <- c(64,78,34,97)                                  #numeric
passing_grade <- c(FALSE, TRUE, FALSE, TRUE)                  #logical

student_test_grades.df <- data.frame(student_names,test_scores,passing_grade)
names(student_test_grades.df) <- c(" Student Name","Test Score","Passing Grade")
head(student_test_grades.df)
```

```
##      Student Name Test Score Passing Grade
## 1      Amelia      64      FALSE
## 2      Sophia      78       TRUE
## 3      Charlie      34      FALSE
## 4         Sam      97       TRUE
```

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

Answer: A data frame is capable of containing different data classes; a matrix can only contain a single data class.

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.

```
calc_passing_grade <- function(x) {
  ifelse(x<50, FALSE, TRUE)
}

pass_or_fail <- calc_passing_grade(student_test_grades.df$`Test Score`); pass_or_fail
```

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

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

Answer: The **ifelse** option worked. The **if** and **else** function will only work with one input value from a vector at a time; the **ifelse** function will evaluate all values in a vector at once.