Assignment 2: Coding Basics

Eric Newton

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.

```r
#1.
seq(1,30,3) #sequence starting at 1 to 30, by 3
```

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

```r
sequencebythree <- seq(1,30,3) #assigning above sequence with descriptive name

#2.
mean(sequencebythree)
```

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

```r
median(sequencebythree) #using functions mean() and median() to calculate values
```

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

```
#3.
mean(sequencebythree) > median(sequencebythree) #used greater than symbol to ask R
```

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

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.
student_names <- c("Peter", "Paul", "Luke", "John")
student_grades <- c(20, 40, 60, 80)
passing_grade <- c(FALSE, FALSE, TRUE, TRUE)

#6.
#student_names is a character vector; student_grades is a numerical vector; passing_grade is a logical

#7.
cbind(student_names, student_grades, passing_grade)
```

```
##      student_names student_grades passing_grade
## [1,] "Peter"       "20"           "FALSE"
## [2,] "Paul"        "40"           "FALSE"
## [3,] "Luke"        "60"           "TRUE"
## [4,] "John"        "80"           "TRUE"
```

```
df_student_performance <- data.frame(student_names, student_grades, passing_grade)


#8.
#columns already listed with informative names?
```

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

   Answer: The data frame has three distinct types of vectors. A matrix can only hold one data type.

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.

```r
#10.
greater_than_50 <- function(x) {
  ifelse(x>50, TRUE, FALSE)
}

#11.
greater_than_50(student_grades)
```

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

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

    Answer: ifelse worked because if and else analyzes individual objects, while ifelse analyzes vectors