Assignment 2: Coding Basics

Analise Lindborg

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 a sequence of numbers from 1 to 100, increasing by fours.
my.sequence.100 <- seq(1, 100, 4)
my.sequence.100 #print the object so it appears in the knitted document
```

```
##  [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 the mean
mean <- mean(my.sequence.100)
mean #print the mean value
```

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

```
#Calculating the median
median <- median(my.sequence.100)
median #print the median value
```

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

```
#3.
#Checking to see if the mean object is greater than the median object (created above)

mean > median
```

```
## [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.

```
#5 & #6
students <- c("Mary", "John", "Phil", "Kate") #character
grades <- c(45, 49, 67, 89) #numeric
pass <- c(FALSE, FALSE, TRUE, TRUE) #logical TRUE/FALSE (boolean)

#7 & #8
student.grades <- data.frame("Students" = students,"Grades" = grades, "Pass" = pass)
student.grades #print
```

```
##   Students Grades  Pass
## 1     Mary     45 FALSE
## 2     John     49 FALSE
## 3     Phil     67  TRUE
## 4     Kate     89  TRUE
```

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

   Answer: This data frame is different than a matrix because it contains multiple data types (characters, numbers, etc.). Matrices only contain one data type, usually numeric.

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.
passing.grade <- function(grade){
  ifelse(grade>50, TRUE, FALSE)
}

passing.grade(grades)
```

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

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

    Answer: I used "ifelse" because it is designed for vectors of length > 1 (more than one value). "if" and "else" is used for one input (i.e. passing.grade(47)), not a vector with multiple values.