Assignment 2: Coding Basics

Jackie Fahrenholz

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.create a sequence; start, stop, by
byfour <- seq(1,100,4)
#let's take a look at it
byfour
```

```
##  [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.compute the mean()
byfour.mean <- mean(byfour)
#print the value
byfour.mean
```

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

```
#compute the median()
byfour.median <- median(byfour)
#print the value
byfour.median
```

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

```
#3. boolean T/F mean > median (should be F b/c they are equal)
byfour.mean > byfour.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.

```
#create a series of vectors (aka lists)

#students is a character vector
students <- c("Jackie", "Cal", "Michael", "Eni")
#scores is numeric
scores <- c(49, 87, 42, 96)
#status (pass/fail) is logical
status <- scores >= 50

#now form these vectors of equal length into data frame
#and make sure that the columns have useful names

df.student.scores <- data.frame("Name" = students, "Score" = scores, "Pass/Fail" = status)

#let's look at what we made!
df.student.scores
```

```
##      Name Score Pass.Fail
## 1  Jackie    49     FALSE
## 2     Cal    87      TRUE
## 3 Michael    42     FALSE
## 4     Eni    96      TRUE
```

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

   Answer: Data frames can have vectors that contain different kinds of data, like in this case we have 3 kinds of vectors. Whereas in a matrix all of the data needs to be of the same "type" i.e. all character or all 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.

```
#create an if/else statement
#if the variable is > or = 50, print TRUE, if not print FALSE
#could edit this to have 'TRUE' be 'pass' and 'FALSE' to be 'fail'
student.grade <- function(x){
  ifelse(x>= 50, 'TRUE', 'FALSE')
}

#this prints the results of the function using the specified vector
# if you just said print(student.grades), it would print the code of the function
student.grade(scores)
```

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

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

    Answer: In this case, `ifelse` worked because you weren't preforming any operations to the values, you just wanted to know if it was 'this' or 'that'. If we wanted the scores of the students to be transformed through an operation then we would want to use `if` and `else` together in the function.