Assignment 2: Coding Basics

Karen Thornton

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 from 1 to 100 by fours
Fours <-seq(1,100,4)
Fours
```

```
##  [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. Get the mean and median of the sequence
mean(Fours)
```

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

```
median(Fours)
```

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

```
#3. Figure out if the mean is greater than the median
mean(Fours) > median(Fours)
```

```
## [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.

```
#student names, this is a categorical vector
names <- c("Lexi","Shirley","Taro","Elise")
#test scores, this is a numerical vector
scores <- c(70,80,49,92)
#did they pass, this is a categorical vector
pass <-c(TRUE,TRUE,FALSE,TRUE)
#make the three vectors into a dataframe
df <-data.frame(names,scores,pass)
df
```

```
##     names scores  pass
## 1    Lexi     70  TRUE
## 2 Shirley     80  TRUE
## 3    Taro     49 FALSE
## 4   Elise     92  TRUE
```

```
#label the column headers
colnames(df)<-c("Names","Scores","Pass")
colnames(df)
```

```
## [1] "Names"  "Scores" "Pass"
```

```
df
```

```
##     Names Scores  Pass
## 1    Lexi     70  TRUE
## 2 Shirley     80  TRUE
## 3    Taro     49 FALSE
## 4   Elise     92  TRUE
```

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

   Answer: The dataframe allows categorical data and numerical data to be in the same place. A matrix would just have numbers (or just have categories).

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 if for getting a passing score
did_they_pass <-function(x){
  ifelse(x>=50,print("Pass"),print("Fail"))
}
#Apply the function to the vector
did_they_pass(scores)
```

```
## [1] "Pass"
## [1] "Fail"
```

```
## [1] "Pass" "Pass" "Fail" "Pass"
```

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

    Answer: The 'ifelse' option worked but the 'if' and 'else if' didn't because it only applied the function to the first test score in the vector. The 'if' and 'else if' only had one oput, but the 'ifelse' option gave an output for all four elements in the vector.