Assignment 2: Coding Basics

Abhay V Rao

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.
risebyfour <- seq(1,100,4)
# created a sequence from 1 to 100, with an increment of four.

#2.
median(risebyfour)
```

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

```
mean(risebyfour)
```

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

```
#3.
mean(risebyfour)>median(risebyfour)
```

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

```
# querying if the mean is greater than the median for this sequence
```

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.

```
# 5-8
Students <- c("Adam", "Bob", "Charizard", "Dirk")
# creating a vector of character type
Scores <-c(53,58,99,40) # creating a vector of numerical type
Passed <- c(TRUE,TRUE,TRUE, FALSE) # creating a vector of logical type
create_df<- data.frame("Name"=Students, "Score"=Scores,"Pass or Fail?"=Passed)
print(create_df)
```

```
##        Name Score Pass.or.Fail.
## 1      Adam    53          TRUE
## 2       Bob    58          TRUE
## 3 Charizard    99          TRUE
## 4      Dirk    40         FALSE
```

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

   Answer: This data frame differs from a matrix, as it contains datasets of multiple types; matrices include data of one type.

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.

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

11. Apply your function to the vector with test scores that you created in number 5.

```
Results <- Passing_Grade(Scores)
  print(Results) # the function renders results in the order of students.
```

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

```
create_df <- data.frame("Student"=Students, "Passed"=Results)
print(create_df) # creating a data frame to visualize the results effectively.
```

```
##     Student Passed
## 1      Adam   TRUE
## 2       Bob   TRUE
## 3 Charizard   TRUE
## 4      Dirk  FALSE
```

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

    Answer: Ifelse worked for me: the 'if' option only considered the first element of the 'scores' data series that I'd created.