ErikPatton_A2_Coding Basics

Erik Patton

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 Day 1

1. Generate a sequence of numbers from one to 100, increasing by fours. Assign this sequence a name.

```
#keep_it_100 <- c(1:100) #creates a basic vector from 1 to 100 in sequence
#(not random). this was just to warm up and because i read the instructions
#wrong the first time
#keep_it_100             #displays basic vector

count_by_four <- seq(1, 100, by = 4) #creates a vector using the 'seq' command,
#starts at 1, goes to 100, counts by 4
count_by_four #displays my vector
```

```
##  [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
```

```
length(count_by_four) #check on the vector, should be 25 integers
```

```
## [1] 25
```

```
#keep_it_crazy <- sample(1:100, size = 100) #creates a vector of 100 integers
#with values between 1 and 100 with a random sequence. just for fun
#keep_it_crazy          #display the random sequence vector
```

2. Compute the mean and median of this sequence.

```
summary(count_by_four) #runs the summary statistics for the vector count_by_four
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1      25      49      49      73      97
```

```
#which includes mean and median
median_my_vector <- median(count_by_four) #assignes just the median to a
#variable name
```

```
mean_my_vector <- mean(count_by_four)   #assigns just the mean to a variable name
median_my_vector  #setting up for the following step and display median
```

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

```
mean_my_vector    #setting up the following step and display mean
```

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

3. Ask R to determine whether the mean is greater than the median.

```
ifelse(mean_my_vector > median_my_vector, "YES", "NO") #answers the question,
```

```
## [1] "NO"
```

```
#returns answer "no" because they are equal
```

4. Insert comments in your code to describe what you are doing.

```
#1. see comments inserted within above r chunk

#2. see comments inserted within above r chunk

#3. see comments inserted within above r chunk
```

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.

```
#load(randomNames) #attempt to do this with a package but could not find the
#package through container?
student_names <- c("Kylie", "Kit", "Damian", "Brooke", "Lilly") #creates a vector
#with the names of my children
student_names #displays the names
```

```
## [1] "Kylie"  "Kit"    "Damian" "Brooke" "Lilly"
```

```
test_scores <- sample(1:100, size=5) #creates a vector of five random integers
#between 1 and 100, note the numbers will change every time the code is run!
test_scores #displays the test scores
```

```
## [1] 26 33 83 97 43
```

```
student_grades.df <- data.frame(student_names,test_scores) #creates a data frame
#that assigns the child to the grade
student_grades.df #displays the data frame
```

```
##   student_names test_scores
## 1         Kylie          26
## 2           Kit          33
## 3        Damian          83
## 4        Brooke          97
## 5         Lilly          43
```

```
passing_analysis <- ifelse(test_scores > 50, "YES", "NO, STUDY MORE!") #determines
#if the score is >50
#passing_analysis #displays the vector showing if they passed
student_analysis.df <- data.frame(student_grades.df,passing_analysis) #creates a
```

```
#data frame to answer (c) by combining the previous data frame with the passing
#analysis. note I wanted to use cbind but couldnt figure it out?
#student_analysis.df #displays the complete data frame, this is part 7
```

6. Label each vector with a comment on what type of vector it is.

```
class(student_names) #returns "character"
```

```
## [1] "character"
```

```
class(test_scores)   #returns "integer"
```

```
## [1] "integer"
```

```
class(passing_analysis)  #returns "character"
```

```
## [1] "character"
```

7. Combine each of the vectors into a data frame. Assign the data frame an informative name.

```
student_analysis.df #the code to do this is in line 89
```

```
##   student_names test_scores passing_analysis
## 1         Kylie          26  NO, STUDY MORE!
## 2           Kit          33  NO, STUDY MORE!
## 3        Damian          83              YES
## 4        Brooke          97              YES
## 5         Lilly          43  NO, STUDY MORE!
```

8. Label the columns of your data frame with informative titles.

```
names(student_analysis.df)[1] <- "Student"  #renames the first column in the
#data frame
names(student_analysis.df)[2] <- "Test Score"  #renames the second column in the
#data frame
names(student_analysis.df)[3] <- "Passed?" #renames the third column in the data
#frame
student_analysis.df #displays the data frame with the new names
```

```
##   Student Test Score         Passed?
## 1   Kylie         26 NO, STUDY MORE!
## 2     Kit         33 NO, STUDY MORE!
## 3  Damian         83             YES
## 4  Brooke         97             YES
## 5   Lilly         43 NO, STUDY MORE!
```

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

   Answer:This data frame contains columns with a mix of classes, the first and #third are "character" and the second in "integer". A matrix must have the same #class for all columns.

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.

See Lines 86 for this code in the r chunk. It is copied here outside an r chunk to answer the questions.

passing_analysis <- ifelse(test_scores > 50, "YES", "NO, STUDY MORE!") #determines if the score is >50 #passing_analysis #displays the vector showing if they passed

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

See line 86. Was I supposed to manually create a vector TRUE and FALSE? I think I jumped ahead a little bit.

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

    Answer: The “ifelse” worked because it will allow a return of either positive #or negative statements based on the arguement.