EmmaBeyer_A02_CodingBasics.Rmd

Emma Beyer

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, Part 1

1. Generate a sequence of numbers from one to 30, increasing by threes. 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.
seq(1, 30, 3)
```

```
##  [1]  1  4  7 10 13 16 19 22 25 28
```

```
#sequence from 1 to 30 increasing by 3
A02_sequence_1to30 <- seq(1, 30, 3)
#assigning name to sequence

#2.
mean(A02_sequence_1to30)
```

```
## [1] 14.5
```

```
#mean of sequence
median(A02_sequence_1to30)
```

```
## [1] 14.5
```

```r
#median of sequence

#3.
mean(A02_sequence_1to30) > median(A02_sequence_1to30)
```

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

```r
#determines if mean is greater than median
```

Basics, Part 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.

```r
names_of_students <- c("Harry", "Ron", "Hermione", "Neville")
#vector of student names
test_scores <- c(70, 40, 100, 60)
#vector of test scores from students
passing_grade <- test_scores > 50
#determines if test scores are above 50

student_grades_df <- data.frame("Student_Names"=names_of_students, "Test_Scores"=test_scores, "Passing_
student_grades_df
```

```
##   Student_Names Test_Scores Passing_Grade
## 1         Harry          70          TRUE
## 2           Ron          40         FALSE
## 3      Hermione         100          TRUE
## 4       Neville          60          TRUE
```

```r
#creating a data frame with students, grades, and their passing grades
```

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

   Answer:A matrix must have the same modes and the same length for all columns, but a data frame can have different modes.

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

```r
student_test_scores <- function(test_scores) {
ifelse(test_scores>50, "TRUE", "FALSE")

}
#creating a function where if test_scores is >50 then its true and <50 false

ifelse_test_scores <- student_test_scores(test_scores); ifelse_test_scores
```

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

```r
#testing to see if students had passing scores using ifelse function
```

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

    Answer:The ifelse function worked because I wanted to evaulate the whole vector (test_scores), and not just a single condition. If and else statements are better when I just want to evaluate one condition, and cannot evaluate a whole vector in one operation.