SusannaJenkins_A02_CodingBasics.Rmd

Susanna Jenkins

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.

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(1,100,4)
seq <- seq(1,100,4) # this is generating a sequence of numbers from 1 to 100
#increasing by 4 and then naming it "seq"

#2.
mean(seq) # this is finding the mean of the sequence
```

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

```
median(seq) # this is finding the median of the sequence
```

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

```
#3.
mean(seq)>median(seq) # this is showing whether the mean is greater than the
```

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

```
#median
```

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
#a.
name <- c('susanna', 'irene', 'jordan', 'lucy') # character vector (student
#names)
testscore <- c(100, 90, 94, 46) # numeric vector (student test scores)
passing <- c(TRUE, TRUE, TRUE, FALSE) # logical vector (TRUE if  score is
#passing aka greater than 50)

class(testscore)
```

```
## [1] "numeric"
```

```
class(passing)
```

```
## [1] "logical"
```

```
class(name)
```

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

```
examscores <- data.frame(name=name, testscore=testscore, passing=passing)

examscores
```

```
##       name testscore passing
## 1 susanna       100    TRUE
## 2   irene        90    TRUE
## 3  jordan        94    TRUE
## 4    lucy        46   FALSE
```

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

   Answer: The data frame contains different types of data in the column, but a matrix can only contain a single class of data

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.

```
PassingScores <- function(scores) {
  reviewscores <- ifelse(scores>=50, TRUE, FALSE)
print(reviewscores)
}
PassingScores(testscore)
```

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

```
# PassingScores <- function(scores) {
#   if(scores <= 50) {
#     FALSE
#   }
#   else {
#     TRUE
#   }
# }
# PassingScores(testscore)
```

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

    Answer: ‘ifelse’ works here. ‘if’ ‘else’ doesn’t work here because you can’t run a vector greater than 1. When I ran ‘if’ ‘else’, it gave the following error: Error in if (scores <= 50) { : the condition has length > 1