YueZhang_A02_CodingBasics.Rmd

Yue Zhang

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.
seq1 <- seq(1,100,4) #from, to, by
seq1
```

```
##  [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.
meanseq1 <- mean(seq1)
meanseq1
```

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

```
medianseq1 <- median(seq1)
medianseq1
```

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

```
#3.
meanseq1 < medianseq1
```

```
## [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.

```
#5.
a <- c('yue','jilin','will','fangyi','zoe') #character
a
```

```
## [1] "yue"    "jilin"  "will"   "fangyi" "zoe"
```

```
b <- c(90,93,89,94,49) #numeric
b
```

```
## [1] 90 93 89 94 49
```

```
c <- c(TRUE,TRUE,TRUE,TRUE,FALSE) #logical
c
```

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

```
#6.
class(a)
```

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

```
class(b)
```

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

```
class(c)
```

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

```
#7.
df_studenttest <- data.frame(a,b,c)
df_studenttest
```

```
##        a  b     c
## 1    yue 90  TRUE
## 2  jilin 93  TRUE
## 3   will 89  TRUE
## 4 fangyi 94  TRUE
## 5    zoe 49 FALSE
```

```
#8.
names(df_studenttest) <- c('name','test_score','pass')
df_studenttest
```

```
##     name test_score  pass
## 1    yue         90  TRUE
## 2  jilin         93  TRUE
## 3   will         89  TRUE
## 4 fangyi         94  TRUE
## 5    zoe         49 FALSE
```

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

   Answer: Matrices can only contain a single class of data, while data frames can consist of many different classes 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.

```
#10.
test_score <- function(x){
  ifelse(x >= 50, TRUE, FALSE)
}

#11.
test_score(b)
```

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

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

    Answer: ‘ifelse’worked. ’if’ and ‘else’ function only deal with a single value. So if we put a vector in the ‘if’ statement, it will only check the very first element and issue a warning. But the ‘ifelse’ can check the condition for every element of a vector and select elements from the specified vector depending upon the result.