rbasic_20190909_101_kim

3.3. Exercises

1. What is the sum of the first 100 positive integers? The formula for the sum of integers 1 through n is n(n+1)/2. Define n = 100 and then use R to compute the sum of 1 through 100 using the formula. What is the sum?

My answer (1)

```
n <- 100
n(n+1)/2
```

```
## Error in n(n + 1):   "n"
```

There was an error. I shouldn't omit the operator * in R. **Skipping the operator and using parentheses caused the program to misjudge it as a `function`.**

My answer (2)

```
n <- 100
n*(n+1)/2
```

```
## [1] 5050
```

It's now a good compilation!

2. Now use the same formula to compute the sum of the integers from 1 through 1,000.

My answer

```
n = 1000
n*(n+1)/2
```

```
## [1] 500500
```

Compiled well without any problems.

3. Look at the result of typing the following code into R:

```
n <- 1000
x <- seq(1, n)
sum(x)
```

```
## [1] 500500
```

Based on the result, what do you think the functions `seq` and `sum` do? You can use the `help` system:

A. `sum` creates a list of numbers and `seq` adds them up.

B. `seq` creates a list of numbers and `sum` adds them up.“‘

C. `seq` computes the difference between two arguments and `sum` computes the sum of 1 through 1000.

D. `sum` always returns the same number.

4. In math and programming, we say that we evaluate a function when we replace the argument with a given number. So if we type sqrt(4), we evaluate the sqrt function. In R, you can evaluate a function inside another function. The evaluations happen from the inside out. Use one line of code to compute the log, in base 10, of the square root of 100.

My answer

```
log10(sqrt(100))
```

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

5. Which of the following will always return the numeric value stored in `x`? You can try out examples and use the help system if you want.

A. `log(10^x)`

B. `log10(x^10)`

C. `log(exp(x))`

D. `exp(log(x, base = 2))`

```
log(exp(3))
```

```
## [1] 3
```

My answer was A. **But R uses `log` with base ‘e’, not ‘10’(not commercial log, but it’s ln).** The right answer was C.

3.6. Exercises

1. Load the US murders dataset.

```
library(dslabs)
```

```
## Warning: package 'dslabs' was built under R version 3.6.1
```

```
data(murders)
```

Use the function `str` to examine the structure of the `murders` object. We can see that this object is a data frame with 51 rows and five columns. Which of the following best describes the variables represented in this data frame?

A. The 51 states.

B. The murder rates for all 50 states and DC.

C. The state name, the abbreviation of the state name, the state's region, and the state's population and total number of murders for 2010.

D. str shows no relevant information.

2. What are the column names used by the data frame for these five variables?

My answer

```
names(murders)
```

```
## [1] "state"      "abb"        "region"     "population" "total"
```

3. Use the accessor `$` to extract the state abbreviations and assign them to the object `a`. What is the class of this object?

My answer

```
a <- murders$abb
class(a)
```

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

4. Now use the square brackets to extract the state abbreviations and assign them to the object `b`. Use the `identical` function to determine if `a` and `b` are the same.

My answer

```
b <- murders[ , 2]
a == b
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [15] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [29] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [43] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Let's memorize how to extract data with a square bracket!