

# Assessment 01 - R Basics

Phaneendra Valaboju - Harvard Data Science Professional

## Using variables 1

What is the sum of the first  $n$  positive integers? We can use the formula  $n(n + 1)/2$  to quickly compute this quantity.

### Instructions

- Define 'n=100'
- Then use R to compute the sum of 1 through 100 using the formula  $n(n + 1)/2$ . What is the sum?

*# Here is how you compute the sum for the first 20 integers*

```
20*(20+1)/2
```

```
## [1] 210
```

*# However, we can define a variable to use the formula for other values of n*

```
n <- 20
```

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

```
## [1] 210
```

```
n <- 25
```

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

```
## [1] 325
```

*# Below, write code to calculate the sum of the first 100 integers*

```
n <- 100
```

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

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

## Using variables 2

What is the sum of the first 1000 positive integers?

We can use the formula  $n(n + 1)/2$  to quickly compute this quantity.

### Instructions

- Use the same formula as the last exercise but substitute n
- Instead of typing the result, use the formula and defined variable.

*# Below, write code to calculate the sum of the first 1000 integers*

```
n <- 1000
```

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

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

## Functions

Run the following code in the R console:

```
n <- 1000
```

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

```
sum(x)
```

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

### Possible Answers

- `sum` creates a list of numbers and `seq` adds them up.
- `seq` creates a list of numbers and `sum` adds them up. [X]
- `seq` computes the difference between two arguments and `sum` computes the sum of 1 through 1000.
- `sum` always returns the same number

### Nested function calls 1

In math and programming we say we evaluate a function when we replace arguments with specific values. So if we type `log2(16)` we evaluate the `log2` function to get the log to the base 2 of 16 which is 4.

In R it is often useful to evaluate a function inside another function. For example, `sqrt(log2(16))` will calculate the log to the base 2 of 16 and then compute the square root of that value. So the first evaluation gives a 4 and this gets evaluated by `sqrt` to give the final answer of 2.

### Instructions

- Use one line of code to compute the log, to the base 10, of the square root of 100.
- Make sure your code includes the `log10` and `sqrt` functions.

```
# log to the base 2  
log2(16)
```

```
## [1] 4
```

```
# sqrt of the log to the base 2 of 16:  
sqrt(log2(16))
```

```
## [1] 2
```

```
# Compute log to the base 10 (log10) of the sqrt of 100. Do not use variables.  
log10(sqrt(100))
```

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

### Nested functions call 2

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

### Possible Answers

- `log(10^x)`
- `log10(x^10)`
- `log(exp(x))` [X]
- `exp(log(x, base = 2))`