

STAT 33A Lecture - April 20

Topics:

- Iteration in R
 - for-loops
 - while-loops
 - Recursion
- How to Write Iterative Code

Iteration in R

Iterating (or looping) means doing the same operation repeatedly.

4 ways to iterate in R:

- Vectorization (Feb 3 lecture)
- Apply Functions (Apr 13 lecture)
- Loops (this lecture)
- Recursion (this lecture)



For-loops

A **for-loop** iterates over a vector or list.

- One iteration per element.

Differences compared to apply functions:

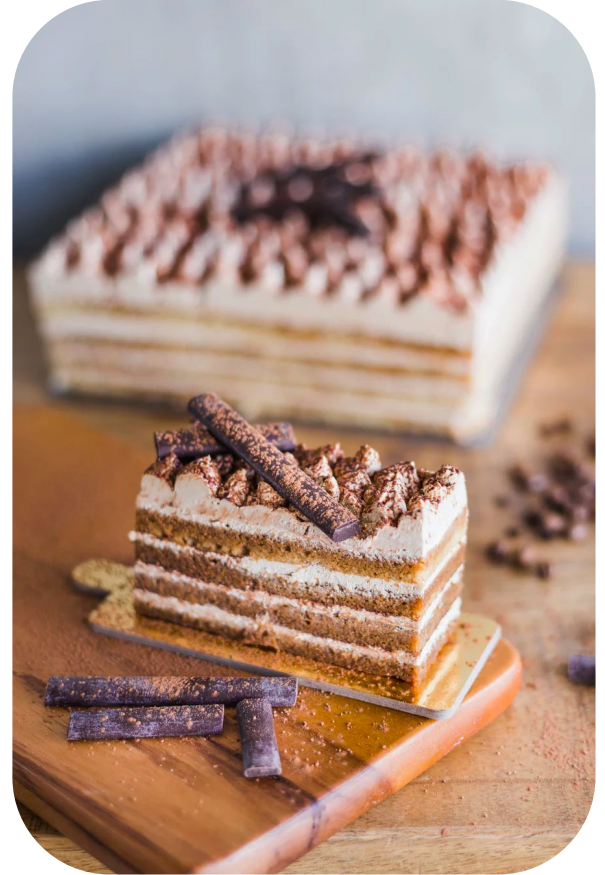
- Iterations can depend on earlier iterations.
- Can skip some iterations, or stop iterations early.
- Results are **not** stored automatically!

Baking a Cake

A slow, silly way to bake a cake:

- Read 1st ingredient: flour
- Go buy flour.
- Read 2nd ingredient: baking soda
- Go buy baking soda.
- Read 3rd ingredient: salt
- Go buy salt.
- ...

Just get the ingredients ahead of time!

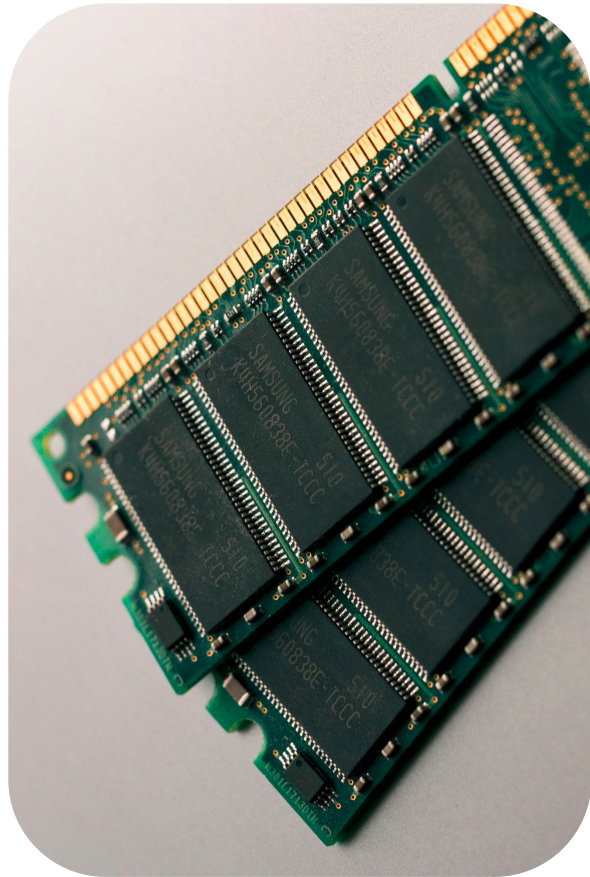


Storing Loop Results

A slow, silly way to store loop results:

- Compute the 1st result.
- Get memory for 1 result.
- Compute 2nd result.
- Get memory for 1 result.
- Compute 3rd result.
- Get memory for 1 result.
- ...

Preallocate: just get the memory ahead of time!



While-loops

A **while-loop** iterates while some condition is TRUE.

- Condition is checked at the **beginning** of each iteration.
- More general than a for-loop.

Use when the number of iterations depends on the iterations.

Example: searching for zeros in a mathematical function.

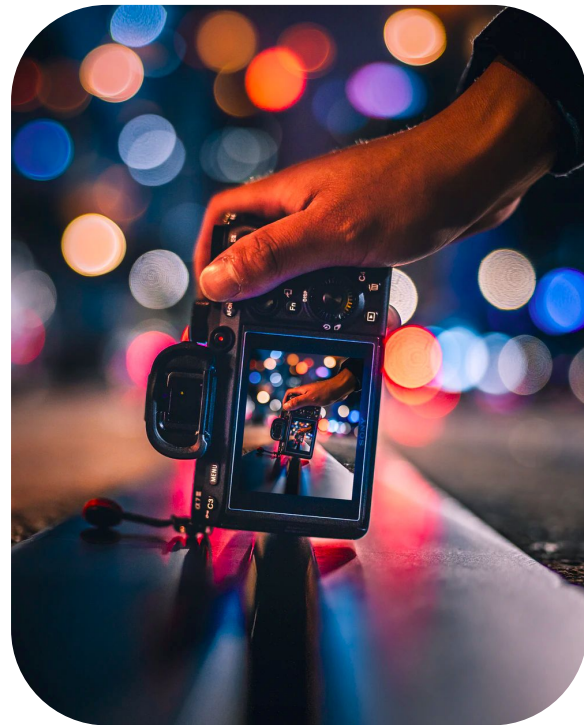
Recursion

A **recursive** function is one that calls itself.

The result is a kind of iteration.

Some problems are naturally recurrent:

- Factorials: $n! = n(n - 1)!$
- Fibonacci: $f_n = f_{n-1} + f_{n-2}$



How to Write Iterative Code

1. Write the code for just one iteration.
2. Test the code! Make sure it works!
3. Edit the code to iterate more than once.
4. Test the code!
 - The `message()` function is helpful for finding problematic iterations.

Choosing an Iteration Strategy

Does the **number of iterations** depend on the iterations?

- While-loop (if yes)

Are some **iterations dependent** on others?

- For-loop (if yes)

Otherwise (the most likely case):

- Vectorization (fastest)
- Apply Function

What about recursion?

Convenient for naturally recurrent problems:

- Recurrent sequences (factorials, Fibonacci numbers, ...)
- Recurrent data structures (file systems, ...)

In R, recursion is often slower than other strategies.