



Code Performance

Nick Ulle



What is code performance?

The primary concerns are how much **time** and **memory** code uses.

In many cases:

- Making code faster makes it use more space (memory or storage).
- Making code use less space makes it slower.

This *space-time tradeoff* arises in many programming problems.

For example, memoization saves time at the expense of memory.

Simple Fixes

“Premature optimization is the root of all evil” — D. Knuth

Fix performance only when performance becomes a problem.

What to do:

1. Find and use packages
2. Find and use specialized functions
3. Vectorize
4. Do less

Elaborate Fixes

Run code in parallel:

- Use the built-in **parallel** package for simple parallelism
- Many problems are difficult to parallelize

Call C, C++, or Fortran code:

- Greater potential for unpleasant bugs
- See Wickham, Ch. 19 & 20



Profiling

Nick Ulle



Profiling

Profiling means analyzing code as it runs to determine:

- Time spent in each function
- Number of calls to each function
- Memory usage

R provides a *statistical profiler* that periodically samples these quantities.





Profvis

Nick Ulle





Profiling Case Study

Nick Ulle

