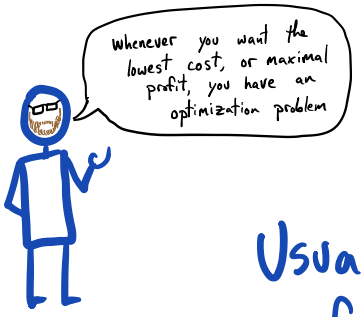
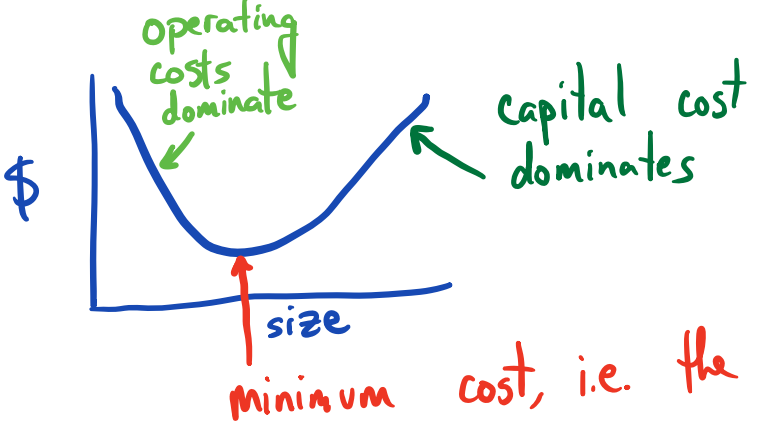
Optimization

AKA Give me the best solution



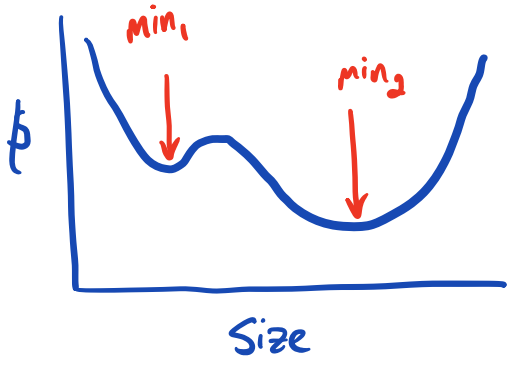
Usually we seek to minimize a function. Take the cost of a process. The capital costs often decrease as a process gets smaller, but operating costs increase because it takes longer to run



minimum cost, i.e. the optimal size

The minimum here is defined by the first derivative $\frac{d\$}{d\text{size}} = 0$ and by the second derivative at the minimum being positive (i.e. the cost function is concave up).

Depending on your cost function there may be many minima:



min_1 and min_2 are both minima but min_2 is lower than min_1. Our job as engineers is to decide which one is a better solution.

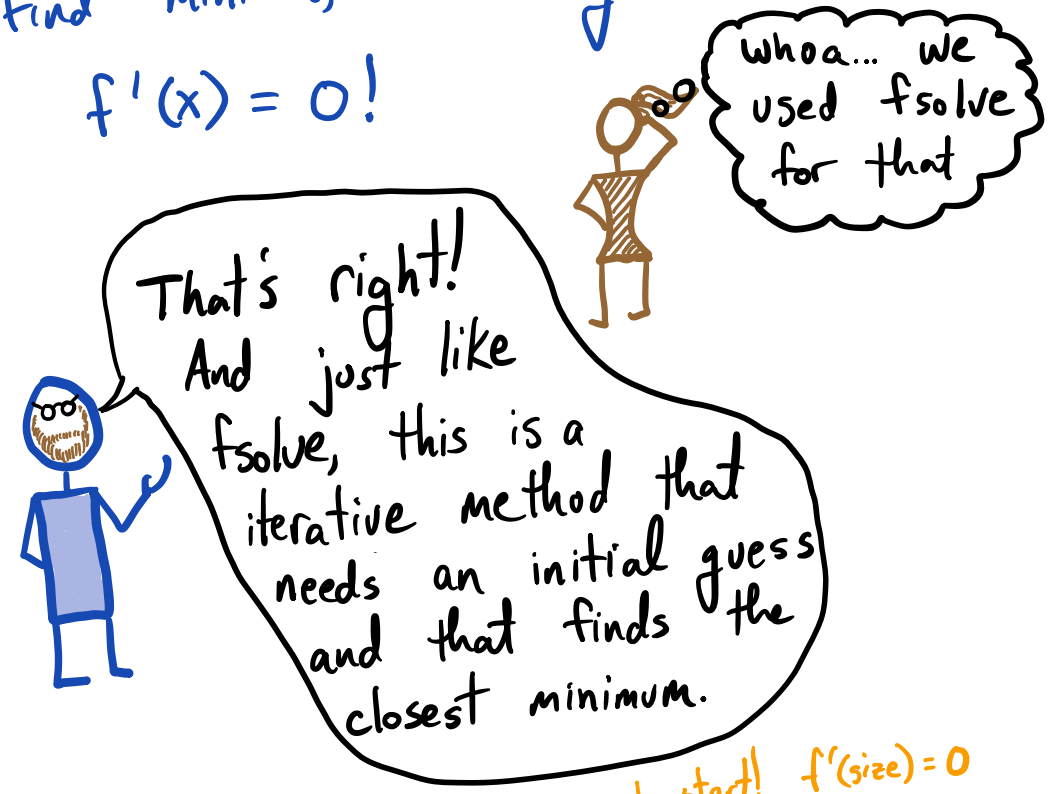
min_1 is called a local minimum

min_2 (at least in this interval) is the global minimum.

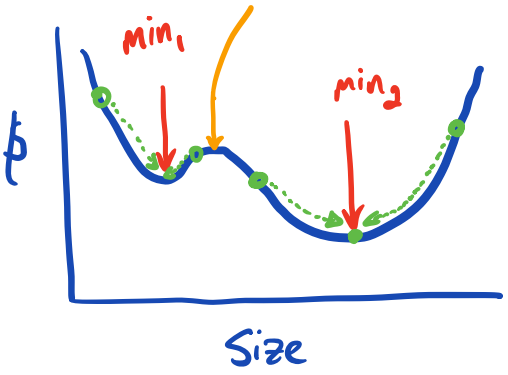
Finding minima

Newton's method can be adapted to find minima; basically we solve

$f'(x) = 0$!



Bad place to start! $f'(size) = 0$



Similar to fsolve, there is

scipy.optimize.minimize

```
def objective(x):
    return some_func(x)
```

minimize(objective, guess) ⇒ sol

sol is a data structure containing the solution.

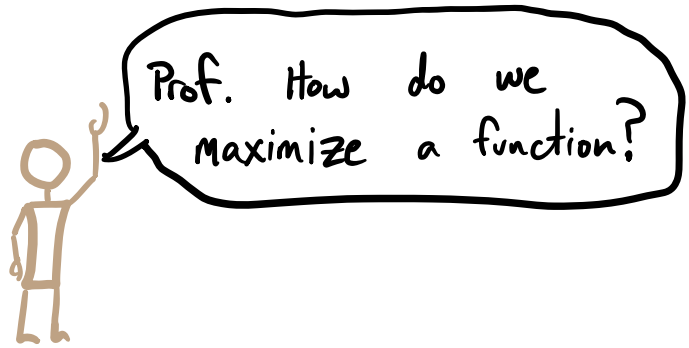
sol.status == 0 when the minimization succeeded

sol.success == True

sol.message == 'Optimization terminated successfully'

sol.x contains an array of the solution

sol.fun is the value of objective(sol.x)



Maximizing a function

Maximizing a function is equivalent to minimizing the negative of that function.