# Lecture 16 - Optimization

* AMA / Quiz / Prayer

## I. Intro to Optimization

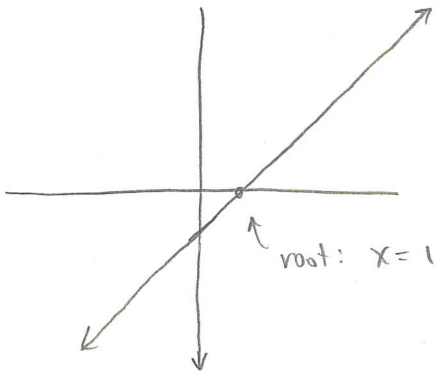
* So far we talked about iterative fixed-point methods as a way to find roots in non-linear equations. Scipy.optimize.root uses something very similar to what you learned to code by hand.

* There is another approach that is also quite common using "optimization" methods. This is most easily illustrated with an example.

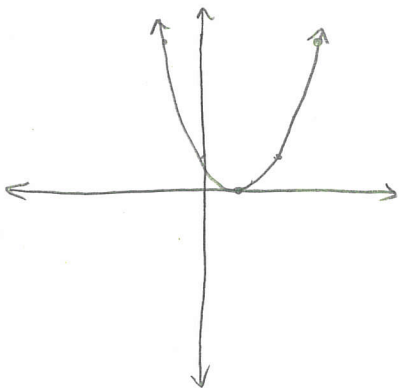
**Example** Suppose we want to find the root of

$$f(x) = x - 1 = 0$$

(a)




(b)



$$S(x) = f^2(x) = (x-1)^2$$

* One way is to look where $f(x)$ crosses zero. This leads us to Newton's method.

* Another way is to look at square of the residual:

$$S(x) = [f(x)]^2$$

Now, instead of looking for the place where $f(x)$ is zero, we look for a place where $S(x)$ is a minimum.

* $S(x)$ is called the objective function or the cost function.

* The process of minimizing the objective function is called optimization. There are many methods to do this, but we don't have time to learn them :(. However, tools for doing this are built into Excel and Scipy in Python.

## II. Optimization in Excel: Solver

* Excel has an optimization tool called "Solver." The best way to learn it is by doing an example.

**Example & Activity** * use solver to solve a single equation & a system of equations.

**Turning on Solver**

Windows: File → Options → Add-Ins → Manage Excel Add-Ins → Go → Check Box for Solver → OK

(To use: Data Tab → Analyze → Solver)

Mac: Tools → Add-Ins → Solver Add-In → OK

(To use: Data Tab → Solver)

## III. Optimization in Python: Scipy.optimize.minimize

* Python has an optimization function called "minimize." The best way to learn it is by doing an example.

**Example & Activity**

* use minimize to solve a single equation & a system of equations.