Where we are

- Supervised Learning
- Rules/Trees
- Overfitting
- Ensembles
- Ex: Random Forests
- k Nearest Neighbors
- *Next*
  - *Optimization with Gradient Descent*
  - *Quick Intuition for Logistic Regression*
  - *Quick Intuition for Support Vector Machines*
  - *Regularization*

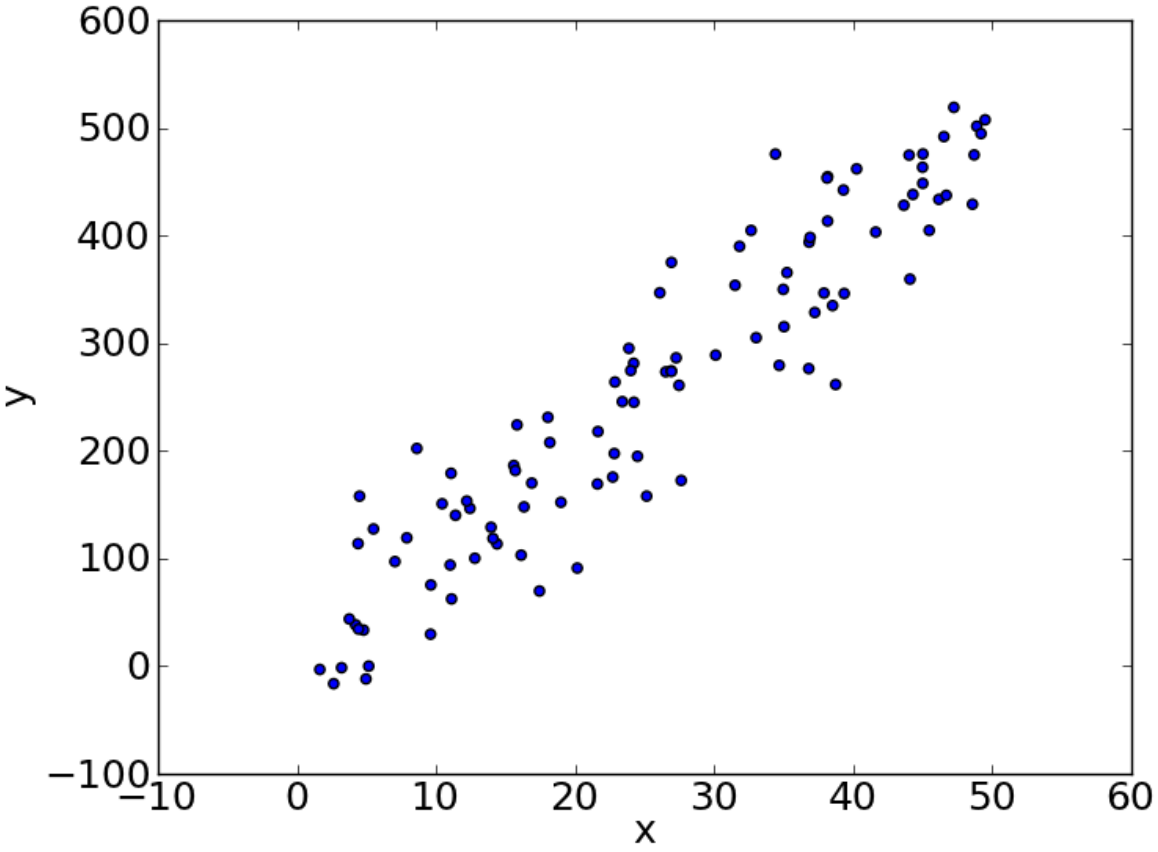
Learning = Three Core Components

- Representation
- Evaluation
- Optimization
  - Gradient Descent

Pedro Domingos, A Few Useful Things to Know about Machine Learning, CACM 55(10), 2012

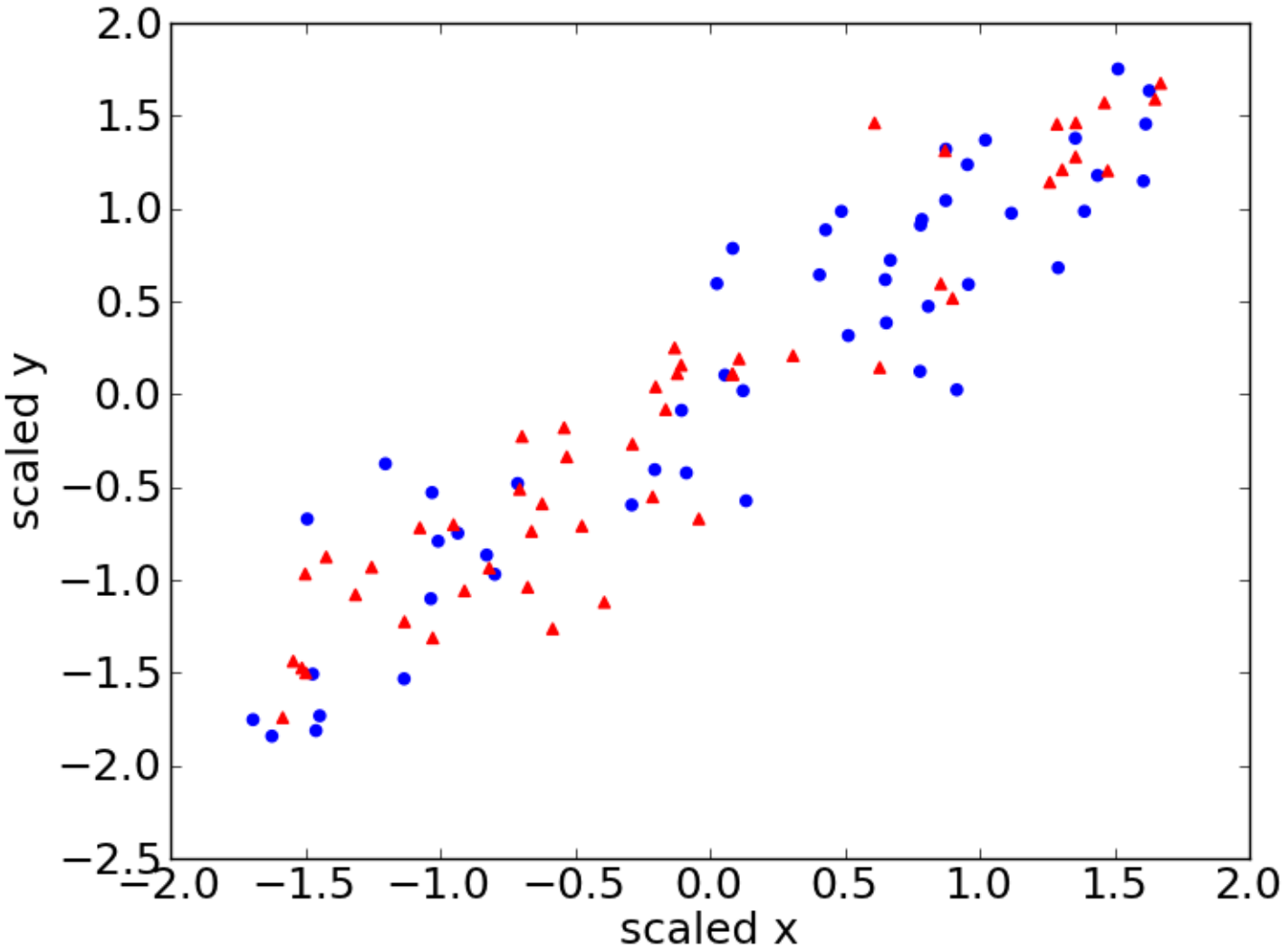
Gradient Descent in a nutshell

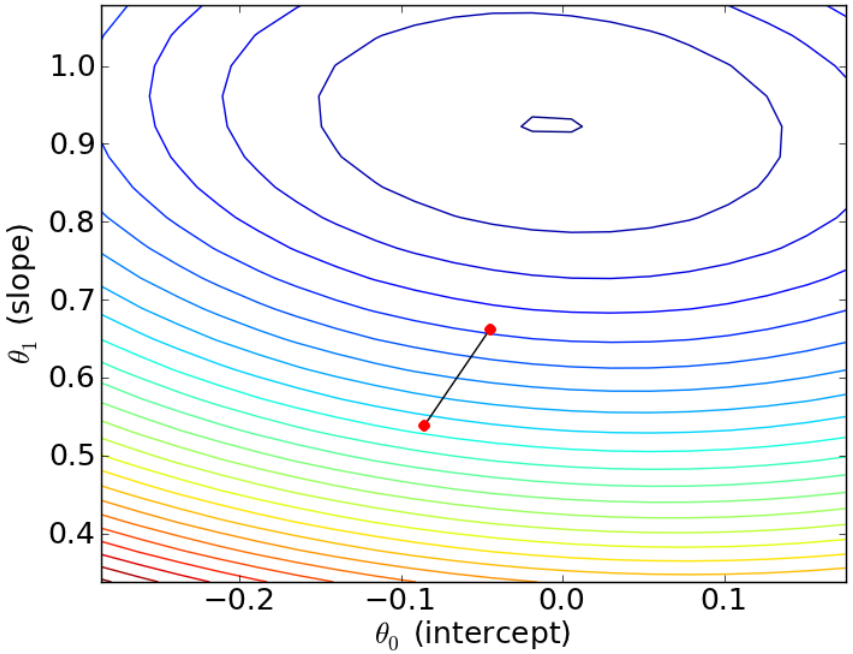
- Express your learning problem in terms of a cost function that should be minimized
- Starting at initial point, step “downhill” until you reach a minimum
- Some situations offer a guarantee that the minimum is the global minimum; others don’t

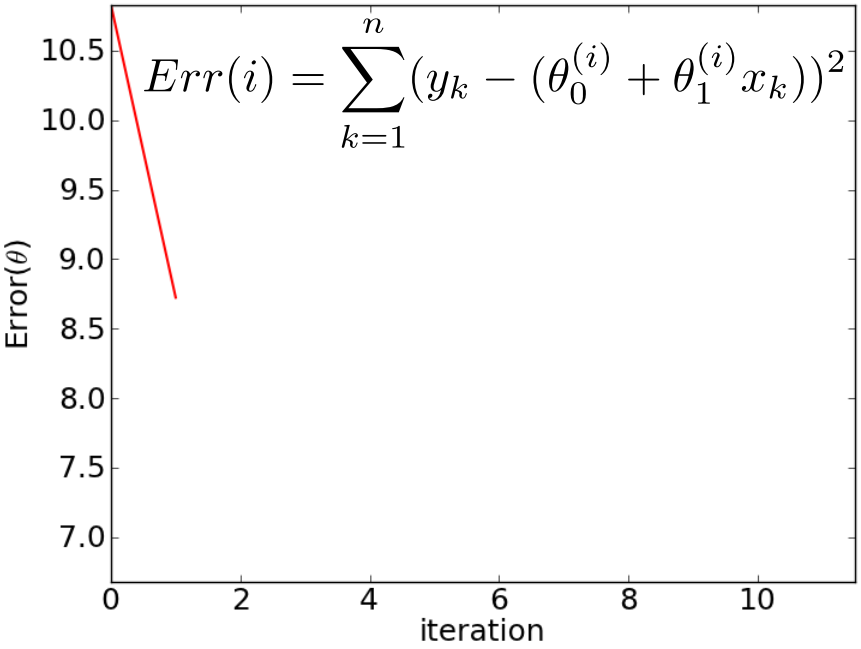


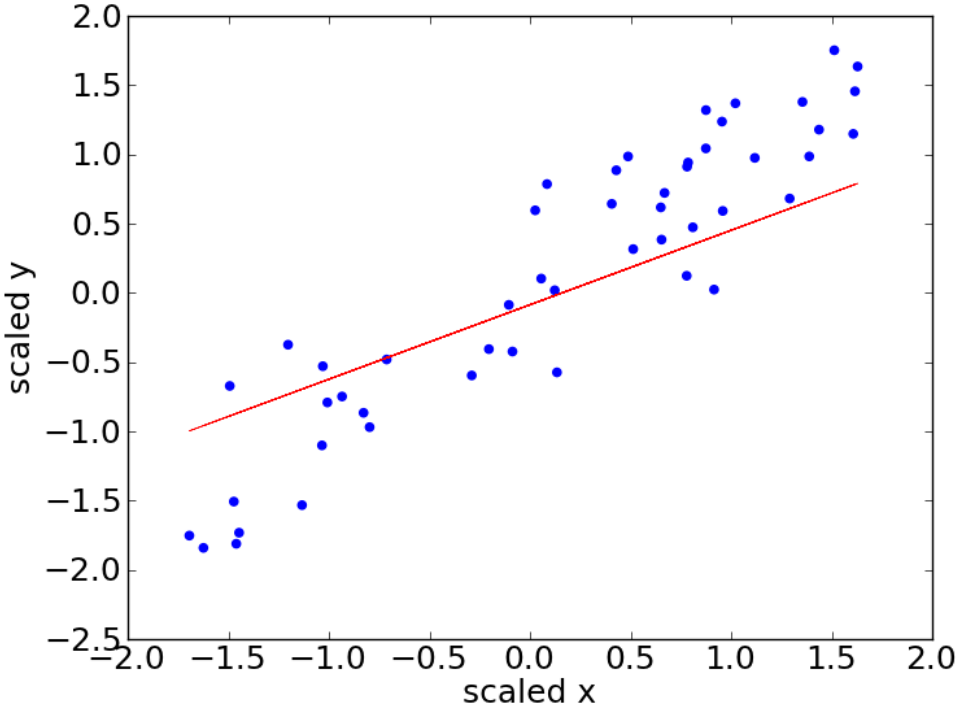
input variable | response variable

x	y
3.1	84.2
19.6	175.8
45.9	448.3
6.8	50.4
3.5	81.9
…	…

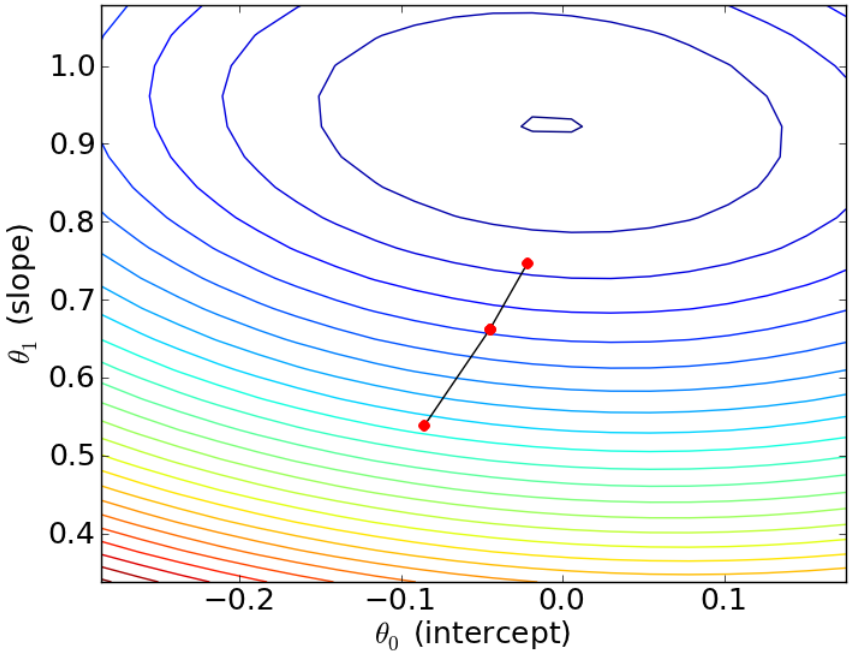

2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
scaled y
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
scaled x

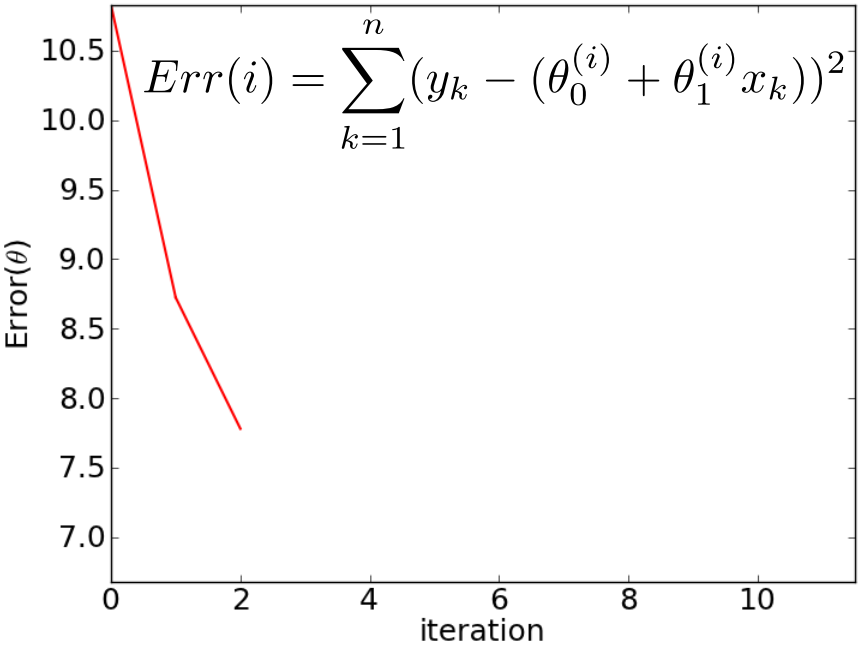


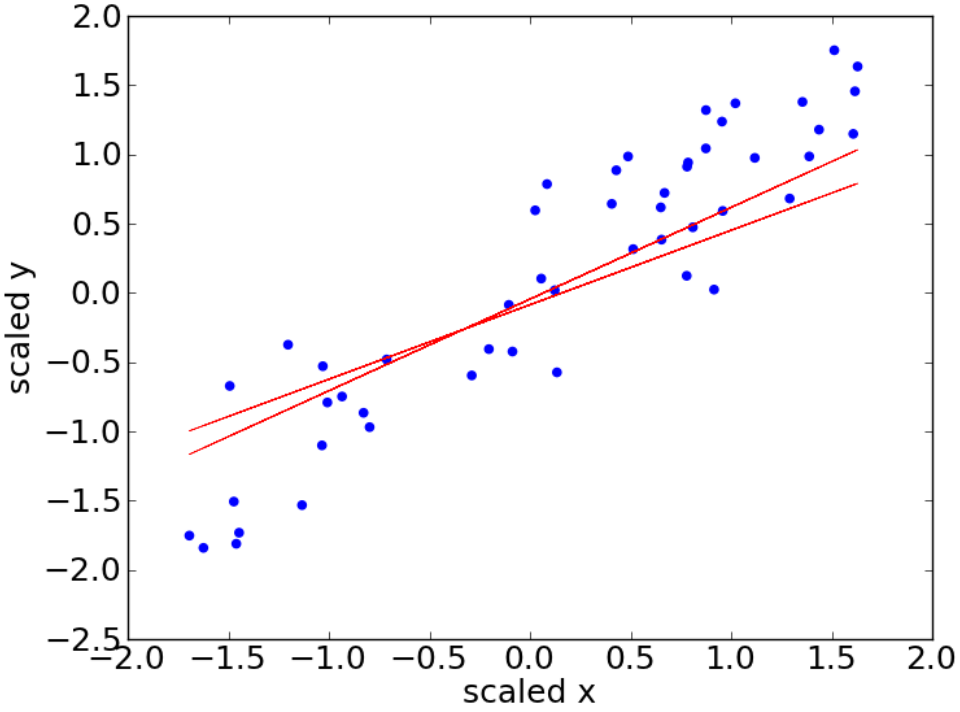




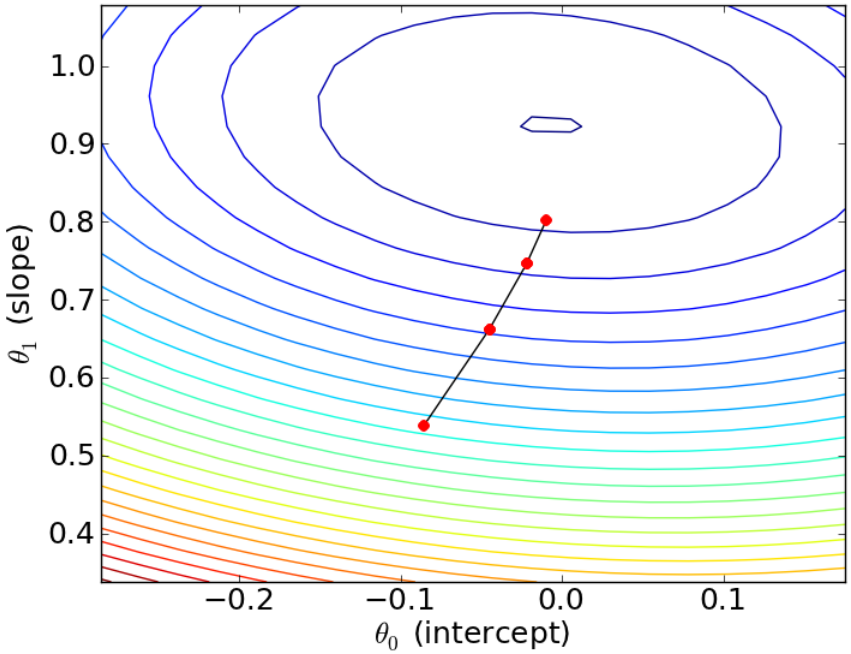
$$y = \theta_0^{(i)} + \theta_1^{(i)} x$$

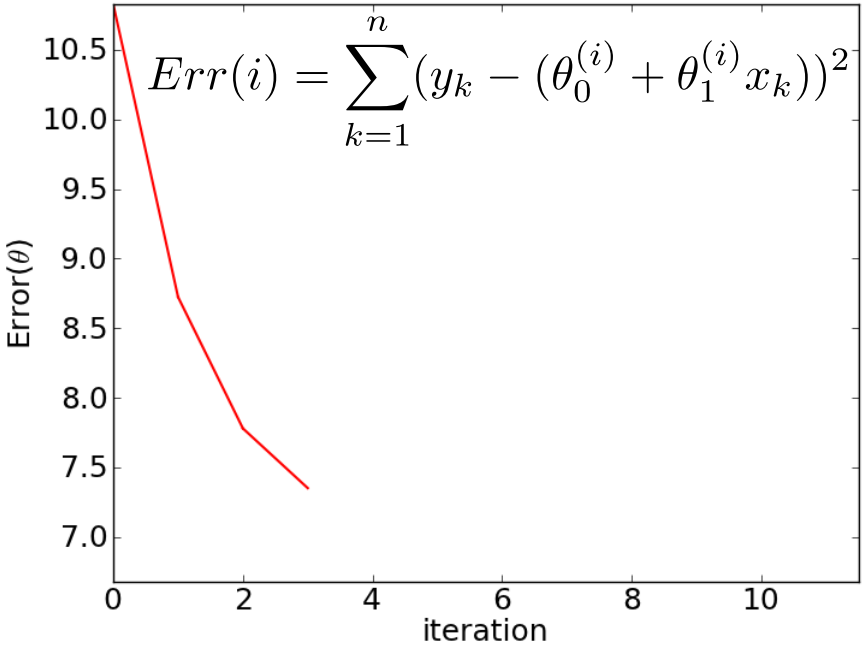


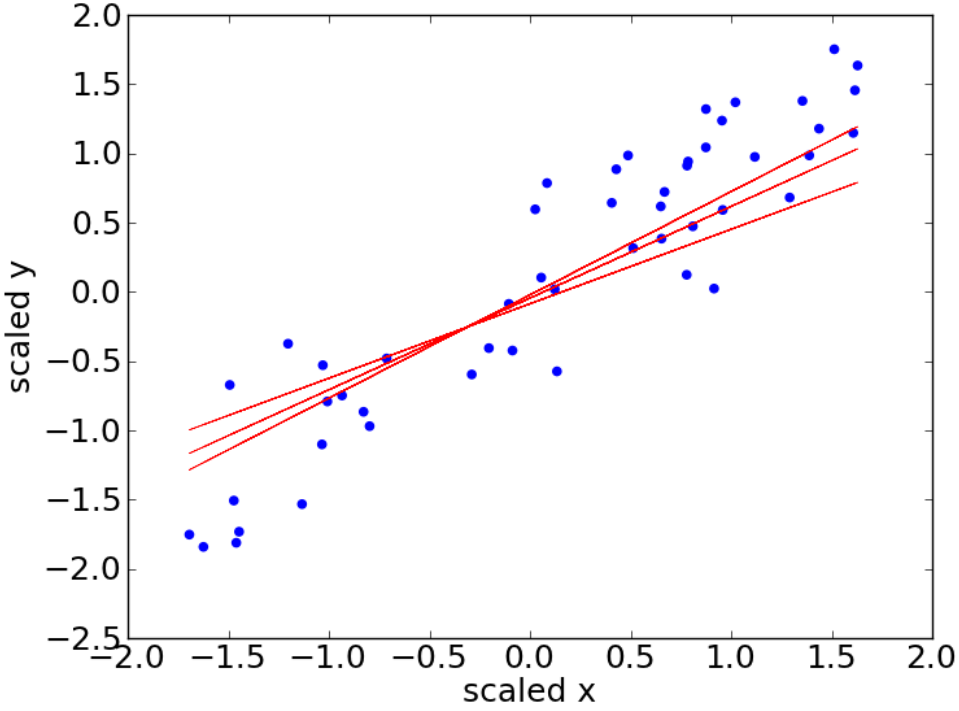




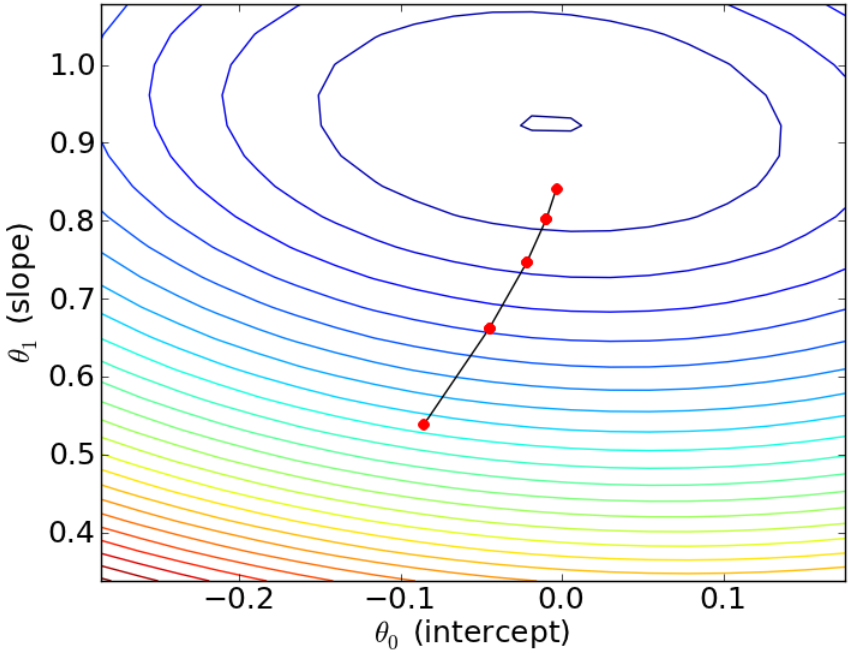
$$y = \theta_0^{(i)} + \theta_1^{(i)} x$$

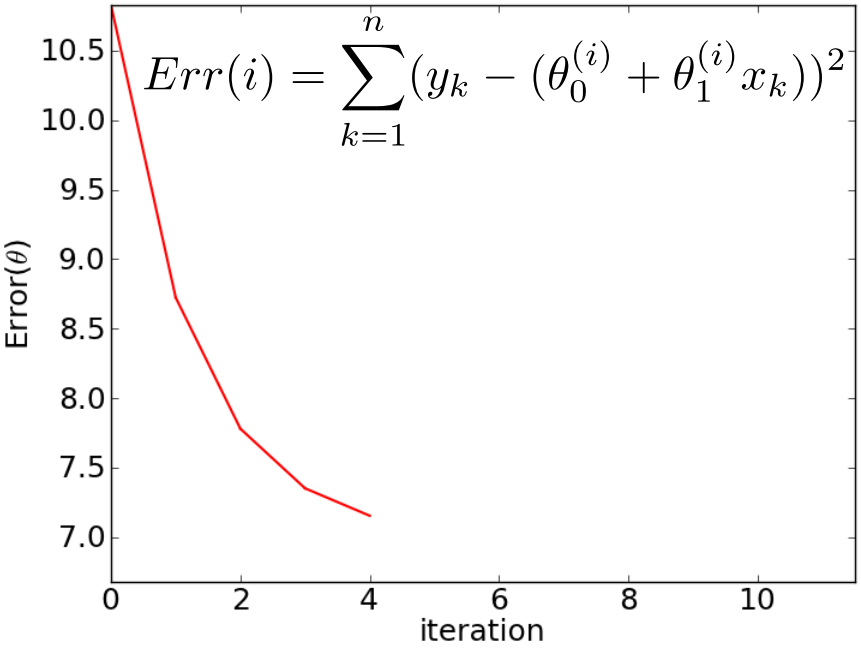


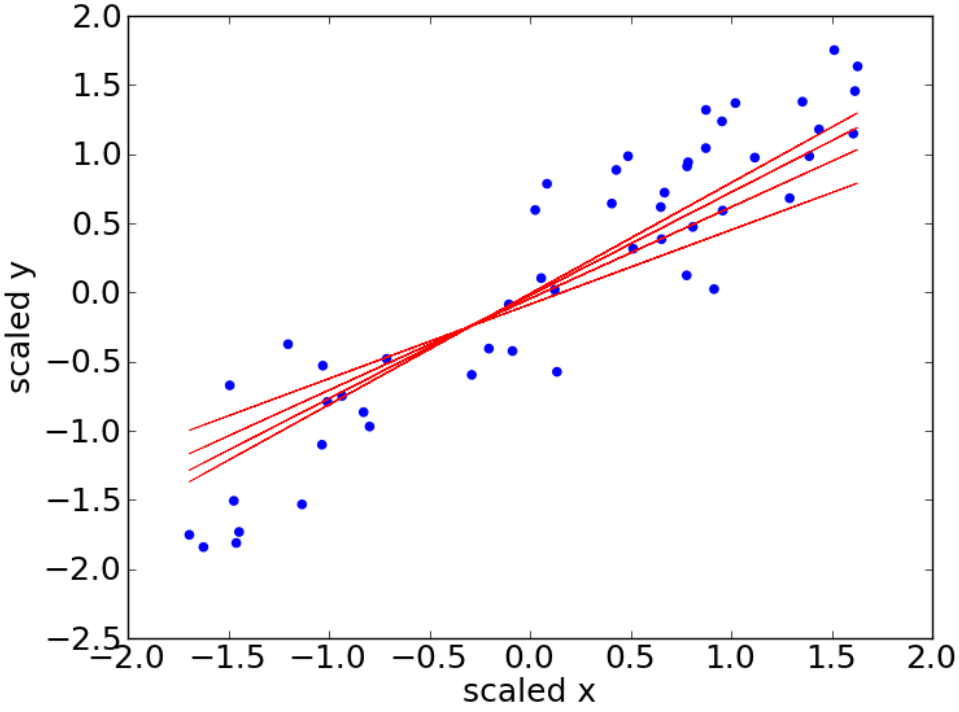




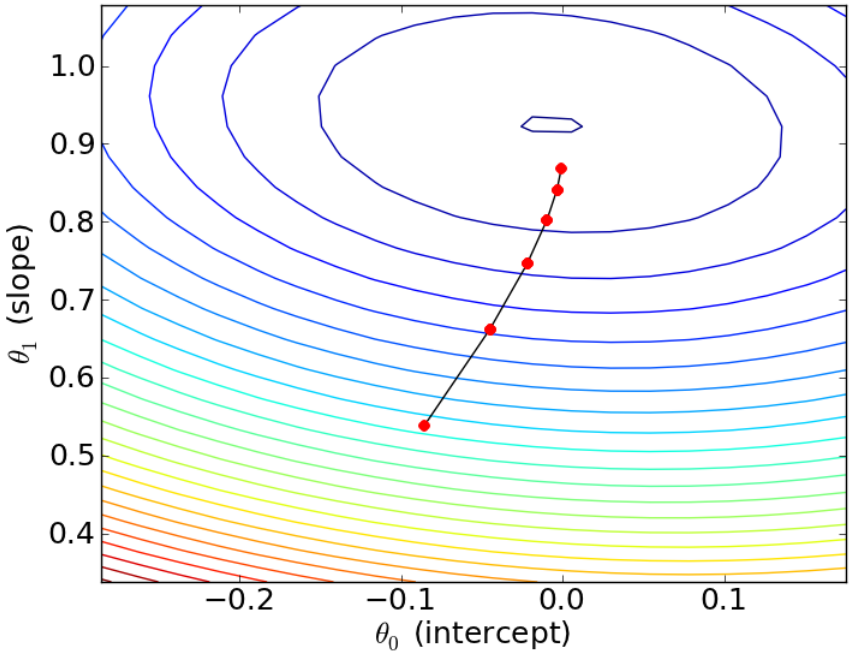
$$y = \theta_0^{(i)} + \theta_1^{(i)} x$$

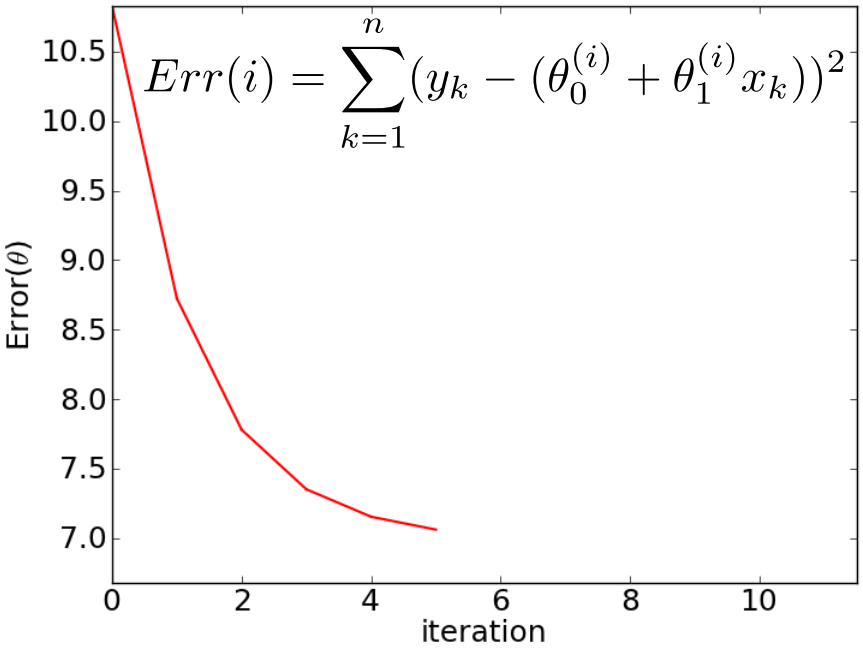


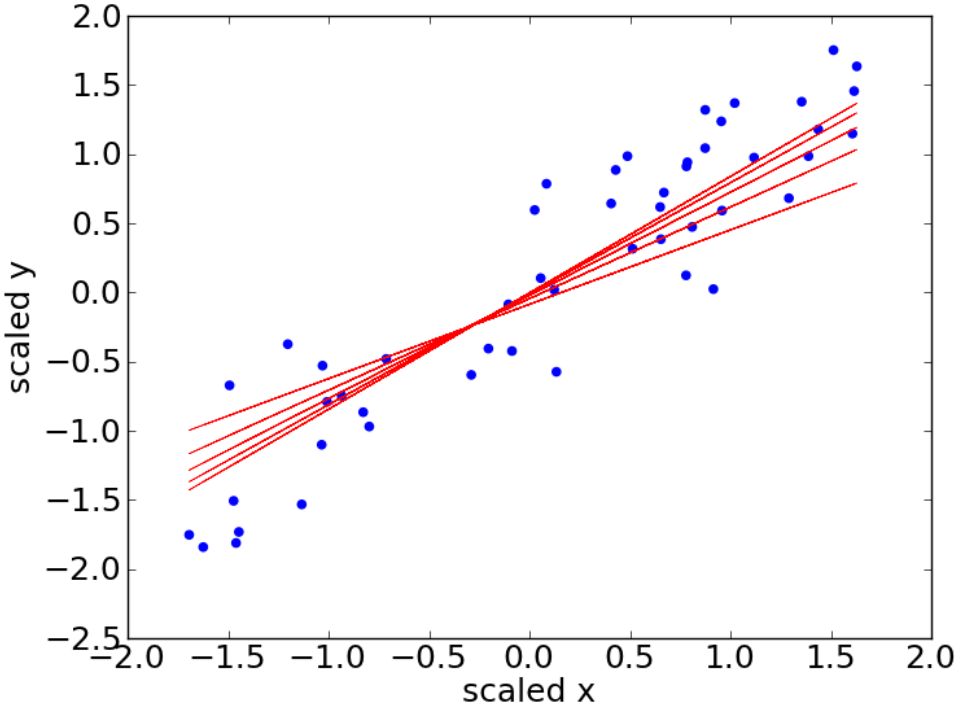




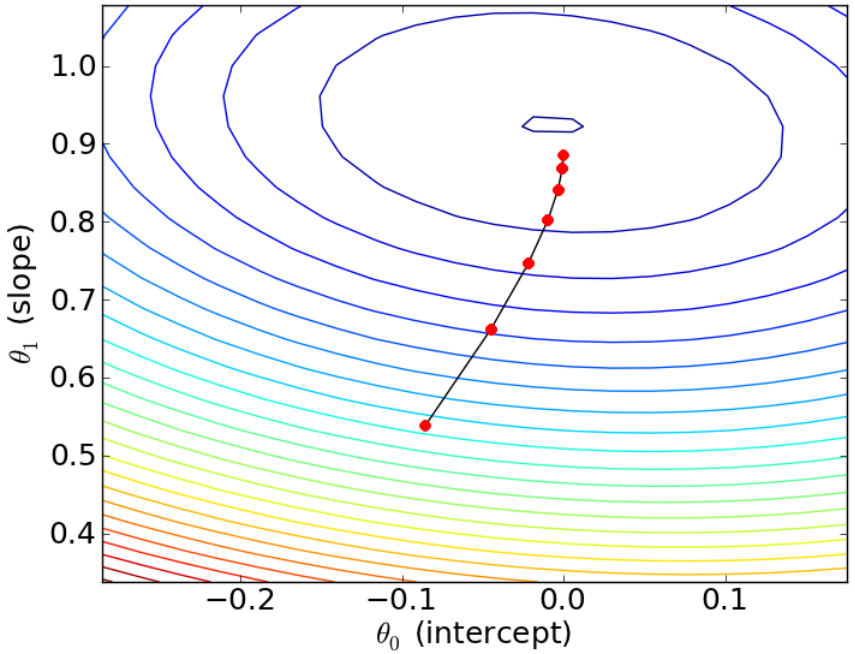
$$y = \theta_0^{(i)} + \theta_1^{(i)} x$$

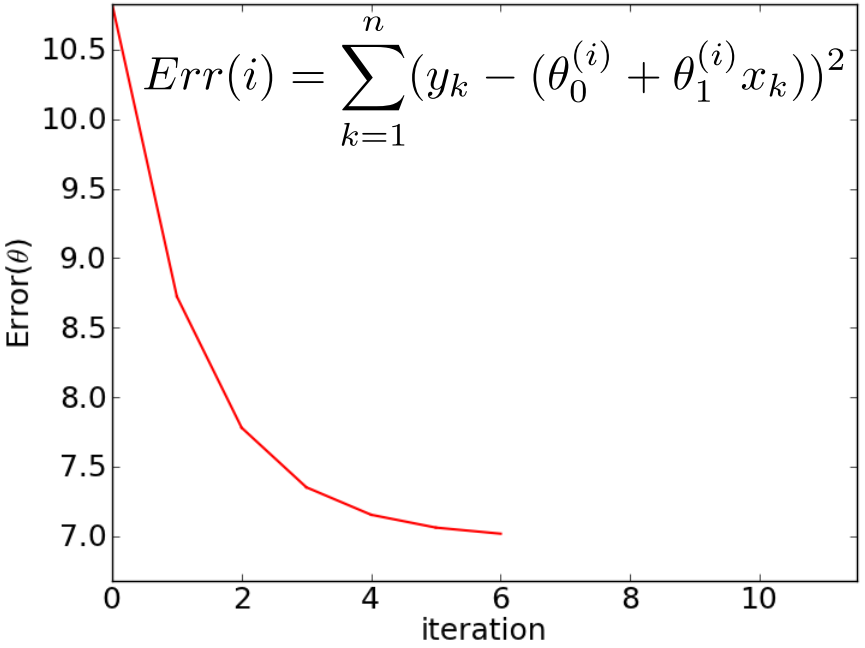


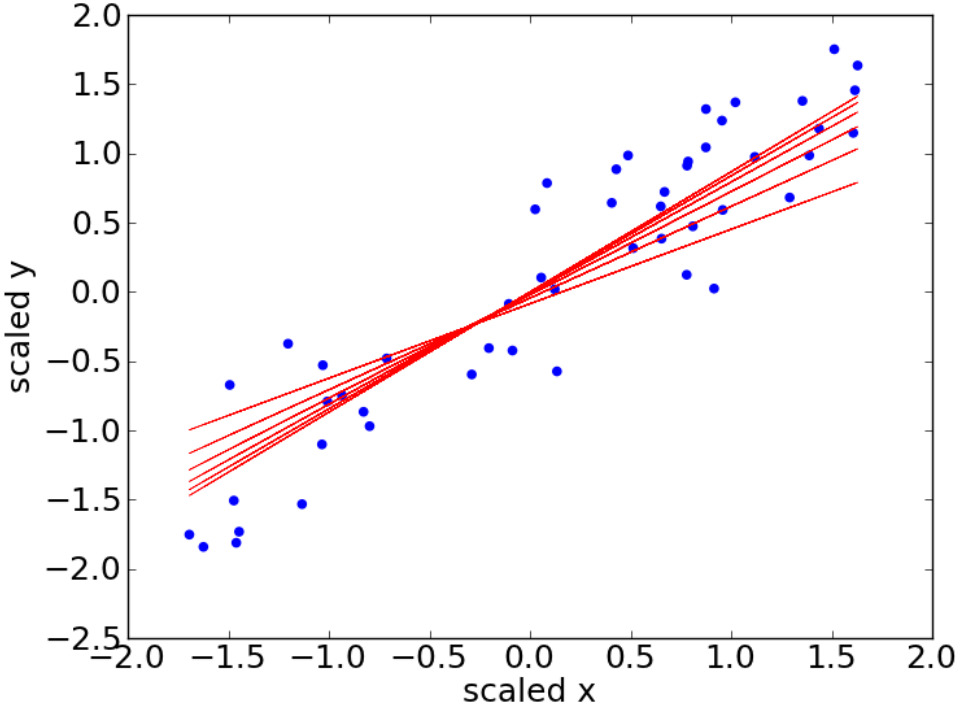




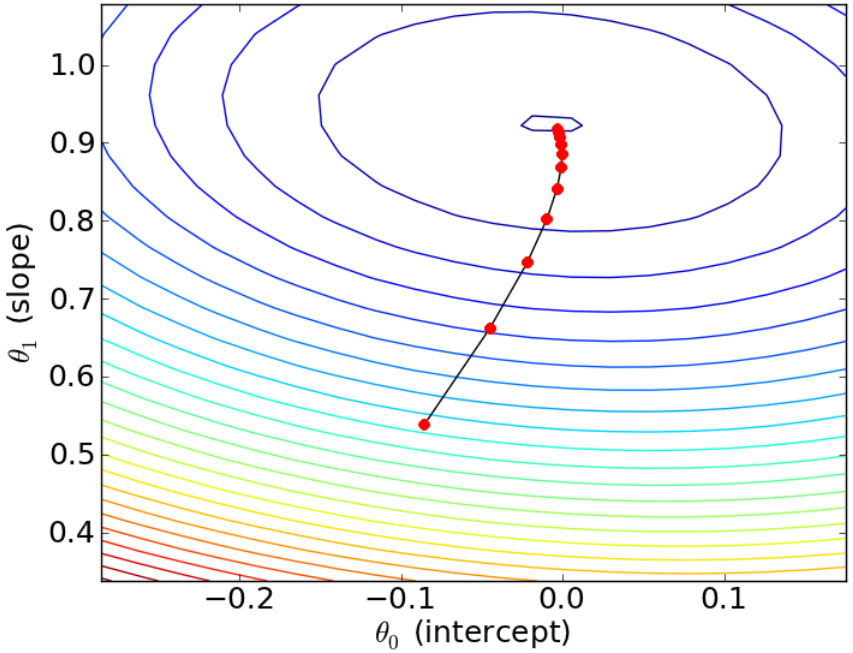
$$y = \theta_0^{(i)} + \theta_1^{(i)} x$$

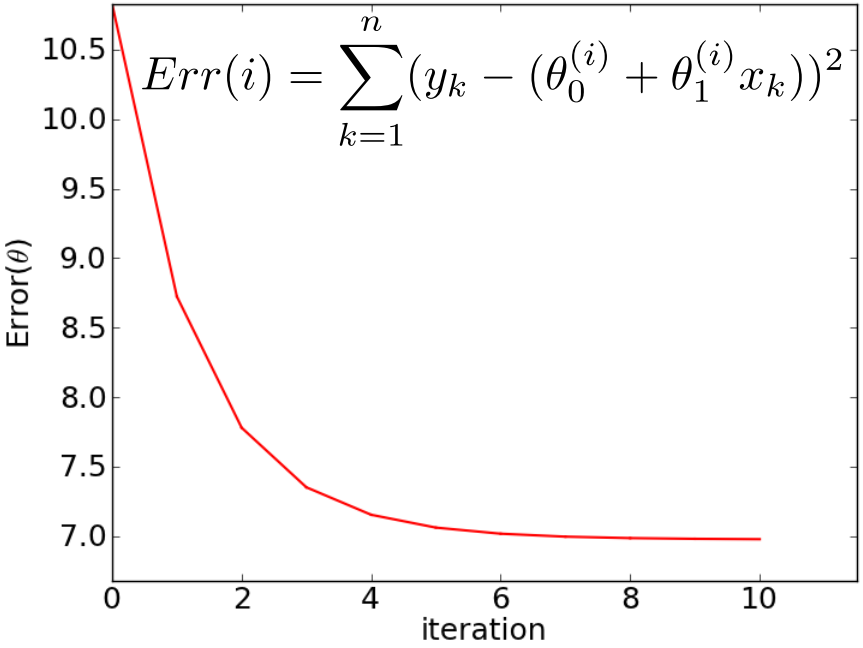


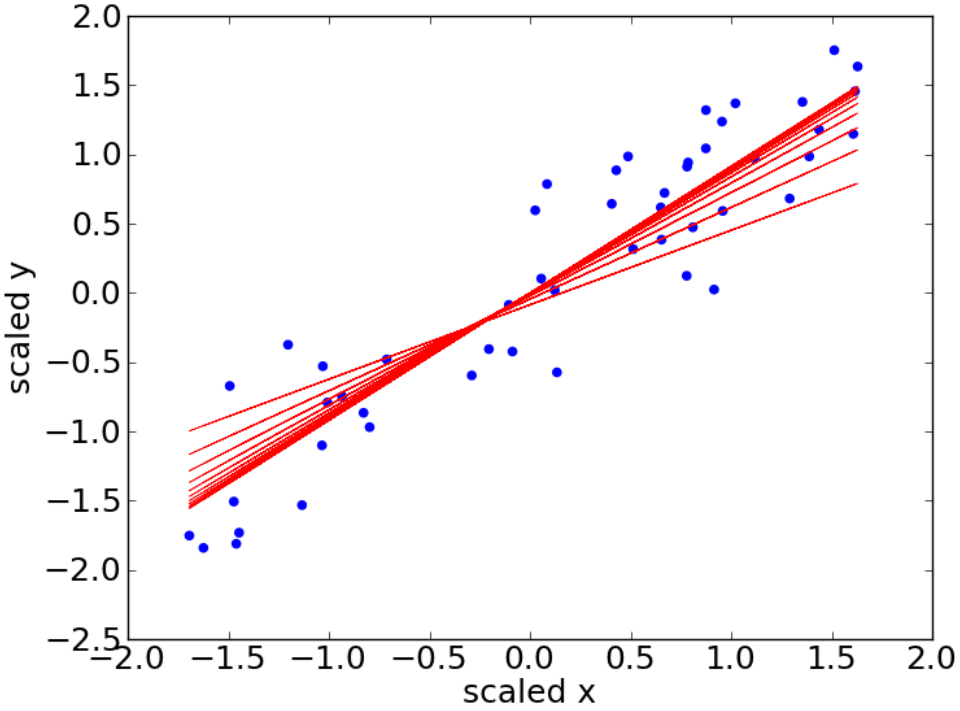




$$y = \theta_0^{(i)} + \theta_1^{(i)} x$$







$$y = \theta_0^{(i)} + \theta_1^{(i)} x$$