CS 383: Machine Learning

Prof Adam Poliak

Fall 2024

09/24/2024

Lecture 08

Announcements

HW02 is due tonight night

HW03 is due next Tuesday night

- **Reading quiz: Thursday**
  - Duame 9.3 (2 pages)

Proposed updated schedule

Midterm 1 was Thursday October 3rd

3 lectures this week

lecture on Wednesday 10/02 but no lecture on Thursday 10/03 (was supposed to be midterm 1)

No lecture Wednesday 10/09

HW02 decision trees due tonight, HW03 polynomial regression due next Tuesday 10/01, HW04 naive Bayes due Tuesday 10/08 (it’ll be a shorter assignment)

Midterm 1 on Thursday 10/10

Outline

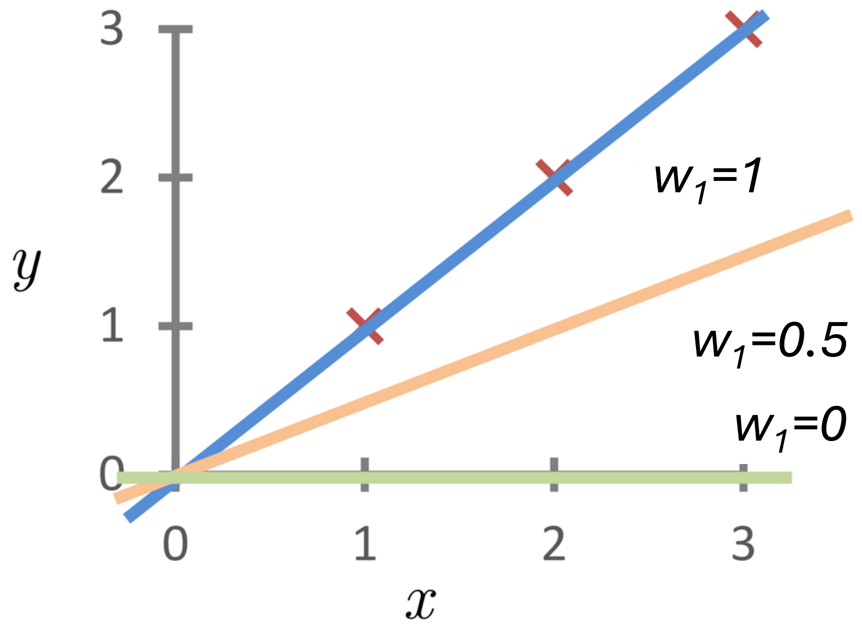
Normal equations solution

Regularization

Cost Function

$h_w(x)$

(assume $w_0=0$ for this example)

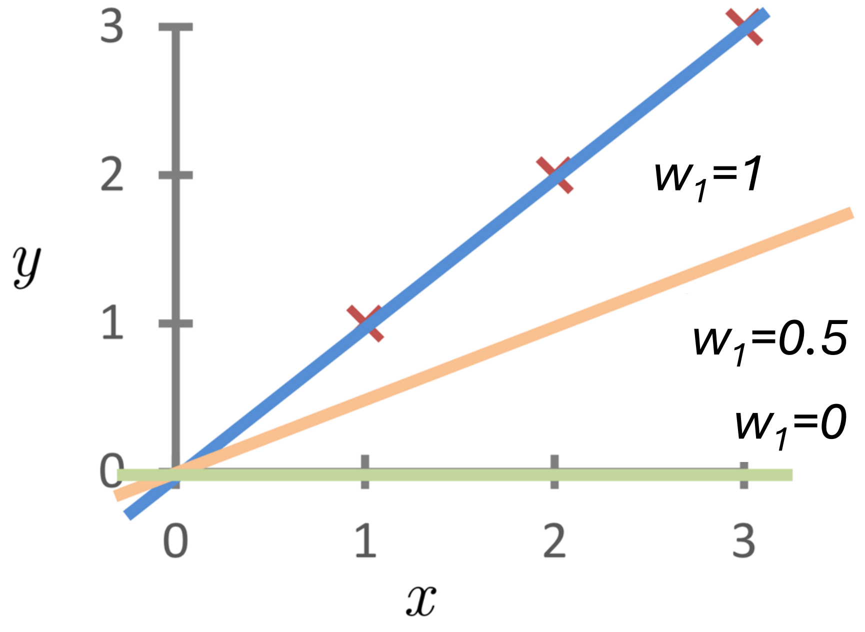


Based on slide by Eric Eaton

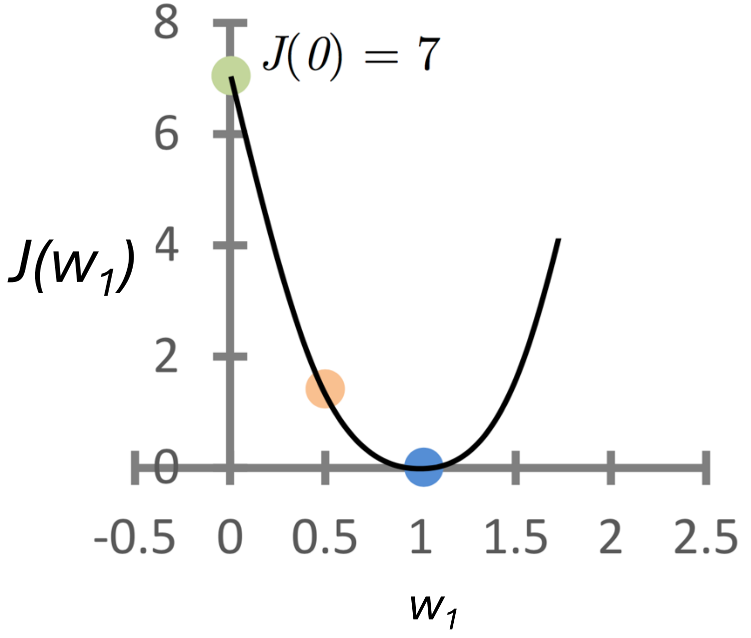
Cost Function

$h_w(x)$

(assume $w_0=0$ for this example)



$J(w_1)$



$$J(0.5) = \frac{1}{2}\left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right] = 1.75$$

Slide: modified from Jessica Wu
Based on slide by Eric Eaton
[Example by Andrew Ng]

Outline

Normal equations solution

Regularization

Assumptions of linear regression

Explanatory (independent) and response (dependent) variables have a linear relationship

Instances are independent of each other

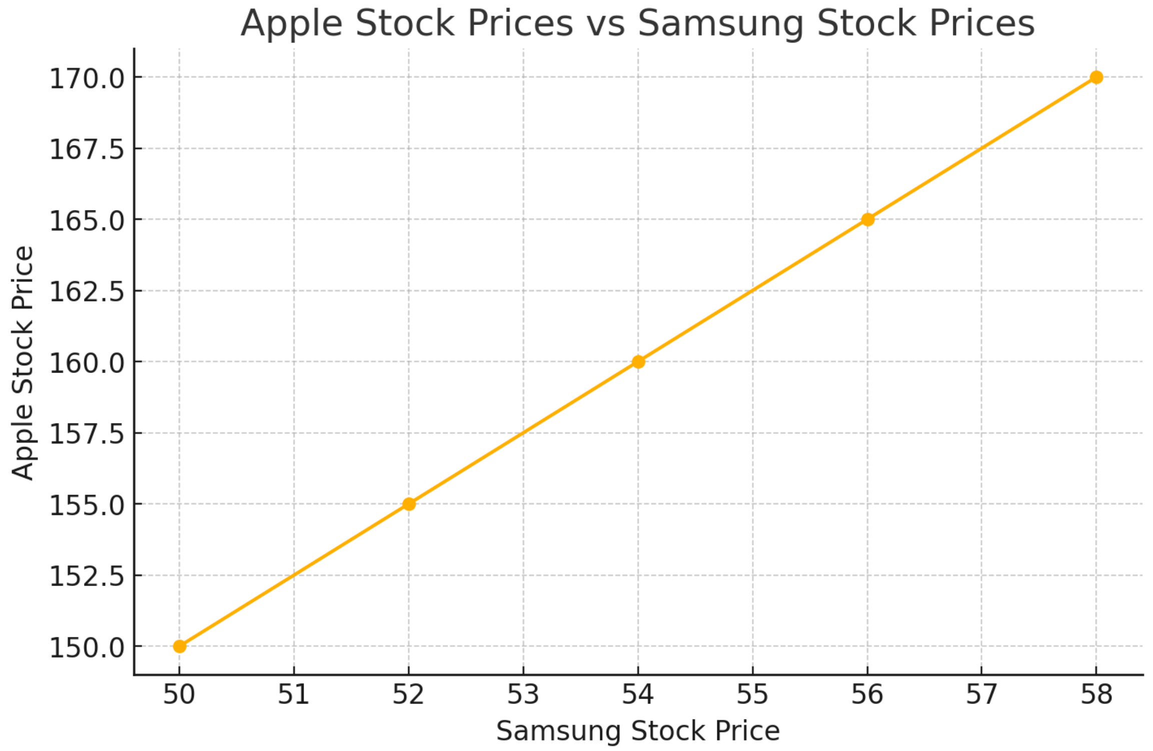
Residuals have a normal distribution with mean 0

The variance of the residuals is the same for an X (independent variable) - Homoscedacity

Linear Regression example with ChatGPT

https://chatgpt.com/share/66f2e089-95a0-8011-a567-7e75bd93261c

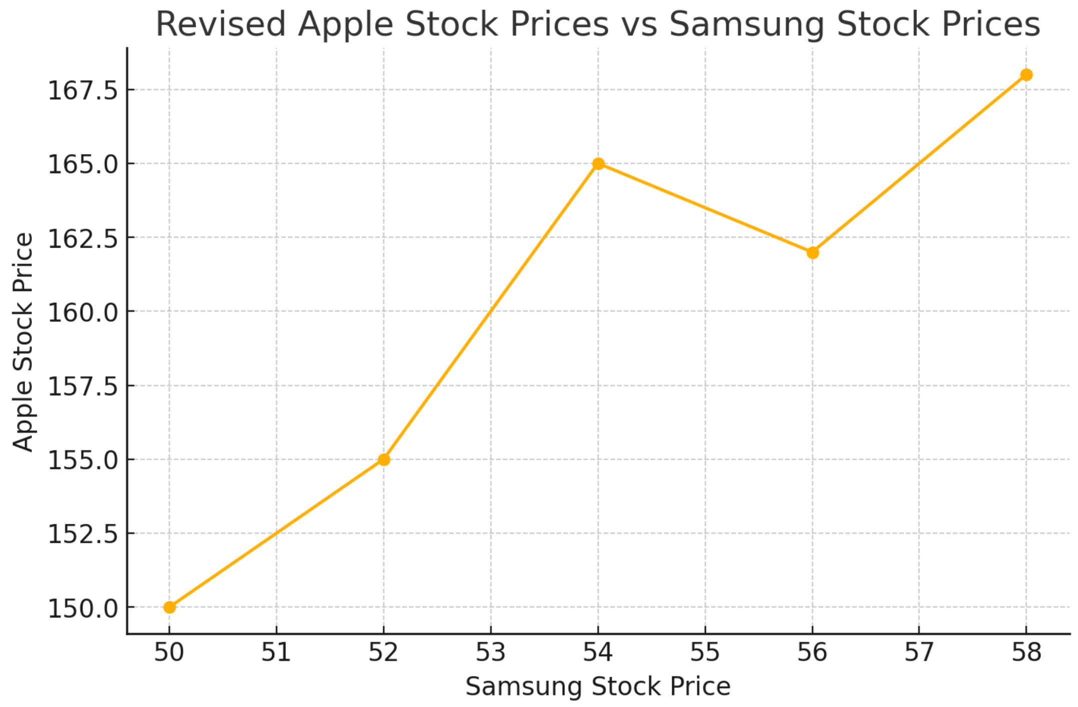
5 examples predicting Apple stock from Samsung stock



Linear Regression example with ChatGPT

https://chatgpt.com/share/66f2e089-95a0-8011-a567-7e75bd93261c

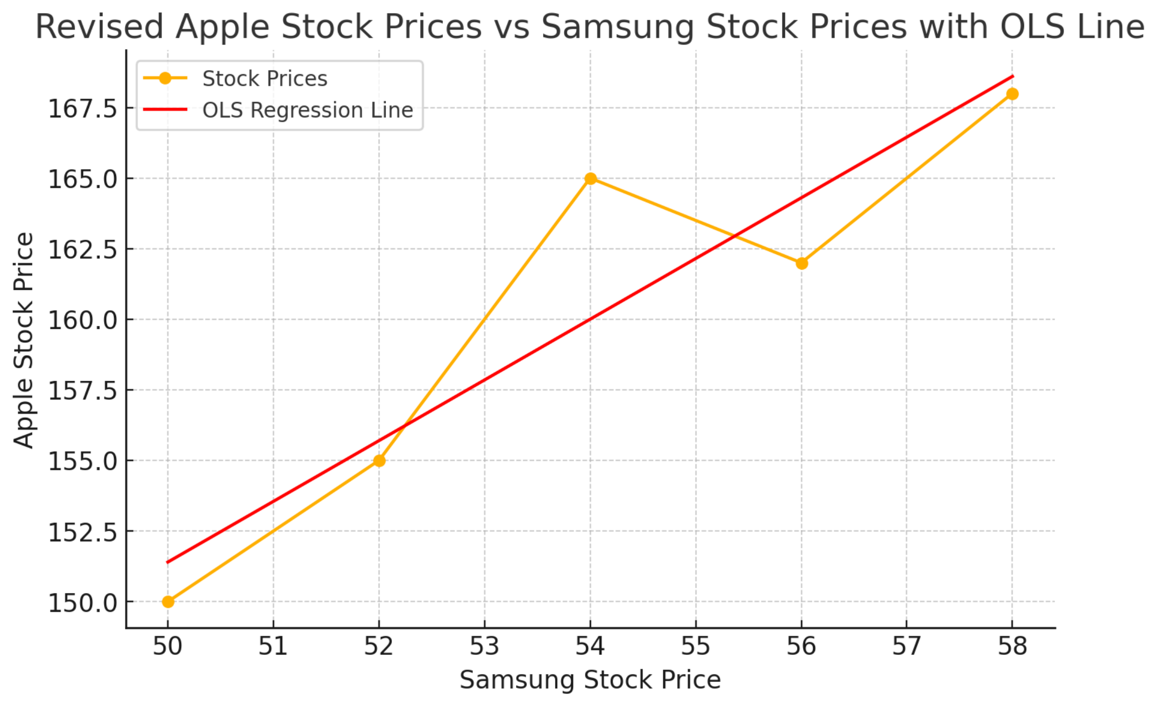
5 revised examples predicting Apple stock from Samsung stock



Linear Regression example with ChatGPT

https://chatgpt.com/share/66f2e089-95a0-8011-a567-7e75bd93261c

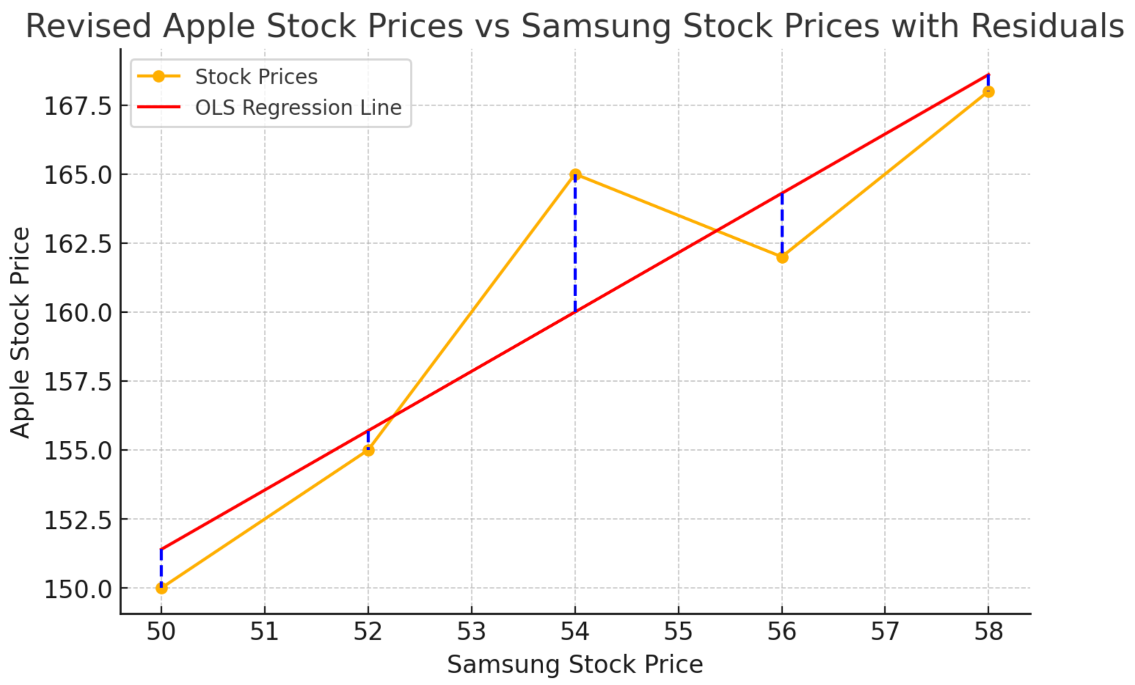
5 revised examples predicting Apple stock from Samsung stock



Linear Regression example with ChatGPT

https://chatgpt.com/share/66f2e089-95a0-8011-a567-7e75bd93261c

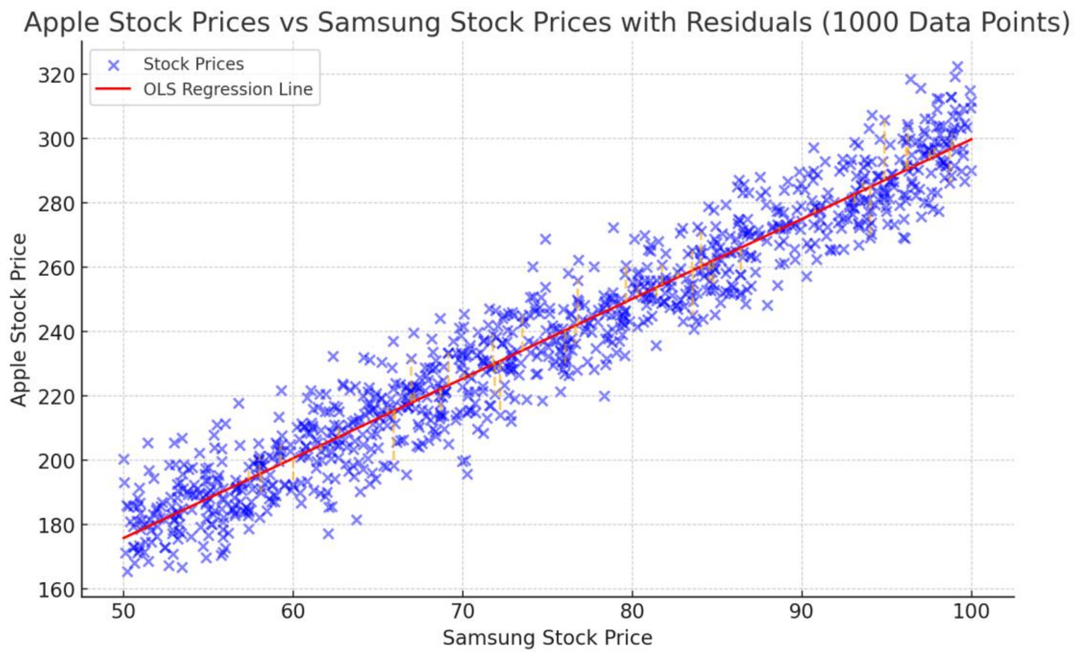
5 revised examples predicting Apple stock from Samsung stock



Linear Regression example with ChatGPT

https://chatgpt.com/share/66f2e089-95a0-8011-a567-7e75bd93261c

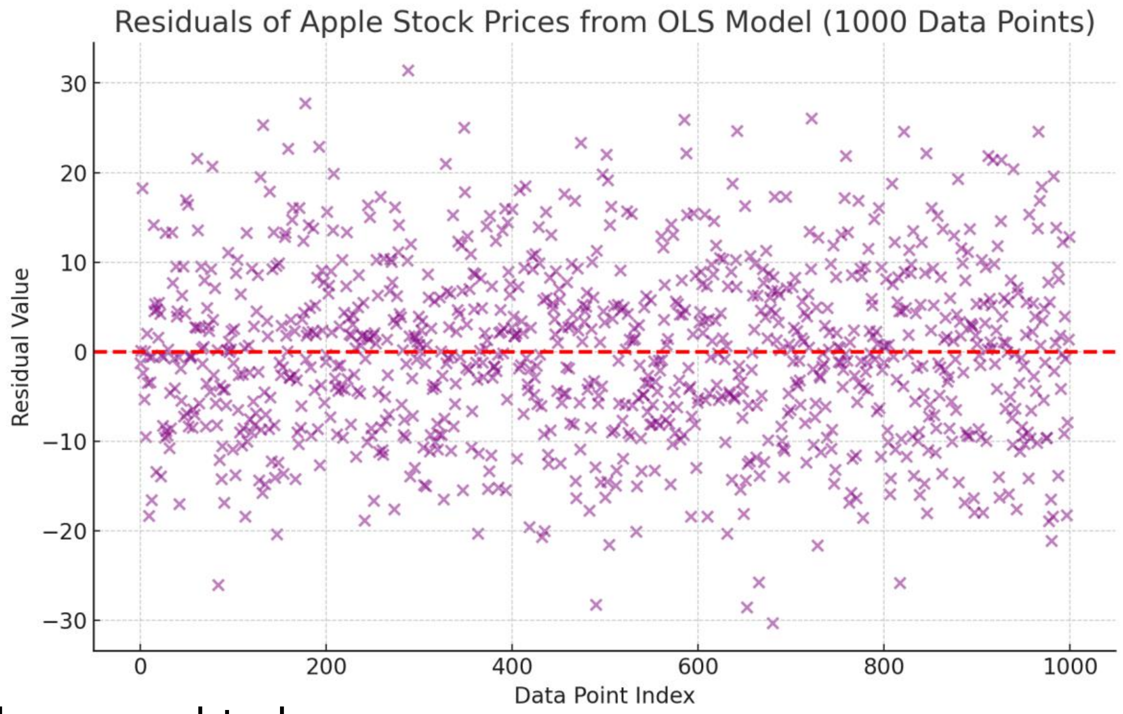
1000 examples predicting Apple stock from Samsung stock



Linear Regression example with ChatGPT

https://chatgpt.com/share/66f2e089-95a0-8011-a567-7e75bd93261c

Residuals and feature are uncorrelated

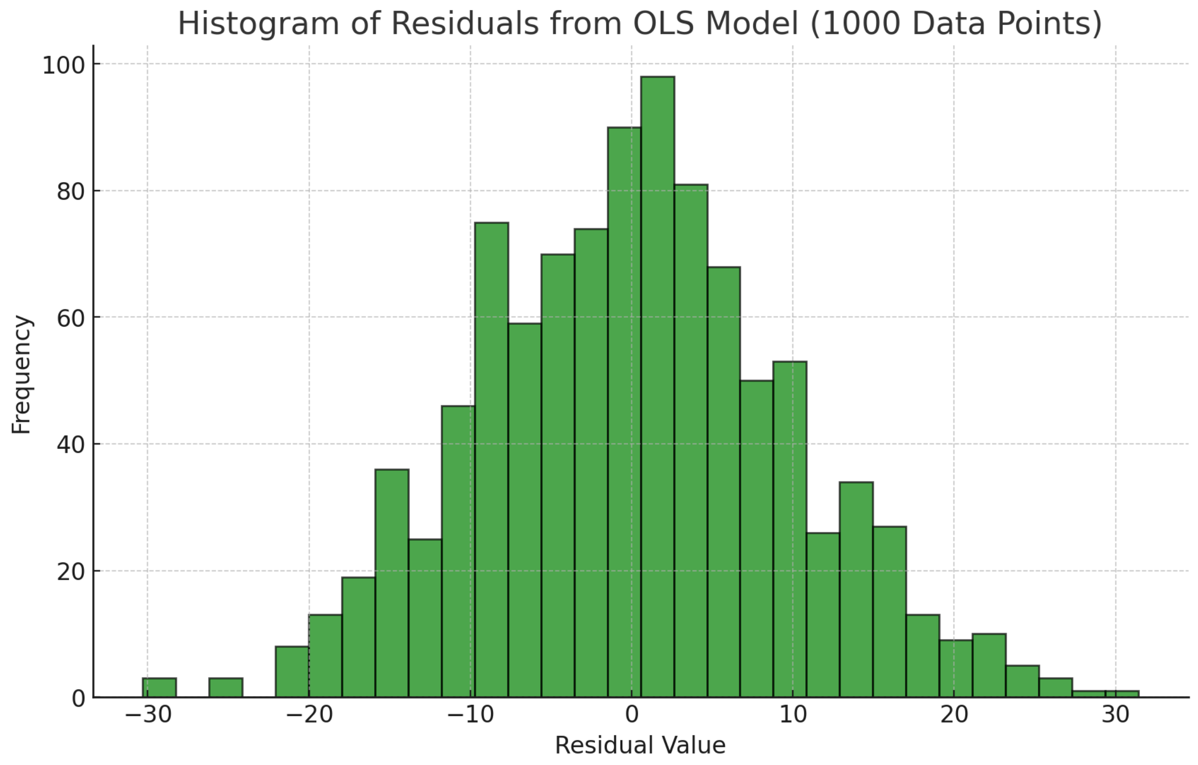


Proof: https://statproofbook.github.io/P/slr-rescorr.html

Linear Regression example with ChatGPT

https://chatgpt.com/share/66f2e089-95a0-8011-a567-7e75bd93261c

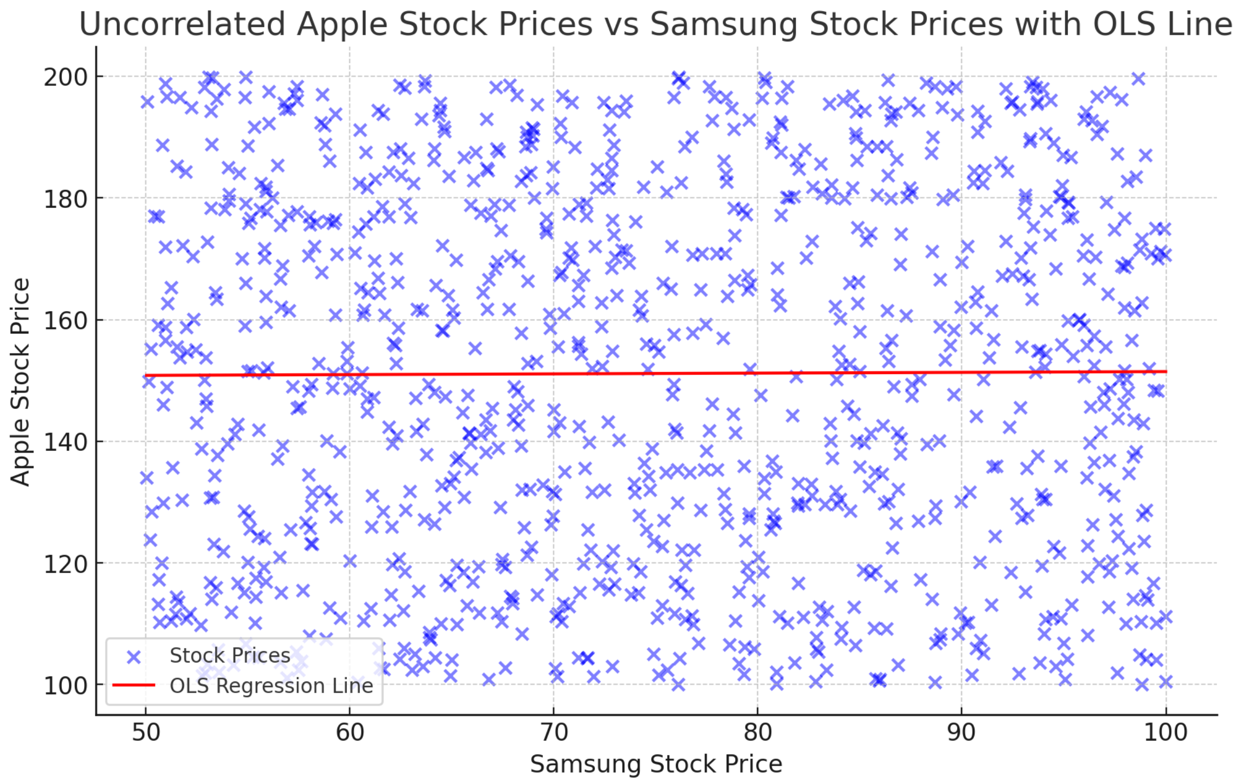
Distribution of residuals is normal, with mean 0



Linear Regression example with ChatGPT

https://chatgpt.com/share/66f2e089-95a0-8011-a567-7e75bd93261c

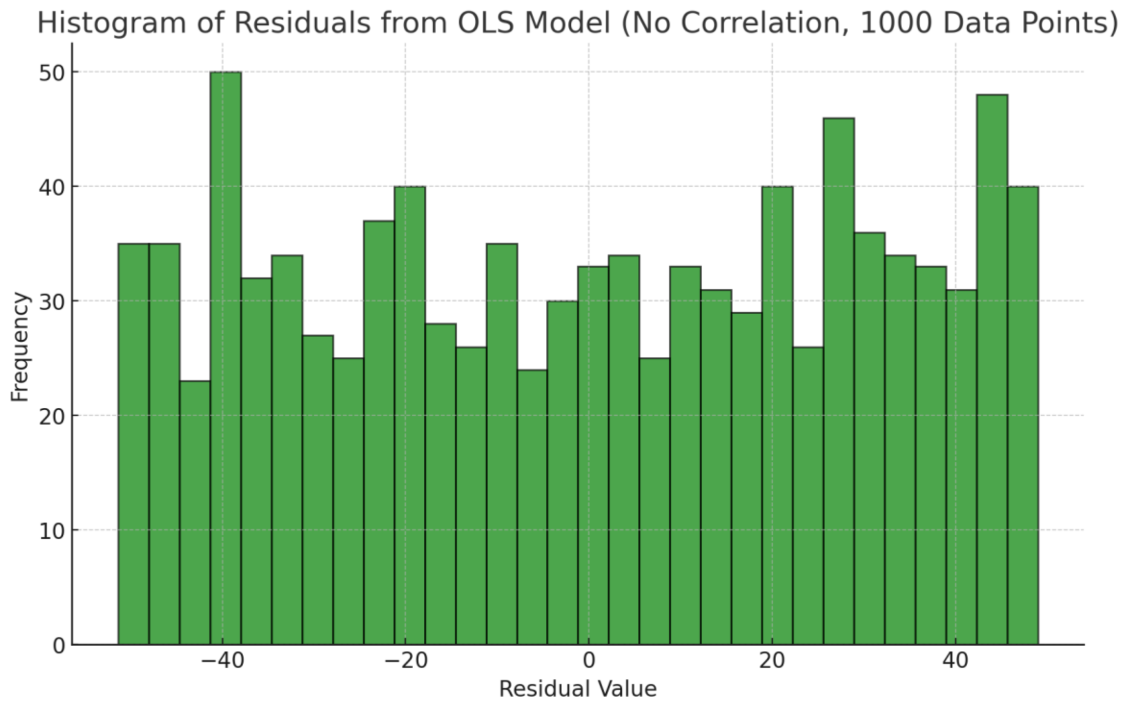
Predicting (red line) Apple price from Samsung price where variables are uncorrelated



Linear Regression example with ChatGPT

https://chatgpt.com/share/66f2e089-95a0-8011-a567-7e75bd93261c

Distribution of residuals is not normal when response and predictor variable are uncorrelated



Normal Equation

$$J(\vec{w}) = \frac{1}{2}\sum_{i}^{n}\sum_{k}^{p}(w_0 + w_1 x_{i,1} + w_2 x_{i,2} + \ \ldots + w_p x_{i,p} - y_i)^2$$

$$J(\vec{w}) = \frac{1}{2}\ (\vec{w}X \ - y)^2$$

Normal Equation

$$J(\vec{w}) = \frac{1}{2}\sum_{i}^{n}(w_0 + w_1 x_i - y_i)^2$$

Now we have more than 1 feature:

$$J(\vec{w}) = \frac{1}{2}\sum_{i}^{n}\sum_{k}^{p}(w_0 + w_1 x_{i,1} + w_2 x_{i,2} + \ \ldots + w_p x_{i,p} - y_i)^2$$

How many partial derivatives do we need to compute?

Linear Algebra Review

- Transpose

- A^T swap all columns and rows

$$\begin{aligned}(Ax)^T &= \\ &= x^T A^T\end{aligned}$$

Pros and Cons

Gradient Descent

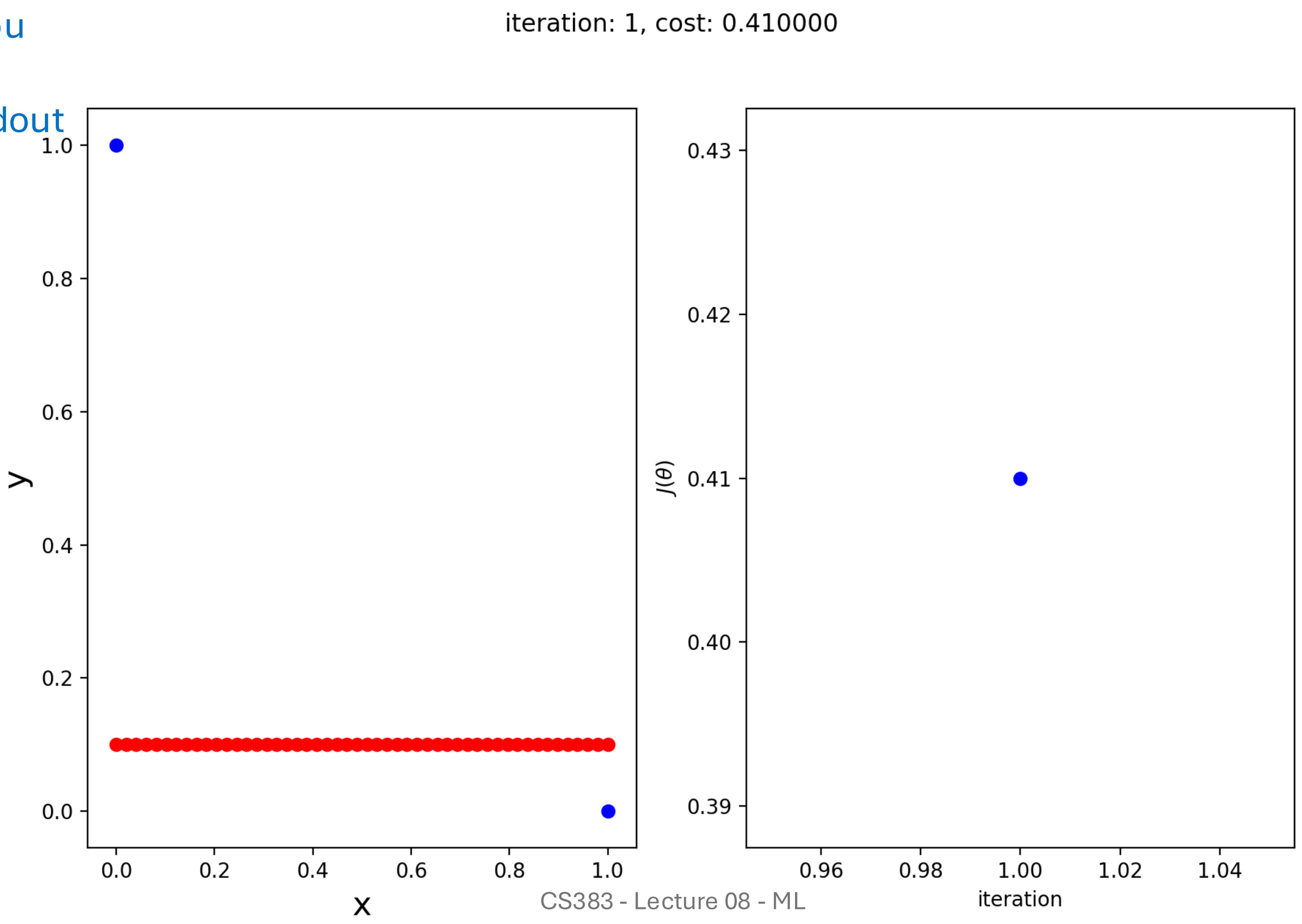
- Requires multiple iterations
- Need to choose η
- Works well when *n* is large
- Can support online learning

Normal Equation

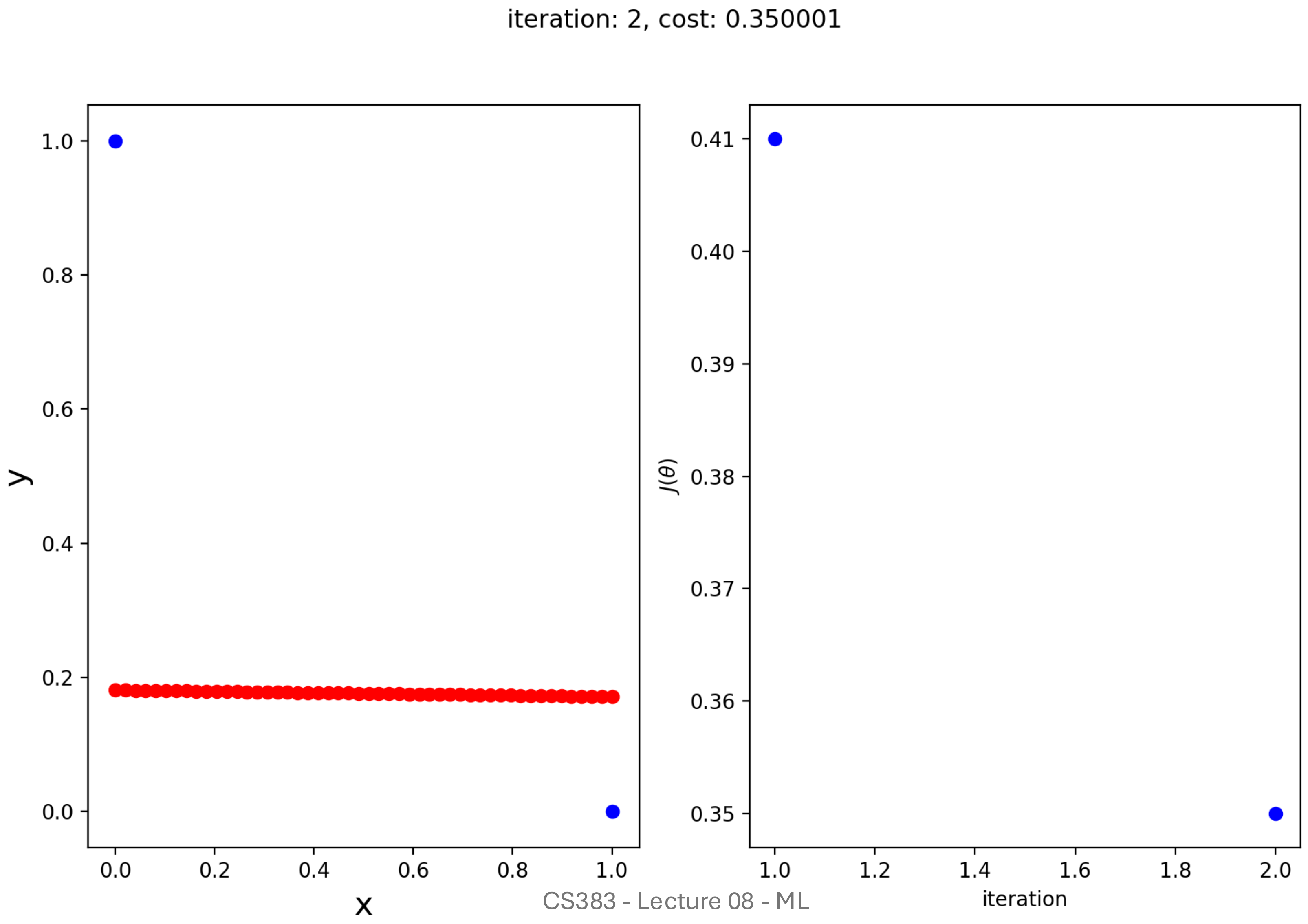
- Non-iterative
- No need to choose η
- Slow if *p* is large
  - Matrix inversion is $O(p^3)$

Toy example, iteration 1

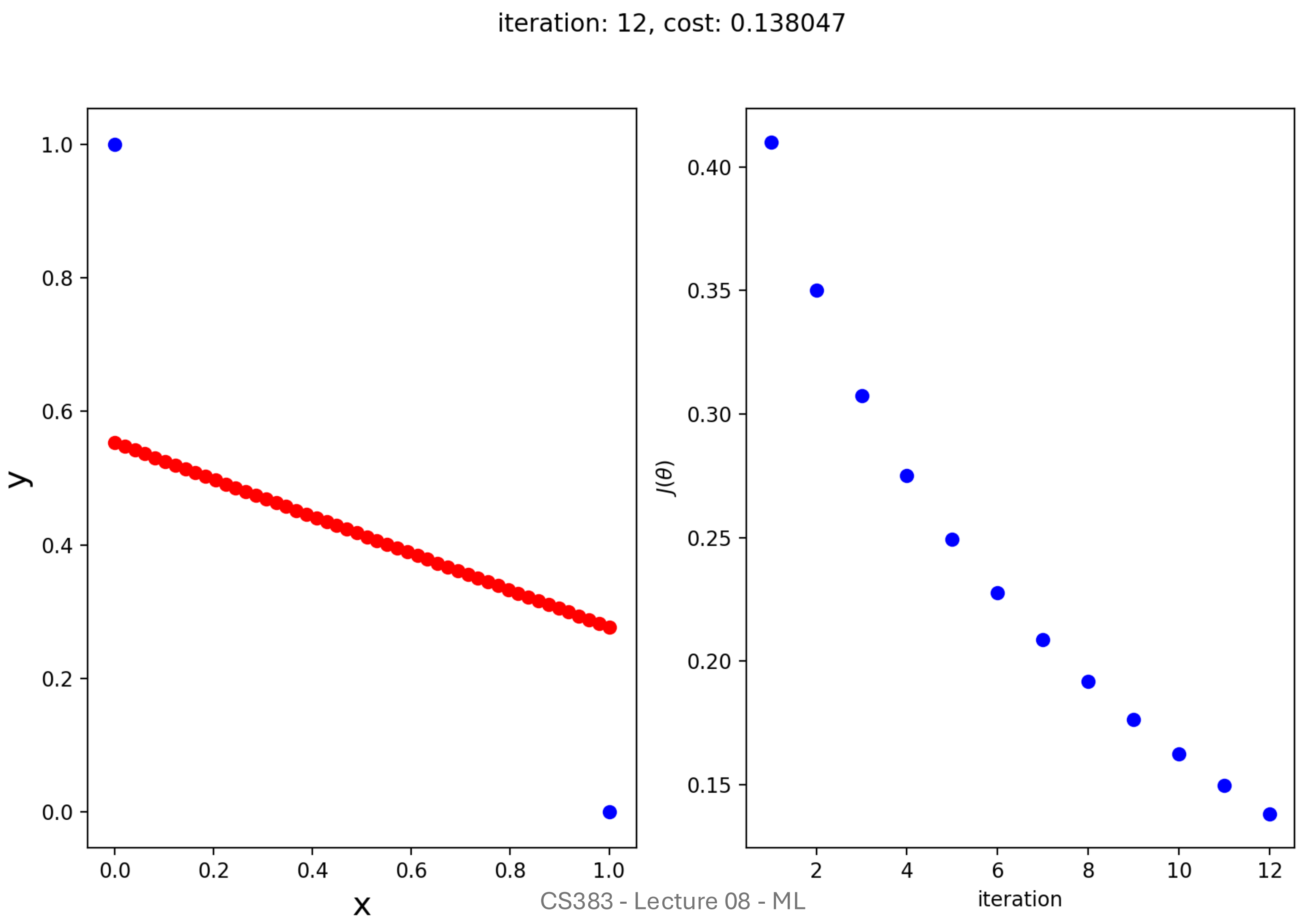
This is what you should have obtained in Handout 7!



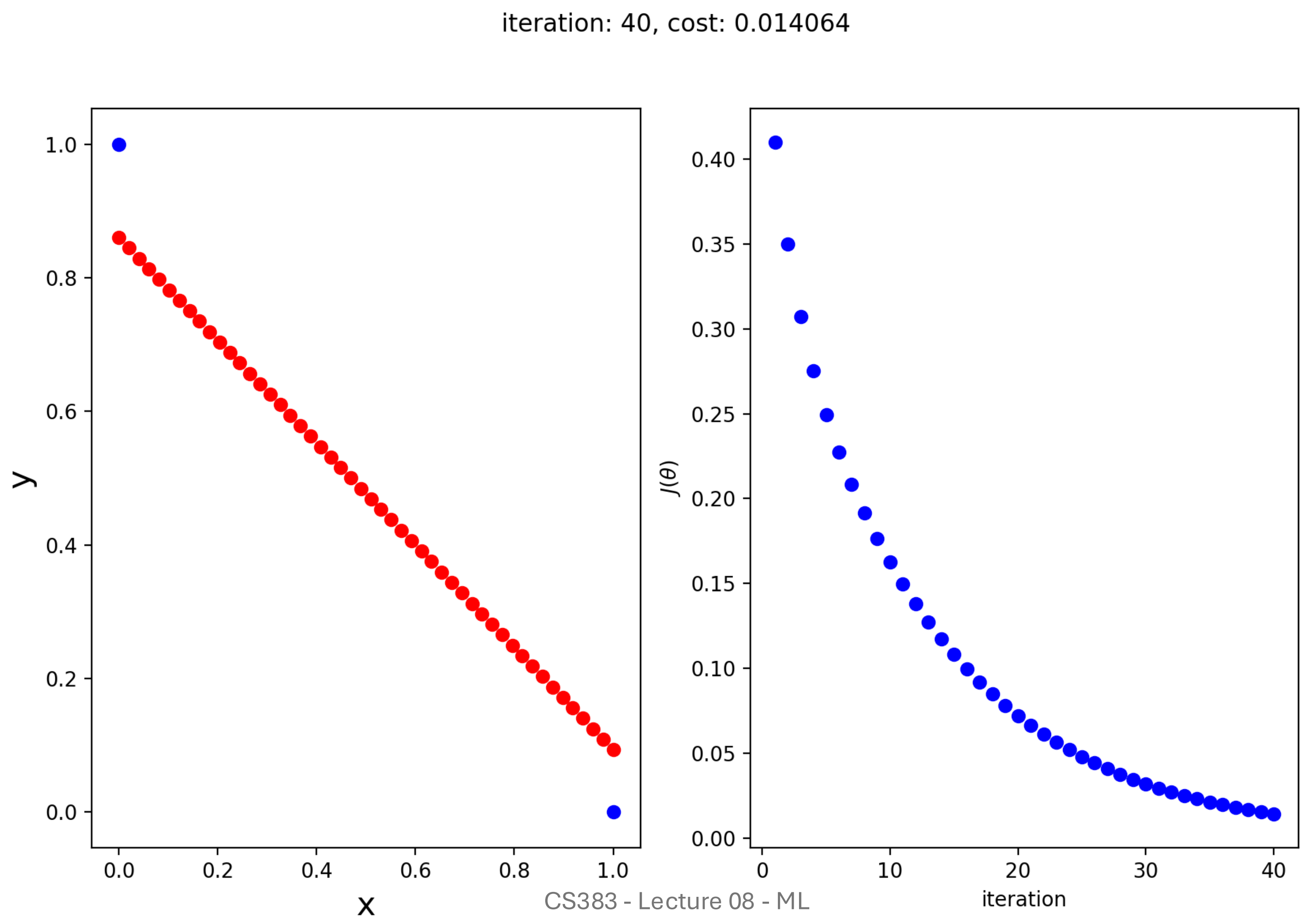
Toy example, iteration 2



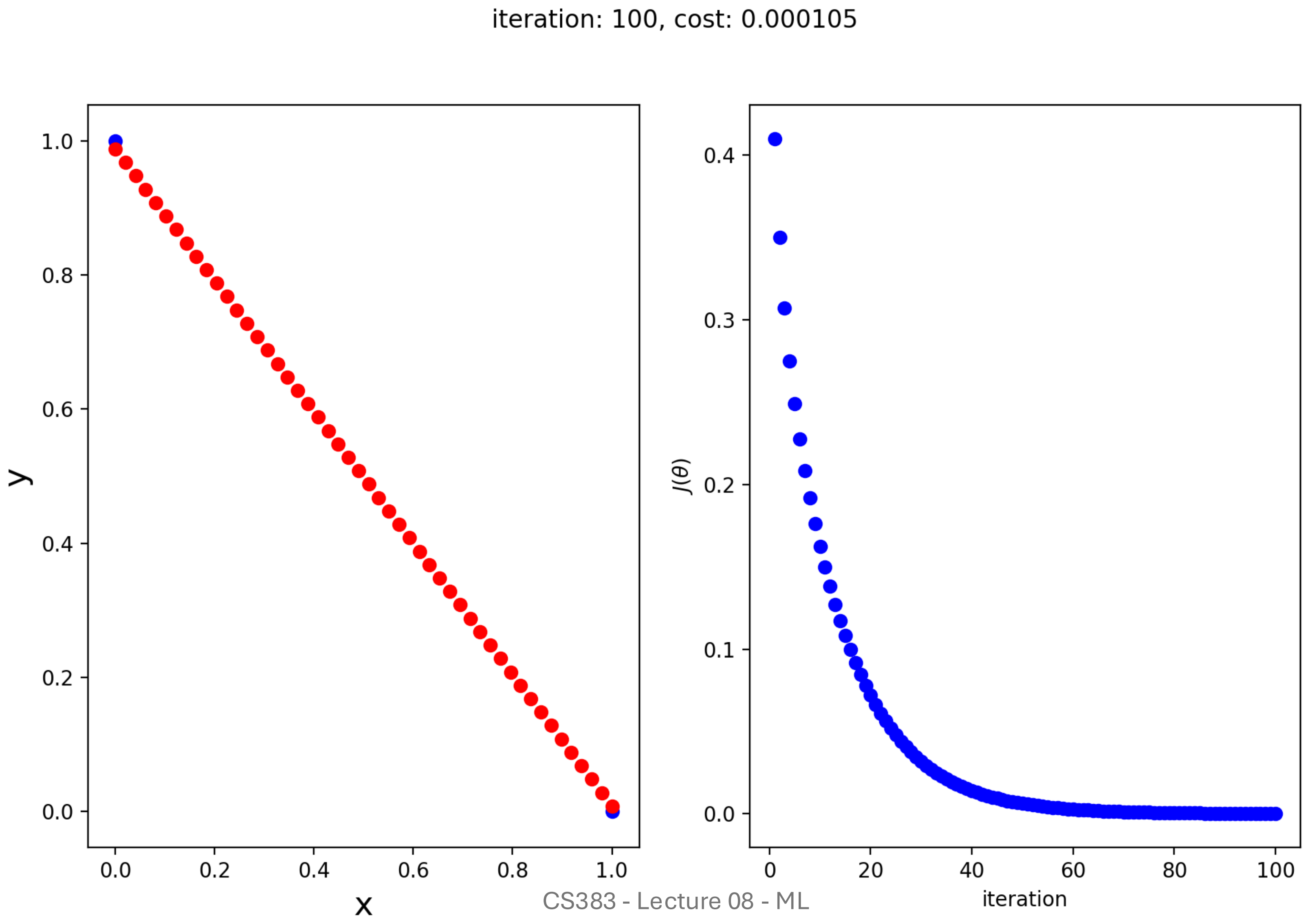
Toy example, iteration 12



Toy example, iteration 40



Toy example, iteration 100



Outline

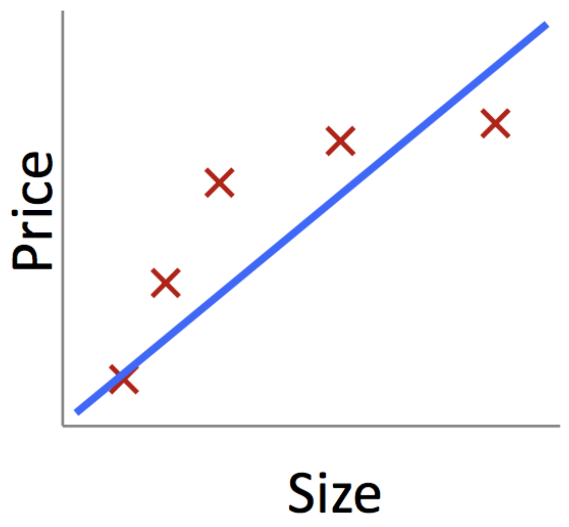
Normal equations solution

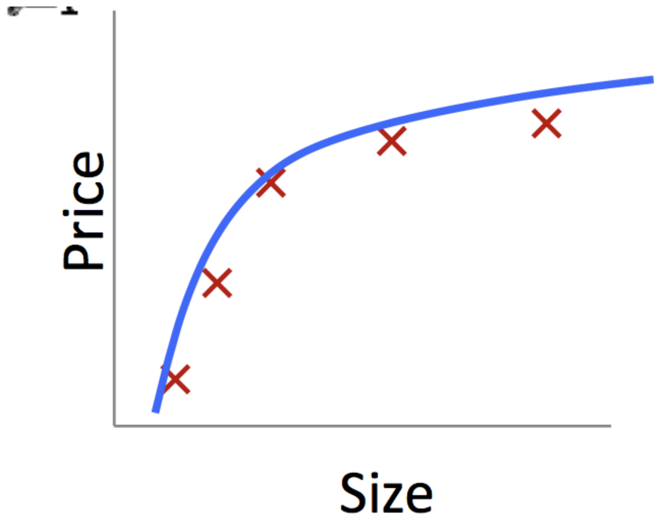
Regularization

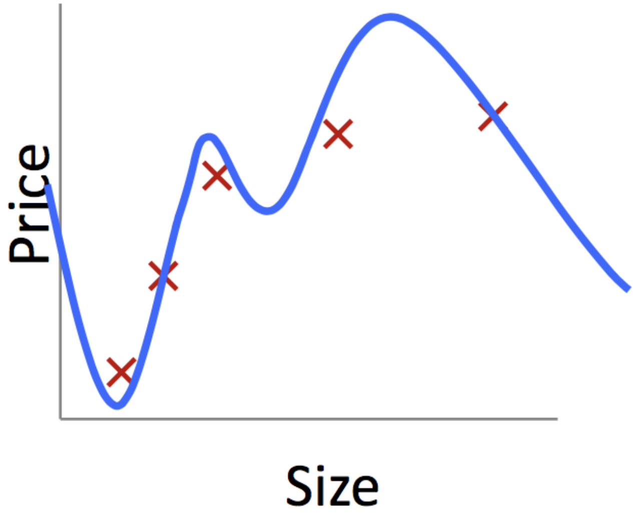
Adapted from slide by Jessica Wu
[example by Andrew Ng]

Generalization Error

Example: price vs. size (i.e. of a house or car)



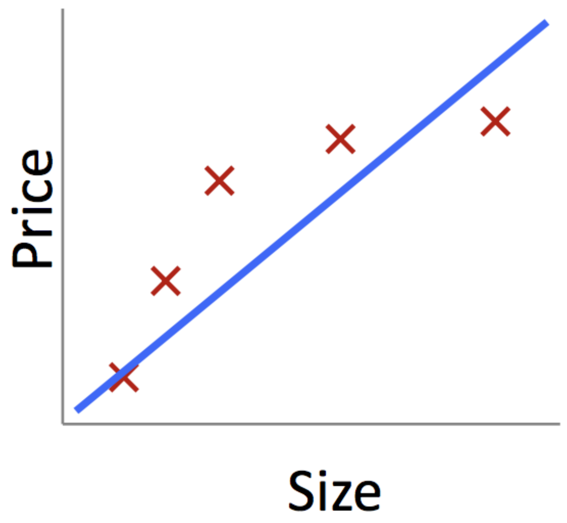




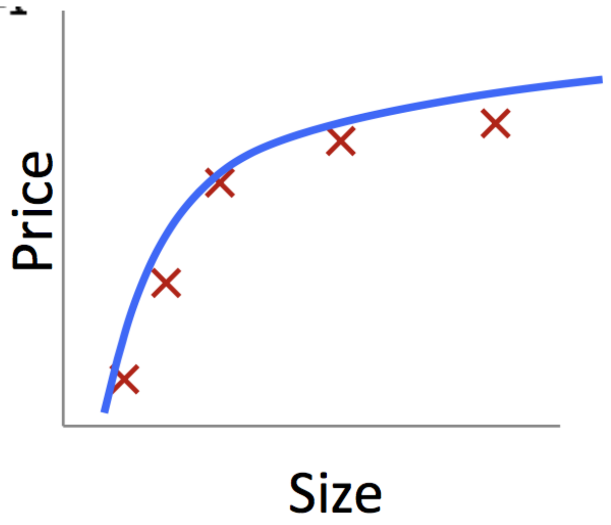
Adapted from slide by Jessica Wu
[example by Andrew Ng]

Generalization Error

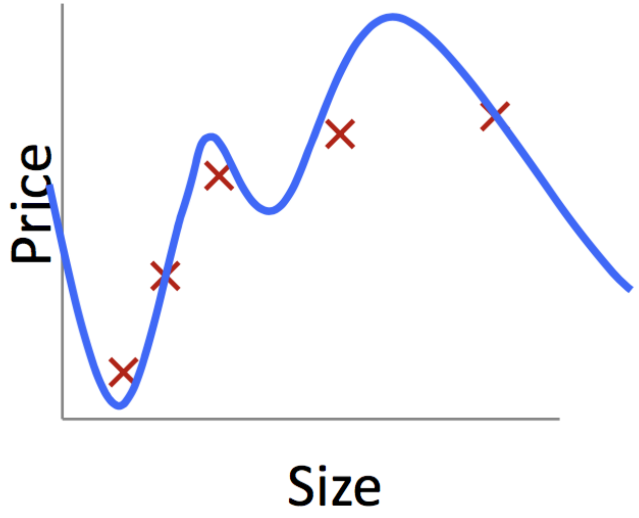
Example: price vs. size (i.e. of a house or car)



underfitting
(high bias)



correct fit



overfitting
(high variance)

Adapted from slide by Jessica Wu
[example by Andrew Ng]

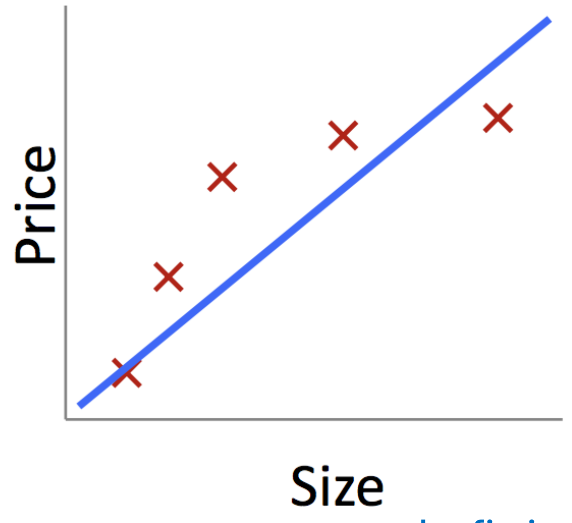
Generalization Error

<u>Structural error:</u>
Hypothesis space cannot model true relationship

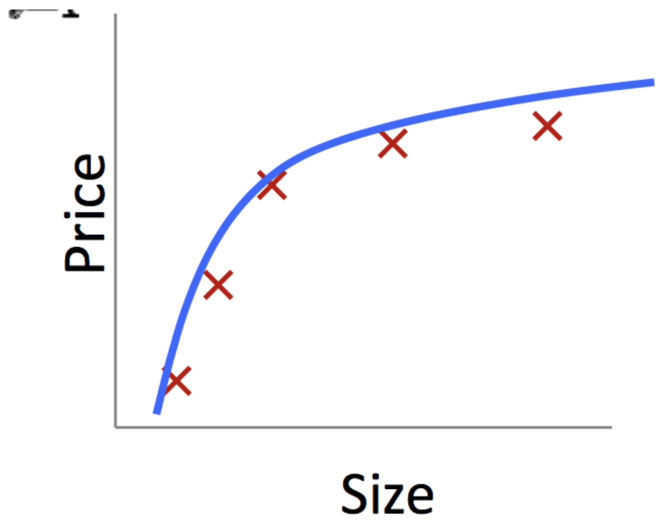
-More data doesn’t help
-Need a more flexible model

balance
⇔

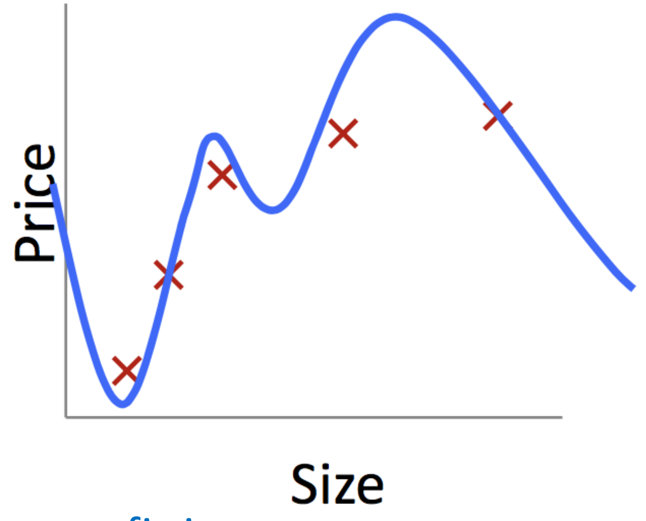
<u>Estimation (approximation) error:</u>
Hypothesis space *can* model true relationship, BUT hard to identify correct model due to large hypothesis space, small *n*, or noise
⍰Reduce hypothesis space
⍰Add more data



underfitting
(high bias)



correct fit



overfitting
(high variance)

Regularization

What if …

- we have a limited # of training examples ($n<p$), or
- we want to automatically control the complexity of the learned hypothesis?

Regularization

What if …

- we have a limited # of training examples ($n<p$), or
- we want to automatically control the complexity of the learned hypothesis?

Idea: penalize large values of w_j

Why prefer small weights?

Regularization

What if …

- we have a limited # of training examples ($n<p$), or
- we want to automatically control the complexity of the learned hypothesis?

Idea: penalize large values of w_j

Why prefer small weights?

- if large weights, small change in feature can result in large change in prediction
- prevent giving too much weight to any one feature
- might prefer zero weight for useless features

Common Regularizers

$$||\vec{w}||_0 = \sum_{j:w_j \neq 0} 1$$

L_0 norm

- Number of non-zero entries
- Minimizing L_0 norm is NP hard

$$||\vec{w}||_1 = \sum_{j=1}^{p} |w_j|$$

L_1 norm

- Sum of magnitude of weights
- Not differentiable

$$||\vec{w}||_2 = \sqrt{\sum_{j=1}^{p} w_j^2}$$

L_2 norm

- Sum of squared weights
- Differentiable

Slide adapted from Jessica Wu