Linear regression

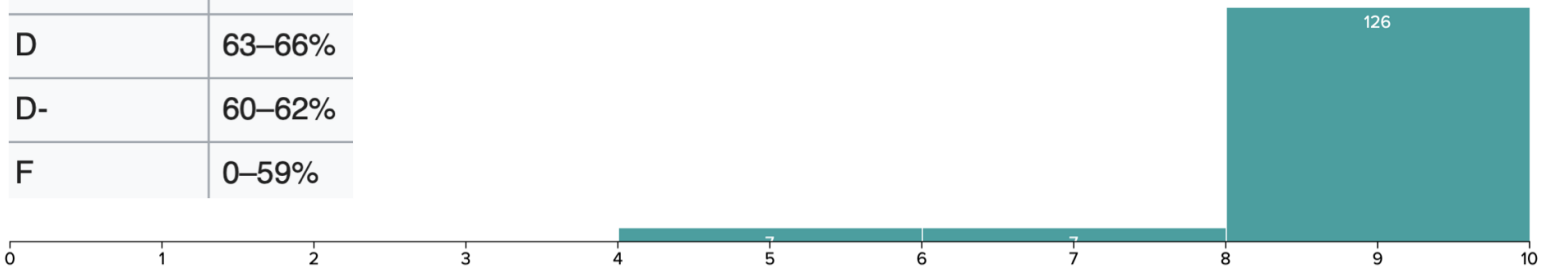
Instructor: Taylor Berg-Kirkpatrick
Slides: Sanjoy Dasgupta

Quiz 0 Grades

A	93– %
A–	90–92%
B+	87–89%
B	83–86%
B–	80–82%
C+	77–79%
C	73–76%
C-	70–72%
D+	67–69%
D	63–66%
D-	60–62%
F	0–59%

Overall letter grade cutoffs

Quiz 0 Grade Distribution



MINIMUM	MEDIAN	MAXIMUM	MEAN	STD DEV
4.5	10.0	10.0	9.4	1.25

Quiz Schedule

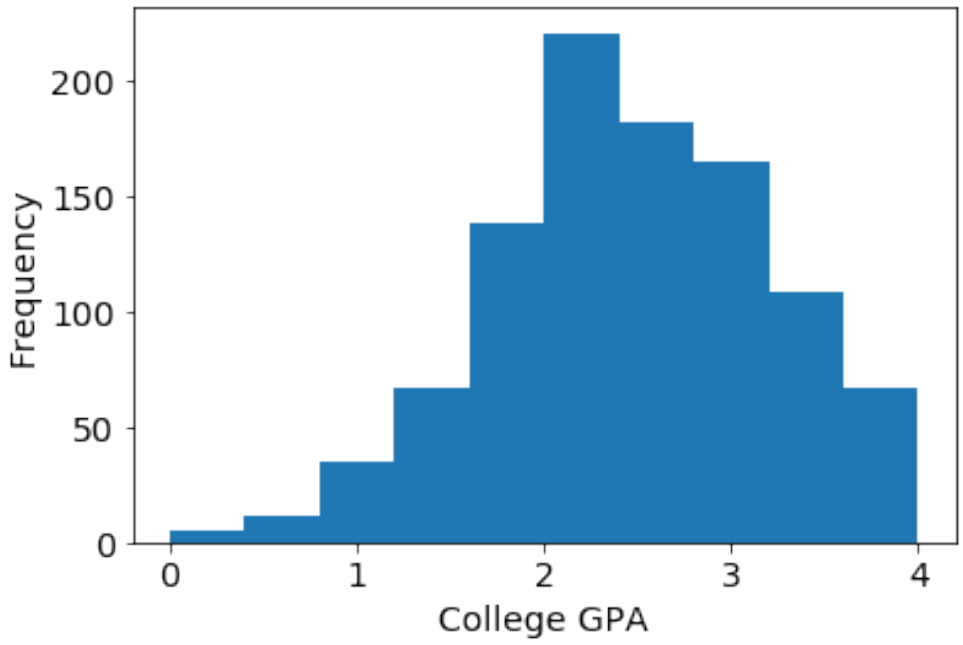
- Quiz 1: Oct 28
- Quiz 2: Nov 12
- Quiz 3: Nov 21
- Quiz 4: Dec 5 (last day of class)
- Future HWs: TBA
- Check updated schedule on course website!

Linear regression

Fitting a line to a bunch of points.

Example: college GPAs

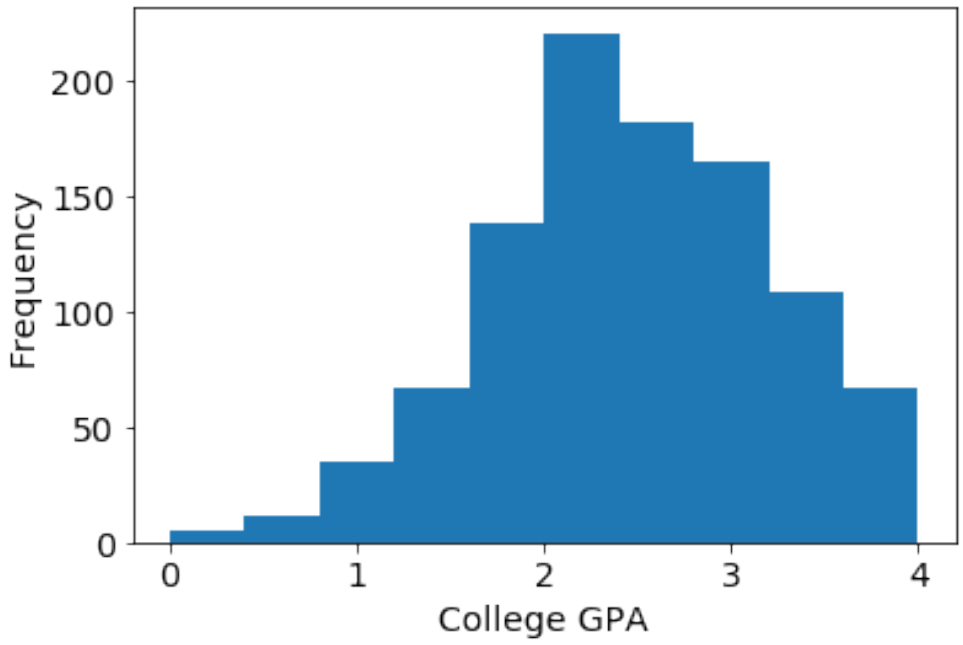
Distribution of GPAs of students at a certain Ivy League university.



What GPA to predict for a random student from this group?

Example: college GPAs

Distribution of GPAs of students at a certain Ivy League university.

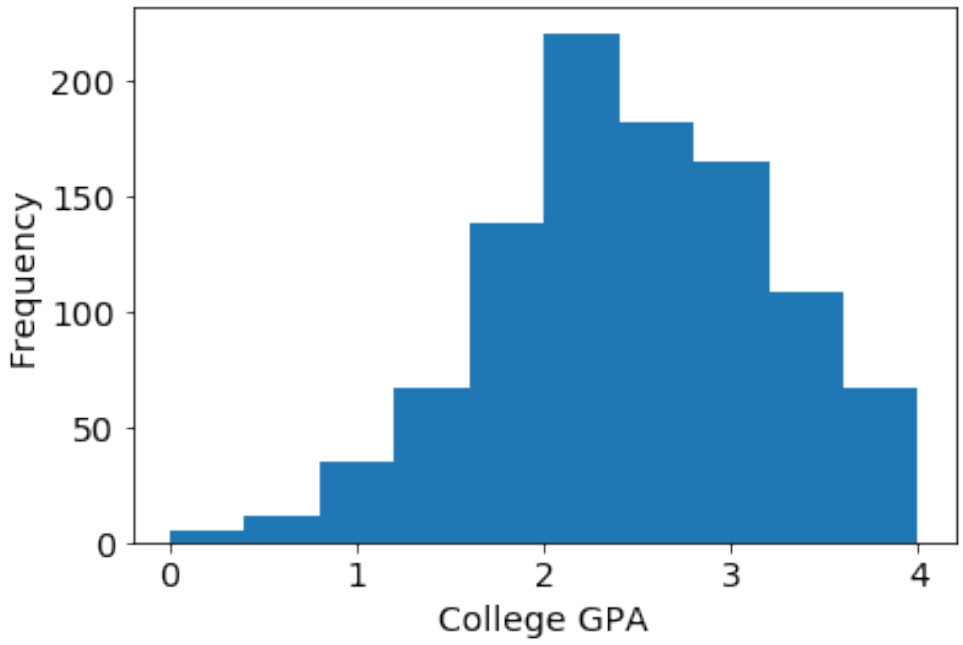


What GPA to predict for a random student from this group?

- Without further information, predict the **mean**, 2.47.
- What is the average squared error of this prediction?

Example: college GPAs

Distribution of GPAs of students at a certain Ivy League university.

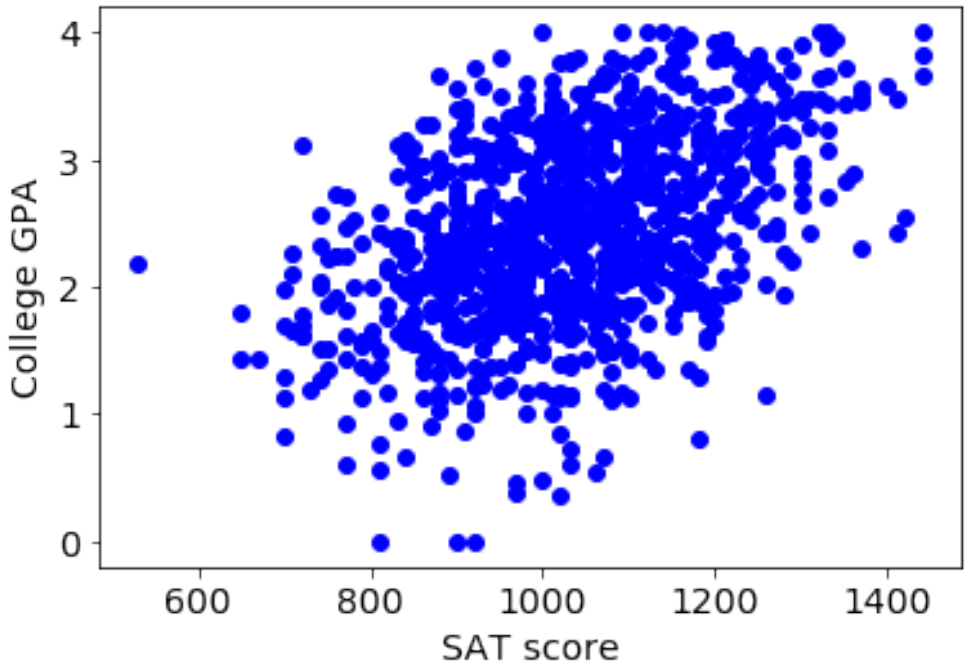


What GPA to predict for a random student from this group?

- Without further information, predict the **mean**, 2.47.
- What is the average squared error of this prediction?
  The **variance** of the distribution, 0.55.

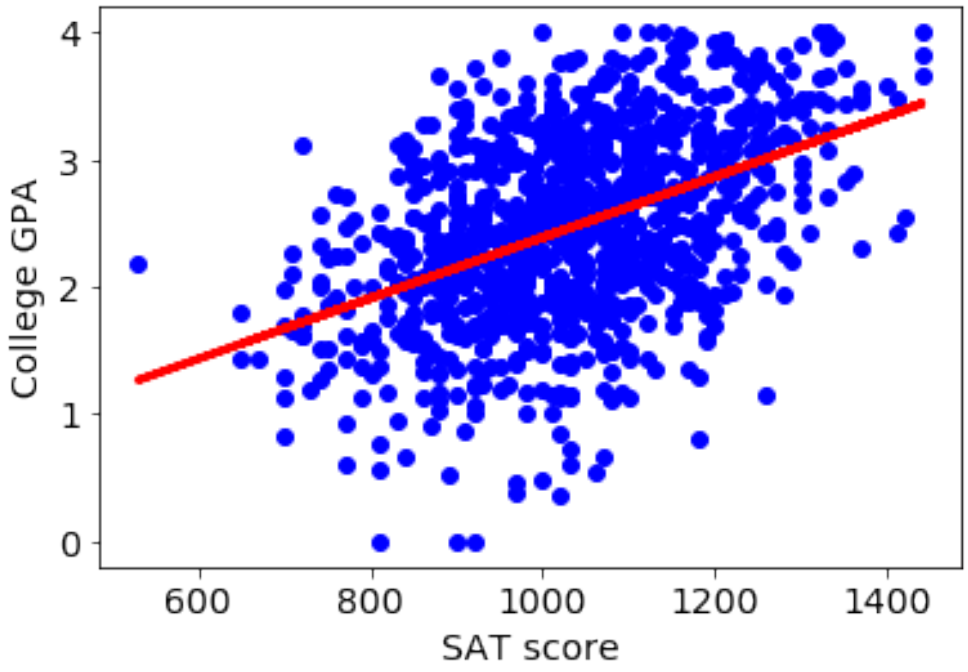
Better predictions with more information

We also have SAT scores of all students.



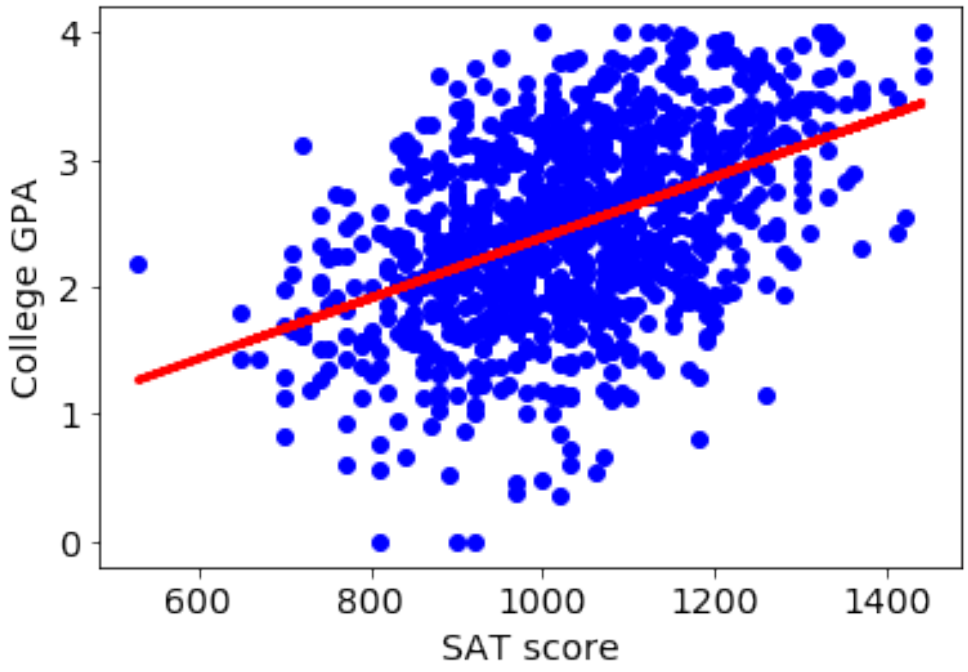
Better predictions with more information

We also have SAT scores of all students.



Better predictions with more information

We also have SAT scores of all students.



Mean squared error (MSE) drops to 0.43.

This is a **regression** problem with:

- **Predictor variable**: SAT score
- **Response variable**: College GPA

The line fitting problem

A line can be parameterized as $y = ax + b$ (a: slope, b: intercept).

Pick a, b that are suited to the data, $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$:

- x_i, y_i are the SAT score and GPA of the ith student.
- Minimize the mean squared error,

$$\text{MSE}(a, b) = \frac{1}{n} \sum_{i=1}^{n} (y^{(i)} - (ax^{(i)} + b))^2.$$

This is the **loss function**.

Minimizing the loss function

Given $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$, minimize

$$L(a, b) = \sum_{i=1}^{n} (y^{(i)} - (ax^{(i)} + b))^2.$$

Moving to higher dimension

Data from $n = 442$ **diabetes** patients.

For each patient:

- 10 features $x = (x_1, \ldots, x_{10})$

  age, sex, body mass index, average blood pressure, and six blood serum measurements.

- A real value y: the progression of the disease a year later.

Regression problem:

- **response** $y \in \mathbb{R}$
- **predictor variables** $x \in \mathbb{R}^{10}$

A linear predictor

Linear function of 10 variables:

$$f(x) = w_o + w_1 x_1 + w_2 x_2 + \cdots + w_{10} x_{10}.$$

A linear predictor

Linear function of 10 variables:

$$f(x) = w_o + w_1 x_1 + w_2 x_2 + \cdots + w_{10} x_{10}.$$

To simplify things, add a new feature $x_o = 1$. Now $x \in \mathbb{R}^{11}$ and

$$f(x) = w_o x_o + w_1 x_1 + \cdots + w_{10} x_{10}.$$

So: $f(x) = w \cdot x$ where $w \in \mathbb{R}^{11}$.

A linear predictor

Linear function of 10 variables:

$$f(x) = w_o + w_1 x_1 + w_2 x_2 + \cdots + w_{10} x_{10}.$$

To simplify things, add a new feature $x_o = 1$. Now $x \in \mathbb{R}^{11}$ and

$$f(x) = w_o x_o + w_1 x_1 + \cdots + w_{10} x_{10}.$$

So: $f(x) = w \cdot x$ where $w \in \mathbb{R}^{11}$.

Goal: find vector w so that this linear function fits the data well, i.e., y is well approximated by $w \cdot x$.

Squared loss

Want to find w so that $y \approx w \cdot x$. How much to penalize the error?

Squared loss

Want to find w so that $y \approx w \cdot x$. How much to penalize the error?

Most common choice: **squared loss** $(y - w \cdot x)^2$.

Least-squares regression:

- *Given:* data set $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$, where each $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}$.
- *Find:* the vector $w \in \mathbb{R}^d$ that minimizes the loss function

$$L(w) = \sum_{i=1}^{n} (y^{(i)} - w \cdot x^{(i)})^2.$$

Squared loss

Want to find w so that $y \approx w \cdot x$. How much to penalize the error?

Most common choice: **squared loss** $(y - w \cdot x)^2$.

Least-squares regression:

- *Given:* data set $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$, where each $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}$.
- *Find:* the vector $w \in \mathbb{R}^d$ that minimizes the loss function

$$L(w) = \sum_{i=1}^{n} (y^{(i)} - w \cdot x^{(i)})^2.$$

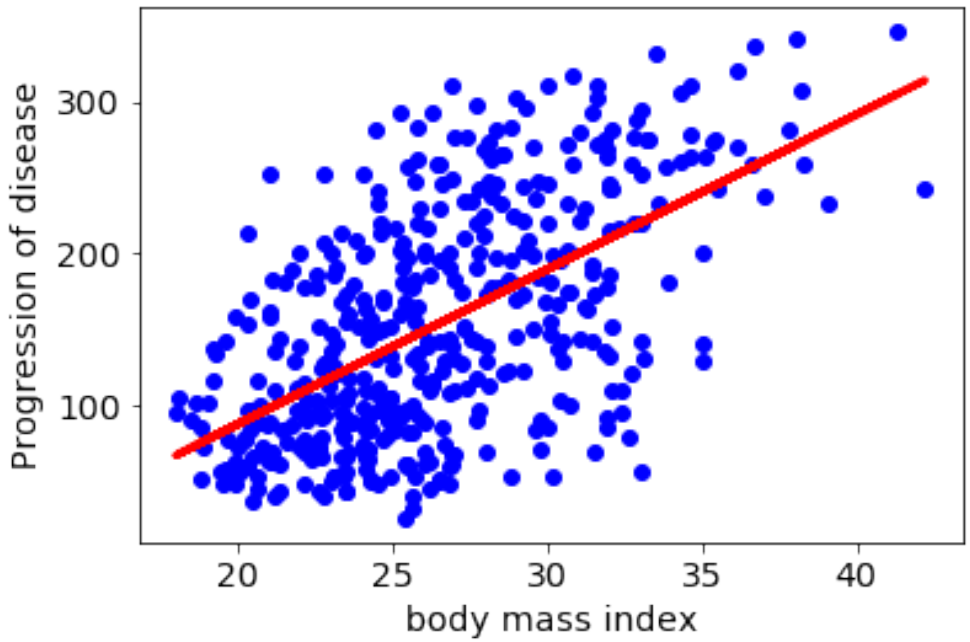
There is a simple formula for w.

Back to the diabetes data

- No predictor variables: MSE $= 5930$

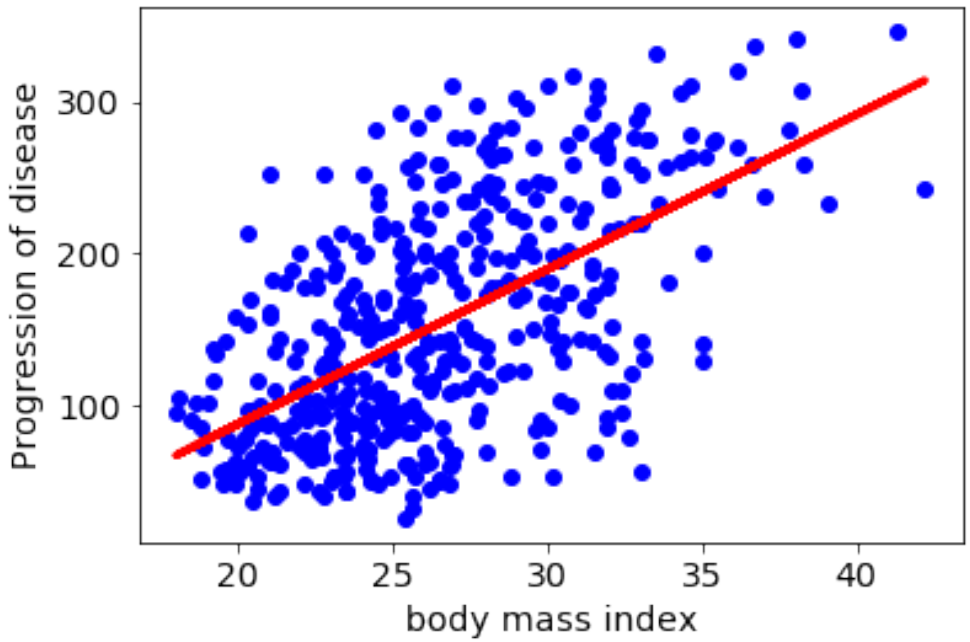
Back to the diabetes data

- No predictor variables: MSE = 5930
- One predictor ('bmi'): MSE = 3890



Back to the diabetes data

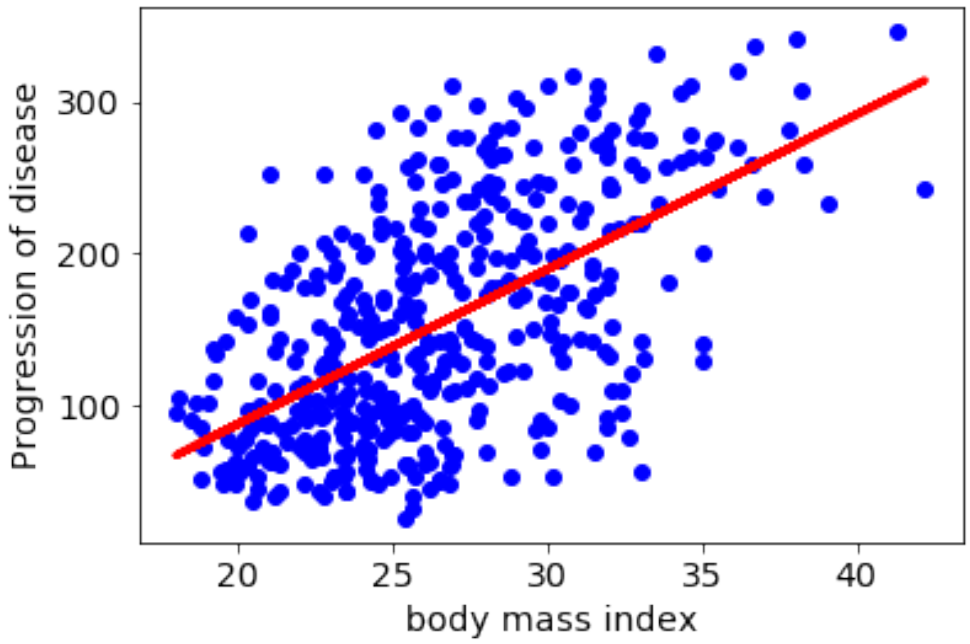
- No predictor variables: MSE = 5930
- One predictor ('bmi'): MSE = 3890



- Two predictors ('bmi', 'serum5'): MSE = 3205

Back to the diabetes data

- No predictor variables: MSE = 5930
- One predictor ('bmi'): MSE = 3890



- Two predictors ('bmi', 'serum5'): MSE = 3205
- All ten predictors: MSE = 2860

The least-squares fit

$$w = \left(\sum_i x^{(i)} x^{(i)T} \right)^{-1} \left(\sum_i y^{(i)} x^{(i)} \right)$$

Deriving the solution

$$L(w) = \sum_{i=1}^{n} (y^{(i)} - w \cdot x^{(i)})^2.$$

Matrix-vector notation

Write

$$X = \begin{pmatrix} \longleftarrow & x^{(1)} & \longrightarrow \\ \longleftarrow & x^{(2)} & \longrightarrow \\ & \vdots & \\ \longleftarrow & x^{(n)} & \longrightarrow \end{pmatrix}, \quad y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix}$$

Matrix-vector notation

Write

$$X = \begin{pmatrix} \longleftarrow & x^{(1)} & \longrightarrow \\ \longleftarrow & x^{(2)} & \longrightarrow \\ & \vdots & \\ \longleftarrow & x^{(n)} & \longrightarrow \end{pmatrix}, \quad y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix}$$

Then the loss function is

$$L(w) = \sum_{i=1}^{n} (y^{(i)} - w \cdot x^{(i)})^2 = \|y - Xw\|^2$$

and it minimized at

$$w = (X^T X)^{-1} (X^T y)$$

Generalization behavior of least-squares regression

Given a **training set** $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \mathbb{R}$, find regressor $w \in \mathbb{R}^d$ that minimizes the squared loss

$$L(w) = \sum_{i=1}^{n} (y^{(i)} - (w \cdot x^{(i)}))^2.$$

Is the squared loss on the training set a good estimate of performance on **future** data?

Generalization behavior of least-squares regression

Given a **training set** $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \mathbb{R}$, find regressor $w \in \mathbb{R}^d$ that minimizes the squared loss

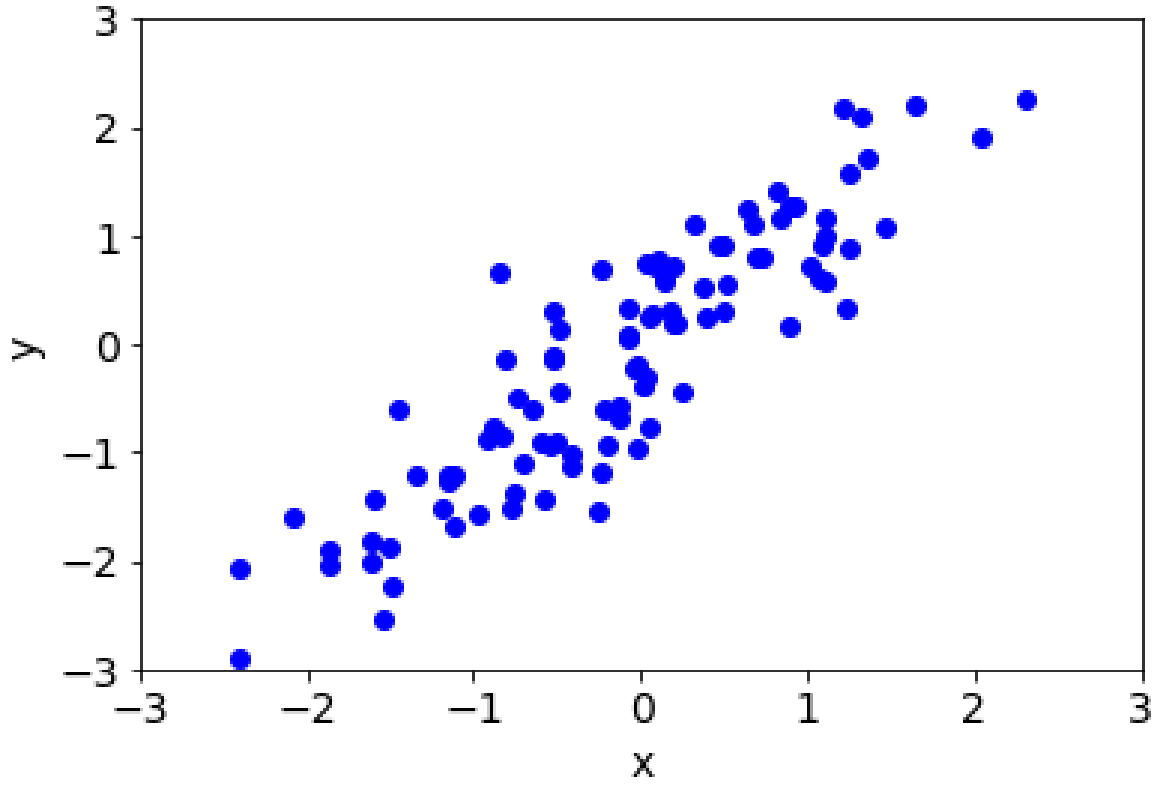
$$L(w) = \sum_{i=1}^{n} (y^{(i)} - (w \cdot x^{(i)}))^2.$$

Is the squared loss on the training set a good estimate of performance on **future** data?

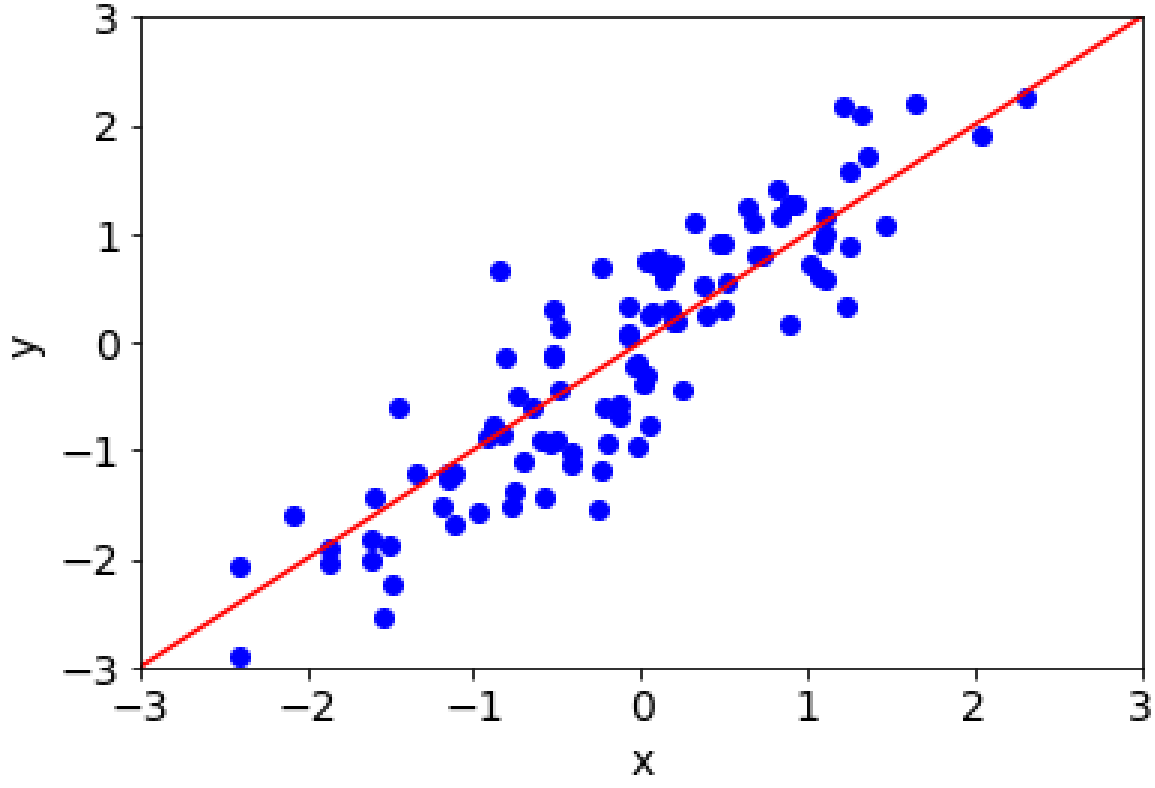
Short answer:

- If n is large enough: yes.
- Otherwise: probably an underestimate.

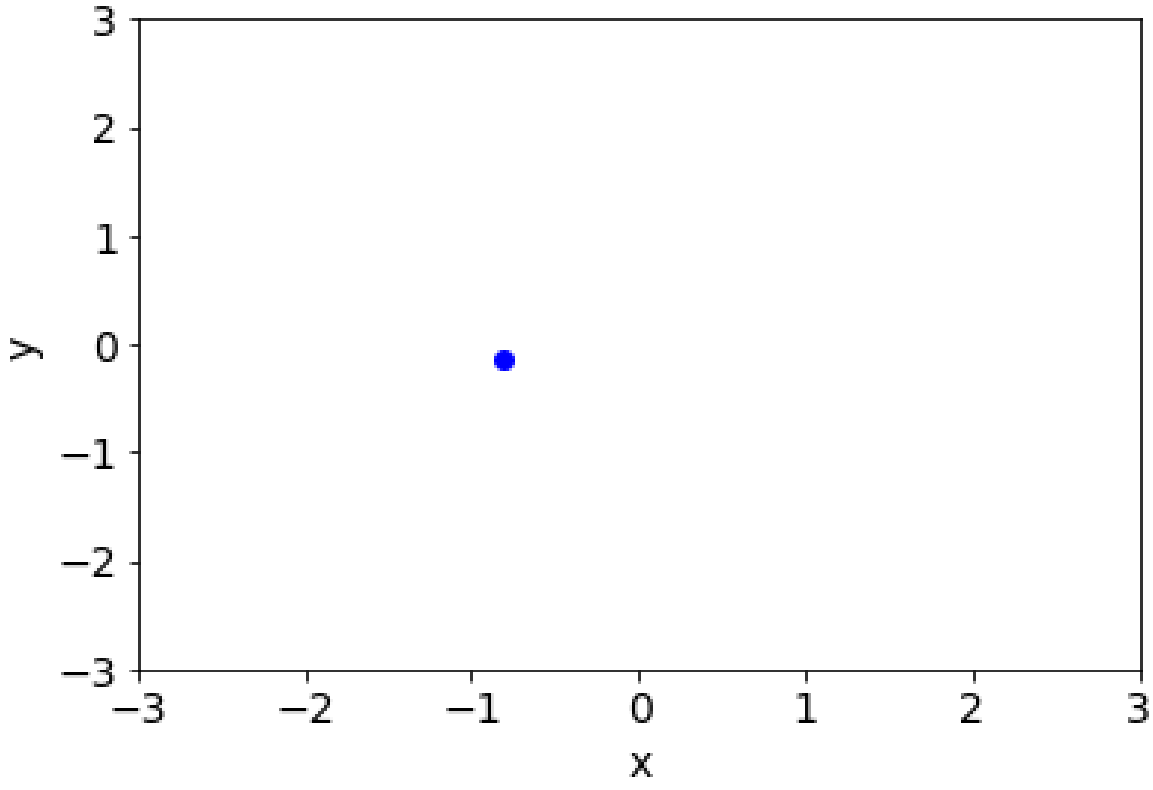
Example



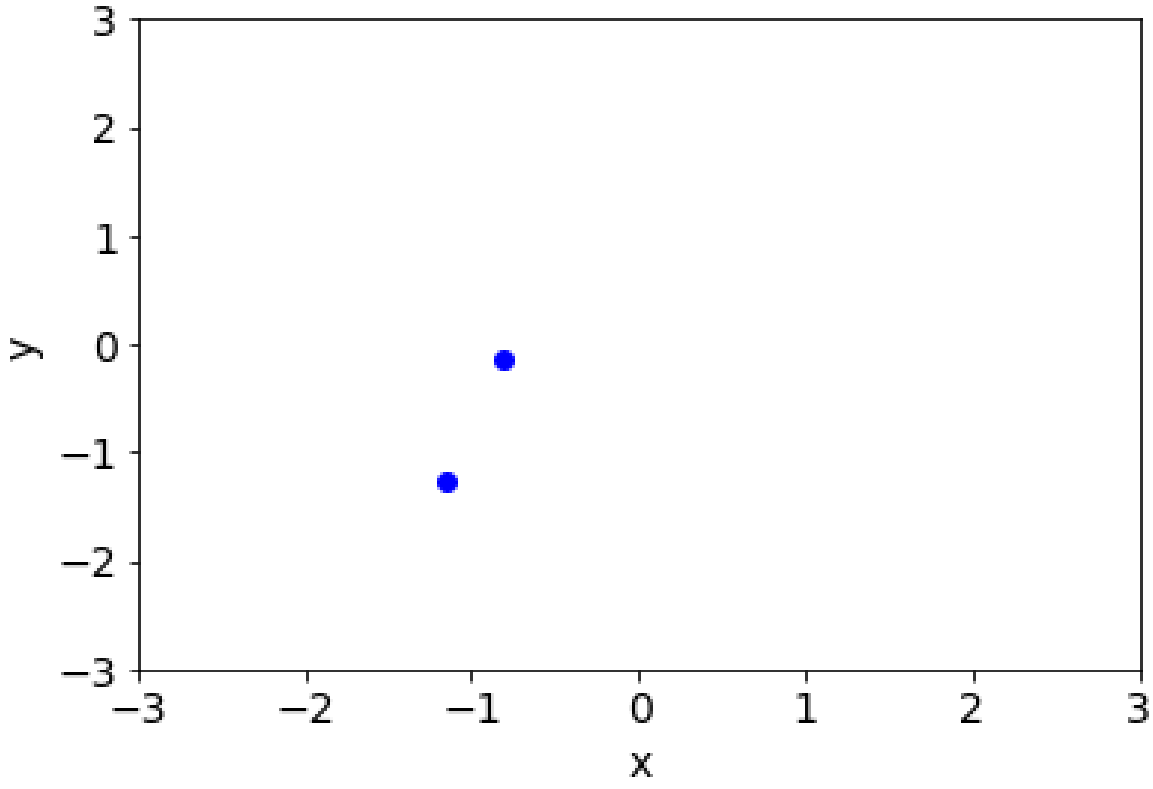
Example



Example



Example



Better error estimates

Recall: k-**fold cross-validation**

- Divide the data set into k equal-sized groups $S_1, \ldots, S_k$
- For $i = 1$ to k:
  - Train a regressor on all data except S_i
  - Let E_i be its error on S_i
- Error estimate: average of $E_1, \ldots, E_k$

Ridge regression

Minimize squared loss **plus** a term that penalizes "complex" w:

$$L(w) = \sum_{i=1}^{n} (y^{(i)} - (w \cdot x^{(i)}))^2 + \lambda \|w\|^2$$

Adding a penalty term like this is called **regularization**.

Toy example

Training, test sets of 100 points

- $x \in \mathbb{R}^{100}$, each feature x_i is Gaussian $N(0, 1)$
- $y = x_1 + \cdots + x_{10} + N(0, 1)$

Toy example

Training, test sets of 100 points

- $x \in \mathbb{R}^{100}$, each feature x_i is Gaussian $N(0, 1)$
- $y = x_1 + \cdots + x_{10} + N(0, 1)$

λ	training MSE	test MSE
0.00001	0.00	585.81
0.0001	0.00	564.28
0.001	0.00	404.08
0.01	0.01	83.48
0.1	0.03	19.26
1.0	0.07	7.02
10.0	0.35	2.84
100.0	2.40	5.79
1000.0	8.19	10.97
10000.0	10.83	12.63

The lasso

Popular "shrinkage" estimators:

- **Ridge regression**

$$L(w) = \sum_{i=1}^{n} (y^{(i)} - (w \cdot x^{(i)}))^2 + \lambda \|w\|_2^2$$

- **Lasso**: tends to produce sparse w

$$L(w) = \sum_{i=1}^{n} (y^{(i)} - (w \cdot x^{(i)}))^2 + \lambda \|w\|_1$$

The lasso

Popular "shrinkage" estimators:

- **Ridge regression**

$$L(w) = \sum_{i=1}^{n} (y^{(i)} - (w \cdot x^{(i)}))^2 + \lambda \|w\|_2^2$$

- **Lasso**: tends to produce sparse w

$$L(w) = \sum_{i=1}^{n} (y^{(i)} - (w \cdot x^{(i)}))^2 + \lambda \|w\|_1$$

Toy example:
Lasso recovers 10 relevant features plus a few more.