Machine Learning
CS342

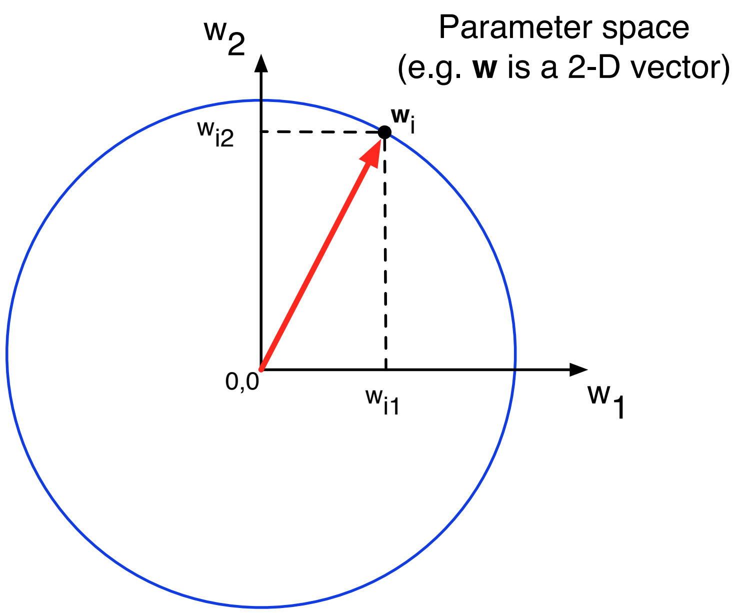
Lecture 8: The Maximum Likelihood framework

Dr. Theo Damoulas
T.Damoulas@warwick.ac.uk

Office hours: Mon & Fri 10-11am @ CS 307

Recap: Regularised Linear regression (PLS vs Lasso)

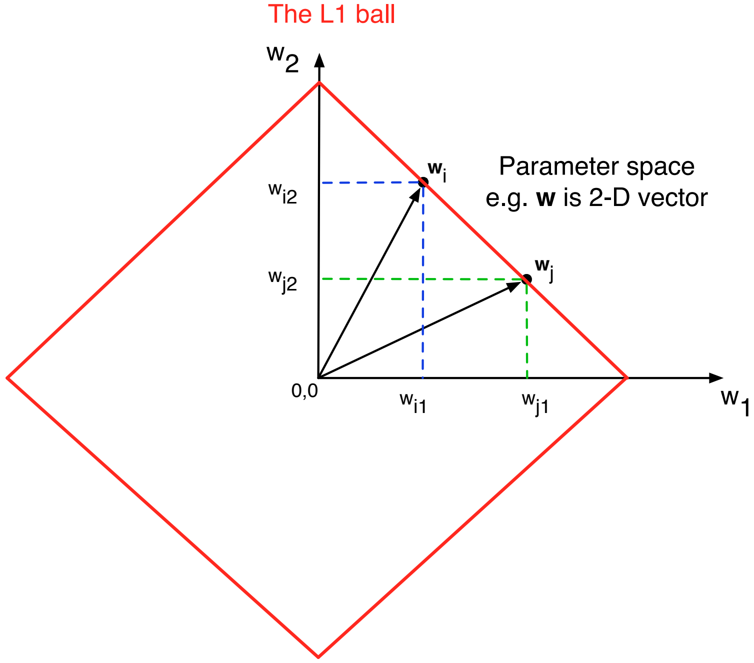
Regularisation to avoid overfitting in OLS



PLS/Ridge regression

$$\mathrm{L}_2^2(\mathbf{w}) = \sum_d w_d^2 \qquad \mathcal{L}' = \mathcal{L} + \lambda \mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\text{Minimise } \mathcal{L} \qquad \text{s.t. } \sum_d w_d^2 = \mathbf{w}^{\mathrm{T}}\mathbf{w} \leq t$$



The Lasso

$$\mathrm{L}_1(\mathbf{w}) = \sum_{d=1}^{D} |w_d| \qquad \mathcal{L}' = \mathcal{L} + \lambda \sum_d |w_d|$$

$$\text{Minimise } \mathcal{L} \qquad \text{s.t. } \sum_d |w_d| \leq t$$

Recap: PLS versus The Lasso

PLS / Ridge regression

`sklearn.linear_model`.Ridge

- We “couple” the parameter magnitudes to constrain them
- We constraint parameters by regularising with squared **L_2 norm**
- Lambda controls the strength of regularisation (the volume of the ball)

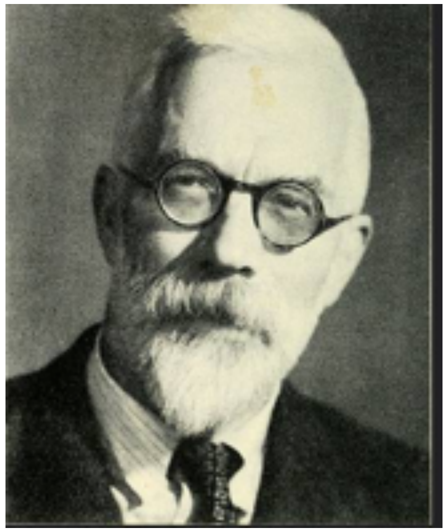
The Lasso

`sklearn.linear_model`.Lasso

- We “couple” the parameter magnitudes to constrain them
- We constraint parameters by regularising with the **L_1 norm**
- Sparse solutions with some parameters at 0
- Great for Interpretation - Lambda again controls regularisation strength
- Will under-fit if our problem is not really sparse (use PLS instead)
- Will outperform PLS when many attributes are irrelevant

Other variants (Elastic Net) with mixed norms!

Maximum Likelihood: Errors as random noise



Sir R. A. Fisher

Statistical framework - not a model!
A way of thinking about ***"errors" as random variables***

"The Maximum Likelihood principle" - can be applied to most SL problems

We will study this principle in the context of a setting we understand:
Linear regression! (exciting?)

In this lecture we will derive the **exact same solution as OLS**
but through The Maximum Likelihood principle

We will think ***Generatively*****:** *How has our data been generated?*

Errors as Noise

Rogers & Girolami, Ch. 2

Requires familiarity with random variables and probability…
Support: R&G book 2.2.1 - 2.7 and module website material

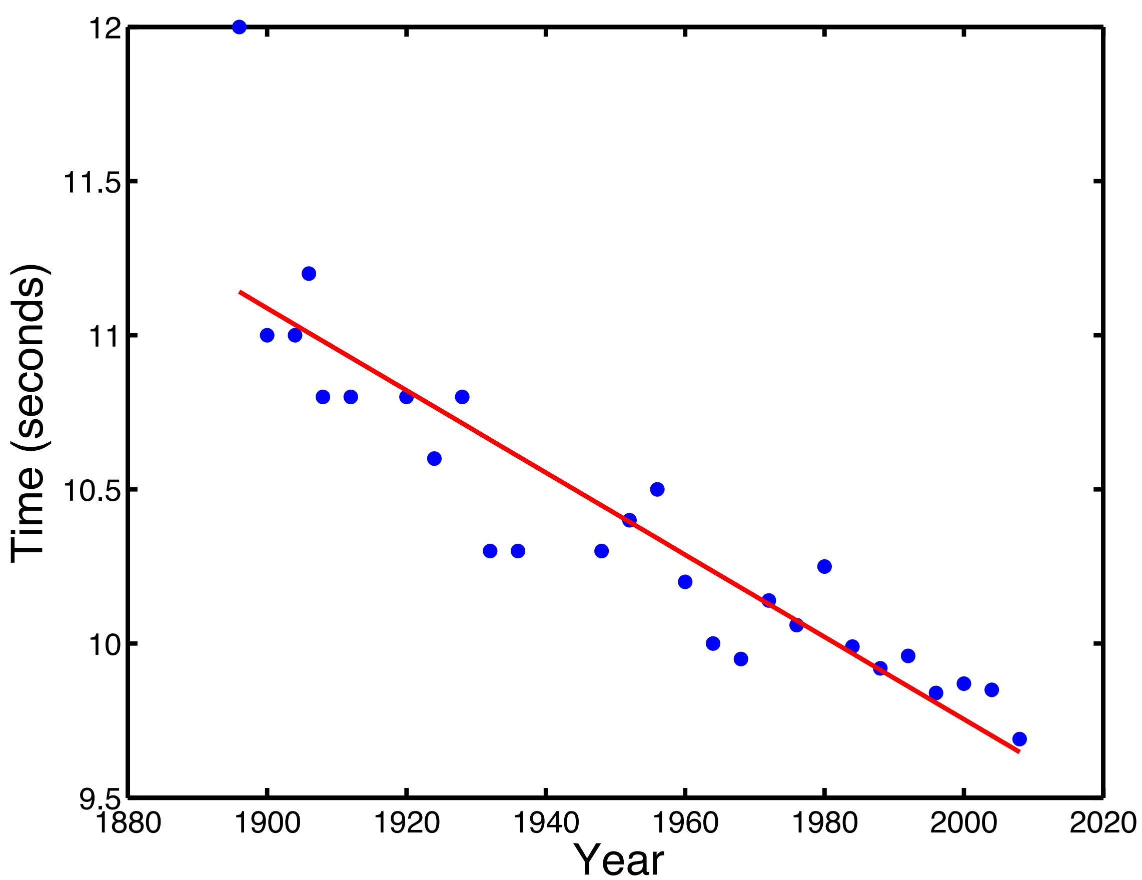
What was the "framework" we followed so far in LinReg (OLS/PLS/Lasso)?

Choose a Loss function (squared error), perhaps add regulariser.
Then Minimise it to get final parameters/solution

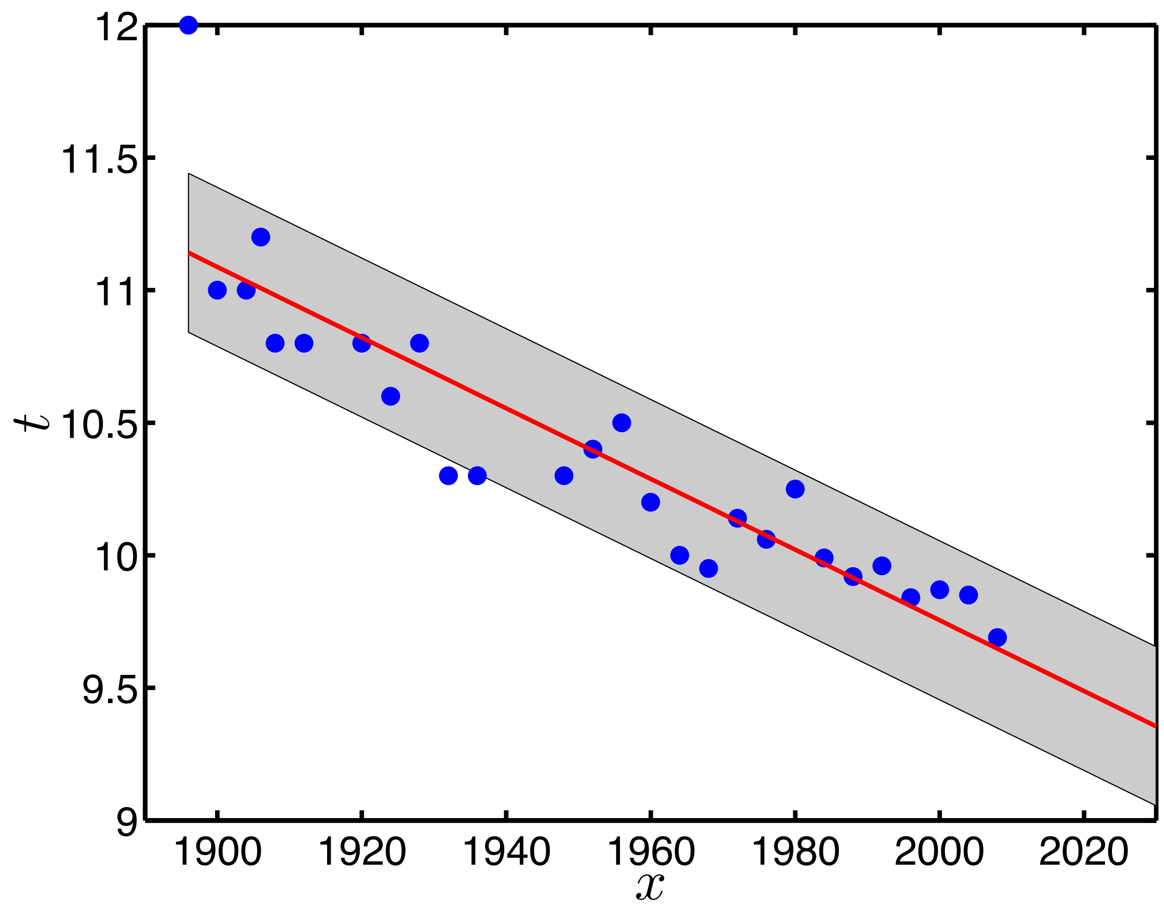
Why linear hypothesis?

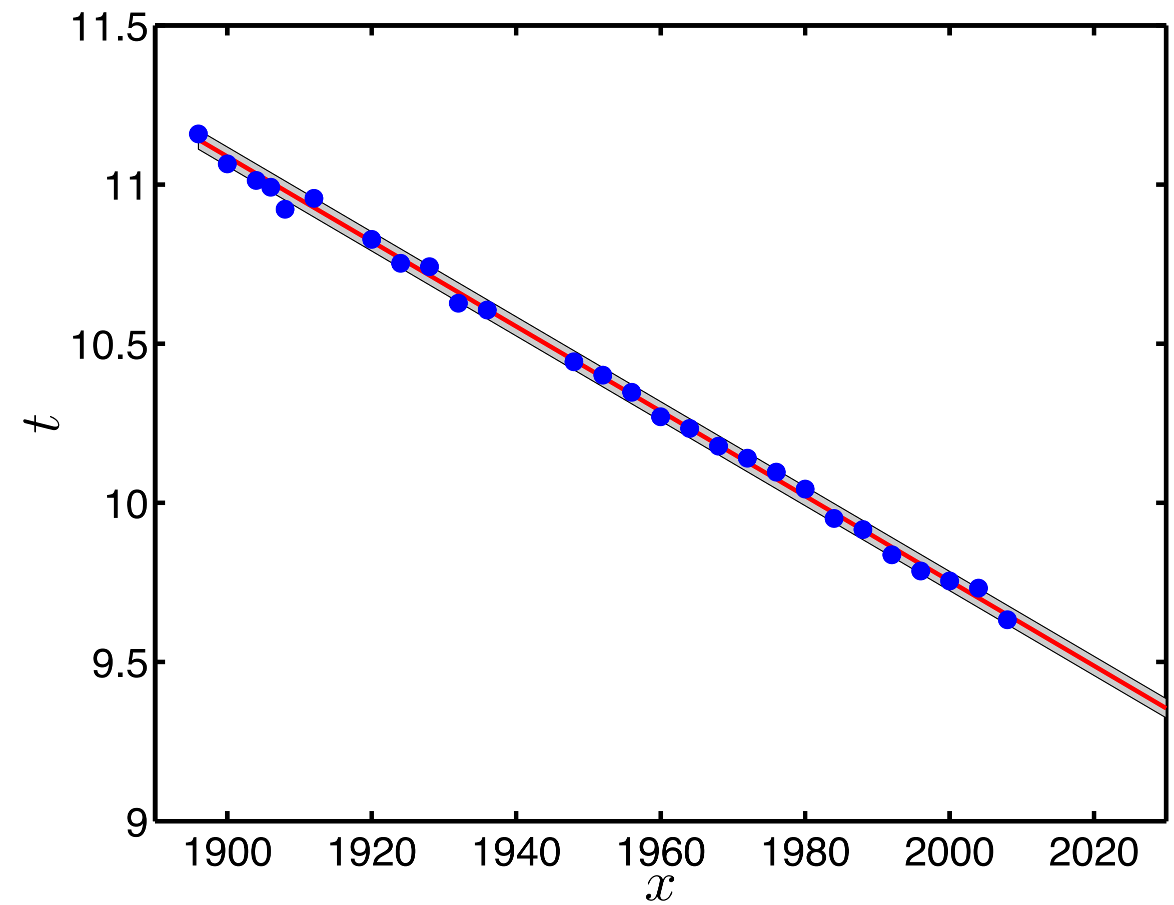
What are we saying about the underlying process that generated our data?

What is "noise"?



Errors as Noise





Same linear fit

How confident should we be for our predictions in each case?

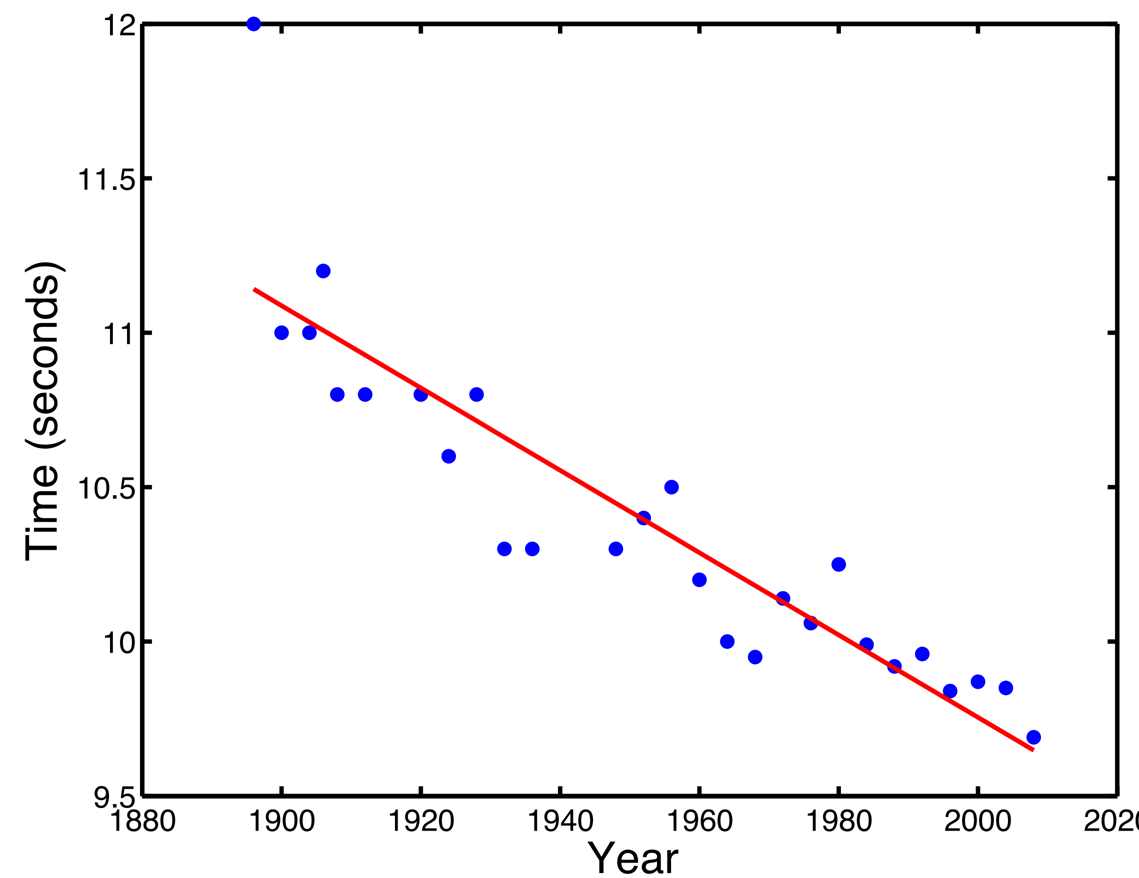
Errors as Noise

We will think ***Generatively*: *How has our data been generated?***

Ok lets call it noise.
Is there an obvious pattern?

Looks deviations from line are random
Noise is “random”

What is “random”?



Theo’s working definition of random:
Anything that we lack the information and/or the computational capabilities in order to compute/predict.

Randomness parenthesis

How do we compute/use random numbers in our computers?

Random number generators

Wikipedia: A **random number generator** (**RNG**) is a computational or physical device designed to generate a sequence of numbers or symbols that can not be reasonably predicted better than by a random chance

Pretty circular?… lets summarise what an RNG does

RNG: Produces “pseudo-random” numbers based on increasingly complex ***patterns***

Food for thought…

Randomness

Entropy: A measure of structure/order/homogeneity

Out of the box: High Entropy
(very "random")

As we build it we reduce Entropy.
(less "random")

Ok enough with "philosophy" lets go back to linear regression

Random variables 101

- A discrete random variable has a Probability Distribution Function (PDF)
- e.g. Rolling a dice (discrete events)

$$0 \leq P(X = x) \leq 1 \qquad \sum_x P(X = x) = 1$$

- What is the expected value of rolling a fair dice?

$$\mathbb{E}_{P(X)} = \sum_x x P(x) = ?$$

- A continuous random variable has a probability density function (pdf)

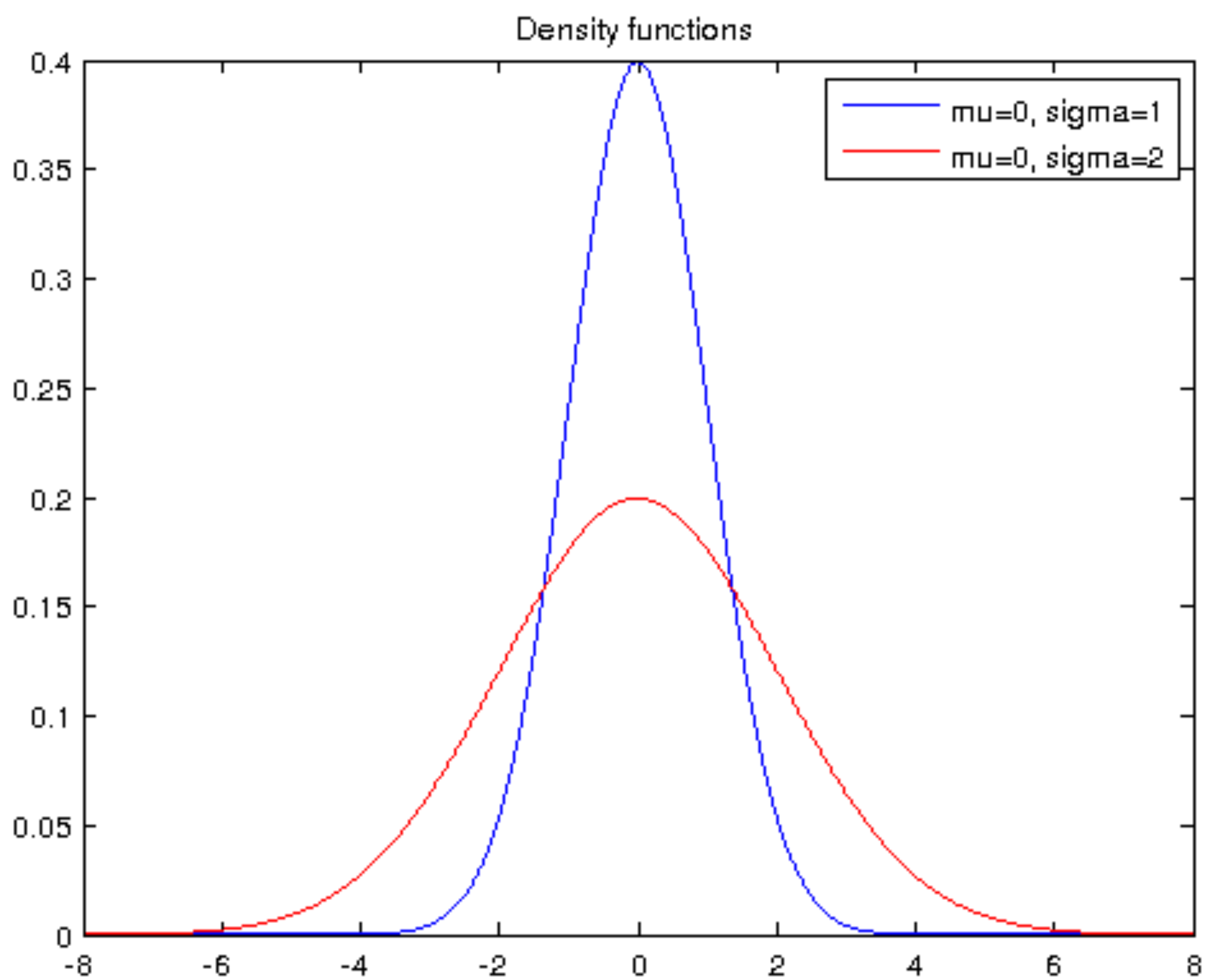
e.g. The Normal or Gaussian distribution

$$p(x) \sim \mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}$$

Random variables 101

Gaussian white noise: 0-mean Normal/Gaussian distribution

$$p(x) \sim \mathcal{N}(0, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{x^2}{2\sigma^2}\right\}$$



What is the expected value of x?

Random variables 101

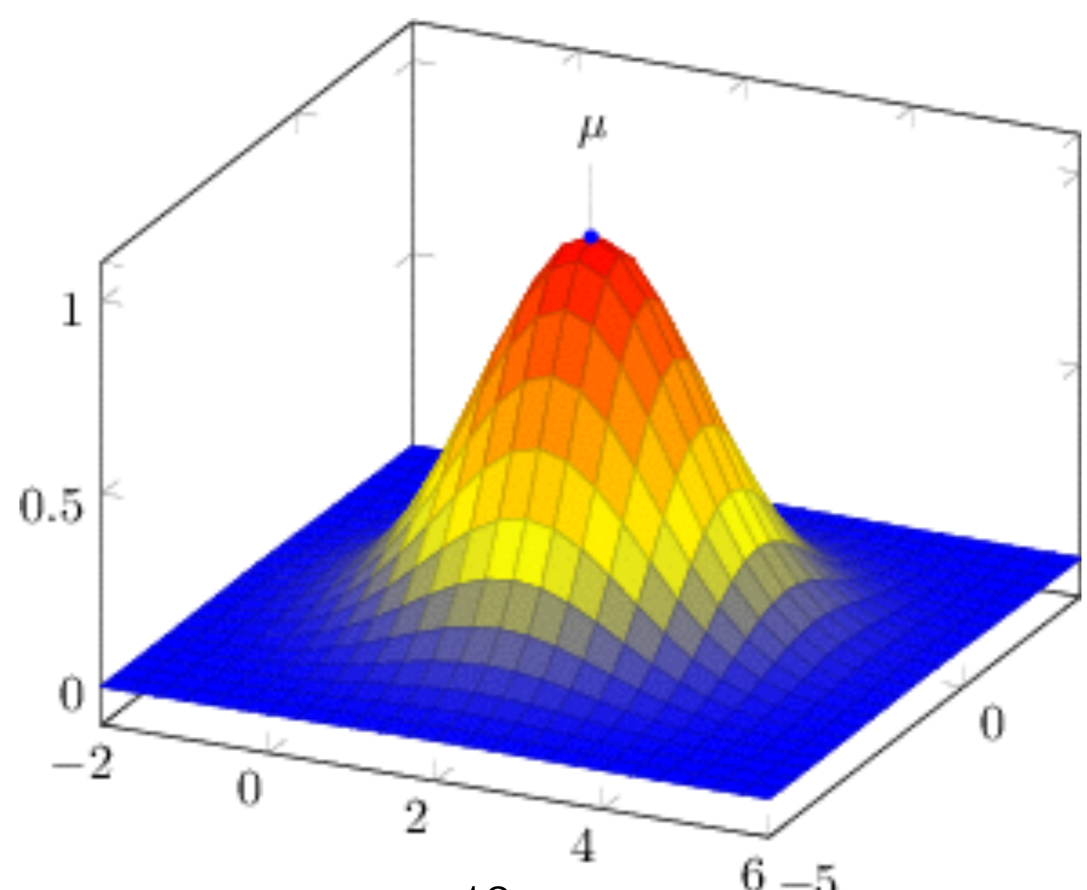
So far X was a scalar. Can we place distributions over vectors?

Higher-dimensional space so distributions become also higher-D

So a "Multivariate Gaussian distribution" is the generalisation to higher-D

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right\}$$

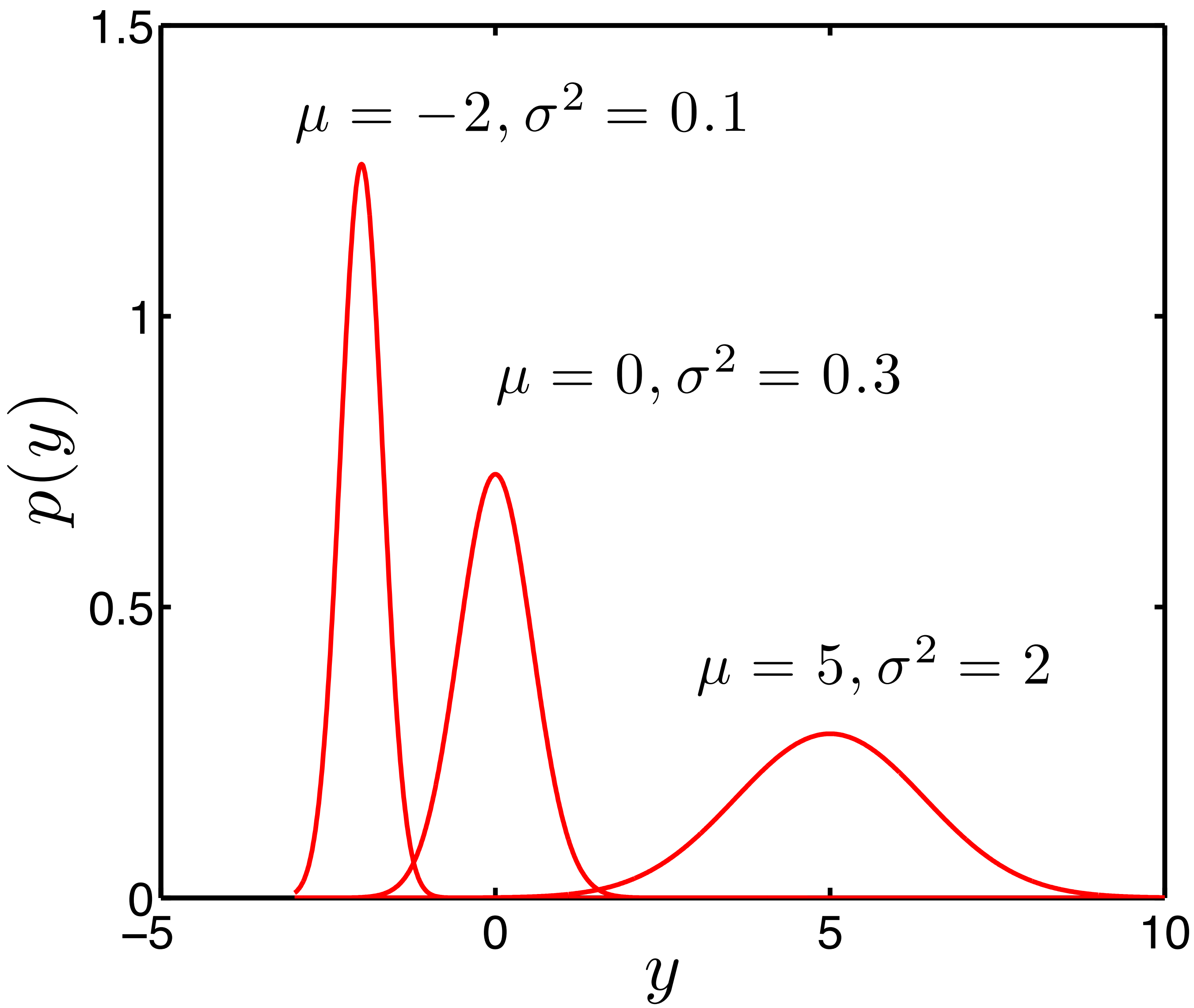
Covariance matrix defines the skewness



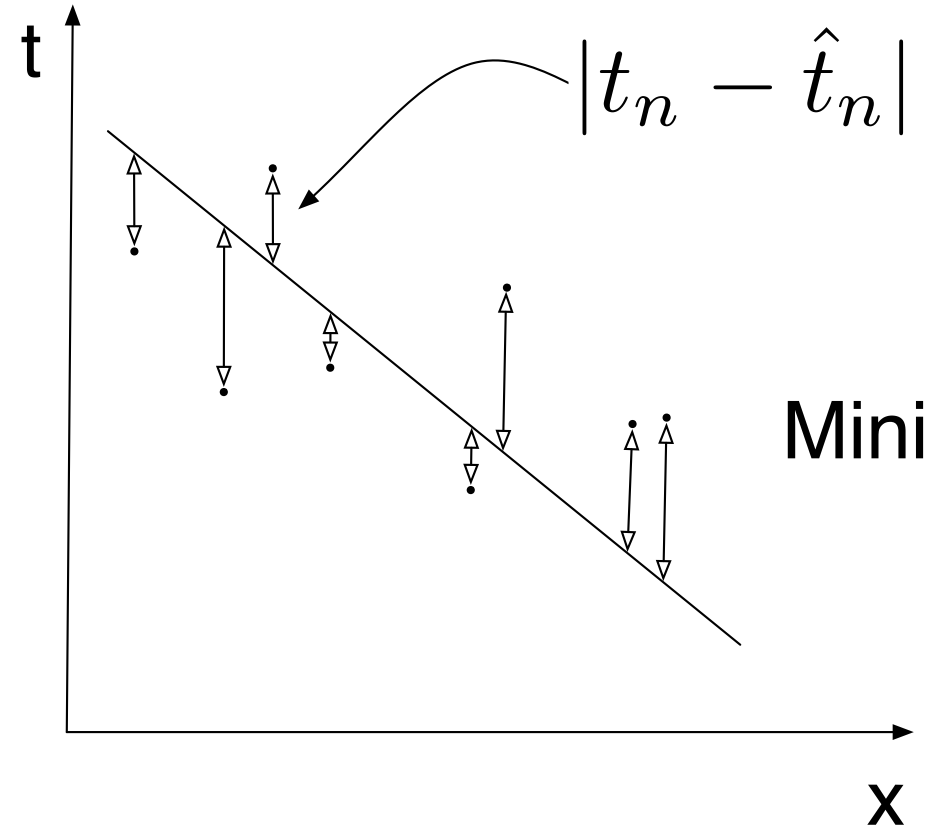
We will run into that a lot

Random variables 101

Effect of varying the mean and variance of the Gaussian distribution



Errors as Noise



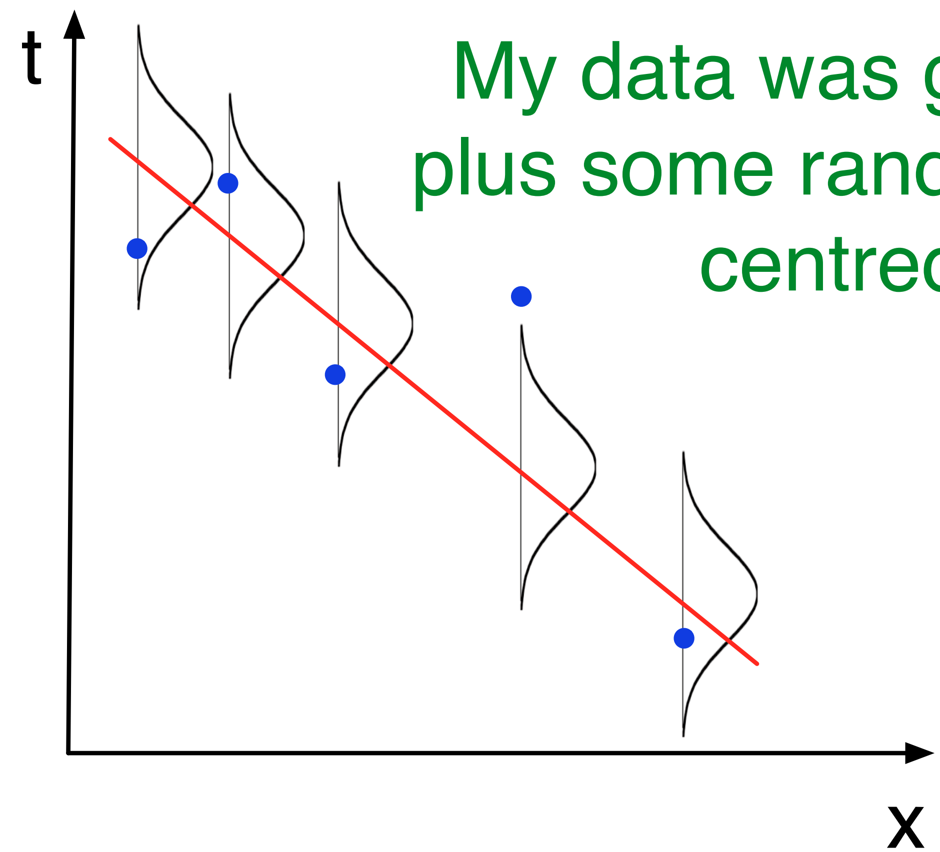
$$|t_n - \hat{t}_n| = |t_n - f(x_n; \mathbf{w})|$$

Error

OLS:
Minimise squared error

Think Generatively!

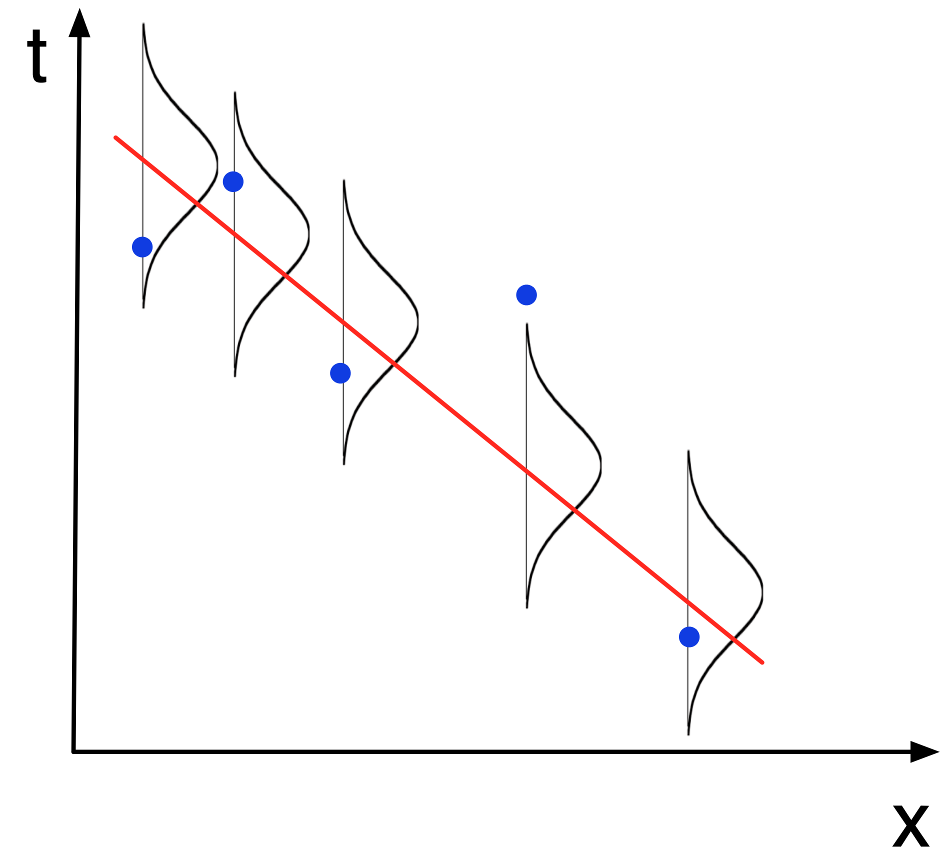
White Gaussian Noise



My data was generated by a line plus some random Gaussian noise centred on the line

Noise and Likelihood

So my model of what happened is:



$$t_n = f(\mathbf{x}_n; \mathbf{w}) + \epsilon_n, \quad \epsilon_n \sim \mathcal{N}(0, \sigma^2)$$

"My data was generated from a line (or plane in higher-D) plus some noise"

$$t_n = \mathbf{x}_n \mathbf{w} + \epsilon_n$$

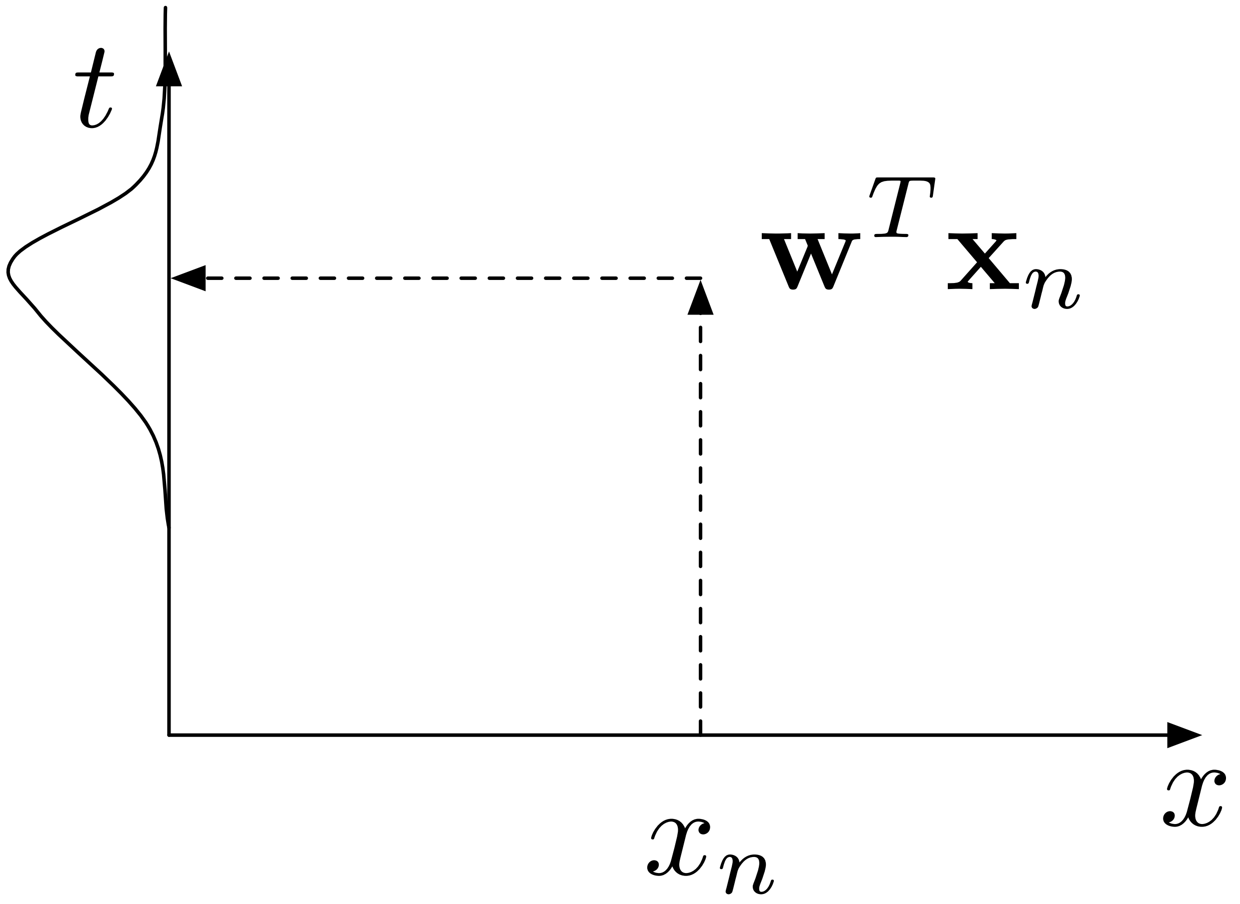
deterministic component
(a.k.a. trend or drift)

random component
(a.k.a. noise term)

Noise and Likelihood

Generate your own synthetic data:

- Create a line (Fix **w**, choose some x values)
- For every point, add Gaussian noise on t dimension



Noise and Likelihood

Assume that noise values are *independent* and *homoscedastic*:

$$p(\epsilon_1, \ldots, \epsilon_N) = \prod_{n=1}^{N} p(\epsilon_n) = \prod_{n=1}^{N} \mathcal{N}(0, \sigma^2)$$

Q: Why does independence lead to a product?

Substitute for noise term $$t_n = \mathbf{x}_n \mathbf{w} + \mathcal{N}(0, \sigma^2)$$

When adding a constant to a normal distribution what happens?

$$t_n = \mathcal{N}(\mathbf{x}_n \mathbf{w}, \sigma^2)$$ So it is $$p(t_n | \mathbf{x}_n, \mathbf{w}, \sigma^2) = \mathcal{N}(\mathbf{x}_n \mathbf{w}, \sigma^2)$$

Noise and Likelihood

$$p(t_n|\mathbf{x}_n, \mathbf{w}, \sigma^2) = \mathcal{N}(\mathbf{x}_n\mathbf{w}, \sigma^2)$$

And using independence of noise variables to talk about all the data:

Likelihood

$$\boxed{p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n\mathbf{w}, \sigma^2)}$$

It is a function of the parameters

$$L(\boldsymbol{\Theta}) \text{ in this case : } L(\mathbf{w}, \sigma^2)$$

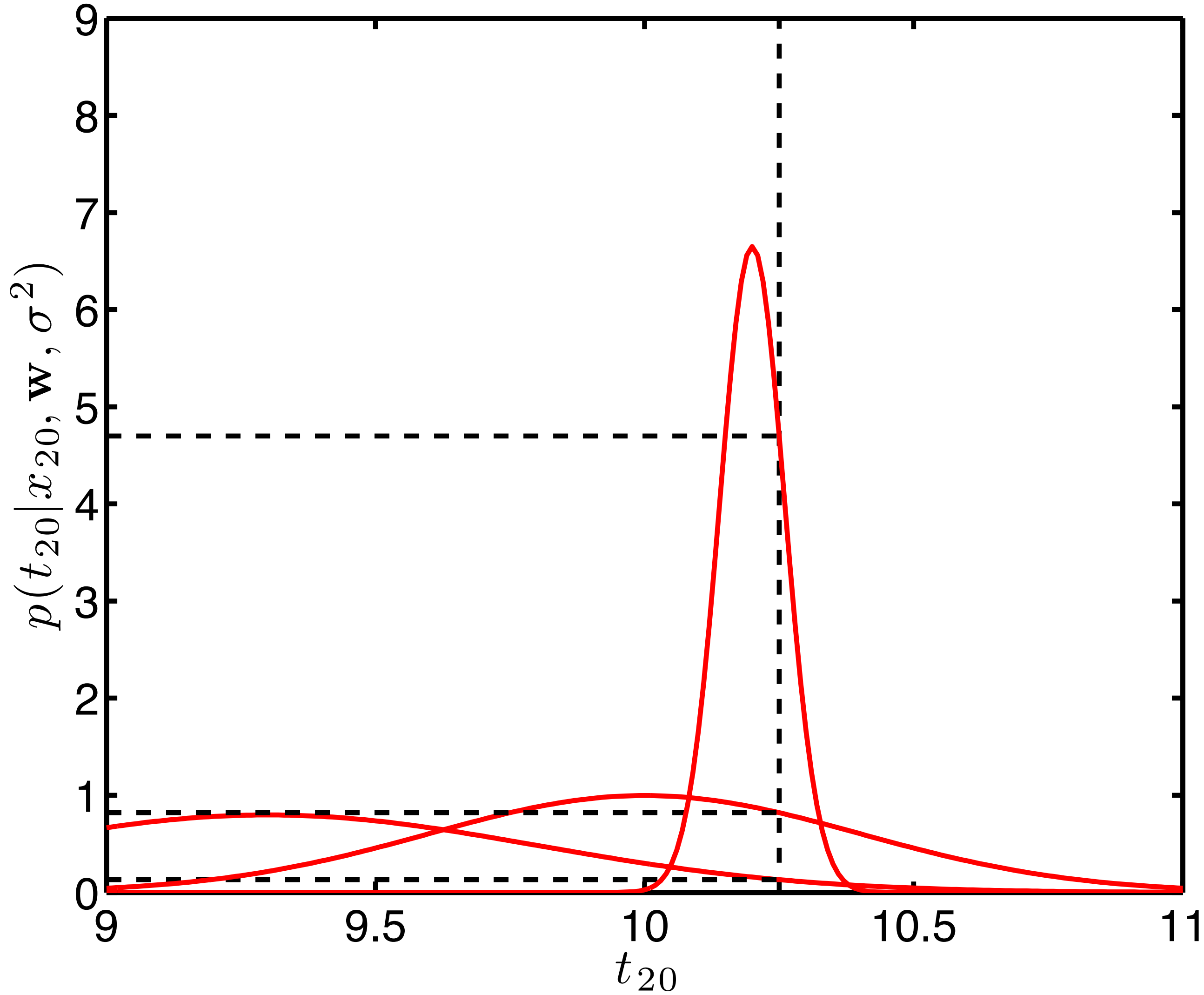
"How likely is that my model with these parameters can generate the data?"
Likelihood of observing the data under my model

frequentist vs Bayesian views

Likelihood: More examples: Olympic data

Let's look at the 1980 Olympics (n=20).
Dashed vertical line shows t_{20}

Looking at a
single observation
under different Likelihoods



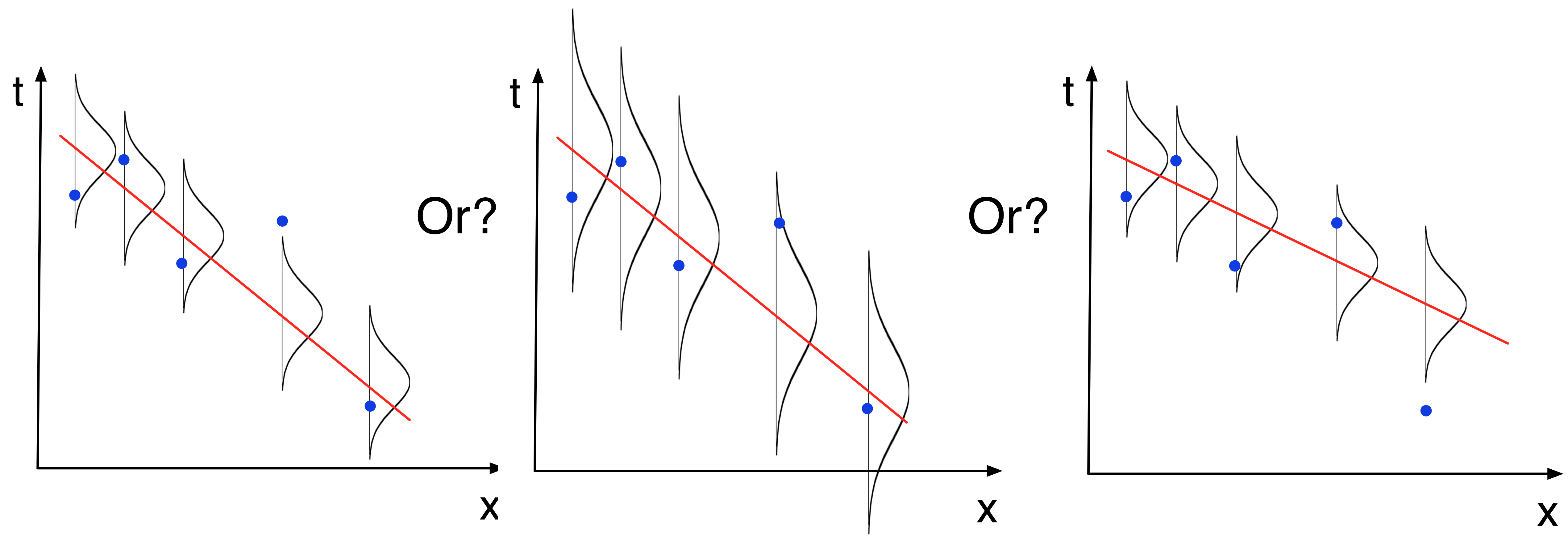
Third model (highest peak) looks better

Department of COMPUTER SCIENCE

Likelihood

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n \mathbf{w}, \sigma^2)$$

"How likely is that my model with these parameters can generate the data?



Different parameters = different Likelihood of model generating the data

Do I want my model to have high likelihood or low? What do I do?

Maximum Likelihood

Rogers & Girolami, Ch. 2: 2.7.2

Learning: Find the parameters that maximise the Likelihood function

$$\mathbf{w}, \sigma \leftarrow \underset{\mathbf{w},\sigma}{\operatorname{argmax}} \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n \mathbf{w}, \sigma^2)$$

Any analogies to other frameworks we have learned so far?

In fact we will maximise the natural logarithm (*ln* really) of the likelihood

$$\mathbf{w}, \sigma \leftarrow \underset{\mathbf{w},\sigma}{\operatorname{argmax}} \ \log \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n \mathbf{w}, \sigma^2)$$

What will happen?

Similar derivation strategy as with OLS derivation:
1st derivative to 0, examine 2nd derivative matrix

Maximum Likelihood

Rogers & Girolami, Ch. 2: 2.7.2

Substituting for the normal pdf

$$\mathbf{w}, \sigma \leftarrow \underset{\mathbf{w},\sigma}{\operatorname{argmax}} \sum_{n=1}^{N} \log \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(t_n - \mathbf{x}_n \mathbf{w})^2}{2\sigma^2} \right\} \right\}$$

I will have a term that looks like the sum of squared errors!

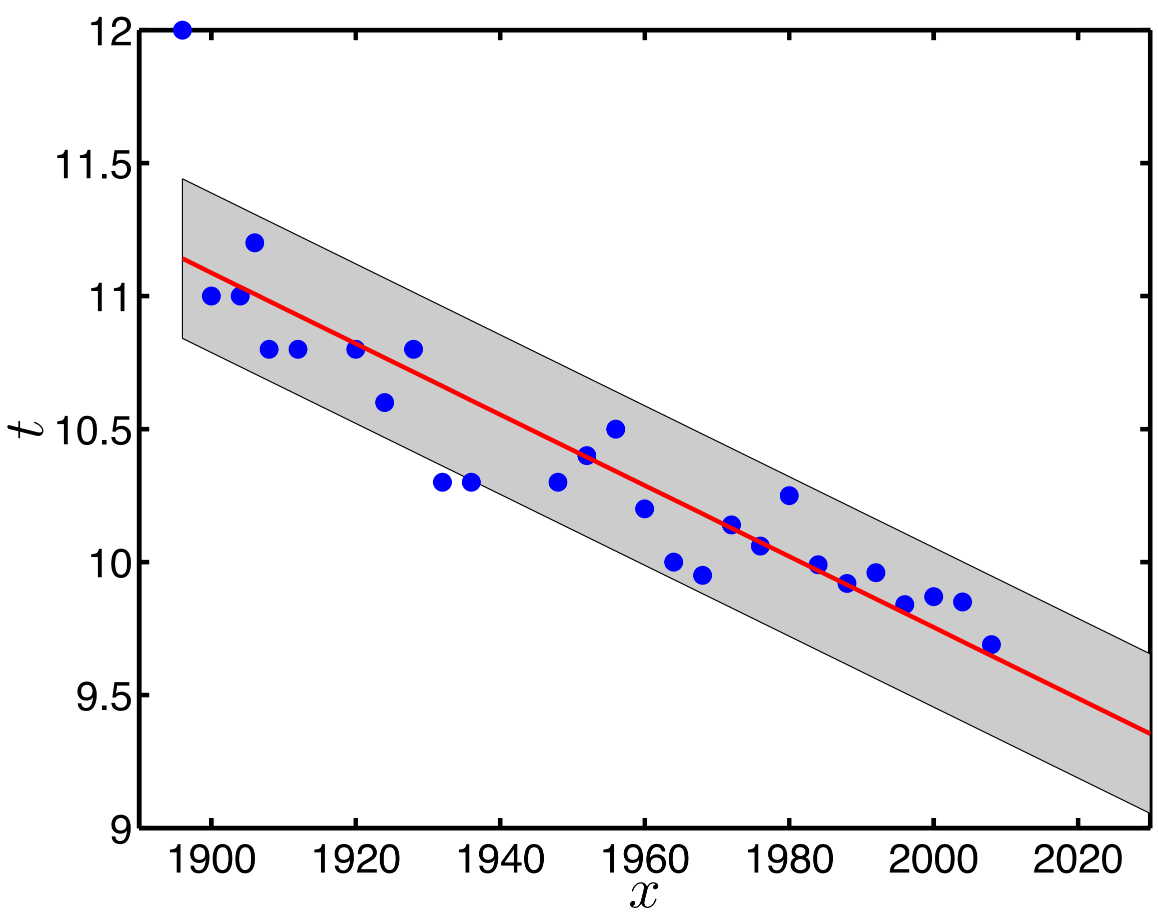
The MaxLike solution for w: $\boxed{\hat{\mathbf{w}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}}$

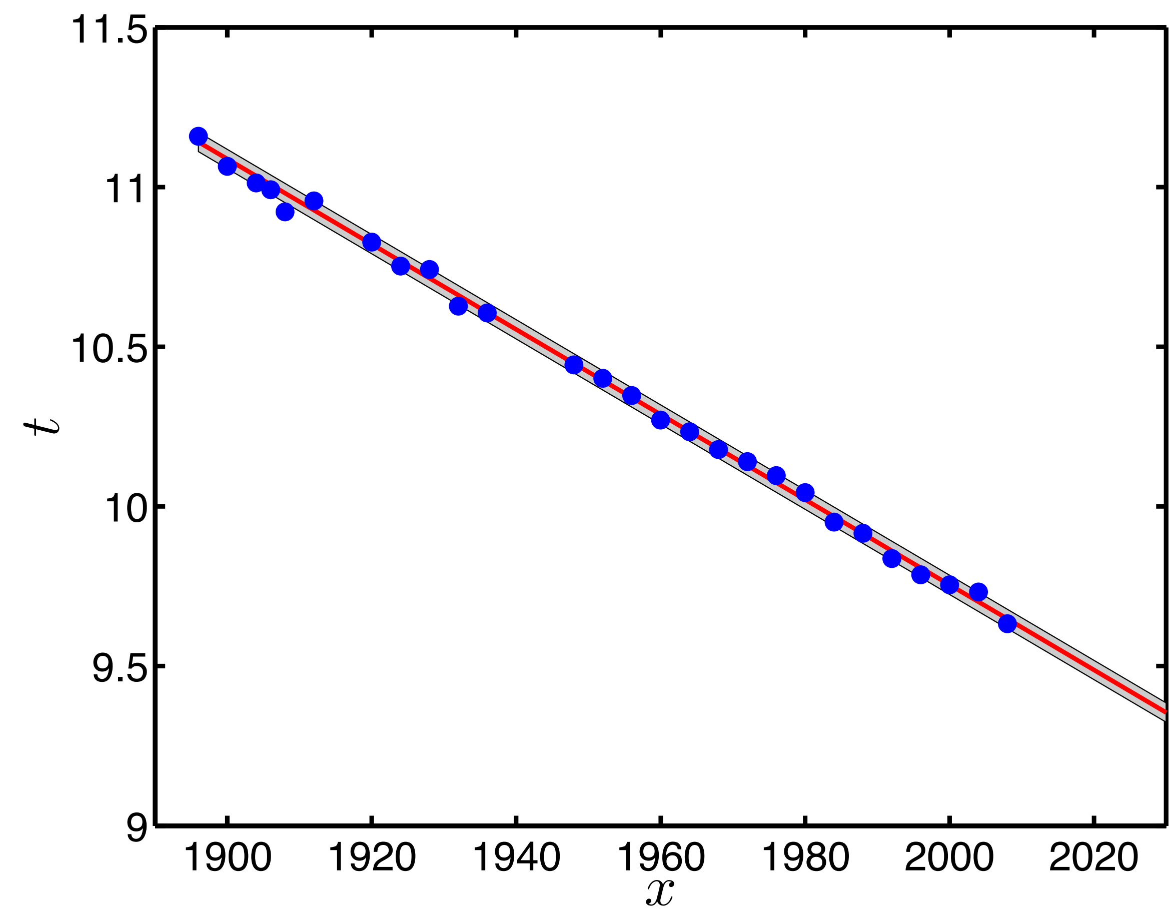
We arrived at the some **w** solution for linear regression as OLS via the Maximum Likelihood framework!

We also learn a noise model (variance). This will give us benefits later on!

Maximum Likelihood

We also learn a noise model (variance). This will give us benefits later on! Can you guess?





Summary for Maximum Likelihood

- Thinking Generatively
- Errors as Noise
- Linear Model has a deterministic and a random (noise) component
- Noise term leads to Likelihood function $L(w,\sigma^2)$
- Likelihood of observing the data under/given my model
- Find parameters (learning) by maximising the Likelihood
- Analogy between minimising Loss and maximising Likelihood
- Equivalence of parameter update in linear regression (OLS-ML)
- Similar problems to OLS (overfitting, outliers)