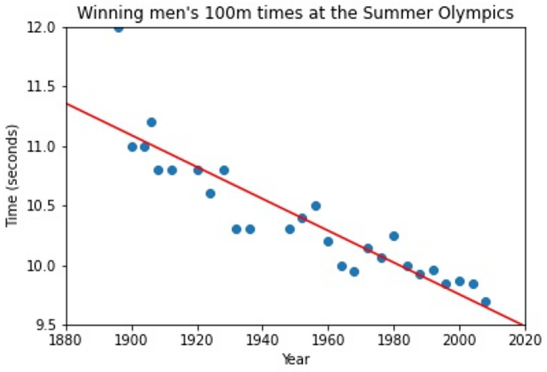
4.1 Maximum likelihood; Introduction to Bayes' rule

By the end of this notebook you should be able to:

1. Understand the maximum likelihood approach to linear regression
2. Describe the differences between Classical vs Frequentist vs Bayesian Probability
3. Understand Bayes' rule

Modelling errors as noise

Recall our example with the 100m data:



Our least squares linear regression model captures the *trend* of the data

But it does not capture the data points perfectly: there are errors between the model and the true values

Note, the errors in our model:
- are sometimes positive, sometimes negative
- are not constant in magnitude
- appear to be independent of the *x* variable (Year)

Suppose we wanted to use our model to **generate** data (e.g. for the years mid-way between consecutive Olympic games).

- It will only generate data along the straight line, which only captures the trend
- Data generated from this straight line will look very unrealistic compared to the real data

In order to use our model to **generate** more realistic looking data, we could add an additional error term ϵ, that behaves like the error

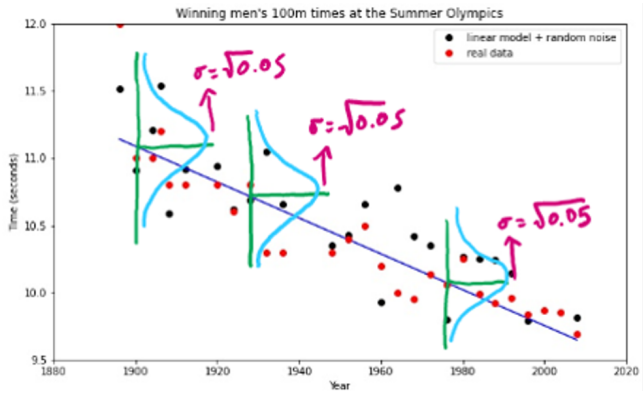
- This error ϵ is a random variable, i.e. it is not deterministic.

We can think of the error term ϵ as being needed due to the fact that the features $\mathbf{x}$ used in the model $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ do not contain all the information needed to determine the value of the dependent variable y.

Errors are very often normally distributed, that is their distribution follows a Gaussian distribution.

As a first attempt to make our model more realistic, let's just sample errors randomly from a Gaussian distribution (using zero mean, and a rough guess at a suitable variance) and add these errors to our linear model.

Suppose we let each error $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$, with $\mu = 0$ and $\sigma^2 = 0.05$



This generated data (black dots) still has a downward trend (captured by the linear model) but now "looks" more realistic, in that it has a random element to its behaviour, similar to the real data (red dots).

Our model is now of the form $\mathbf{w}^T\mathbf{x} + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

There are now two components to our model:

1. A deterministic component $\mathbf{w}^T\mathbf{x}$, referred to as the *trend*.
2. A random component ϵ, referred to as the *noise*.

Our model now has an additional parameter σ^2 that we need to set (in addition to the weights $\mathbf{w}$ as before).

Note that the output of our model is now not deterministic: it is a random variable.

- Therefore there is no single predicted value of y at each point $\mathbf{x}$.
- This means that we cannot use the loss functions from before to optimize both $\mathbf{w}$ and σ^2 (the loss functions requires definite values for the residuals at the training points, in order to make sense).

Likelihood

We can consider the probability density function of the model output at each training point $\mathbf{x}_n$.

- It will be a normal distribution, centred at the value predicted by the linear trend $\mathbf{w}^T\mathbf{x}_n$
- The variance of the normal distribution is σ^2
- The true value of the response y_n at $\mathbf{x}_n$ will be associated with a particular probability density value from this normal distribution: this is called the **likelihood** of the n-th training point.
- The higher the likelihood is for a training point, the more likely it is to be generated by our model
- By varying the weights $\mathbf{w}$ and the value of σ^2, the likelihood of each training point changes.

Dataset likelihood

We are not interested in the likelihood of one single training point, but instead that of the entire training data.

This is the given by the product of the likelihoods of each training example (as the error at each training example is independent)

$$L = \prod_{i=1}^{N} p(y_i|\mathbf{x}_n, \mathbf{w}, \sigma^2) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{w}^T\mathbf{x}_n, \sigma^2)$$

This gives us a value for how *likely* our training data is, given our model (i.e. given values of $\mathbf{w}$ and σ^2)

It makes sense to construct the model for which the training data is most likely.

- This is called the **maximum likelihood** model.
- It has values of $\mathbf{w}$ and σ^2 that maximizes L

It can be shown (see e.g. *First Course in Machine Learning* pp. 71-72) that the maximum likelihood solution for $\mathbf{w}$ is

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

i.e. the same weights that minimize the mean squared errors loss function.

It can also be shown (p. 72) that the maximum likelihood estimate for σ^2 is:

$$\hat{\sigma^2} = \frac{1}{N}(\mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\hat{\mathbf{w}})$$

There is much more could be said about maximum likelihood models (e.g. see p. 75 -91 of *A First Course in Machine Learning*), e.g.:

- maximizing likelihood favour complex models (Section 2.8.4 of *FCML*)
- expressing the uncertainty in our predictions (Section 2.11 of *FCML*)

which you are encouraged to explore in your own time.

But our main purpose in discussing them today is as a stepping stone on the way to Bayesian models.

Introduction to the meaning of *Bayesian* probability

What is the probability of:

- Winning the national lottery, when you buy one ticket?
- A biased coin landing heads?
- Life existing elsewhere in the universe?

Winning the national lottery, when you buy one ticket

- You need to choose 6 numbers from 59
- If your combination of numbers matches the combination drawn, then you win the jackpot
- Each outcome (combination of numbers) is **equally likely**
- Total number of combinations of 6 numbers from 59 is
$$\binom{59}{6} = \frac{59!}{59!(59-6)!} = 45057474$$
- Probability of any single combination winning is then $\frac{1}{45057474}$
- This is an example of **classical** probability

In [3]:
```
## National lottery
import scipy.special
n_total = 59
n_choose = 6
n_combinations = scipy.special.comb(n_total,n_choose) # number of ways of ch
prob_win = 1.0/n_combinations
print("number of combiantions = ",n_combinations)
print("probability of winning with one ticket = ",prob_win)
```

```
number of combiantions =  45057474.0
probability of winning with one ticket =  2.2193876203535066e-08
```

A biased coin landing heads

- If the coin was fair, then we would say each outcome (H/T) was equally likely, and use the approach above.
- As the coin is biased, each outcome is not equally likely.
- One way to work out probability of the coin landing Heads is to flip it a large number of times N_{flips}, and record the number of times it lands Heads N_H.
- The proportion of times it lands Heads is an approximation of $P(H)$:

$$P(H) \approx \frac{N_H}{N_{flips}}$$

- The "true" value of $P(H)$ is given by the limit as the number of trials tends to infinity:

$$P(H) = \lim_{N_{flips} \to \infty} \frac{N_H}{N_{flips}}$$

- This is an example of **frequentist** probabililty

In [4]:
```
## Biased coin landing heads
n_flips = 100
n_heads = 43
prob_heads = n_heads/n_flips
print(prob_heads)
```

```
0.43
```

Intelligent life existing elsewhere in the universe

- We view the existance of intellgent life as "uncertain"
- We then view probability as a measure of that uncertainty
- This is an example of **Bayesian** probability: probability is a measure of our uncertainty in a situation
  - sometimes referred to as *subjective* probability

- We start with our pre-existing *belief* on the matter; known as the *prior* probability
- Our belief may be updated by data on the matter
  - e.g. new planets found, strange signals from space, ...
  - Our updated probability is called the *posterior* probability
- However the strength of our prior belief still matters
  - Some people may be harder to convince with data than others
- This is the gist of how Bayesian analysis works

A proper Bayesian analysis on this for the interested:
http://www.pnas.org/content/109/2/395 (http://www.pnas.org/content/109/2/395) or
https://arxiv.org/pdf/1107.3835.pdf (https://arxiv.org/pdf/1107.3835.pdf)

Main differences between a Frequentist and a Bayesian

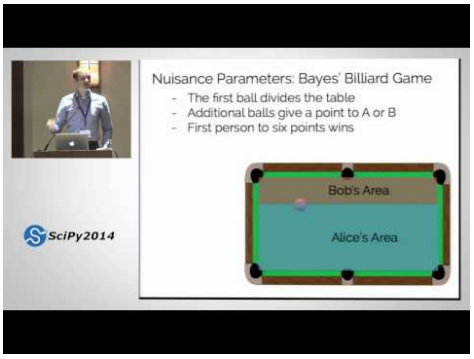
A **Frequentist** considers probability to be a property of the physical world.

- With the biased coin, the coin has a fixed bias value θ
  - a Frequentist aims to determine this "point value" of θ (e.g. $\theta = 0.476$)
- The observations in a trial are "random"
  - Results will vary for another set of flips of the coin
- However, as $N_{flips} \rightarrow \infty$, we will get closer to the true value of θ

A **Bayesian** considers probability to be a measure of uncertainty regarding their knowledge of the physical world.

- The observations have been made and so are "fixed".
- What is unknown is the bias of the coin. Therefore, we model the bias θ as a random variable
- A Bayesian analysis then proceeds to infer a probability distribution over θ (rather than determine a "true fixed" value)

Video on Frequentism vs Bayesianism



(http://www.youtube.com/watch?feature=player_embedded&v=KhAUfqhLakw)

https://arxiv.org/pdf/1411.5018.pdf (https://arxiv.org/pdf/1411.5018.pdf)

Bayes' rule

The probability of an event A, given event B is written as $P(A|B)$ and given as:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Also, the probability of event B, given event A is:

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

From the first equation we have $P(A \cap B) = P(A|B)P(B)$ and from the second equation we have $P(B \cap A) = P(B|A)P(A)$.
But $P(A \cap B) = P(B \cap A)$, so together we have:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

This is **Bayes' rule**

How is Bayes' rule used in machine learning?

Previously, during discussion of maximum likelihood, we had:

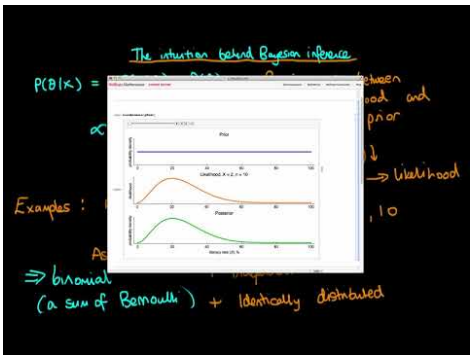
- observed data (training data)
- a probability distribution that we think has generated the data (our model, with parameter θ)

Maximum likelihood proceeded by finding the model parameters $\hat{\theta}$ that maximized $P(data|\theta)$

The Bayesian approach is different:

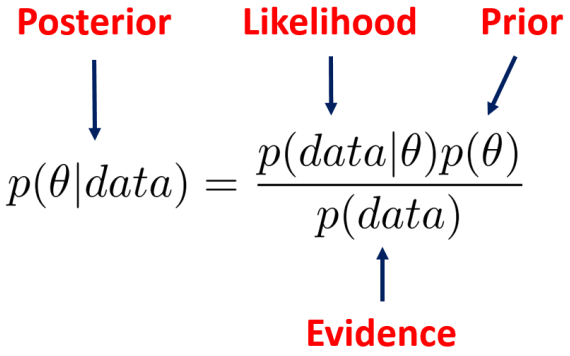
- we are *sure* what our data is, so we model it as "given"
- we are *unsure* what the values of our model parameters θ are (we model θ is a random variable)
- we have a set of *prior* beliefs about the possible values of θ in our model, $P(\theta)$
- but we want to update those beliefs using our observed data
  - we do that using Bayes' rule: $P(\theta|data) = \frac{P(data|\theta)P(\theta)}{P(data)}$

**Video: Explaining the intuition behind Bayesian inference [8:20] **



(https://www.youtube.com/watch?v=yvWlpwnT1nw)

The components of Bayes' rule



Likelihood: the plausibility of the observed data, given the parameters θ

Prior: reflects our belief in the values of θ *before* observing the data

Evidence: Essentially, a normalizing factor, to ensure the Posterior is a probability

Posterior: the result of a Bayesian analysis. The updated belief in the value of θ, given the observed data