Probabilistic Programming

What is Probabilistic Programming

Programming with probability distributions as first class objects.

Makes Bayesian inference possible/accessible.

Bayesian inference - using prior knowledge and data to update beliefs.

Why Bayesian Modeling

Uncertainty estimates

Why Else

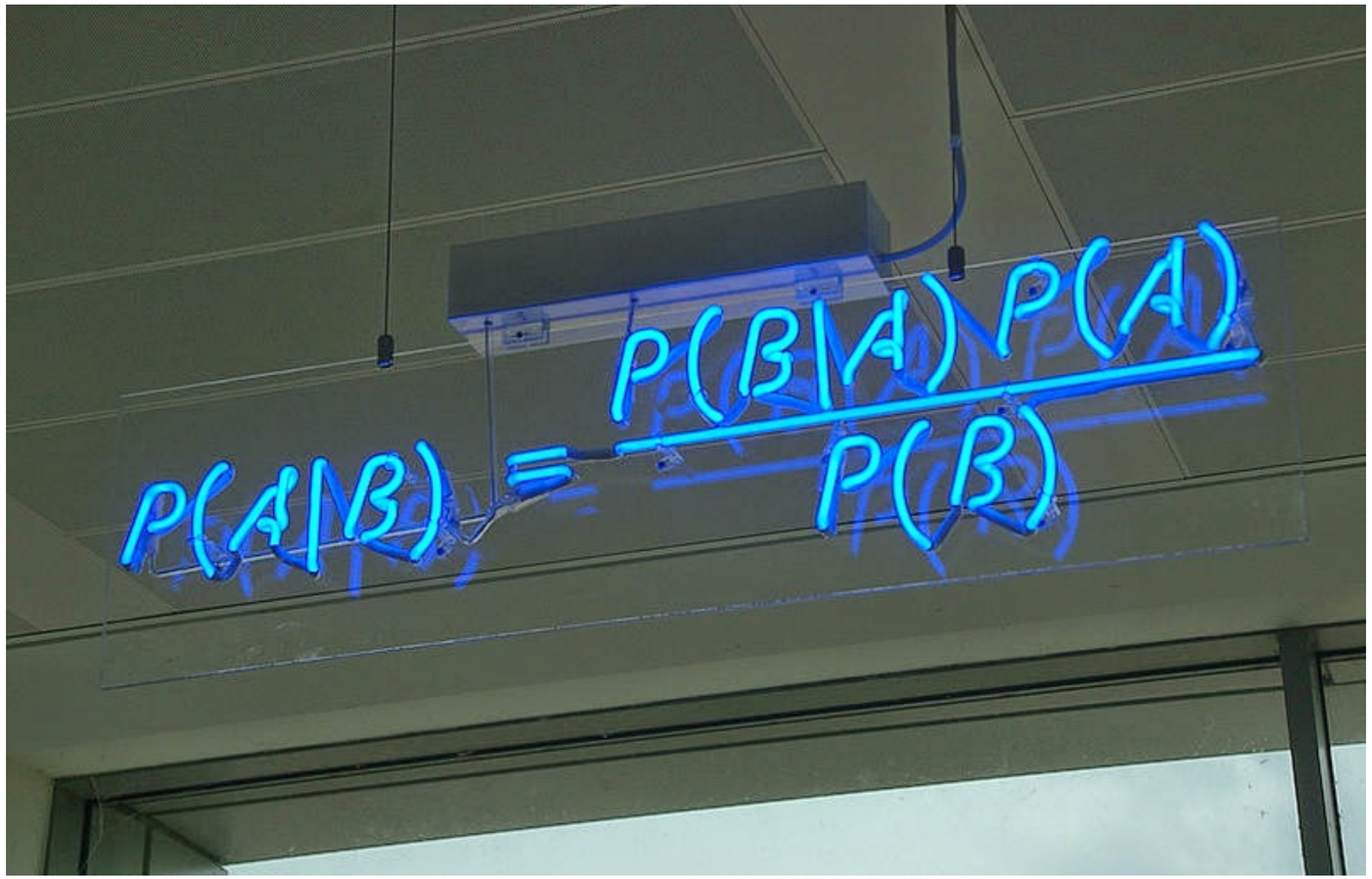
Uncertainty estimates

Structural models - explainable, modular

Incorporate prior/domain knowledge

Useful with less data

Bayes' Theorem



Bayes' Theorem for ML

$$P(\Theta \mid D) = \frac{P(D \mid \Theta) * P(\Theta)}{P(D)}$$

Θ - Model / parameters

D - Observed data

$$Posterior = \frac{Likelihood * Prior}{Evidence}$$

A/B Test Example

Treatment A gets 10 conversions from 30 trials.

Treatment B gets 10 conversions from 25 trials.

Which one is better?

One step at a time - Treatment A

Treatment A gets 10 conversions from 30 trials.

Model - binomial distribution with parameters
- n - number of experiments. Known. 30
- p - probability of success. Unknown.

Data - observed number of conversions. Known. 10

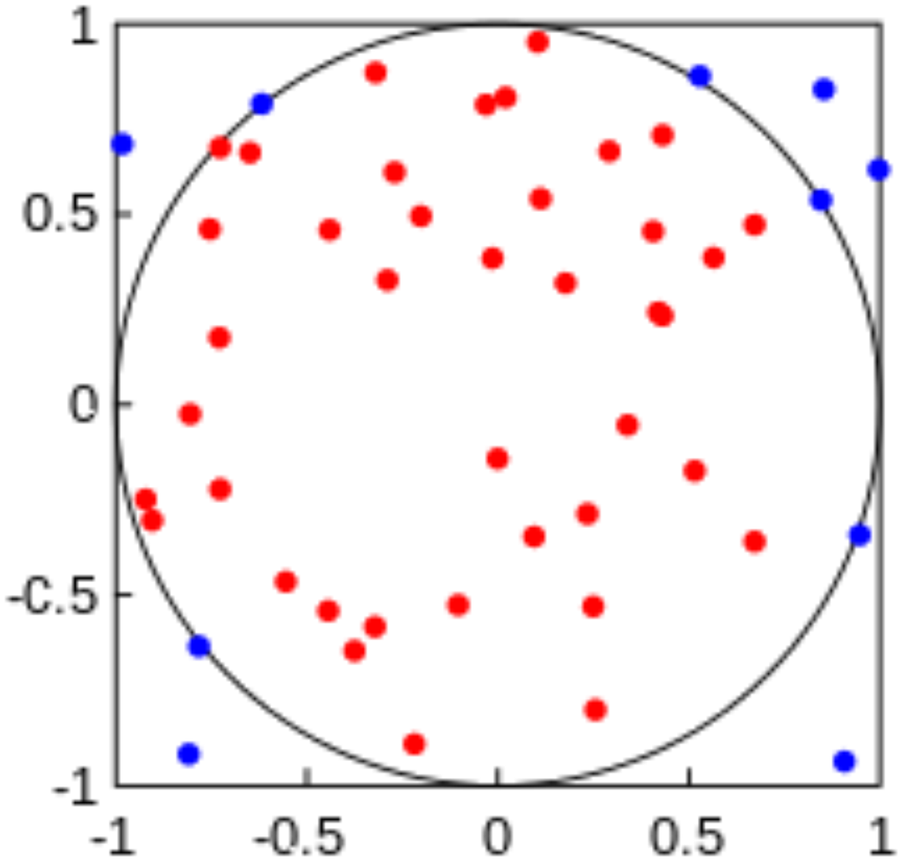
Prior - p could be between 0 and 1.0

Lets take a look at `01_ABTest.ipynb`

Sampling

The process of generating observations that conform to a specified distribution.

Introductory Example - Calculate π



Estimating Conversion Rate

```
def generate_samples(total, num_conversions, num_samples):

    '''Pseudocode to illustrate creating a list of rates that
       generated the observed number of conversions.'''

    trace = []

    while len(trace) < num_samples:
      cur_rate = random.random()

      simulated_num_conversions =
          sum([random.random() < cur_rate for _ in range(total)])

      if simulated_num_conversions == num_conversions
          trace.append(cur_rate)

    return trace
```

Sampling Algorithms

- Rejection sampling / Approximate Bayesian Computation
- Markov Chain Monte Carlo - Metropolis Hastings, Gibbs, Hamiltonian
- NUTS - No U-Turn Sampling
- ADVI - Auto Diff Variational Inference

Probabilistic Programming Frameworks

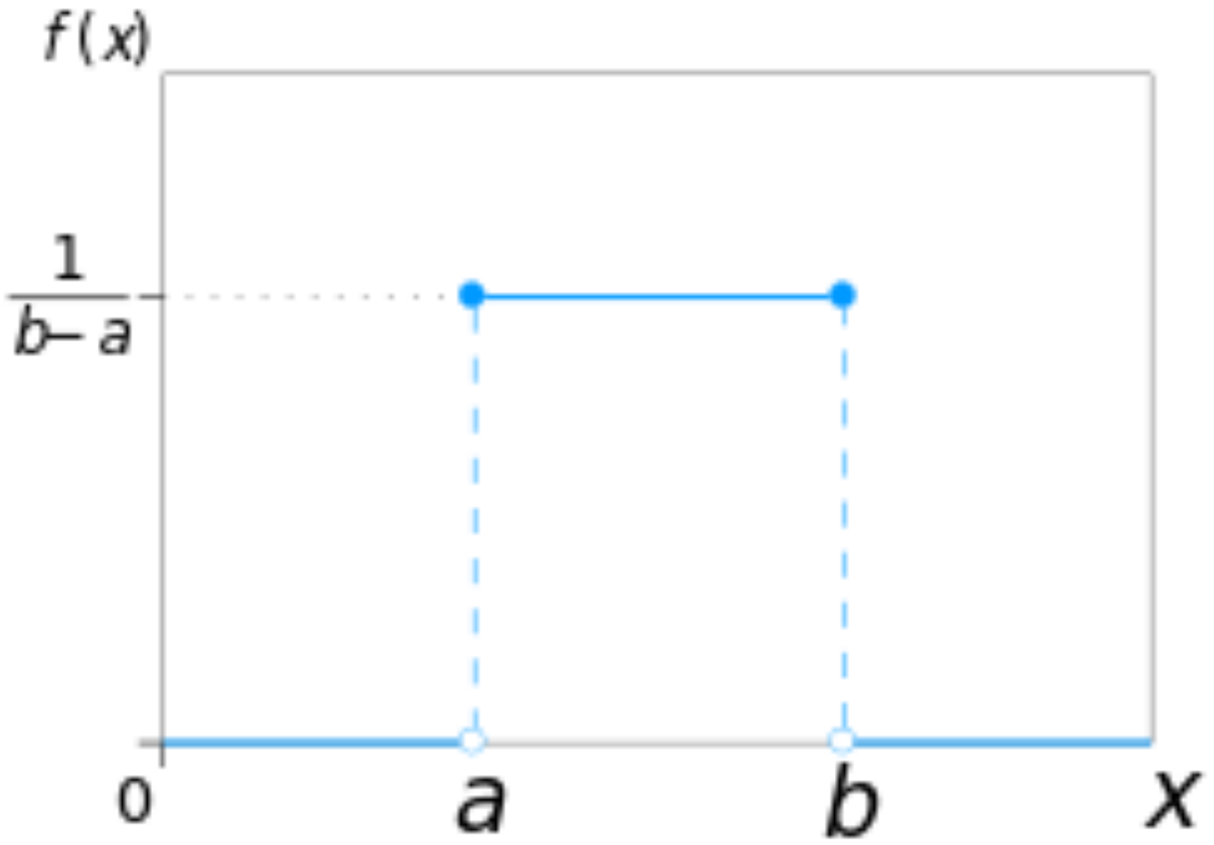
BUGS / JAGS
Stan
PyMC
Edward
Pyro
Web PPL
Figaro, Anglican, BayesDB, ...

Common Distributions

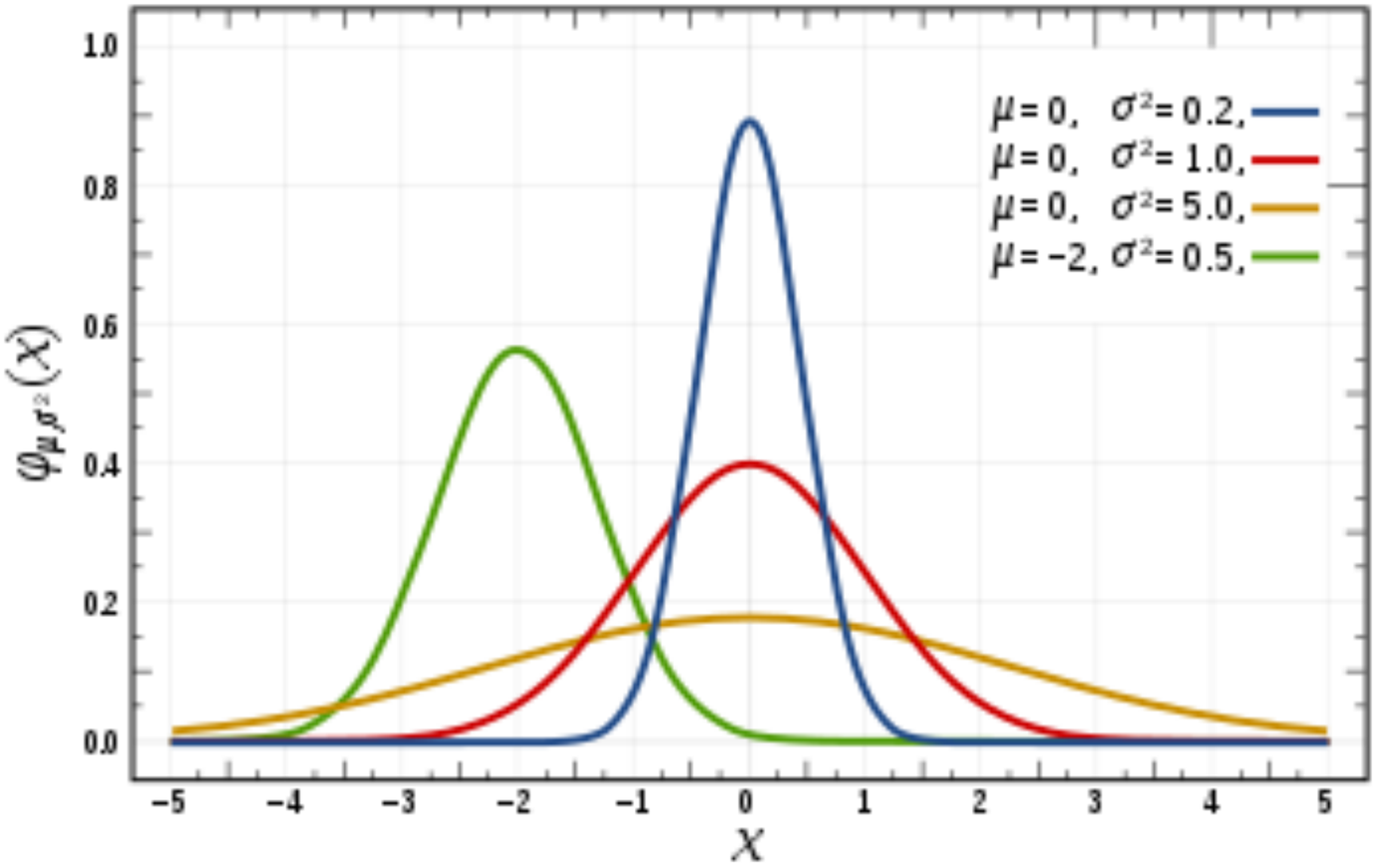
Continous

Discreet

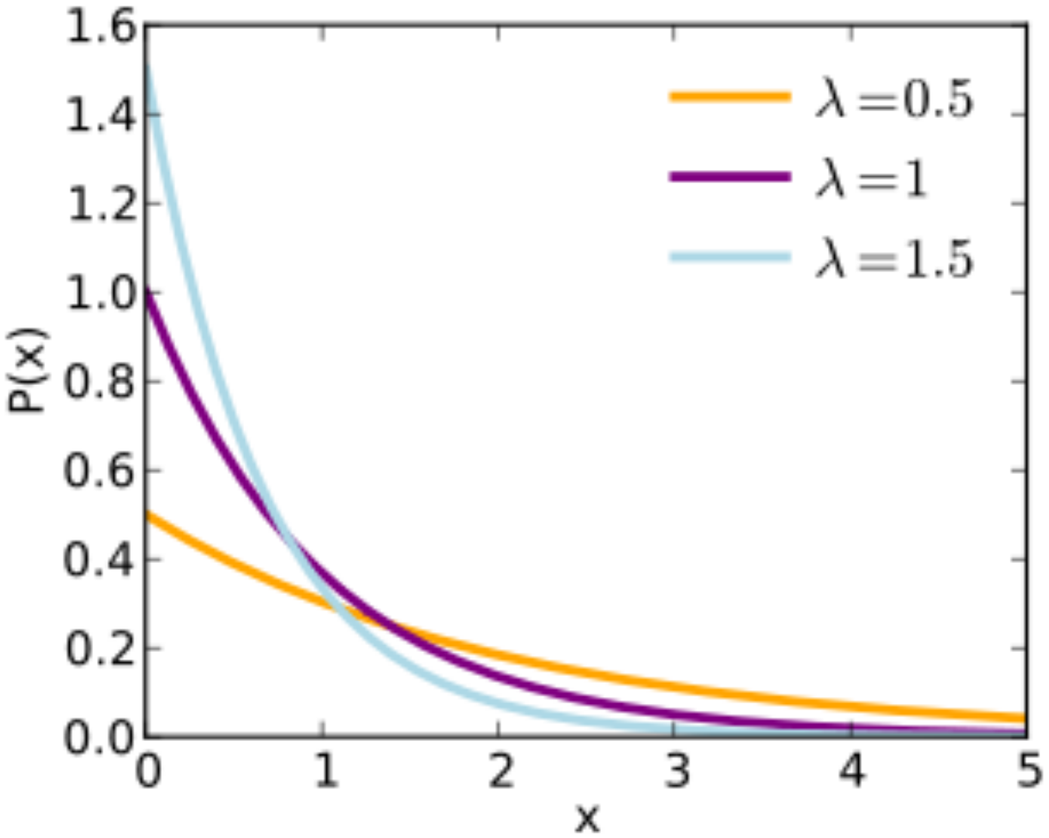
Uniform



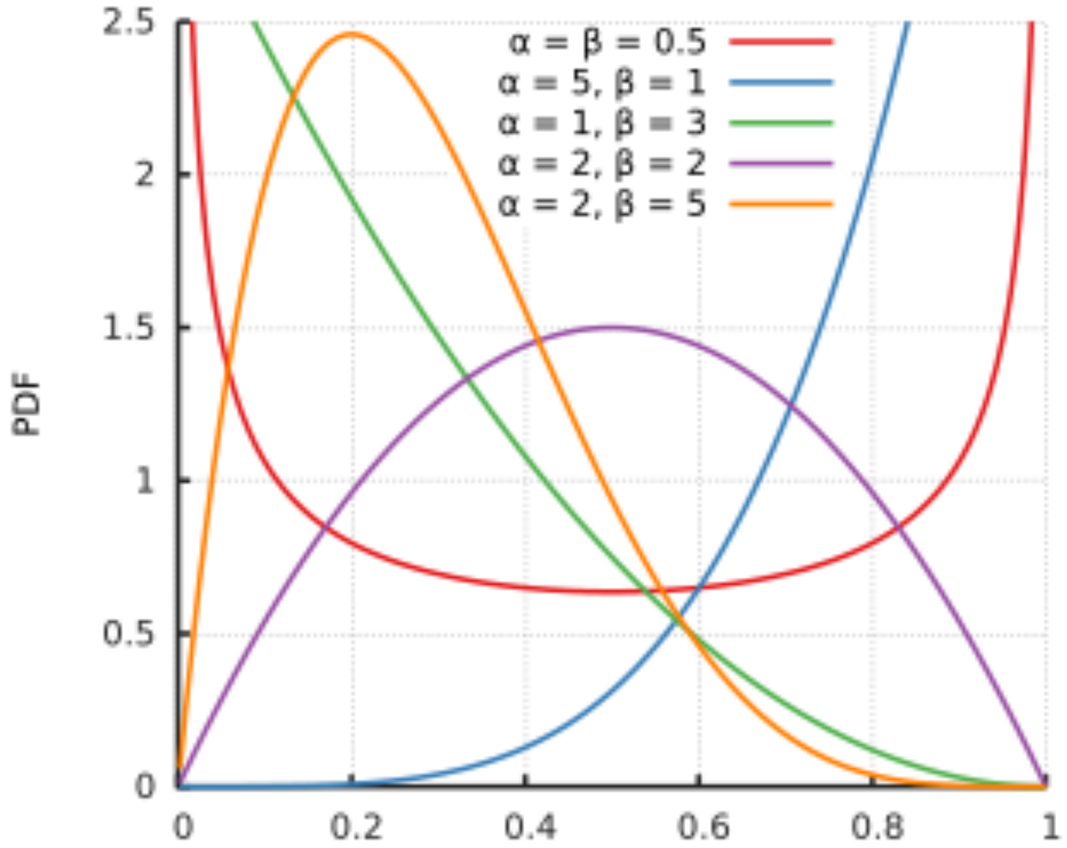
Normal - Gaussian / Half-Gaussian



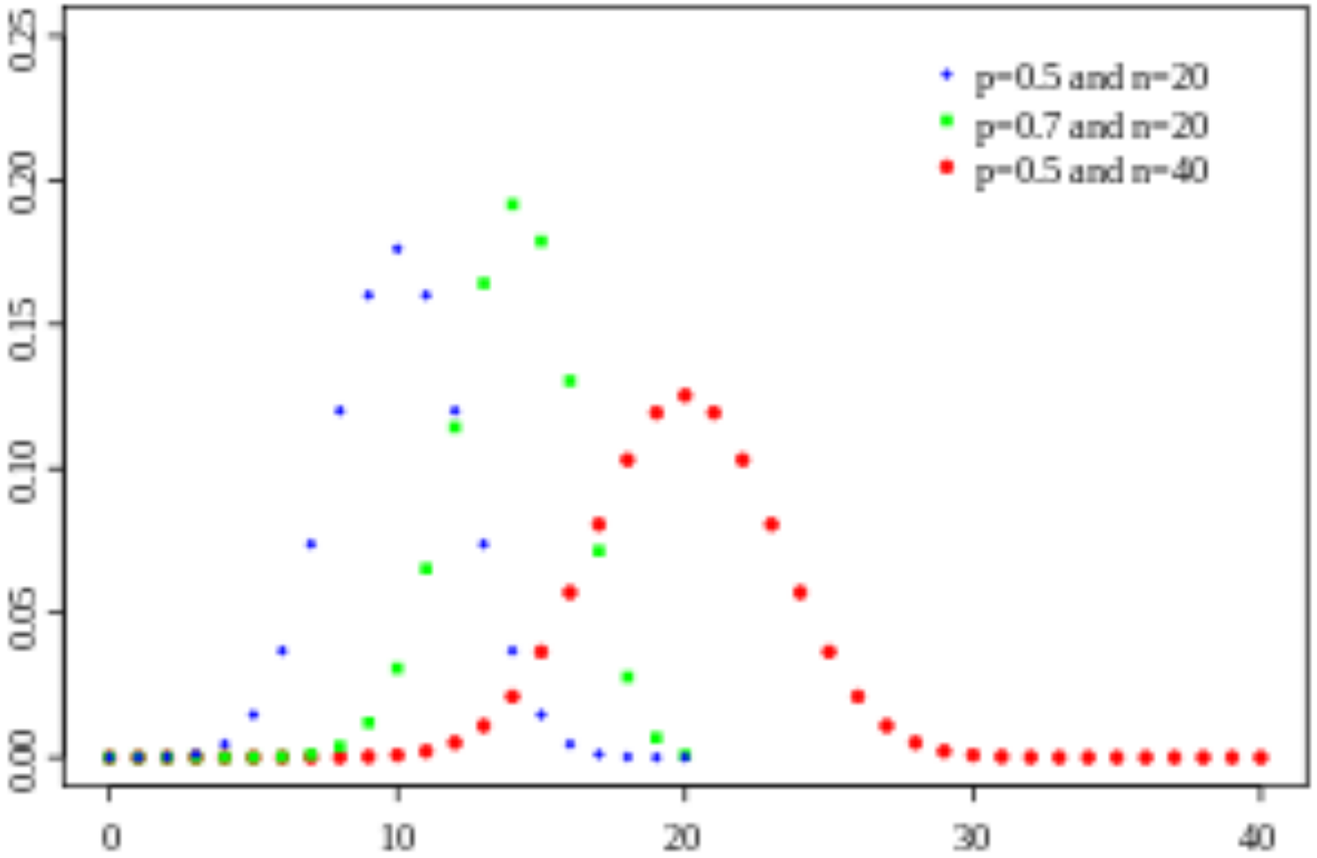
Exponential



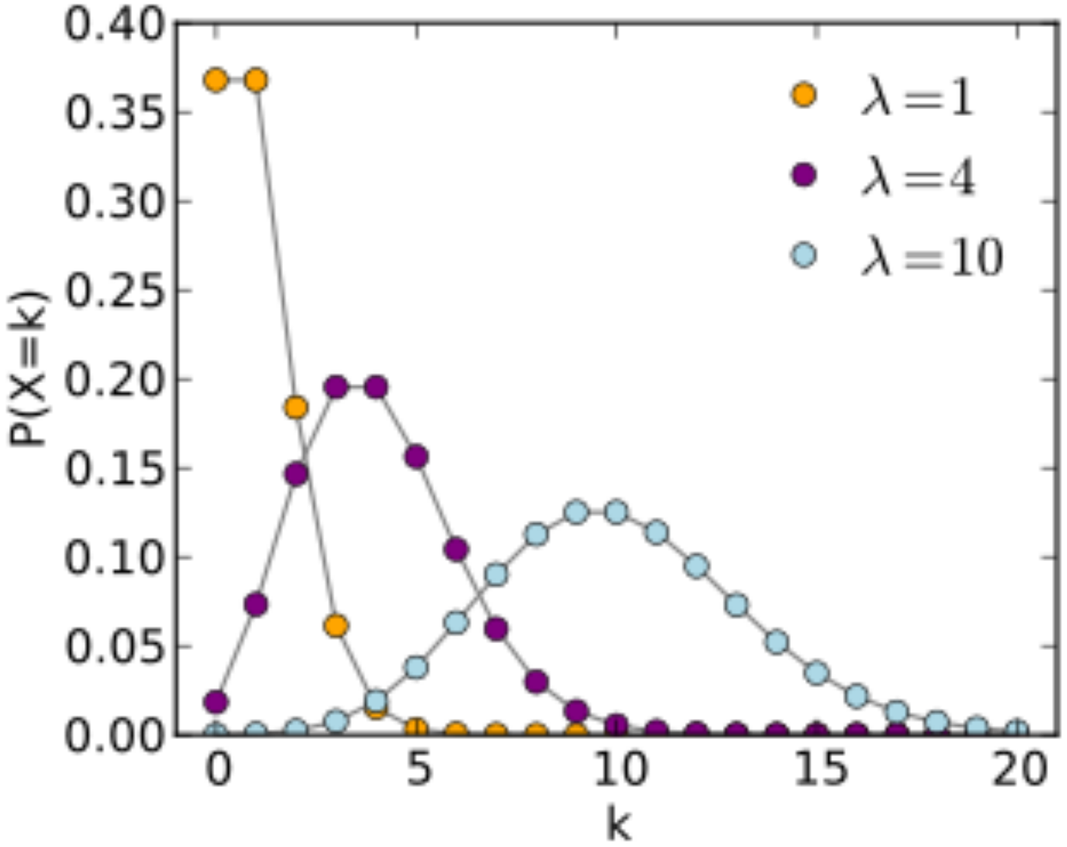
Beta



Bernouli / Binomial



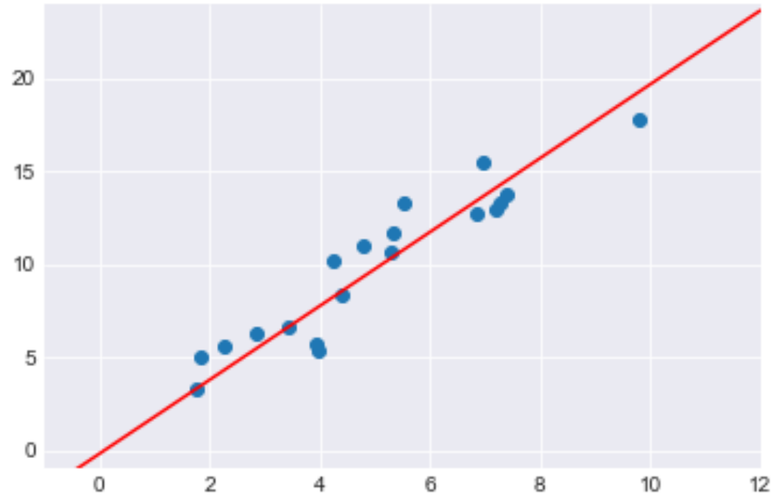
Poisson



Linear Regression Example

$y = m * X + b$

$data \sim N(y,\sigma)$



Lets take a look at `02_Linear.ipynb`

Bayesian Neural Nets

y = activation_fn(w * X + b)

Lets take a look at `03_BayesNets.ipynb`

Summary

Interesting where uncertainty estimates are important.

Bayesians think X is a special case of Bayesian reasoning. Where X is:

- SGD, early stopping, Regularization, Dropout, TTA, ensembles, Neural Nets, ...

(only half joking)

Resources

docs.pymc.io and mc-stan.org

Probabilistic Programming & Bayesian Methods for Hackers, Cam Davidson-Pilon
Bayesian Analysis with Python, Osvaldo Martin

PyMC People: Christopher Fonnesbeck, Thomas Wiecki, Eric Ma, Austin Rochford