Lecture 14

Formal Tests and Gibbs Sampling

Previously

- the bayesian setup
- marginalization and posterior predictives
- globe tossing and beta-binomial
- exchangeability and poisson-gamma
- the coal disasters

Marginalization

Marginal posterior:

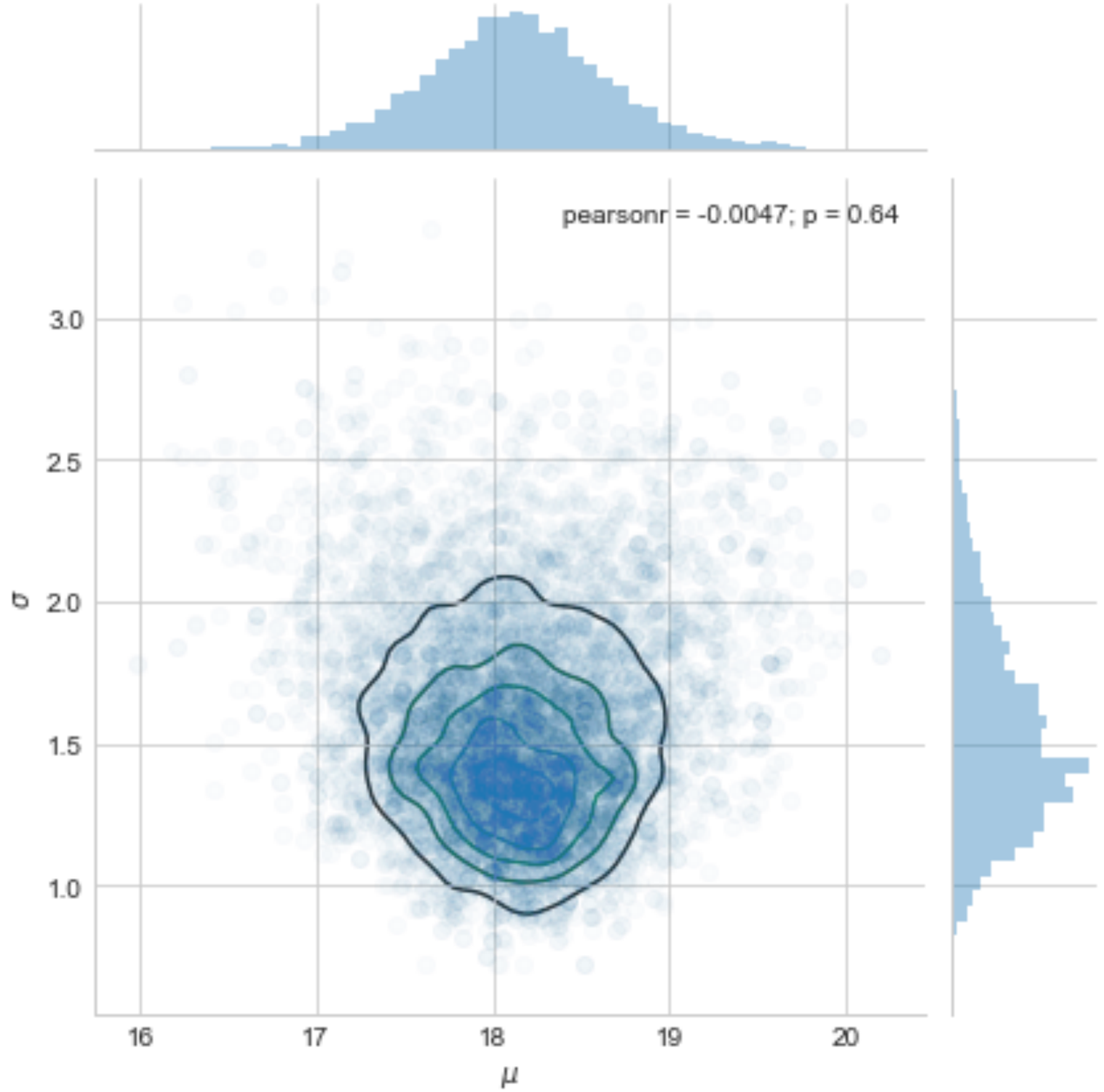
$$p(\theta_1|D) = \int d\theta_{-1} p(\theta|D).$$

```
samps[20000::,:].shape #(10001, 2)
```

```
sns.jointplot(
    pd.Series(samps[20000::,0], name="$\mu$"),
    pd.Series(samps[20000::,1], name="$\sigma$"),
    alpha=0.02)
    .plot_joint(
        sns.kdeplot,
    zorder=0, n_levels=6, alpha=1)
```

Marginals are just 1D histograms

```
plt.hist(samps[20000::,0])
```



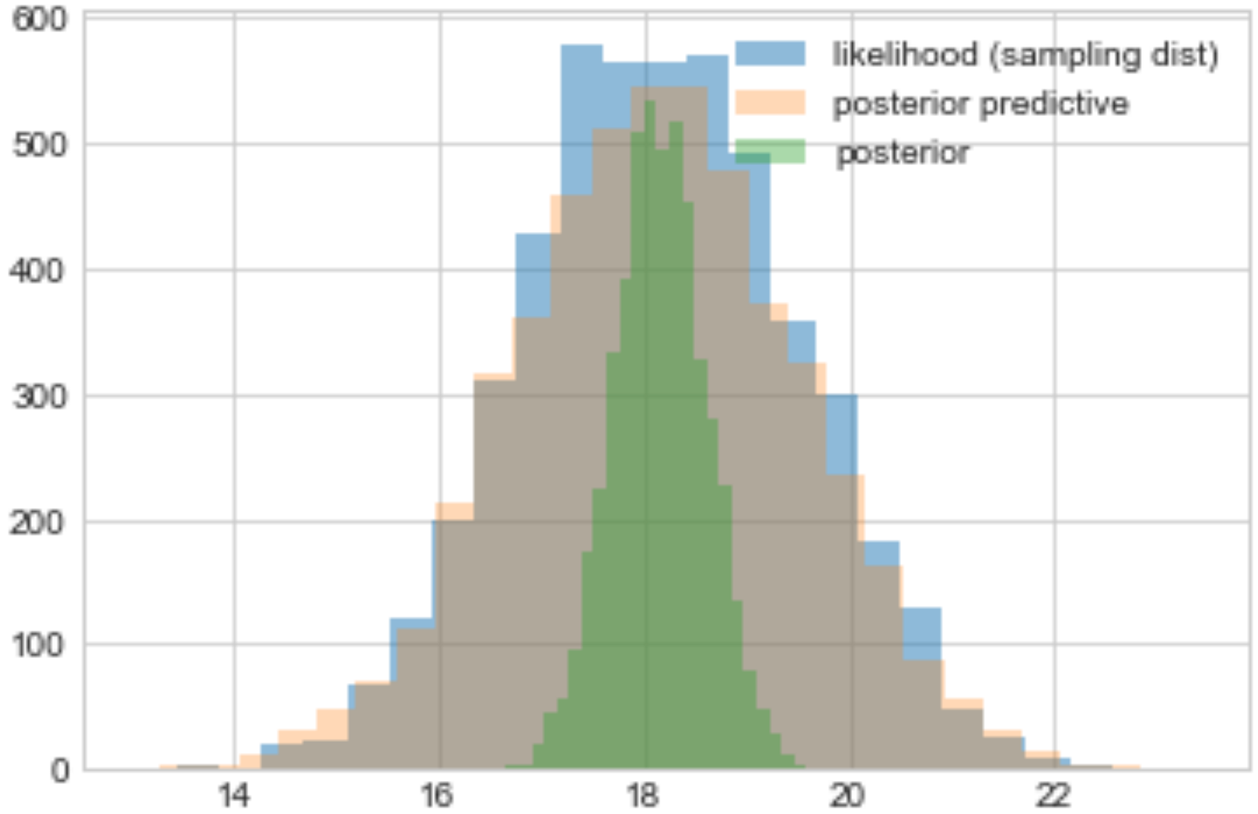
Posterior Predictive

The distribution of a future data point y^*:

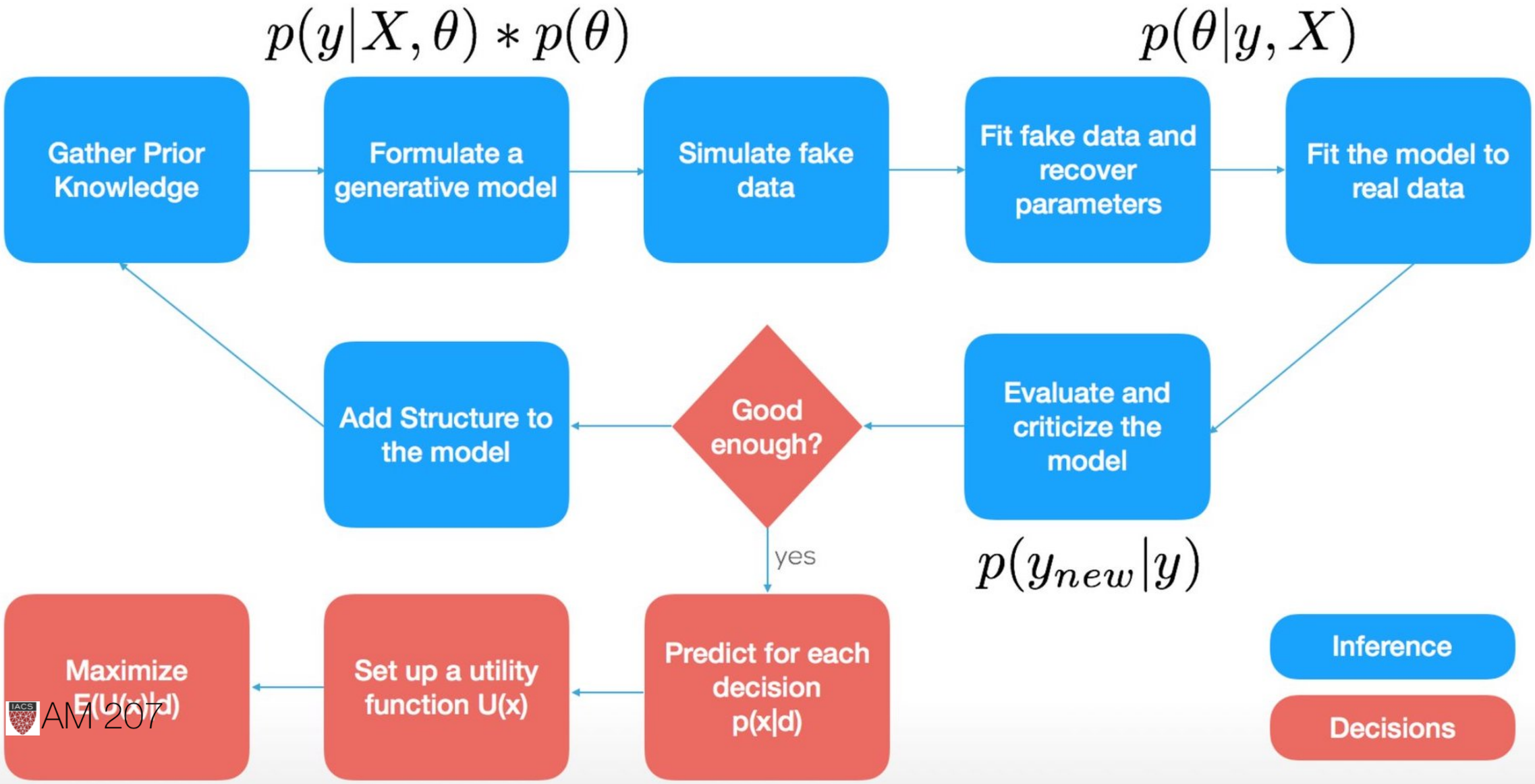
$$p(y^*|D=\{y\}) = E_{p(\theta|D)}[p(y|\theta)]$$
$$= \int d\theta p(y^*|\theta)p(\theta|\{y\}).$$

First draw the thetas from the posterior, then draw y's from the likelihood (these are draws from joint y, θ)

```
post_pred_func = lambda post: norm.rvs(loc = post, scale = sig)
post_pred_samples = post_pred_func(post_samples)
```



Bayesian Workflow
(from @ericnovik)



Exchangeability

Lets assume that the number of children of a women in any one of these classes can me modelled as coming from ONE birth rate.

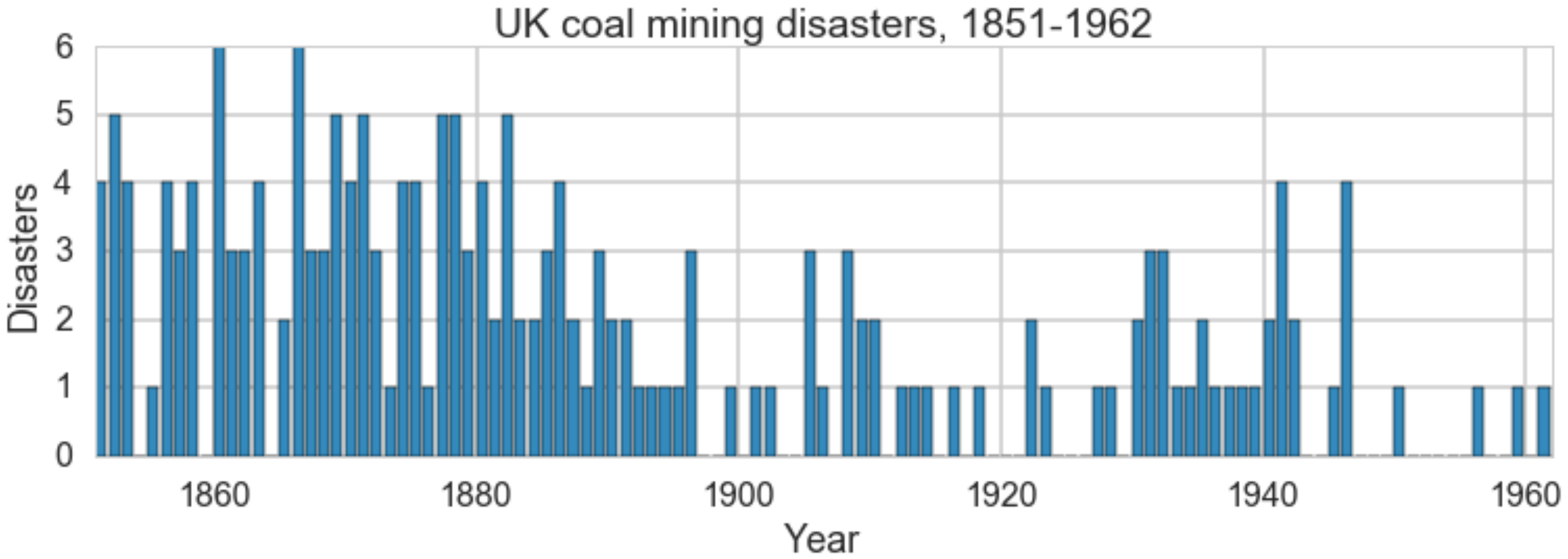
The in-class likelihood for these women is invariant to a permutation of variables.

This is really a statement about what is IID and what is not.

It depends on how much knowledge you have...

Today

- formal convergence criterion (coal disasters)
- convergence paranoia
- gibbs sampling
- hierarchical models
- empirical bayes
- full hierarchical model (in lab)


UK coal mining disasters, 1851-1962
Disasters
0
1
2
3
4
5
6
1860
1880
1900
1920
1940
1960
Year

Model

$$y|\tau, \lambda_1, \lambda_2 \sim Poisson(r_t)$$

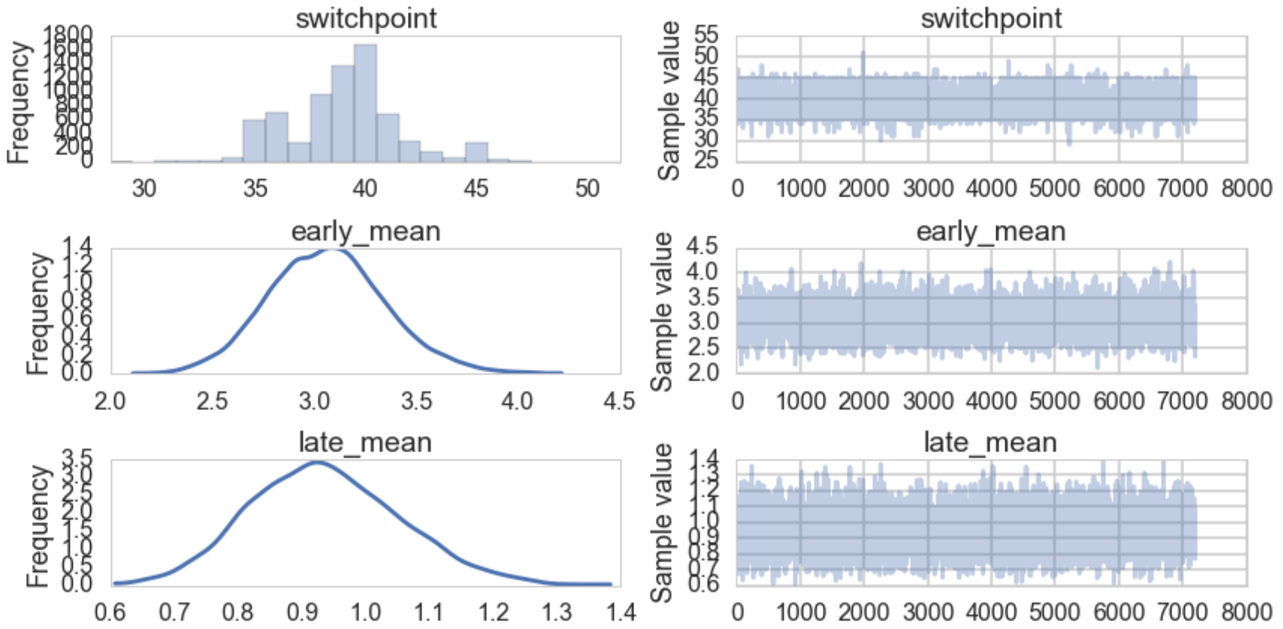
$$r_t = \lambda_1 \text{ if } t < \tau \text{ else } \lambda_2 \text{ for } t \in [t_l, t_h]$$
$$\tau \sim DiscreteUniform(t_l, t_h)$$

$$\lambda_1 \sim Exp(a)$$
$$\lambda_2 \sim Exp(b)$$

```
from pymc3.math import switch
with pm.Model() as coaldis1:
    early_mean = pm.Exponential('early_mean', 1)
    late_mean = pm.Exponential('late_mean', 1)
    switchpoint = pm.DiscreteUniform('switchpoint', lower=0, upper=n_years)
    rate = switch(switchpoint >= np.arange(n_years), early_mean, late_mean)
    disasters = pm.Poisson('disasters', mu=rate, observed=disasters_data)
```

```
with coaldis1:
    stepper=pm.Metropolis()
    trace = pm.sample(40000, step=stepper)
```

```
100%|██████████| 40000/40000 [00:12<00:00, 3326.53it/s] | 229/40000 [00:00<00:17, 2289.39it/s]
```



Imputation

```
>>>disasters_missing = np.array([ 4, 5, 4, 0, 1, 4, 3, 4, 0, 6, 3, 3, 4, 0, 2, 6,
3, 3, 5, 4, 5, 3, 1, 4, 4, 1, 5, 5, 3, 4, 2, 5,
2, 2, 3, 4, 2, 1, 3, -999, 2, 1, 1, 1, 1, 3, 0, 0,
1, 0, 1, 1, 0, 0, 3, 1, 0, 3, 2, 2, 0, 1, 1, 1,
0, 1, 0, 1, 0, 0, 0, 2, 1, 0, 0, 0, 1, 1, 0, 2,
3, 3, 1, -999, 2, 1, 1, 1, 1, 2, 4, 2, 0, 0, 1, 4,
0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1])
>>>disasters_masked = np.ma.masked_values(disasters_missing, value=-999)
```

An array with mask set to True where data is missing.

```python
with pm.Model() as missing_data_model:
    switchpoint = pm.DiscreteUniform('switchpoint', lower=0, upper=len(disasters_masked))
    early_mean = pm.Exponential('early_mean', lam=1.)
    late_mean = pm.Exponential('late_mean', lam=1.)
    idx = np.arange(len(disasters_masked))
    rate = pm.Deterministic('rate', switch(switchpoint >= idx, early_mean, late_mean))
    disasters = pm.Poisson('disasters', rate, observed=disasters_masked)

with missing_data_model:
    stepper=pm.Metropolis()
    trace_missing = pm.sample(10000, step=stepper)

pm.summary(trace_missing, varnames=['disasters_missing'])
```

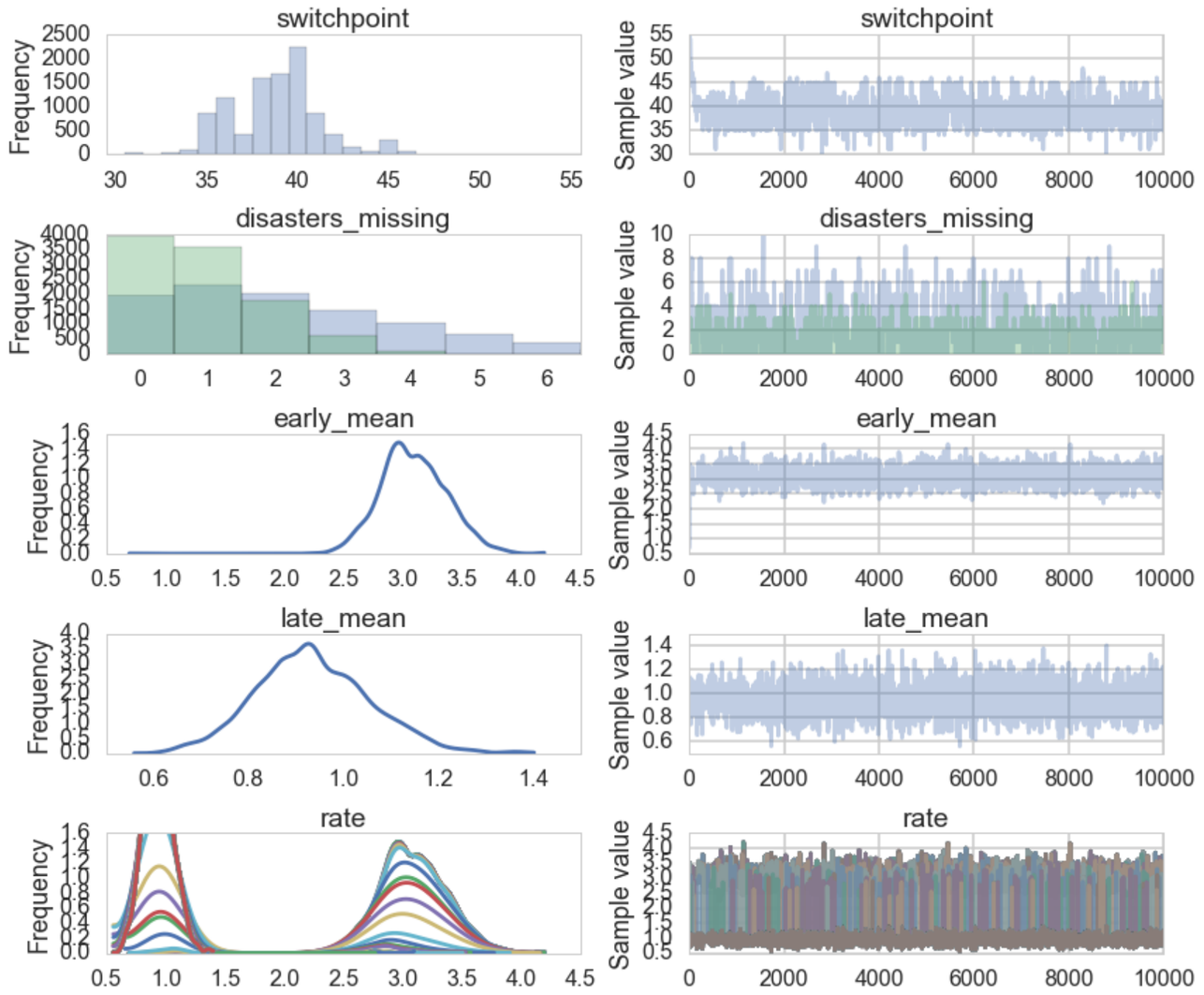
```
disasters_missing:

  Mean             SD               MC Error         95% HPD interval
  ------------------------------------------------------------------


  2.189            1.825            0.078            [0.000, 6.000]
  0.950            0.980            0.028            [0.000, 3.000]

  Posterior quantiles:
  2.5              25               50               75               97.5
  |--------------|===============|===============|--------------|

  0.000            1.000            2.000            3.000            6.000
  0.000            0.000            1.000            2.000            3.000
```

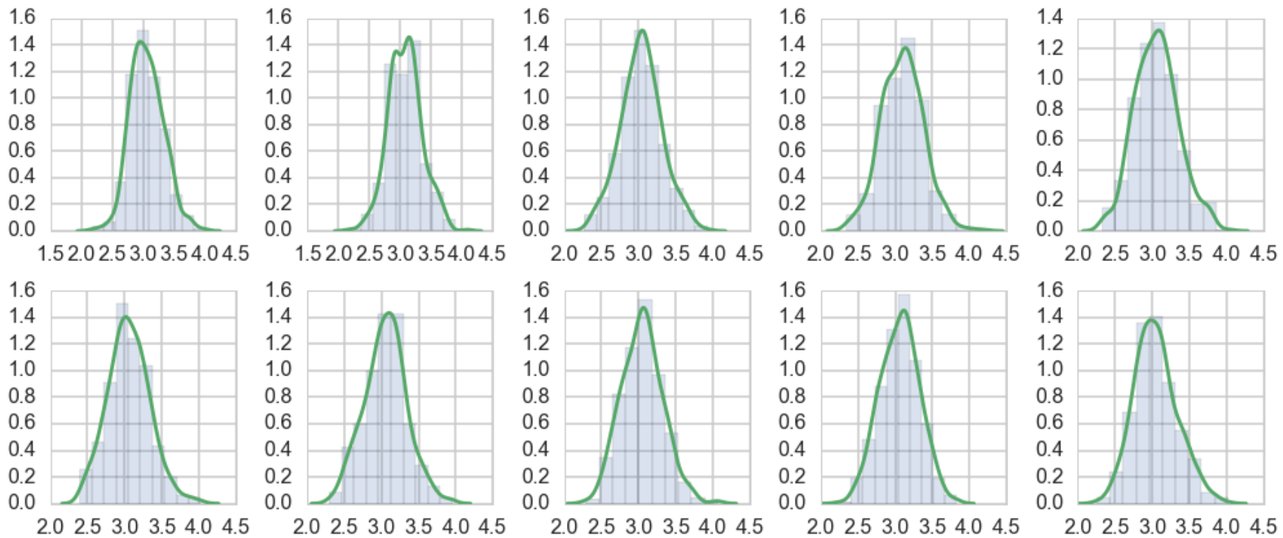


Model convergence

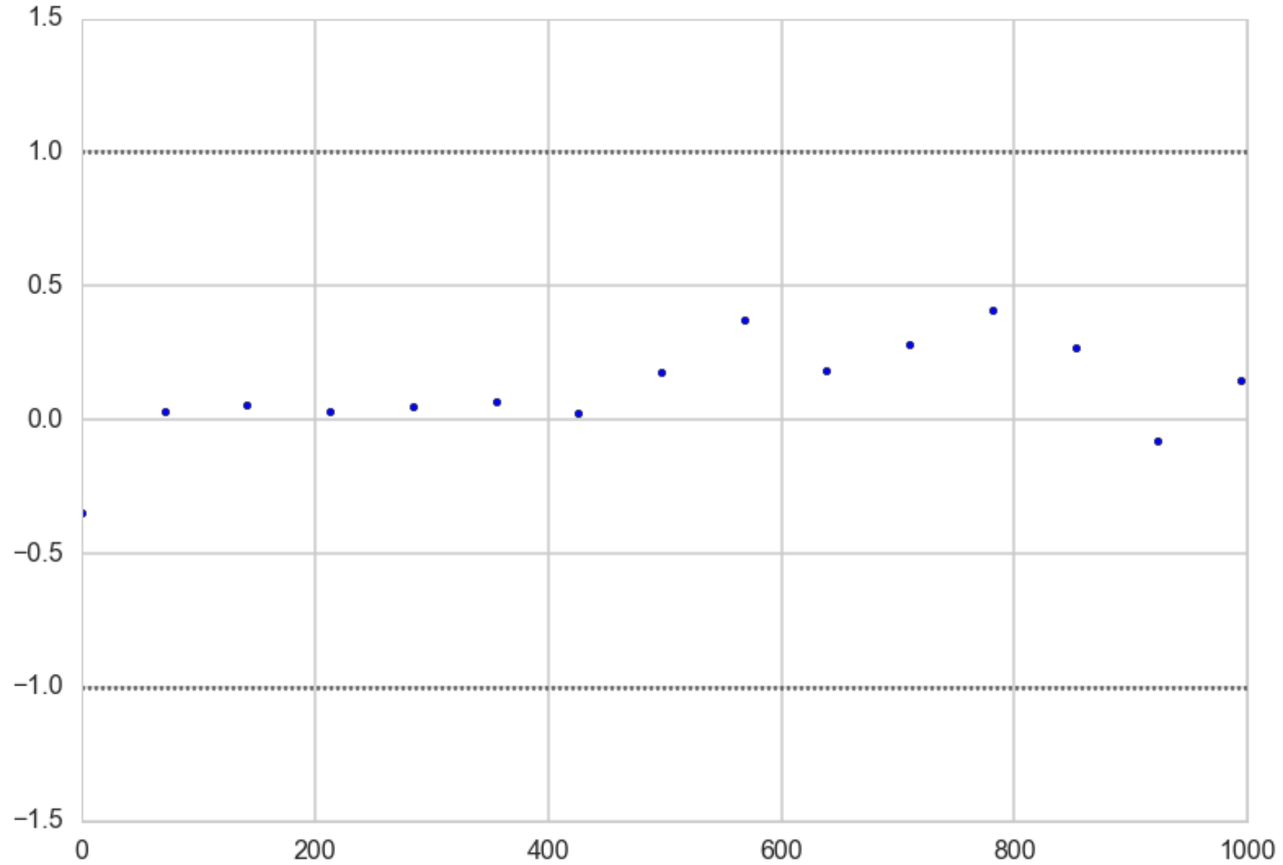
- traces white noisy
- diagnose autocorrelation, check parameter correlations

```
pm.trace_to_dataframe(trace).corr()
```

- visually inspect histogram every m samples
- traceplots from different starting points, different chains
- formal tests: Gewecke, Gelman-Rubin, Effective Sample Size



Gewecke: difference of means



$$H_0 : \mu_{\theta_1} - \mu_{\theta_2} = 0 \implies \mu_{\theta_1 - \theta_2} = 0$$

$$\sigma_{\theta_1 - \theta_2} = \sqrt{\frac{var(\theta_1)}{n_1} + \frac{var(\theta_2)}{n_2}}$$

$$|\mu_{\theta_1} - \mu_{\theta_2}| < 2\sigma_{\theta_1 - \theta_2}$$

```
with coaldis1:
    stepper=pm.Metropolis()
    tr = pm.sample(2000, step=stepper)

z = geweke(tr, intervals=15)

plt.scatter(*z['early_mean'].T)
plt.hlines([-1,1], 0, 1000, linestyles='dotted')
plt.xlim(0, 1000)
```

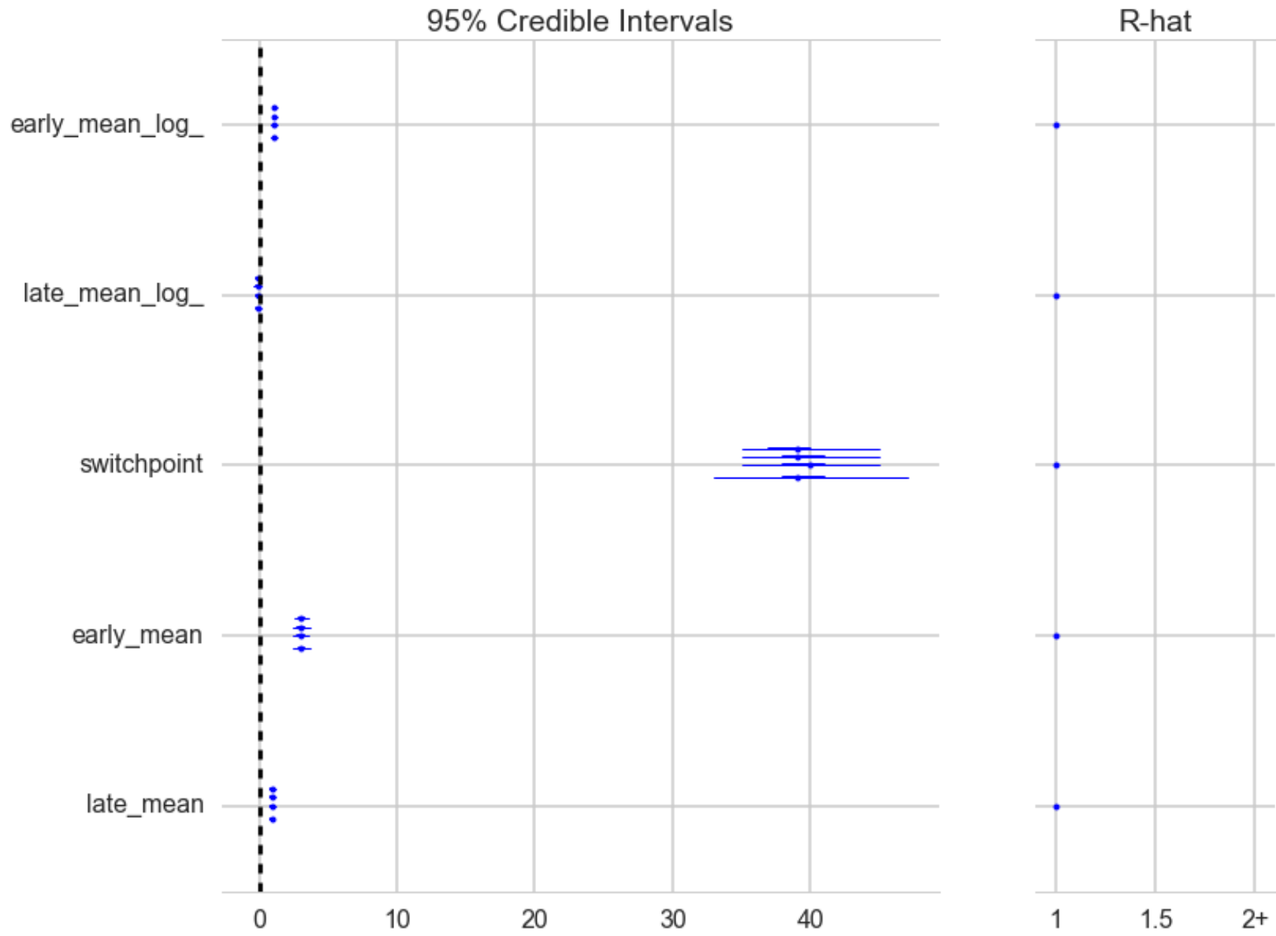
Gelman-Rubin

Multiple chains..compute within chain variance and compare to between chain variance

$$s_j^2 = \frac{1}{n-1}\sum_i (\theta_{ij} - \mu_{\theta_j})^2$$

$$w = \frac{1}{m}\sum_j s_j^2; \;\; \mu = \frac{1}{m}\sum_j \mu_{\theta_j}$$

$$B = \frac{n}{m-1}\sum_j (\mu_{\theta_j} - \mu)^2$$



Use weighted average of w and B to estimate variance of the stationary distribution `pm.gelman_rubin(trace)`:

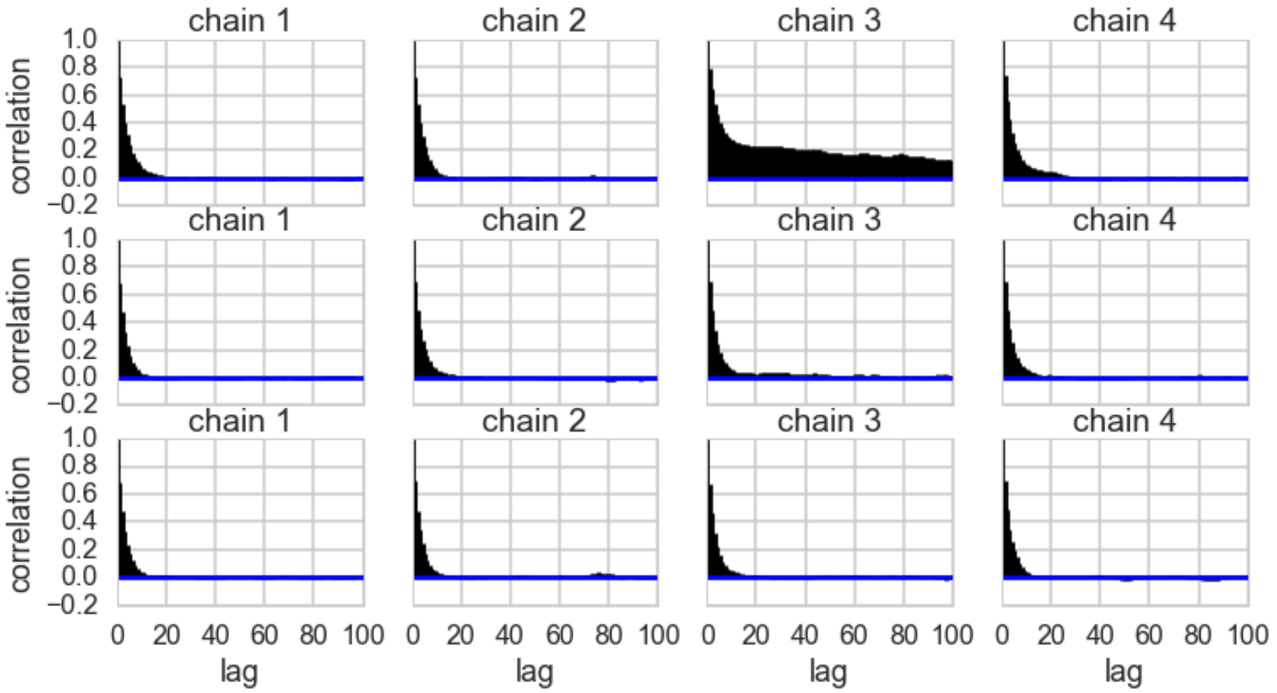
$$\hat{Var}(\theta) = (1 - \frac{1}{n})w + \frac{1}{n}B$$

Overestimates our variance, but unbiased under stationarity.

Ratio of the estimated distribution variance to asymptotic one:

$$\hat{R} = \sqrt{\frac{\hat{Var}(\theta)}{w}}$$

ESS: Effective Sample Size



IIDness of draws decreases

```
pm.effective_n(trace)
```

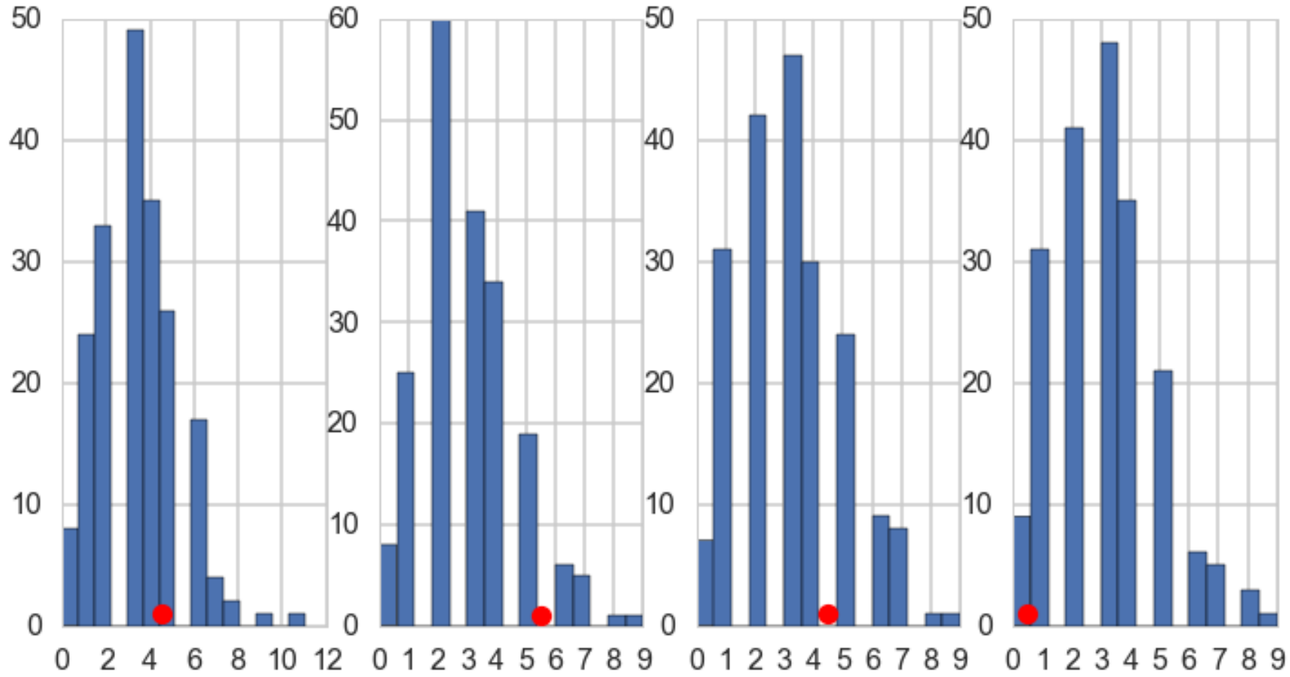
```
{'early_mean': 16857.0,
 'early_mean_log_': 12004.0,
 'late_mean': 27344.0,
 'late_mean_log_': 27195.0,
 'switchpoint': 195.0}
```

(40000 samples)

$$n_{eff} = \frac{mn}{1 + 2\sum_{\Delta t} \rho_{\Delta t}}$$

Posterior Predictive Checks

```
with coaldis1:
    sim = pm.sample_ppc(t2, samples=200)
```



Another sampler issue: Non-Identifiability

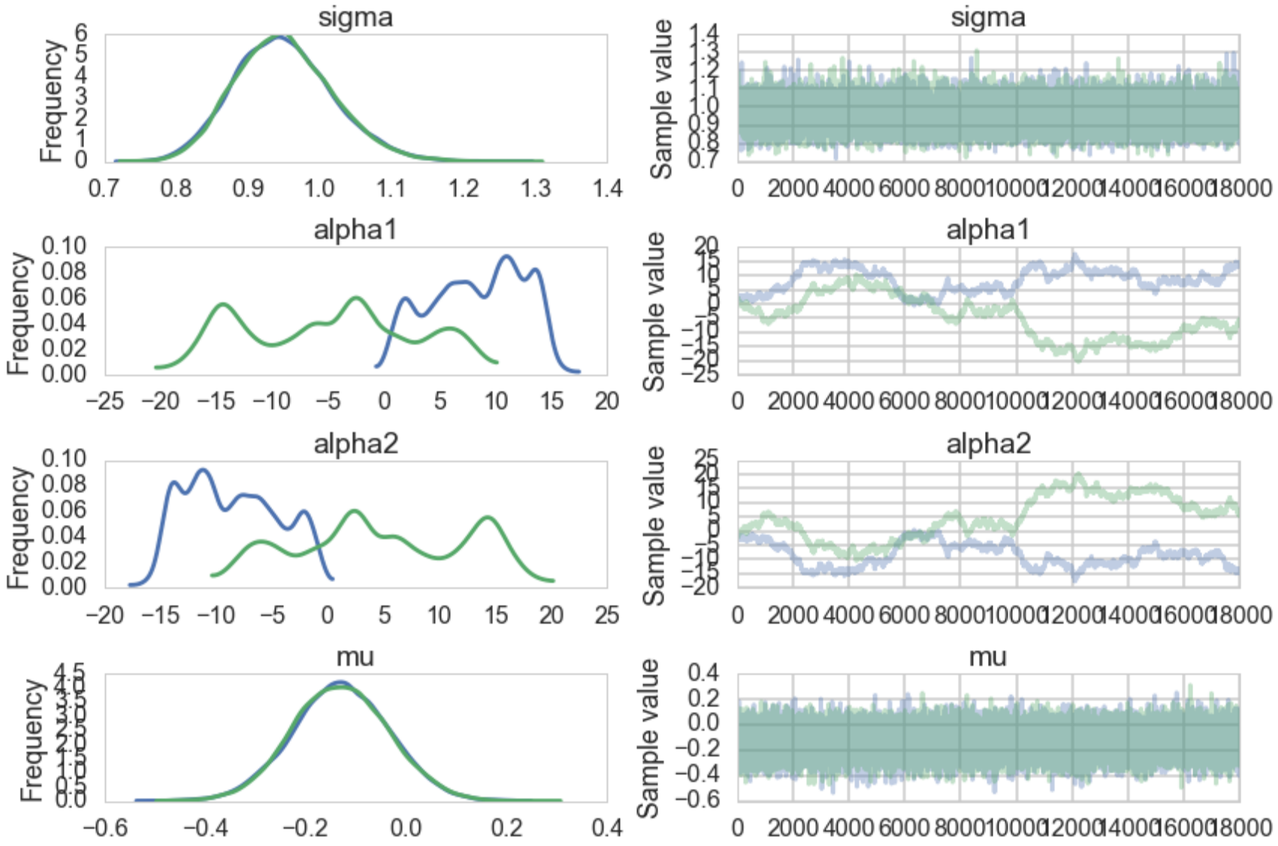
Generate data from N(0,1). Then fit:

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

$$\mu = \alpha_1 + \alpha_2$$

$$\begin{aligned} \alpha_1 &\sim Unif(-\infty, \infty) \\ \alpha_2 &\sim Unif(-\infty, \infty) \\ \sigma &\sim HalfCauchy(0, 1) \end{aligned}$$

Correlation diagnostic




```
sigma = pm.HalfCauchy("sigma", beta=1)
alpha1=pm.Uniform('alpha1', lower=-10**6, upper=10**6)
alpha2=pm.Uniform('alpha2', lower=-10**6, upper=10**6)
mu = pm.Deterministic("mu", alpha1 + alpha2)
y = pm.Normal("data", mu=mu, sd=sigma, observed=data)
```

```
df=pm.trace_to_dataframe(traceni)
df.corr()
```

	sigma	mu	alpha1	alpha2
sigma	1.000000	-0.000115	-0.003153	0.003152
mu	-0.000115	1.000000	0.002844	0.008293
alpha1	-0.003153	0.002844	1.000000	-0.999938
alpha2	0.003152	0.008293	-0.999938	1.000000

```
>>>pm.effective_n(traceni)
{'alpha1': 1.0,
 'alpha1_interval_': 1.0,
 'alpha2': 1.0,
 'alpha2_interval_': 1.0,
 'mu': 26411.0,
 'sigma': 39215.0,
 'sigma_log_': 39301.0}
 >>>pm.gelman_rubin(traceni)
 {'alpha1': 1.7439881580327452,
  'alpha1_interval_': 1.7439881580160093,
  'alpha2': 1.7438626593529831,
  'alpha2_interval_': 1.7438626593368223,
  'mu': 0.9999971018206269 5,
  'sigma': 1.0000248056117549,
  'sigma_log_': 1.0000261752214563}
```

Is autocorrelation bad?

- depends on what you want to do
- this is true for $n_e ff$ in general
- does not matter much for means
- matters for credible intervals as we need tails

Thoughts on Diagnostics

- be paranoid, you only know you have not converged, not if you have
- what if you missed out an entire lobe? Thus multiple chains and multiple starting points.
- check posterior correlations, trace autocorrelation, effective n, the look of the trace, the acceptance rate
- check gewecke and gelman-rubin

Gibbs Sampling and Hierarchical models.

- the idea behind gibbs sampling
- examples of gibbs sampling
- gibbs is an always accepted MH
- hierarchical models as regularizers
- empirical bayes (for rat tumors)
- setting up full bayes for hierarchical models

What did Gibbs do?

He determined the energy states of gases at equilibrium by cycling through all the particles, drawing from each one of them conditionally given the energy levels of the others, taking the time average.

Geman and Geman used this idea to denoise images.

The idea of Gibbs

$$f(x) = \int f(x,y)dy = \int f(x|y)f(y)dy = \int dy f(x|y) \int dx' f(y|x')f(x')$$

Thus: $f(x) = \int h(x,x')f(x')dx'$ integral fixed point equation

where $h(x,x') = \int dy f(x|y)f(y|x')$.

Iterative scheme in which the "transition kernel" $h(x, x')$ is used to create a proposal for metropolis-hastings moves:

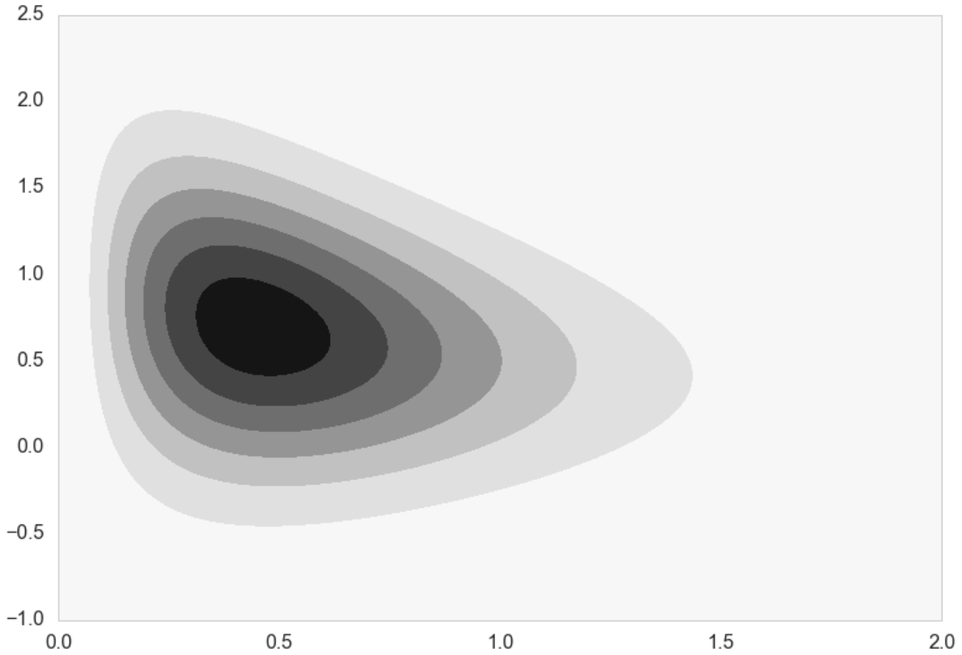
$$f(x_t) = \int h(x_t, x_{t-1}) f(x_{t-1}) dx_{t-1}$$, a Stationary distribution.

$$h(x, x') = \int dy f(x|y) f(y|x')$$.: Sample alternately to get transitions.

Can sample x marginal and $x|y$ so can sample the joint x, y.

Example

Sample from $f(x, y) = x^2 \exp[-xy^2 - y^2 + 2y - 4x]$



Conditionals

$$f(x,y) = x^2 \exp[-xy^2 - y^2 + 2y - 4x] = x^2 \exp[-x(y^2+4)] \exp[-y^2 + 2y]$$

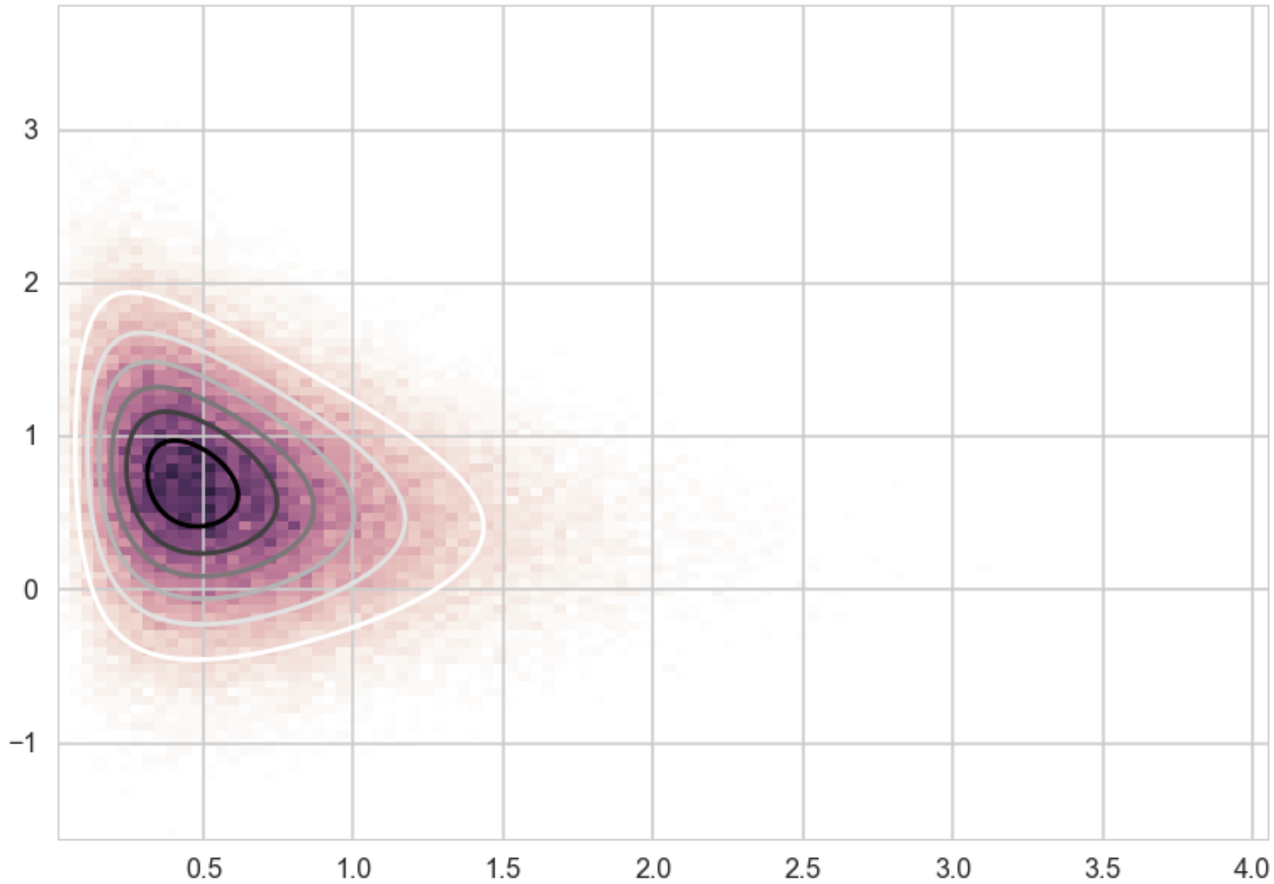
$$= g(y) \text{Gamma}(3, y^2+4) \implies f(x|y) = \text{Gamma}(3, y^2+4)$$

$$f(x,y) = x^2 \exp[-y^2(1+x) + 2y] \exp[-4x]$$

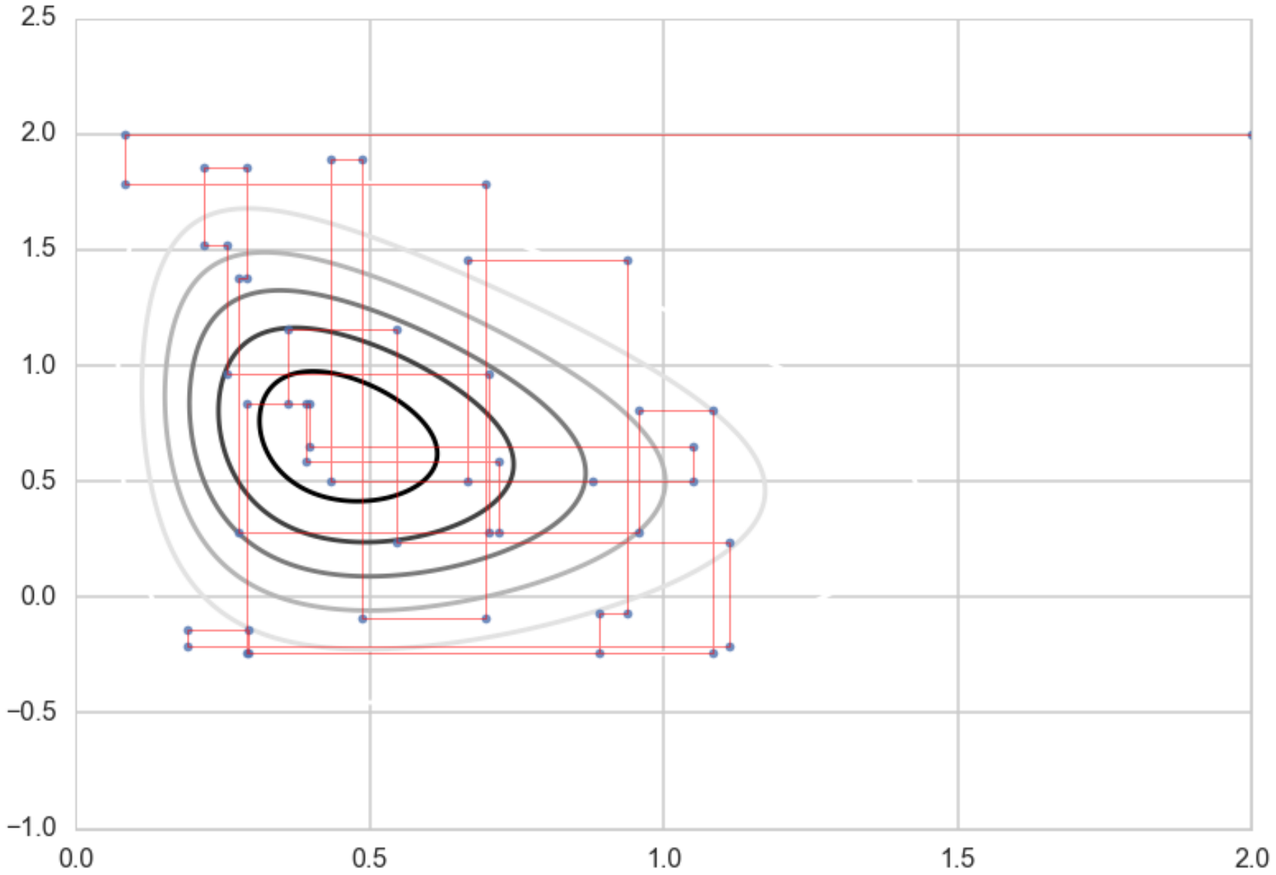
$$\implies f(y|x) = N(\frac{1}{1+x}, \frac{1}{\sqrt{(2(1+x))}})$$

Sampler

```
def xcond(y):
    return gamma.rvs(3, scale=1/(y*y + 4))
def ycond(x):
    return norm.rvs(1/(1+x), scale=1.0/np.sqrt(2*(x+1)))
def gibbs(xgiveny_sample, ygivenx_sample, N, start = [0,0]):
    x=start[0]
    y=start[1]
    samples=np.zeros((N+1, 2))
    samples[0,0]=x
    samples[0,1]=y
    for i in range(1,N,2):
        x=xgiveny_sample(y)
        samples[i,0]=x
        samples[i, 1]=y
        ######################
        y=ygivenx_sample(x)
        samples[i+1,0]=x
        samples[i+1,1]=y
    return samples
out=gibbs(xcond, ycond, 100000)
```

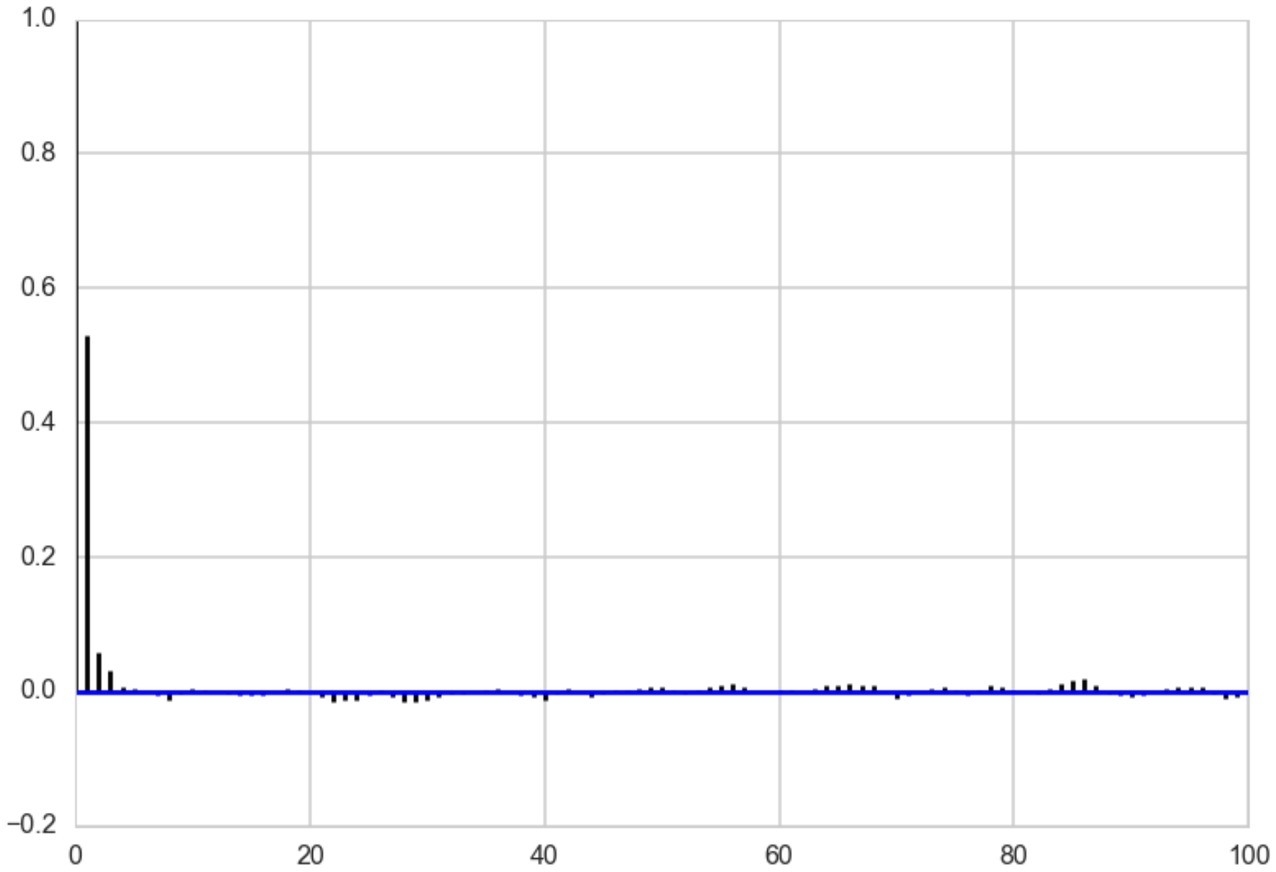


More about gibbs



- easiest is to know how to sample directly from conditionals: **no need for locality**
- moves one component (or one block) at a time
- all is not lost if thats not the case: can use a MH-step once stationarity has been reached
- this makes gibbs a very general idea

Autocorrelation



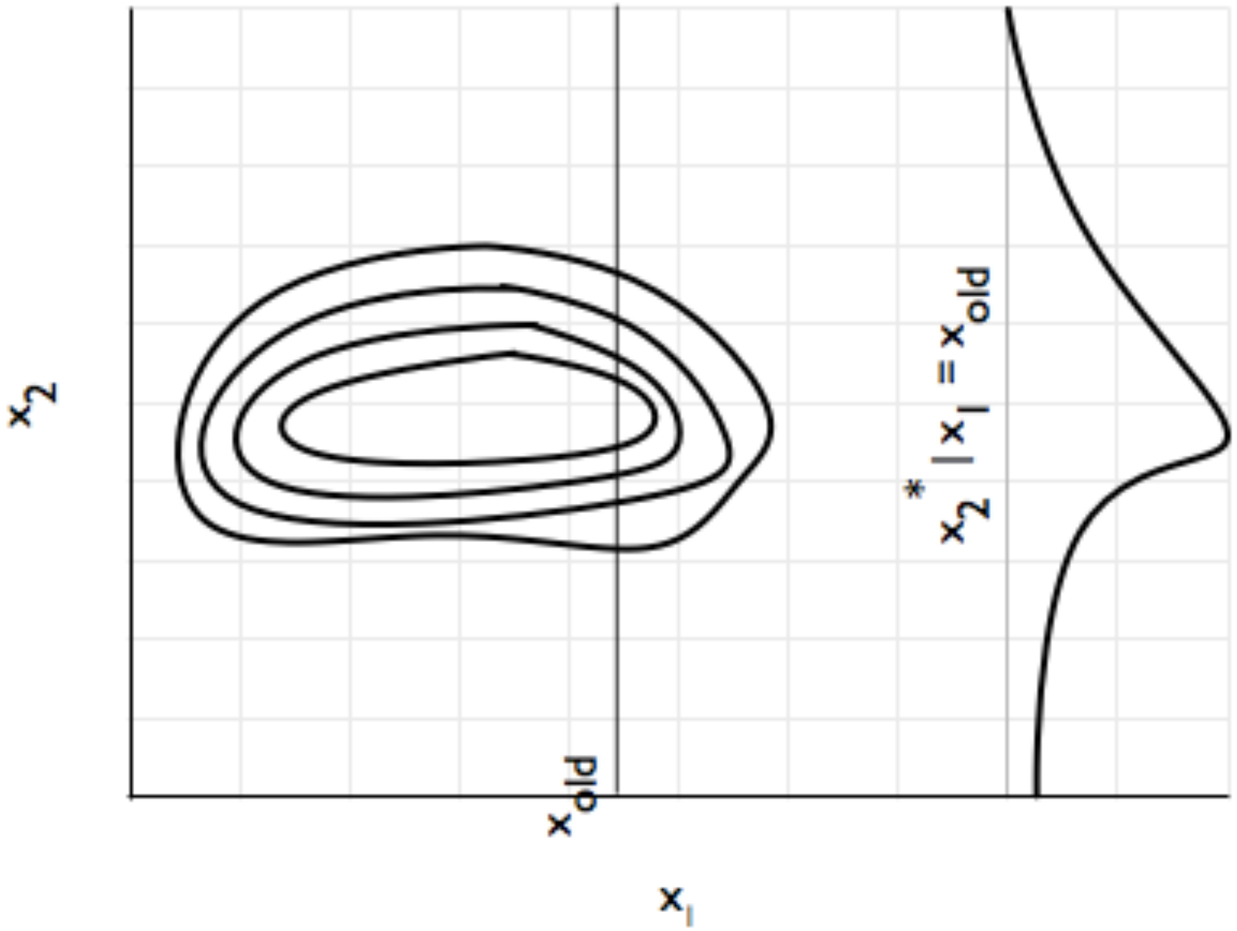
- this joint has very little autocorrelation
- highly correlated joints will have lots of autocorrelation
- thinning/longer chains may be required, but as usual it depends on what you are trying to calculate.
- expectations require far fewer samples
- complete posterior characterization require many more

More Gibbs Theory

The transition kernel corresponds to this proposal:

$$q_k(x^*|x^i) = \begin{cases} p(x^*_k|x^i_{-k}) & for\ x^*_{-k} = x^i_{-k}, \\ 0 & otherwise \end{cases}$$

where x^i_k is the kth component (or block) of x at ith step, while x^i_{-k} is all other components of x at the same step



Gibbs=MH with no rejection

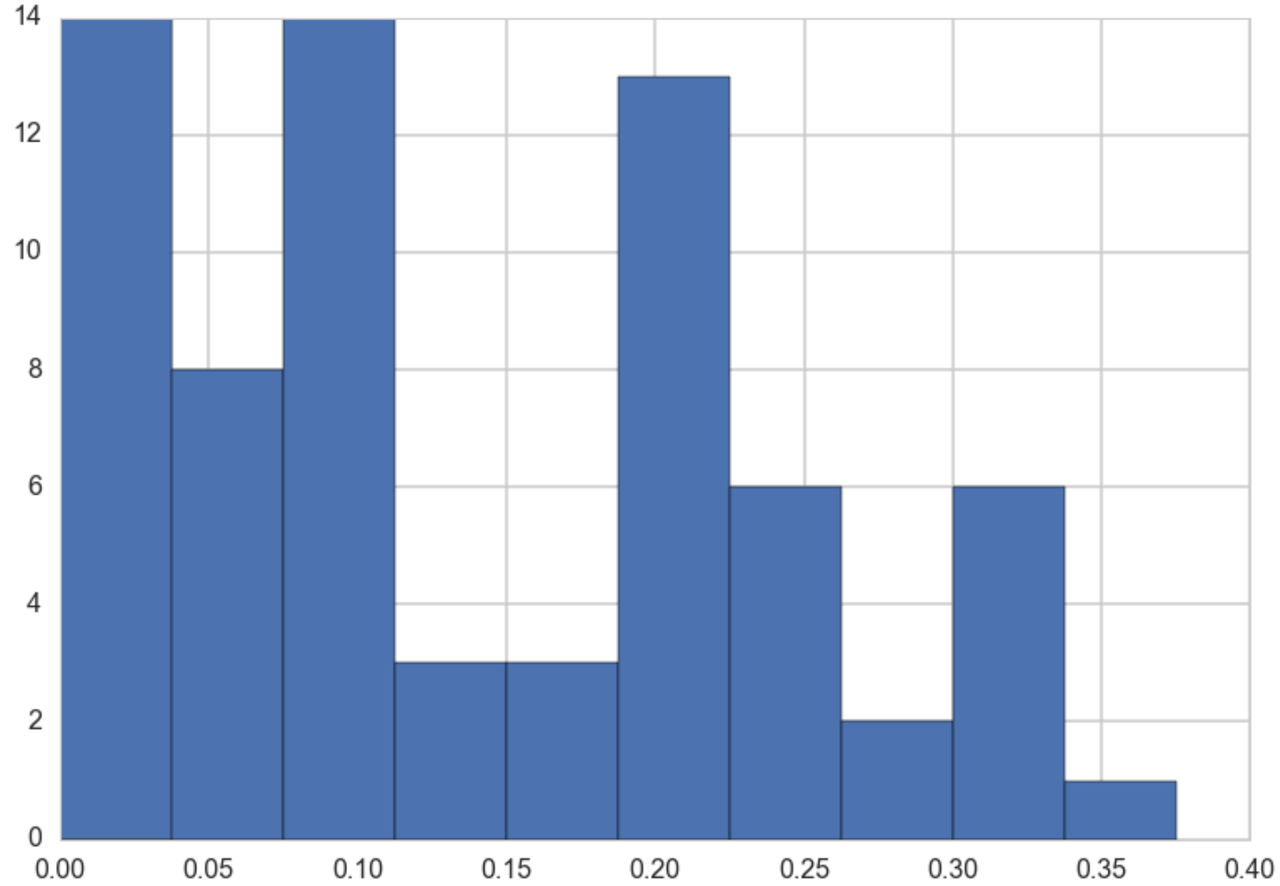
$$A = min(1, \frac{p(x^*)}{p(x^i)} \frac{q_k(x^i|x^*)}{q_k(x^*|x^i)})$$

$$p(x^*) = p(x^*_{-k}, x^*_k) = p(x^*_k|x^*_{-k})p(x^*_{-k})$$

$$A = min(1, \frac{p(x^*_k|x^*_{-k})p(x^*_{-k})}{p(x^i_k|x^i_{-k})p(x^i_{-k})} \frac{q_k(x^i|x^*)}{q_k(x^*|x^i)}) = min(1, \frac{p(x^*_k|x^*_{-k})p(x^*_{-k})}{p(x^i_k|x^i_{-k})p(x^i_{-k})} \frac{p(x^i_k|x^*_{-k})}{p(x^*_k|x^i_{-k})})$$

Componentwise update, $\implies x^*_{-k} = x^i_{-k}$ and A is 1!

Rat Tumors



- tumors in female rats of type "F344" that recieve a particular drug, in 70 different experiments.
- mean and variance of tumor incidence: `0.13600653889043893`, `0.010557640623609196`
- 71st experiment done: 4 out of 14 rats develop tumors. Estimate the risk of tumor in the rats in the 71st experiment

Modeling

$$p(y_i|\theta_i; n_i) = Binom(n_i, y_i, \theta_i)$$

$$p(Y|\Theta; \{n_i\}) = \prod_{i=1}^{70} Binom(n_i, y_i, \theta_i)$$

Need to choose a prior $p(\Theta)$.

No Pooling

Separate priors on each θ_i:

$$\theta_i \sim Beta(\alpha_i, \beta_i).$$

$$p(\Theta|\{\alpha_i\}, \{\beta_i\}) = \prod_{i=1}^{70} Beta(\theta_i, \alpha_i, \beta_i),$$

Very overfit model with 210 parameters. VARIANCE!

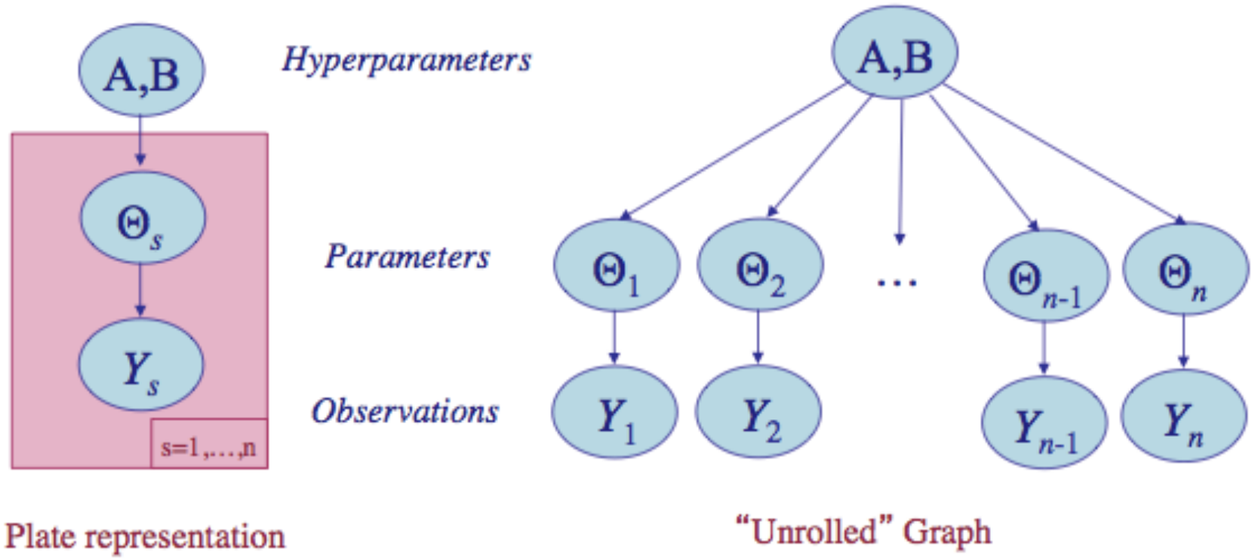
Full Pooling

Assume that there is only one θ in the problem, and set an prior on it.

Ignores any variation amongst the sampling units other than sampling variance.

Underfit model with 3 params. BIAS

Partial pooling: Hierarchical Model



θ_is drawn from "population distribution" given by a conjugate Beta prior $Beta(\alpha, \beta)$ with **hyperparameters** α and β.

$$\theta_i \sim Beta(\alpha, \beta).$$

$$p(\Theta|\alpha, \beta) = \prod_{i=1}^{70} Beta(\theta_i, \alpha, \beta).$$

Priors from data

Where do α and β come from?

Why are we calling them hyperparameters?

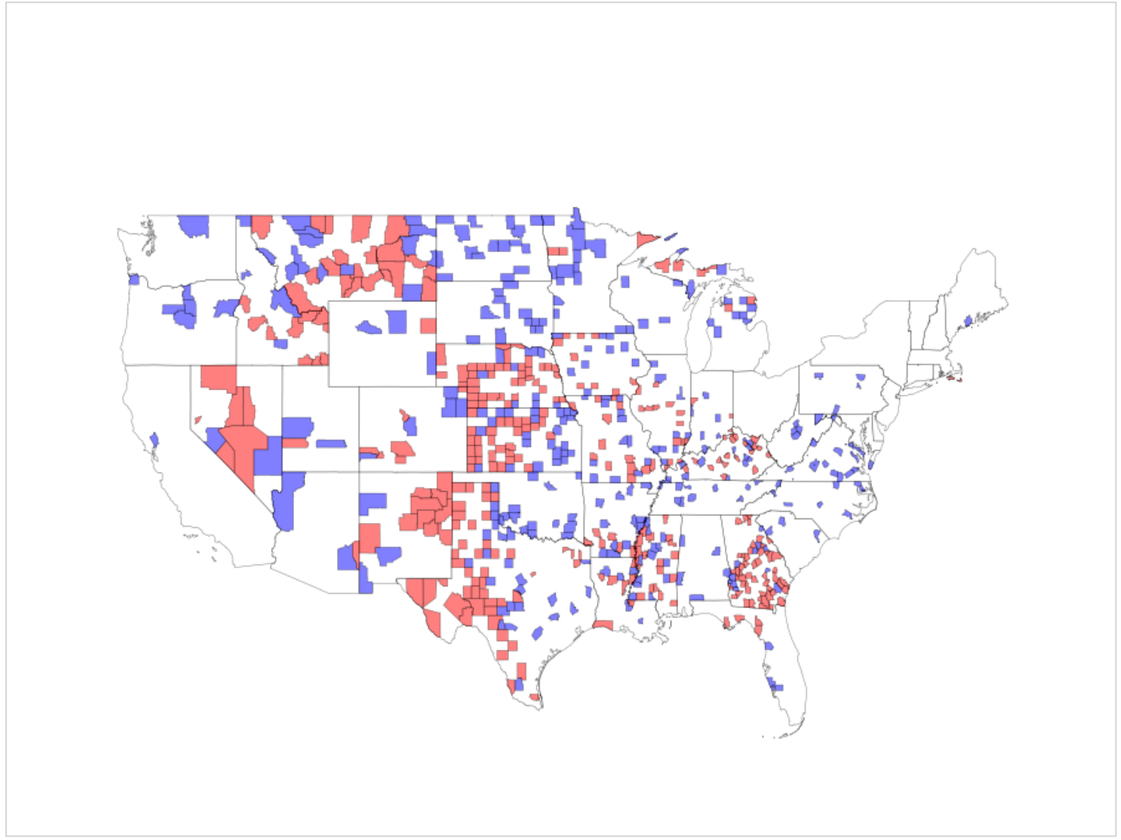
So far have assumed α and β known in priors to be weakly informative.

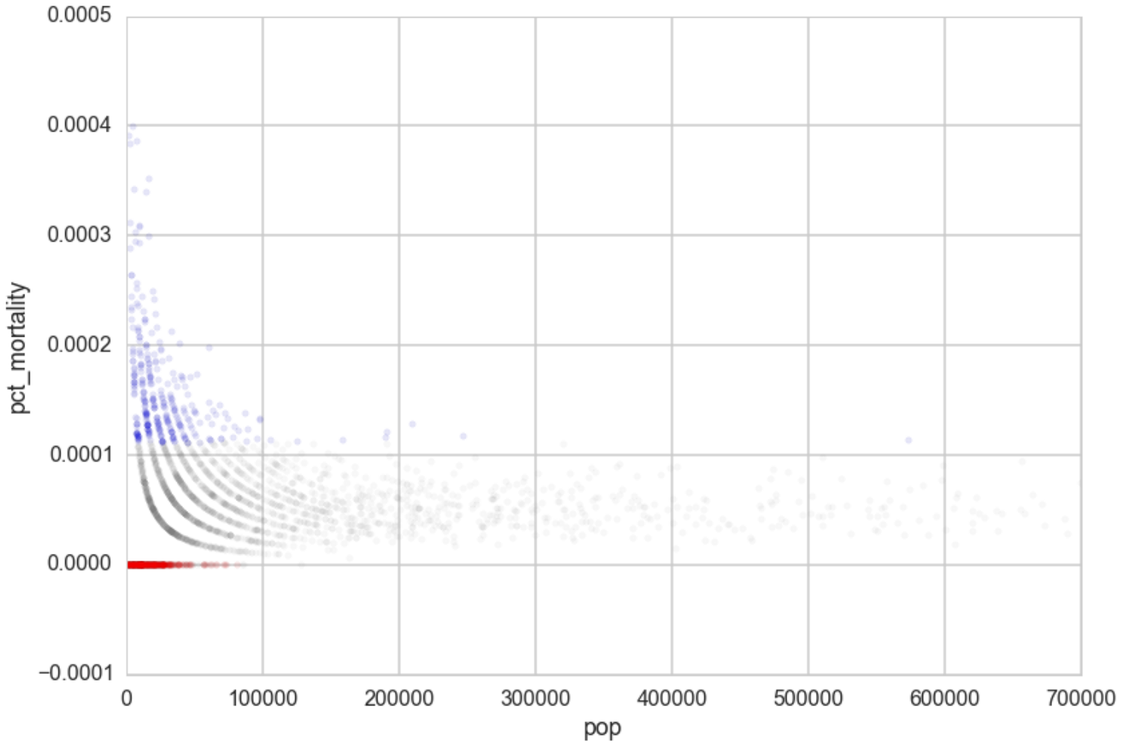
New idea: estimate priors from data. Looks like a cross-validation like setup.

Key Idea: Share statistical strength

- Some **units** (experiments) statistically more robust
- Non-robust experiments have smaller samples or outlier like behavior
- Borrow strength from all the data as a whole through the estimation of the hyperparameters
- **regularized partial pooling model** in which the "lower" parameters (θs) tied together by "upper level" hyperparameters.

Another Example: Kidney cancers





First idea: estimate directly from data

Posterior-predictive distribution, as a function of upper level parameters $\eta = (\alpha, \beta)$.

$$p(y^*|D, \eta) = \int d\theta \, p(y^*|\theta) \, p(\theta|D, \eta)$$

A likelihood with parameters η and simply use maximum-likelihood with respect to η to estimate these η using our "data" y^*

Called Empirical Bayes or Type-2 MLE

- MLE with respect to η
- involves an optimization
- unlike cross-validation, θs not-yet estimated on training set.
- indeed we marginalize over θs so can use training set.
- in practice often match moments of predictive or posterior

EB for rats: prior/prior predictive...

Consider the prior expectation and variance:

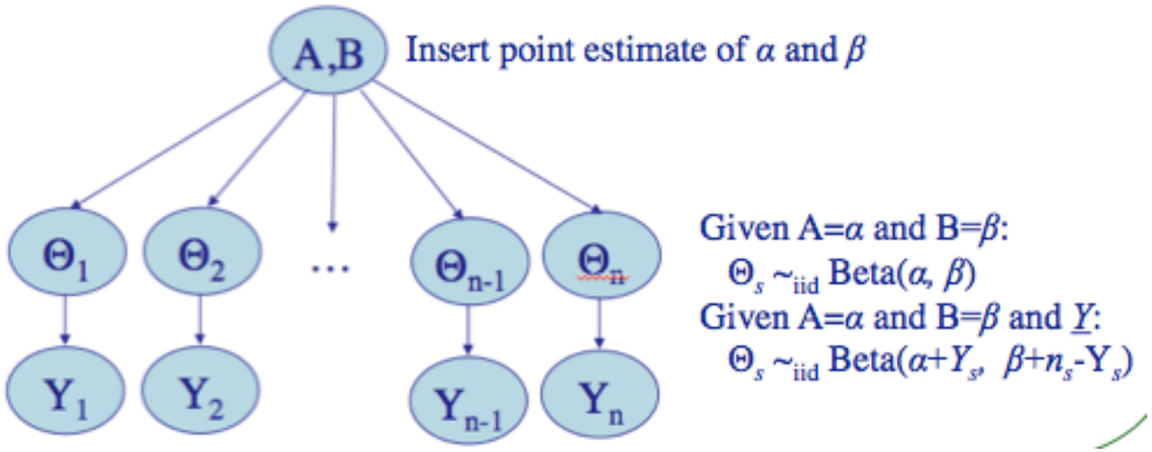
$$\mu = \frac{\alpha}{\alpha+\beta}, V = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

Match empirical mean and variance on y_i/n_i

- Need to be careful what "space" you are working in, predictive (y) or not
- Use prior predictive if in a "predictive space":

$$p(y^*) = E_{p(\theta}[p(y^*|\theta)] = \int d\theta p(y^*|\theta)p(\theta).$$

...to posterior/posterior predictive...

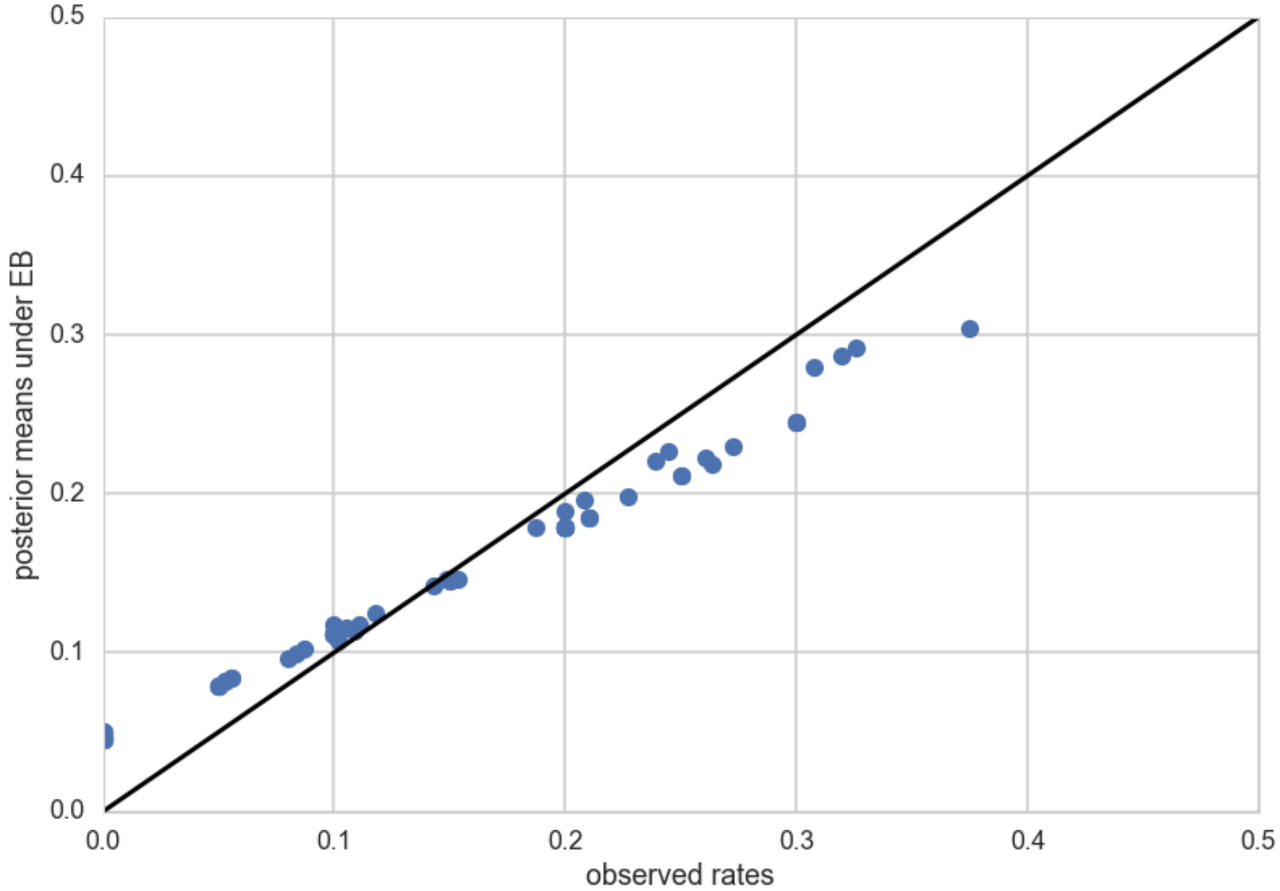


- (α, β) = (`1.3777748392916778`, `8.752435447153129`)
- Conditional posterior distribution for each of the θ_i, given everything else is Beta:.

$$p(\theta_i|y_i, n_i, \alpha, \beta) = Beta(\alpha + y_i, \beta + n_i - y_i)$$

$$\bar{\theta}_{post,i} = \frac{\alpha + y_i}{\alpha + \beta + n_i}$$

Shrinkage in rat (tumors)



Posterior estimates shrink towards full pooling.

Now, for the 71st experiment, we have 4 out of 14 rats having tumors. The posterior estimate for this would be

$$\frac{\alpha + y_{71}}{\alpha + \beta + n_{71}}$$

```
4/14, (4+a_est)/(14+a_est+b_est)
= (0.2857142857142857, 0.22286481449822493)
```

Full Bayesian

- every optimization is a chance to overfit, would like to use integration all the way
- specify a **hyper-prior** $p(\eta)$ ($p(\alpha, \beta)$) on these hyperparameters η (α, β)
- helps us develop a computational strategy of gibbs sampling
- allows estimates of the probabilities of any one unit to borrow strength from all the data as a whole

Fully Bayesian Rat tumors

Joint Posterior:

$$p(\Theta, \alpha, \beta | Y, \{n_i\}) \propto p(\alpha, \beta) \prod_{i=1}^{70} Beta(\theta_i, \alpha, \beta) \prod_{i=1}^{70} Binom(n_i, y_i, \theta_i)$$

Conditionals:

$$p(\theta_i | y_i, n_i, \alpha, \beta) = Beta(\alpha + y_i, \beta + n_i - y_i)$$

More Conditionals

$$p(\alpha|Y,\Theta,\beta) \propto p(\alpha,\beta)\left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)}\right)^N \prod_{i=1}^{N}\theta_i^{\alpha}$$

$$P(\beta|Y,\Theta,\alpha) \propto p(\alpha,\beta)\left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\beta)}\right)^N \prod_{i=1}^{N}(1-\theta_i)^{\beta}$$

These depend on Y and $\{n\}$ via the θ's

Sampling (sampler done in lab)

- Fix α and β, we have a Gibbs step for all of the θ_is
- For α and β, everything else fixed, use stationary metropolis step, as conditionals are not isolatable to simply sampled distributions
- when we sample for α, we will propose a new value using a normal proposal, while holding all the θs and β constant at the old value. ditto for β.

Hierarchy organizes exchangeability

- we use the notion of exchangeability at the level of 'units'.
- for our rats, the y_j were exchangeable since we had no additional information about experimental conditions.
- if specific groups of experiments came from specific laboratories, assume experiments interchangeable if from the same lab.
- lab specific α_{lab} and β_{lab} parameters
- add another level of hierarchy to draw these from hyperprior.