CS395T
Computational Statistics with Application to Bioinformatics

Prof. William H. Press
Spring Term, 2011
The University of Texas at Austin

Lecture 24

Here's the Metropolis-Hastings step function:

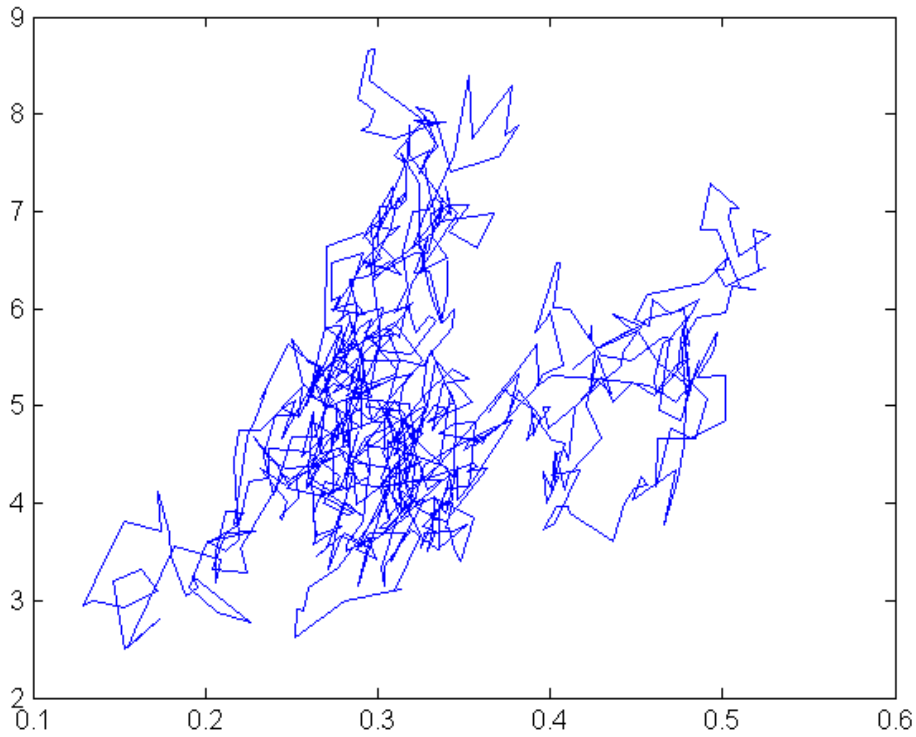
```
function cnew = mcmcstep(cold, covar)
cprop = mvnrnd(cold,covar);
alpha = min(1,exp(loglikefn(cold)-loglikefn(cprop)));
if (rand < alpha)
    cnew = cprop;
else
    cnew = cold;
end

function ll = loglikefn(cc)  %subfunction
global exsamp;
ll = twostudloglikenu(cc,exsamp);
```

Let's see the first 1000 steps:

```
chain = zeros(1000,6);
chain(1,:) = cstart;
for i=2:1000
  chain(i,:) = mcmcstep(chain(i-1,:),covar);
end
plot(chain(:,5),chain(:,6))
```

width of 2nd component

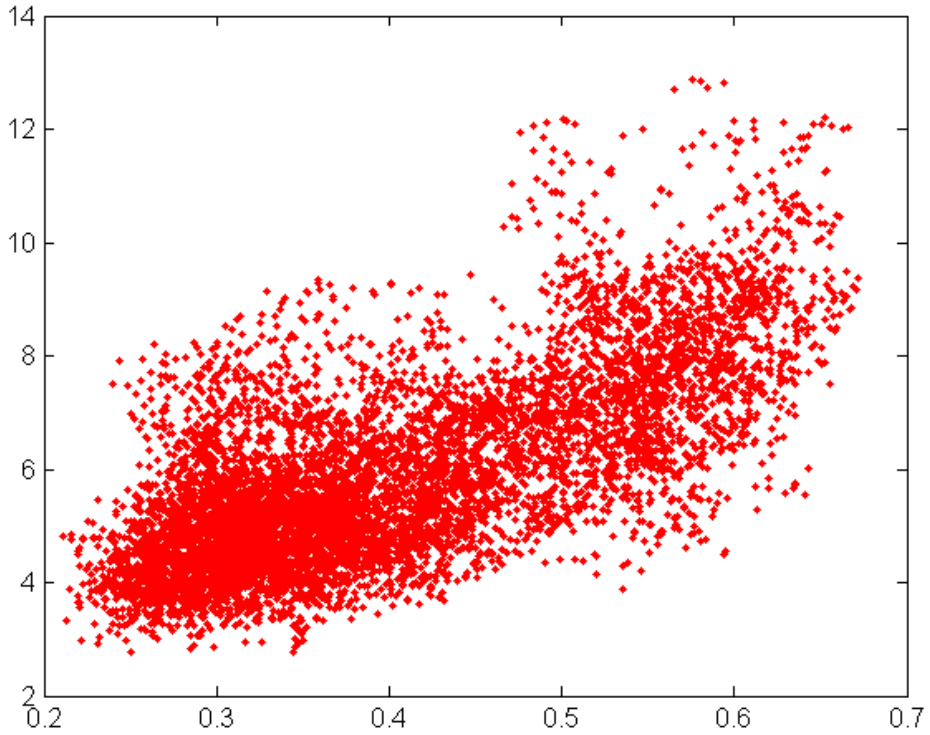
Student-t index



we're only plotting 2 components of chain, but it of course actually is a sample of the joint distribution of all the parameters

Try 10000 steps:

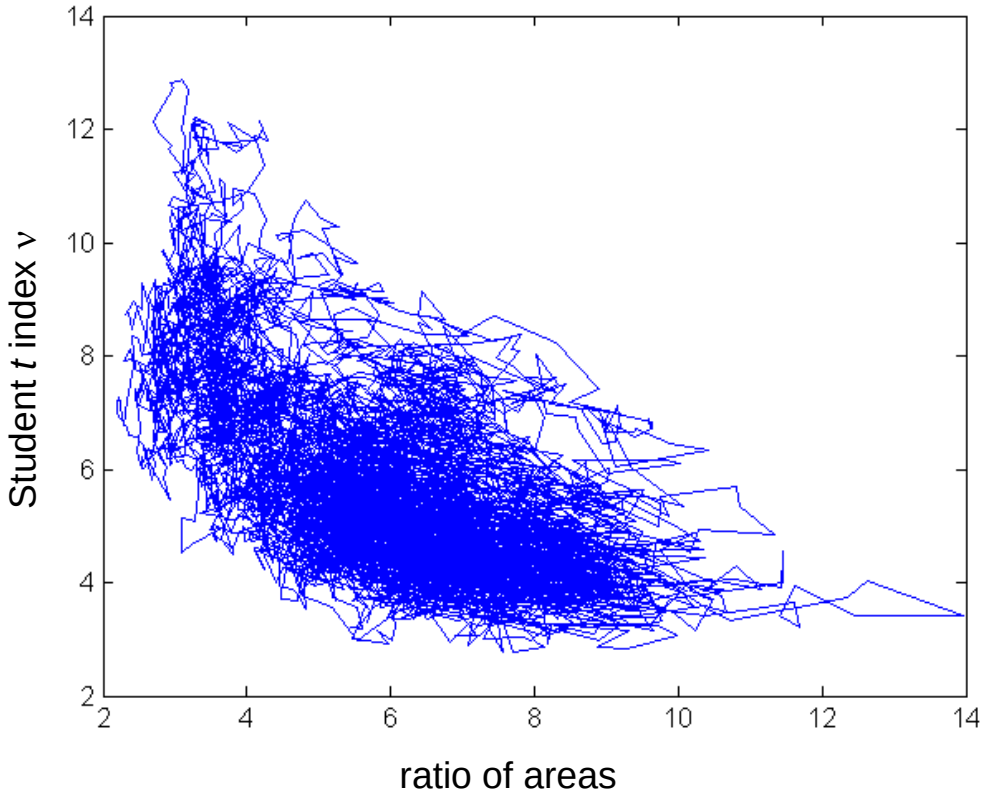
```
chain = zeros(10000,6);
chain(1,:) = cstart;
for i=2:10000, chain(i,:) = mcmcstep(chain(i-1,:),covar); end
plot(chain(:,5),chain(:,6),'.r')
```



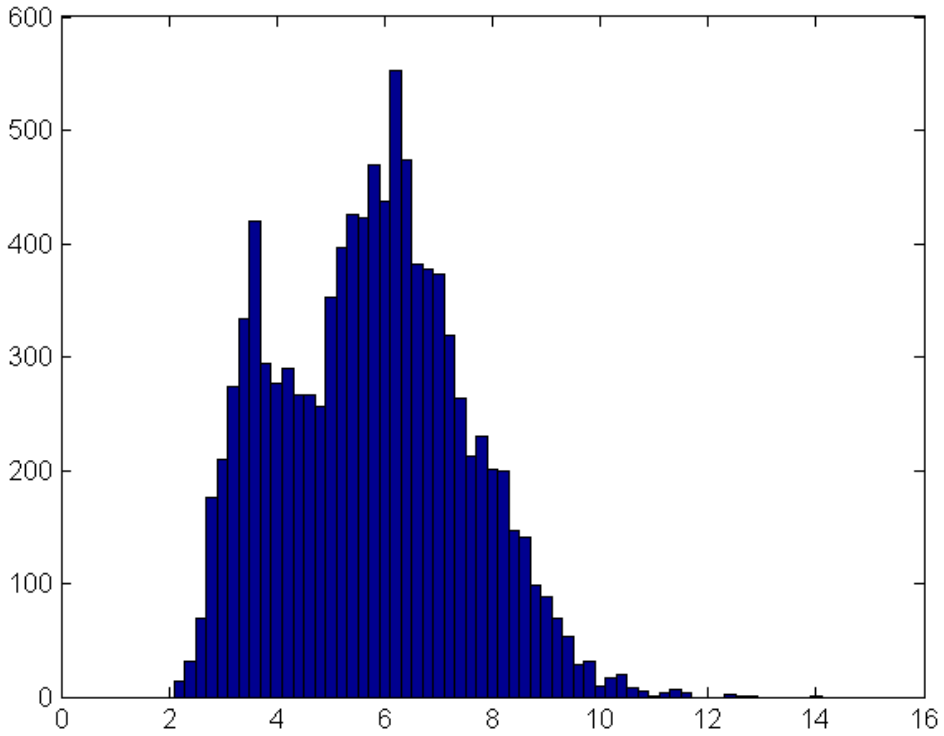
OK, plausibly ergodic. Should probably do 100000 steps, but Matlab is too slow and I'm too lazy to program it in C. (Don't you be!)
There are various ways of checking for convergence more rigorously, none of them foolproof; see NR3.

The big payoff now is that we can look at the posterior distribution of any quantity, or derived quantity, or joint distribution of quantities, etc., etc.

```
areas = chain(:,2)./(chain(:,3).*chain(:,5));
plot(areas,chain(:,6))
```

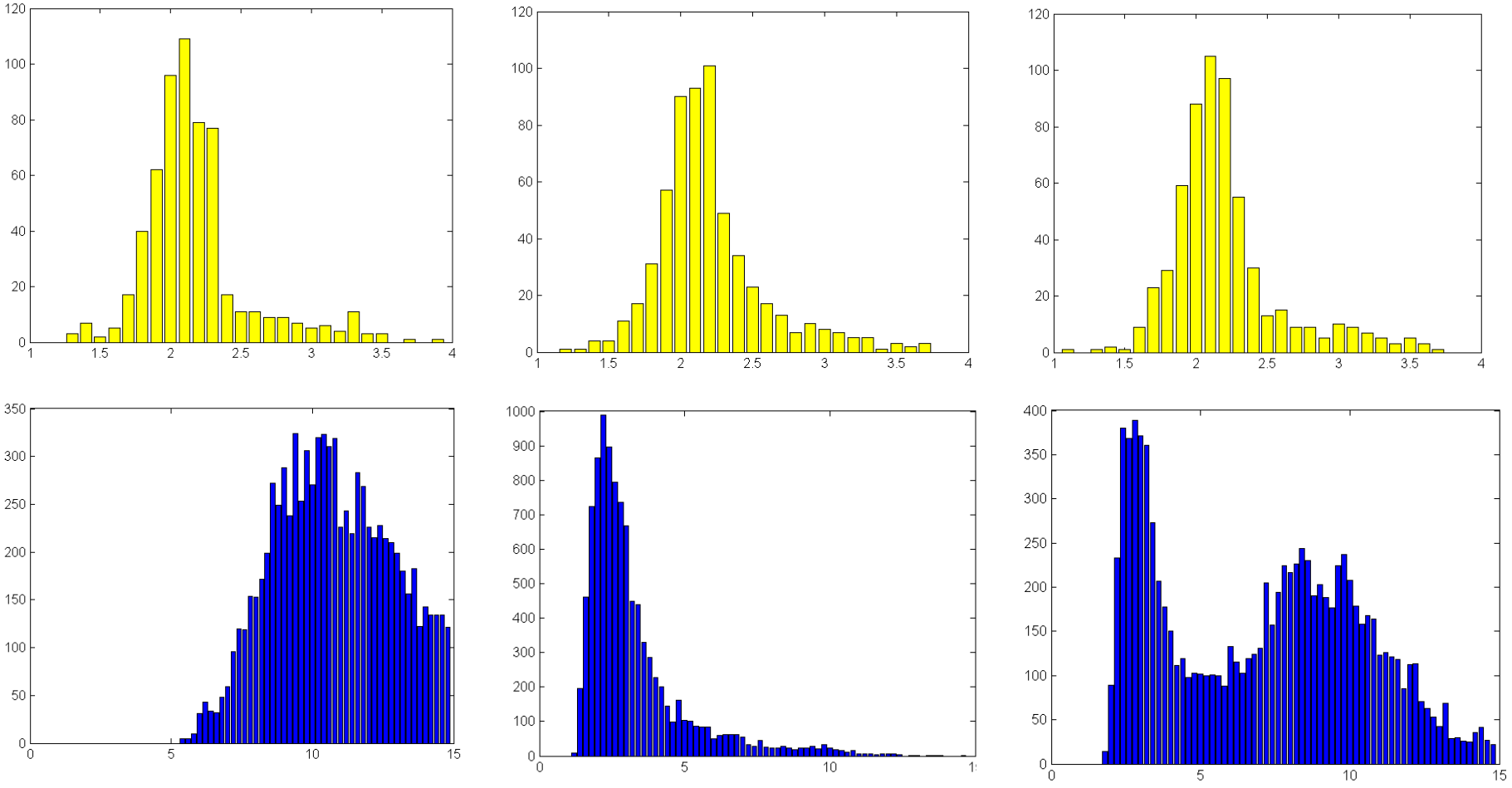


```
hist(areas,1:.2:15)
```



We can now see why the ratio of areas is so hard to determine: For some of our data samples, it can be bimodal. Maximum likelihood and derivatives of the log-likelihood (Fisher information matrix) don't capture this. MCMC and Bayes posteriors do.

Incidentally, the distributions are very sensitive to the tail data.
Different samples of 600 points give (e.g.)



With only one data set, you could diagnose this sensitivity by bootstrap resampling.
Despite this, fierce Bayesians show only the posteriors from all the data actually known.

Let's do another MCMC example to show how it can be used with models that might be analytically intractable (e.g., discontinuous or non-analytic).
[This is the example worked in NR3.]

The lazy birdwatcher problem

- You hire someone to sit in the forest and look for mockingbirds.
- They are supposed to report the time of each sighting t_i
  - But they are lazy and only write down (exactly) every k_1 sightings (e.g., k_1= every 3rd)
- Even worse, at some time t_c they get a young child to do the counting for them
  - He doesn't recognize mockingbirds and counts grackles instead
  - And, he writes down only every k_2 sightings, which may be different from k_1
- You want to salvage something from this data
  - E.g., average rate of sightings of mockingbirds *and* grackles
  - Given only the list of times
  - That is, k_1, k_2, and t_c are all unknown nuisance parameters
- This all hinges on the fact that every second (say) event in a Poisson process is statistically distinguishable from every event in a Poisson process at half the mean rate
  - same mean rates
  - but different fluctuations
  - We are hoping that the difference in fluctuations is enough to recover useful information
- Perfect problem for MCMC

Waiting time to the kth event in a Poisson process with rate λ is distributed as Gamma(k,λ)

$$\tau = t_{i+k} - t_i$$

$$p(\tau|k,\lambda) = \frac{\lambda^k}{(k-1)!}\tau^{k-1}e^{-\lambda\tau}$$

And non-overlapping intervals are independent: $t_{i+k} - t_i$ $t_{i+2k} - t_{i+k}$

Proof:

$$\begin{aligned} p(\tau)d\tau &= P(k-1 \text{ counts in } \tau) \times P(\text{last } d\tau \text{ has a count}) \\ &= \text{Poisson}(k-1, \lambda\tau) \times (\lambda\, d\tau) \\ &= \frac{(\lambda\tau)^{k-1}}{(k-1)!}e^{-\lambda\tau}\lambda\, d\tau \end{aligned}$$

So

$$P(\mathbf{D}|\mathbf{x}) = \prod_{t_i \le t_c} p(t_{i+1} - t_i \,|\, k_1, \lambda_1) \times \prod_{t_i > t_c} p(t_{i+1} - t_i \,|\, k_2, \lambda_2)$$

What shall we take as our proposal generator?
This is often the creative part of getting MCMC to work well!

For t_c, step by small additive changes (e.g., normal)

For λ_1 and λ_2, step by small multiplicative changes (e.g., lognormal)

In the acceptance probability the ratio of the q's in

$$\alpha(\mathbf{x}_1, \mathbf{x}_{2c}) = \min\left(1, \frac{\pi(\mathbf{x}_{2c})\, q(\mathbf{x}_1|\mathbf{x}_{2c})}{\pi(\mathbf{x}_1)\, q(\mathbf{x}_{2c}|\mathbf{x}_1)}\right)$$

is just x_{2c}/x_1, because

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left(-\frac{1}{2}\left[\frac{\log(x) - \mu}{\sigma}\right]^2\right)$$

Bad idea: For $k_{1,2}$ step by 0 or ±1

This is bad because, if the λ's have converged to about the right rate, then a change in *k* will throw them way off, and therefore nearly always be rejected. Even though this appears to be a "small" step of a discrete variable, it is not a small step in the model!

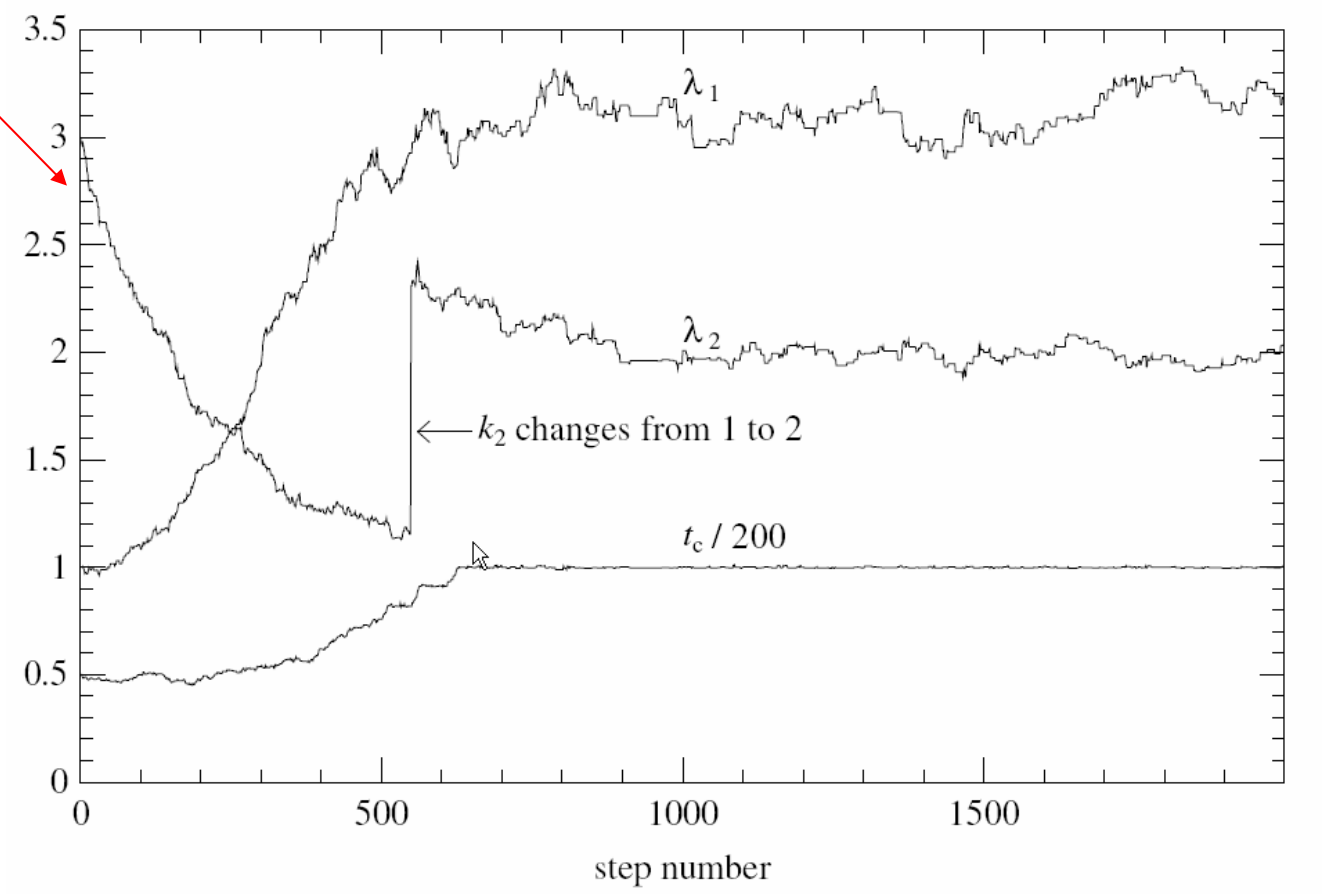
Good idea: For $k_{1,2}$ step by 0 or ±1, also changing $\lambda_{1,2}$ so as to keep λ/k constant in the step

This is genuinely a small step, since it changes only the clumping statistics, by the smallest allowed amount.

Let's try it.

We simulate 1000 t_i's with the secretly known λ_1=3.0, λ_2=2.0, t_c=200, k_1=1, k_2=2

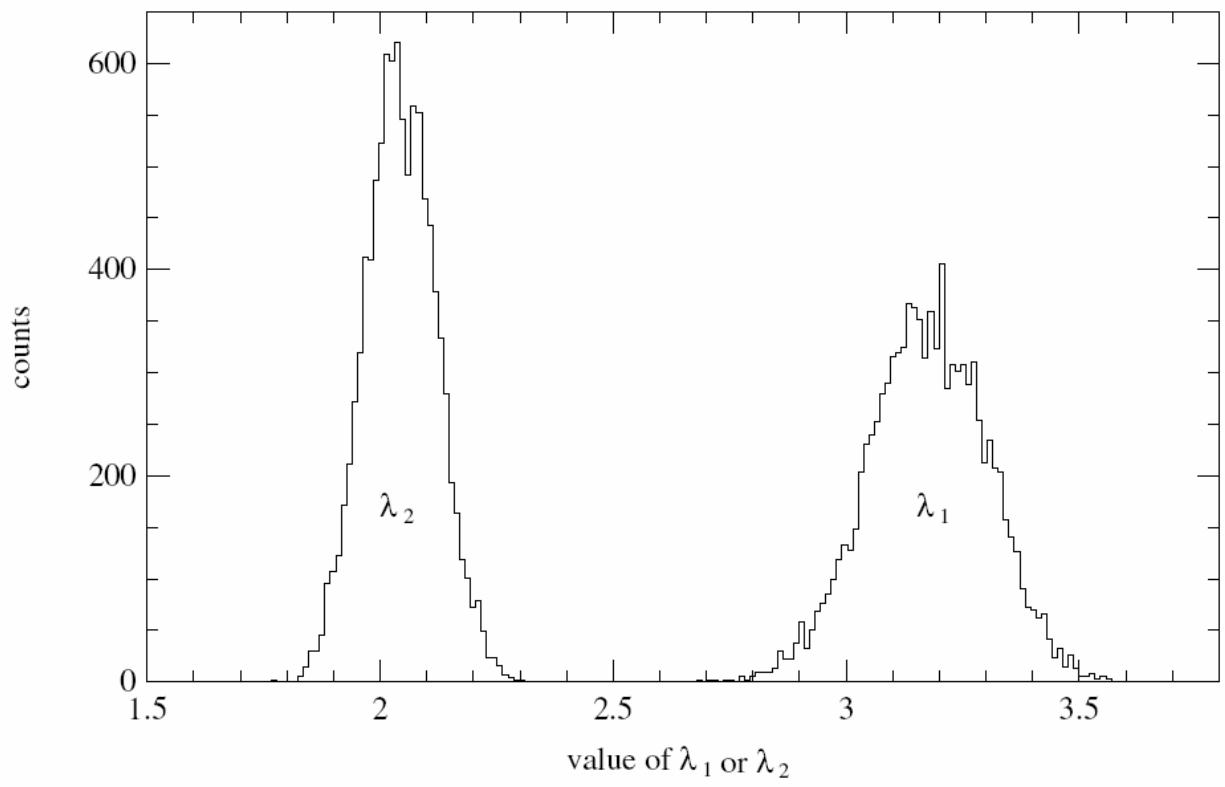
Start with wrong values λ_1=1.0, λ_2=3.0, t_c=100, k_1=1, k_2=1



"burn-in" period while it locates the Bayes maximum

ergodic period during which we record data for plotting, averages, etc.

Histogram of quantities during a long-enough ergodic time

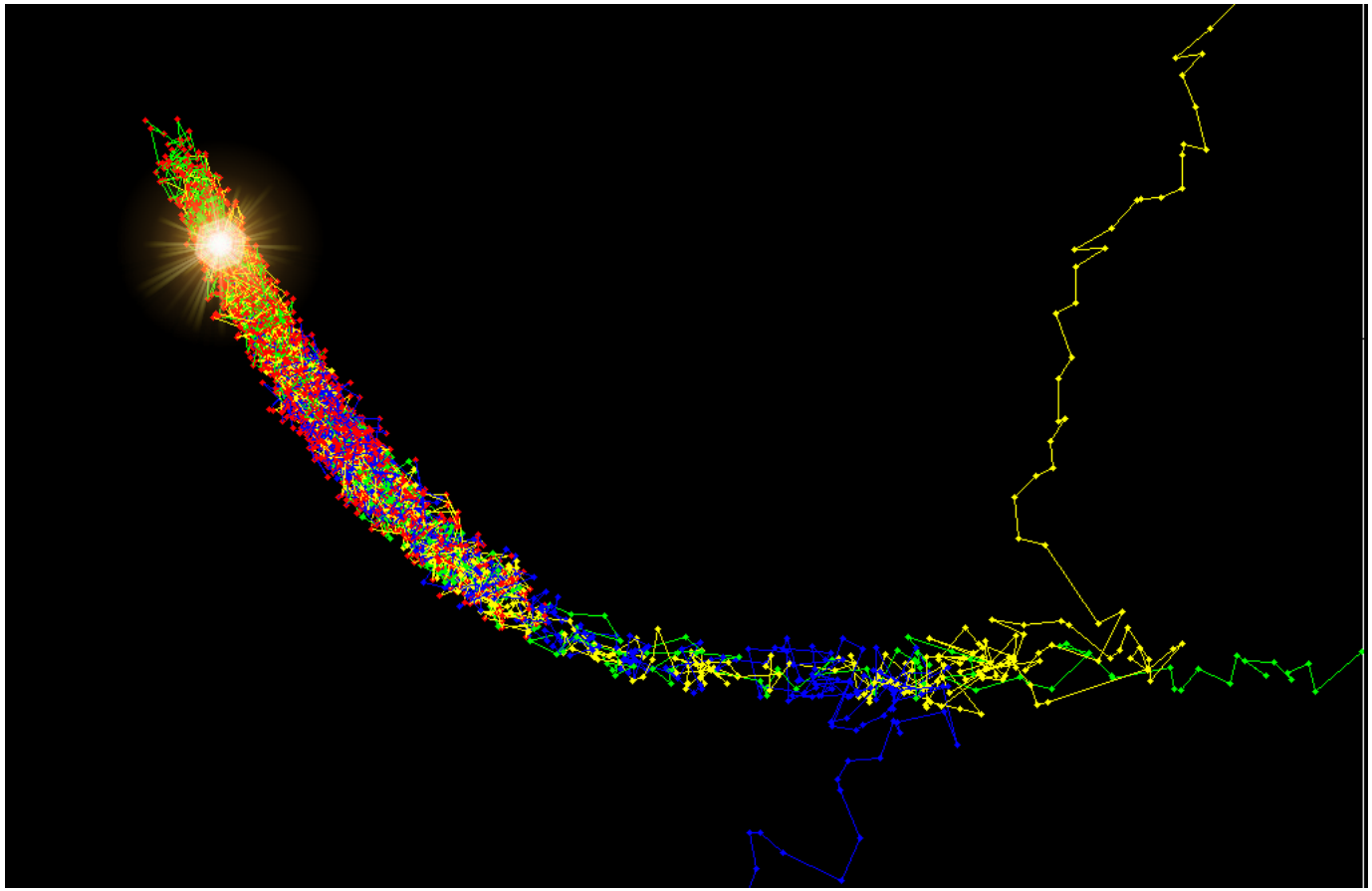


These are the actual Bayesian posteriors of the model!

Could as easily do joint probabilities, covariances, etc., etc.

Notice does not converge to being centered on the true values, because the (finite available) data is held fixed. Convergence is to the Bayesian posterior for that data.

Burn-in can have multiple timescales (e.g., ascent to a ridge, travel along ridge)



Wikipedia