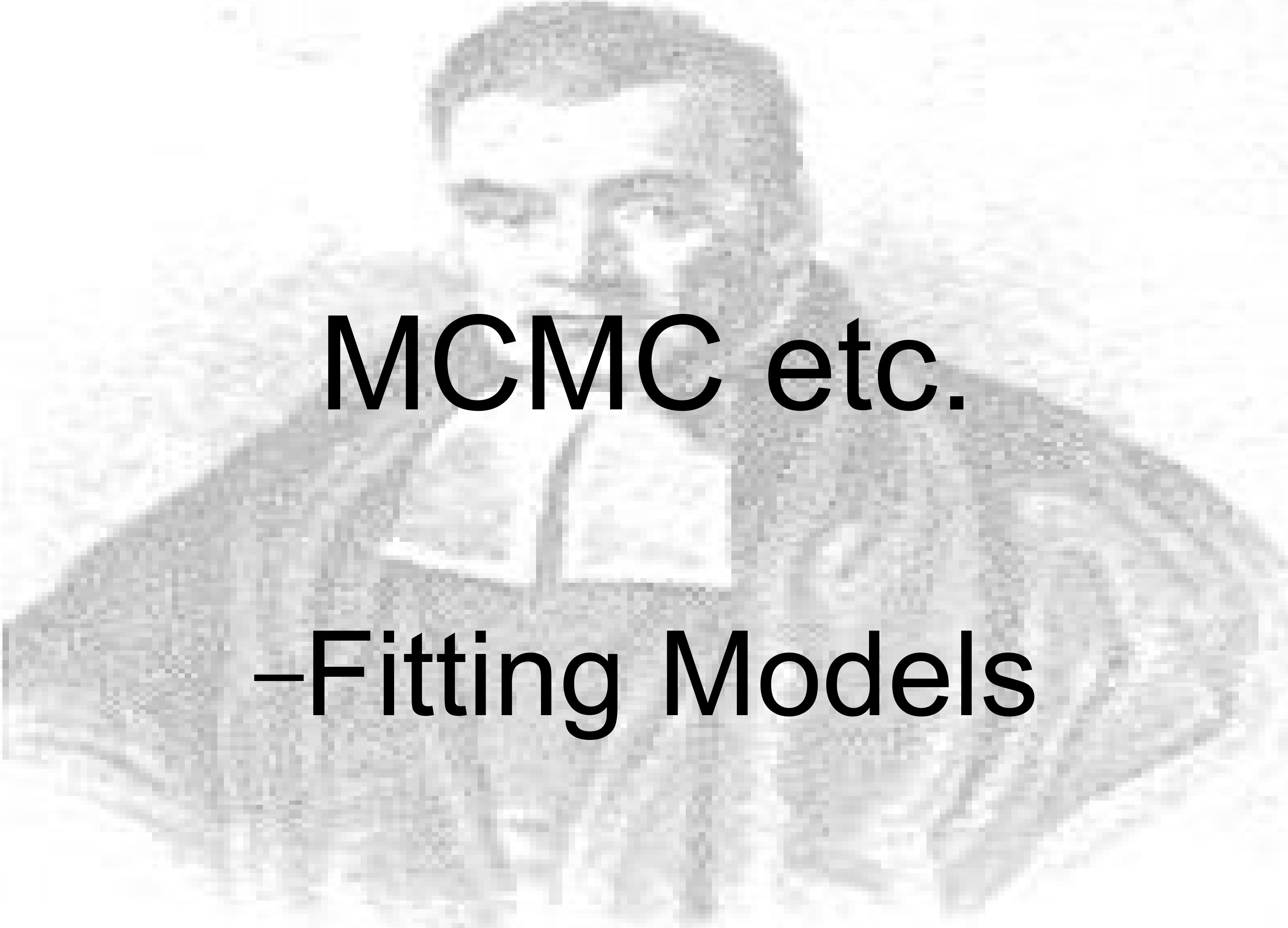


MCMC etc.

–Fitting Models

The Story So Far

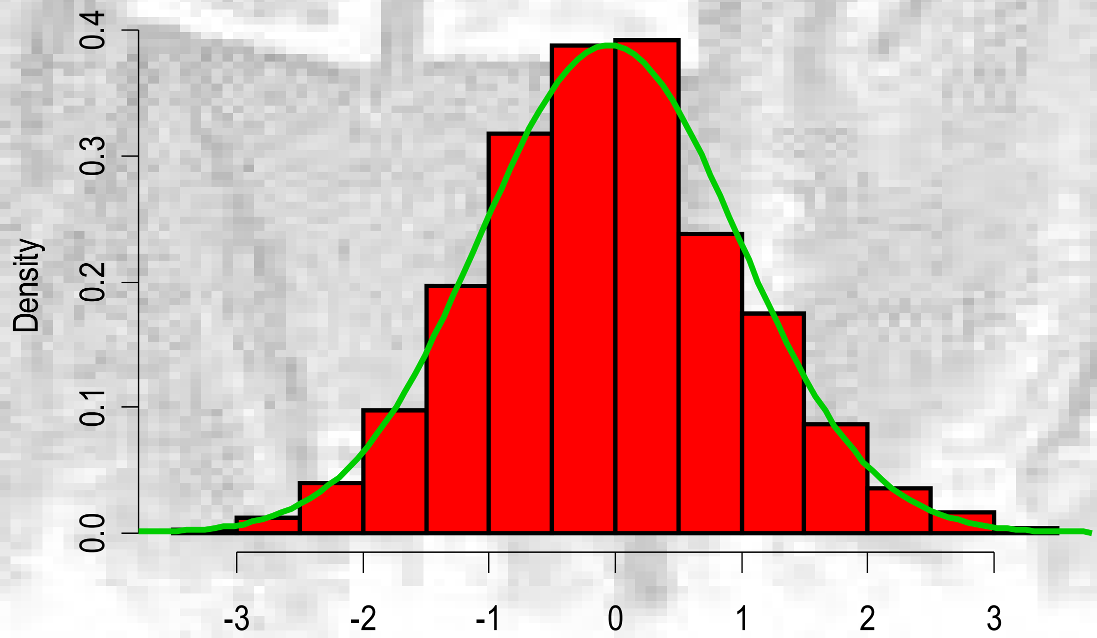
- We can use Bayes' Theorem as the basis for statistical inference:
- P $(\theta|X) \propto$ P$(X|\theta)$.P(θ)
- With any non-trivial problem, it is impossible in practice to derive the equations for P$(\theta|X)$
- This use to be a problem, but no longer!
- Now we use computer intensive algorithms to estimate P$(\theta|X)$

The Basics

- Computers can do lots of boring calculations
- The challenge is to develop algorithms that are efficient
  - i.e. which don't take too much time
- Computers can generate “pseudo-random numbers”
  - numbers that look random
  - Uniform distribution
  - want to use these to simulate other distributions

The Output

- We want to estimate probability distributions
- In particular, the posterior distribution, $P(\theta|X)$
- Simulate the distribution
- Should look like the underlying distribution:

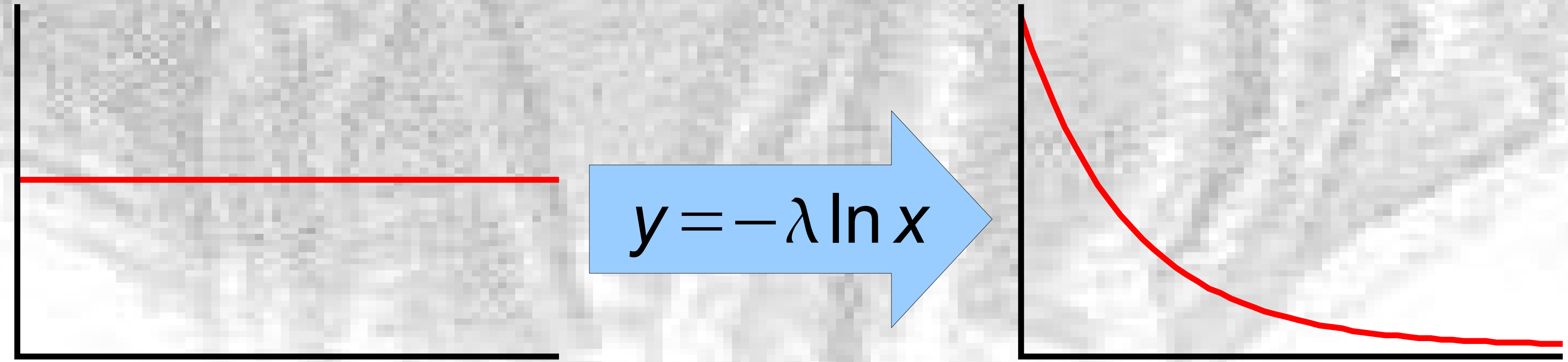


Simple Simulations

- What if we want to simulate an exponential distribution?

$$f(x) = \frac{1}{\lambda} e^{-\frac{x}{\lambda}}$$

- We can do this by transforming a uniform distribution:



Simple Simulations

- Some distributions can be simulated in this way
  - exponential
  - normal
  - uniform
- But the transformation may be difficult to calculate
- Or it may take a long time
  - use quicker approaches

Markov Chain Monte Carlo

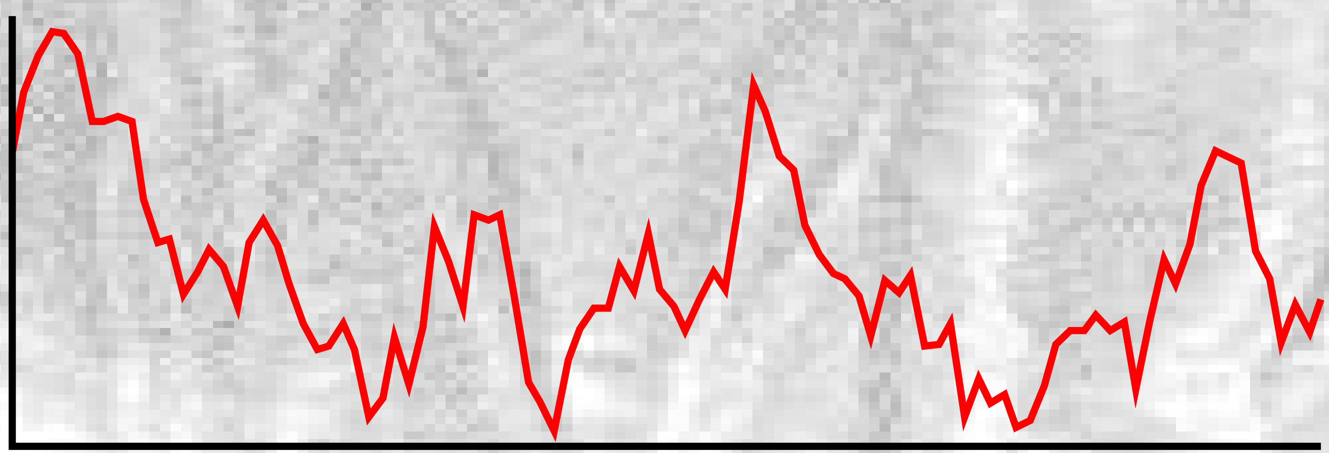
- Often the solution to fitting complex models
  - but not without problems
- Enables us to simulate posterior distributions
  - in many dimensions
- Now used in almost every Bayesian analysis
- Can also be used for frequentist analyses
  - but is often slower than other methods

What MC and MC mean

- Monte Carlo simulation
  - numerical simulation
  - stochastic
- Markov Chain
  - discrete time stochastic model
  - each time step depends on the previous one
  - common population model
  - often interested in the stationary distribution
    - the probability distribution after a long time

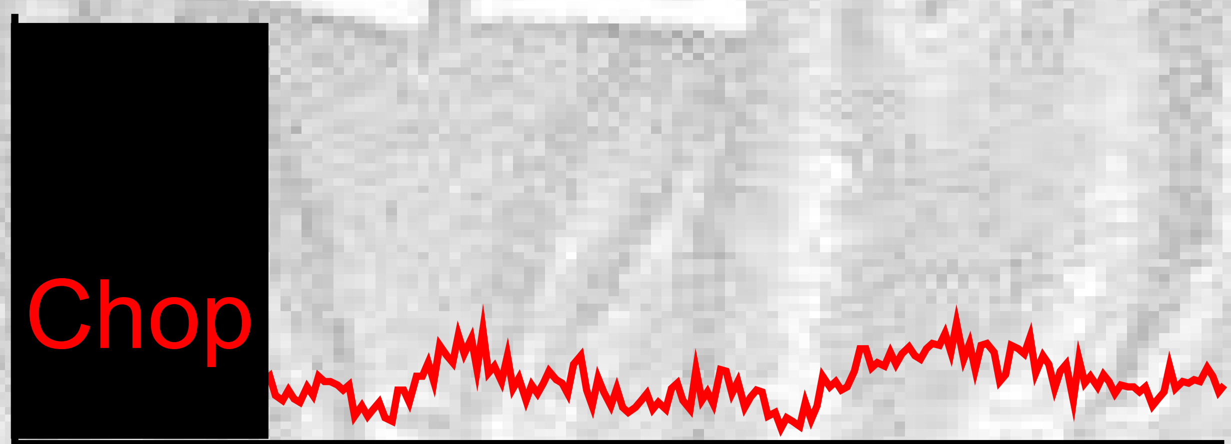
MCMC

- . Create a Markov chain whose stationary distribution is the distribution we want
- . Then run a Monte Carlo simulation for a long time, and store the values
- – this will be from the distribution we want



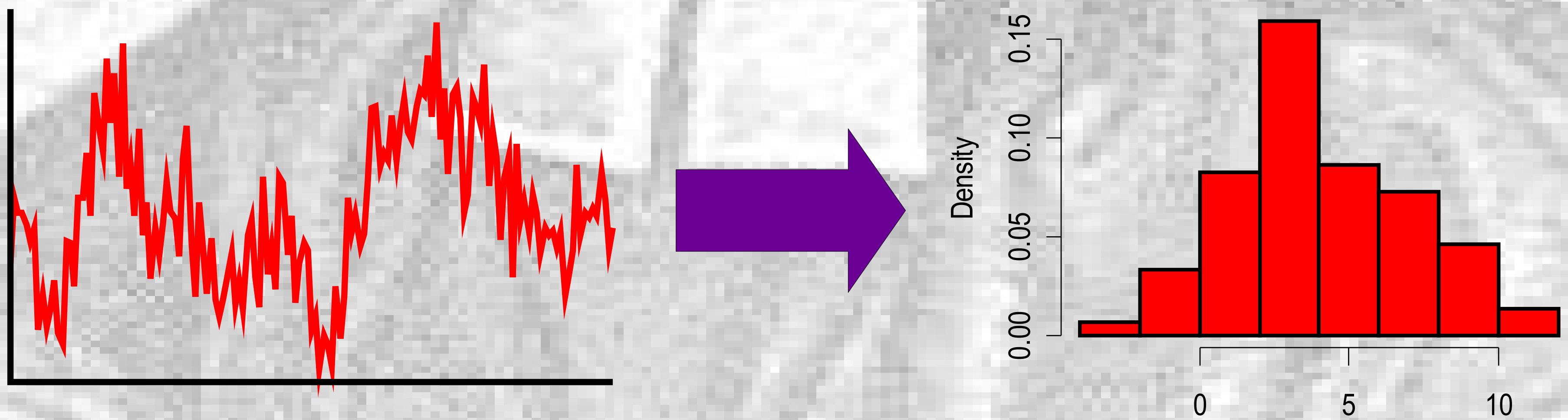
The burn-in

- .Once the chain has got to the stationary state, all values are from the required distribution
- .We remove the first few values, the burn-in
- Chop



The Rest

- . After the burn-in, the values are from the target distribution



Updating Algorithms

- Several algorithms available
  - minor industry in computational statistics
- Metropolis-Hastings
  - propose a new point, accept if good enough
  - if not good enough, stay there
  - can prove it works
- Gibbs Sampler

Gibbs Sampler

- A form of M-H algorithm
- Update each parameter individually
- Propose from the conditional distribution
  - e.g. $P(\beta_0 | \beta_1, \sigma^2)$
- Can prove we will always accept
- Needs an additional assumption
  - "conditional independence"
- Most regression problems fulfil this criterion

Into Several Dimensions

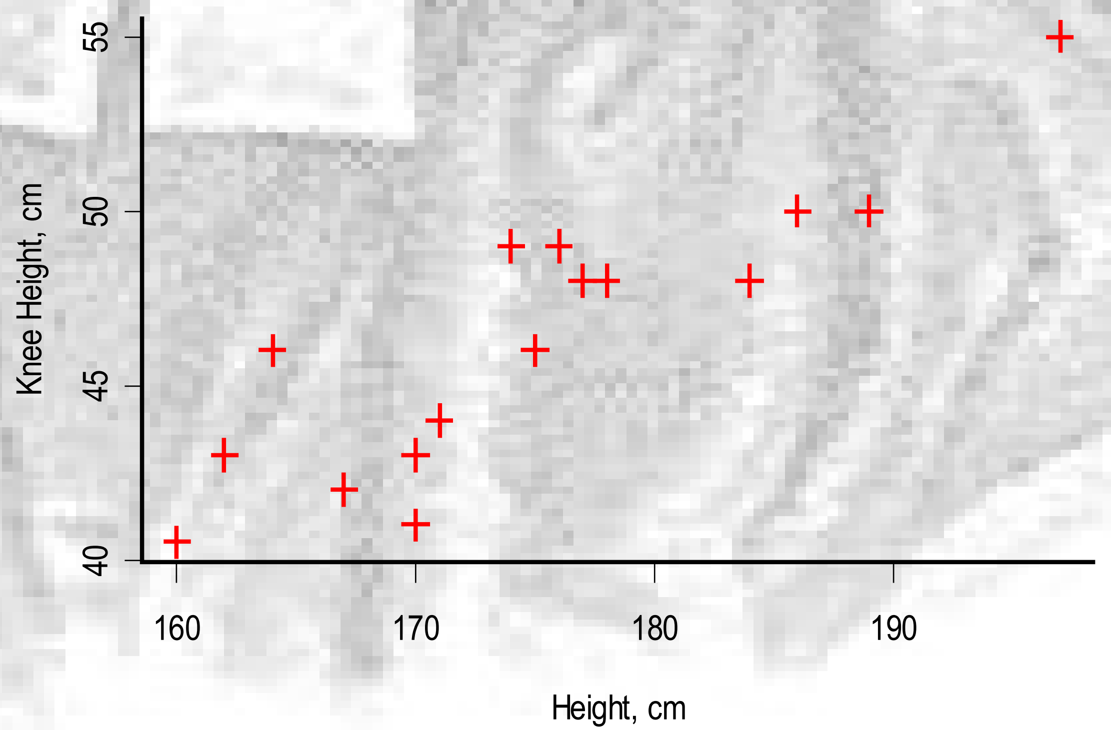
- We could use M-H or Gibbs sampler in several dimensions
  - propose several dimensions simultaneously
- But gets difficult in many dimensions
- Good news: We don't have to update all at once
- We can propose each parameter on its own
- Rotate through the parameters
- Even do some parameters several times

An Example

- Knee heights – Regression
- Model:

$$y_i \sim N\left(\beta_0 + \beta_1 (x_i - \bar{x}), \sigma^2\right)$$

- y_i: knee height
- x_i: total height

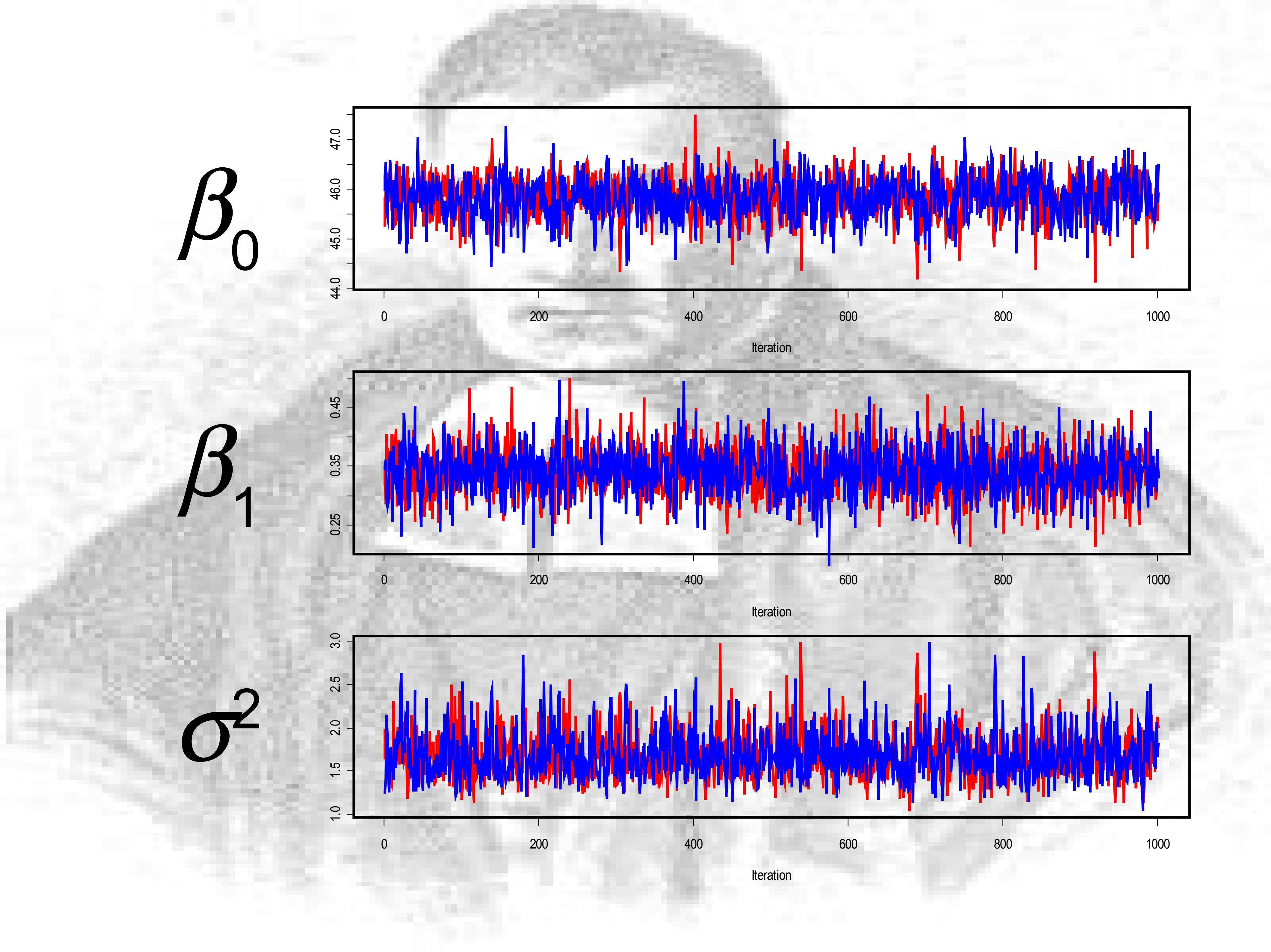


From A Previous Course

- Want to estimate posteriors for β_0, β_1 and σ^2
- Priors:
  - β_0 ~ N(42, 1.52)
  - β_1 ~ N(0.25, 0.022) or β_1 ~ N(0.33, 0.052)
  - σ^2 ~ Inv-χ^2(4, 0.5)
- Mainly interested in β_1
  - look at marginal distribution

What we get

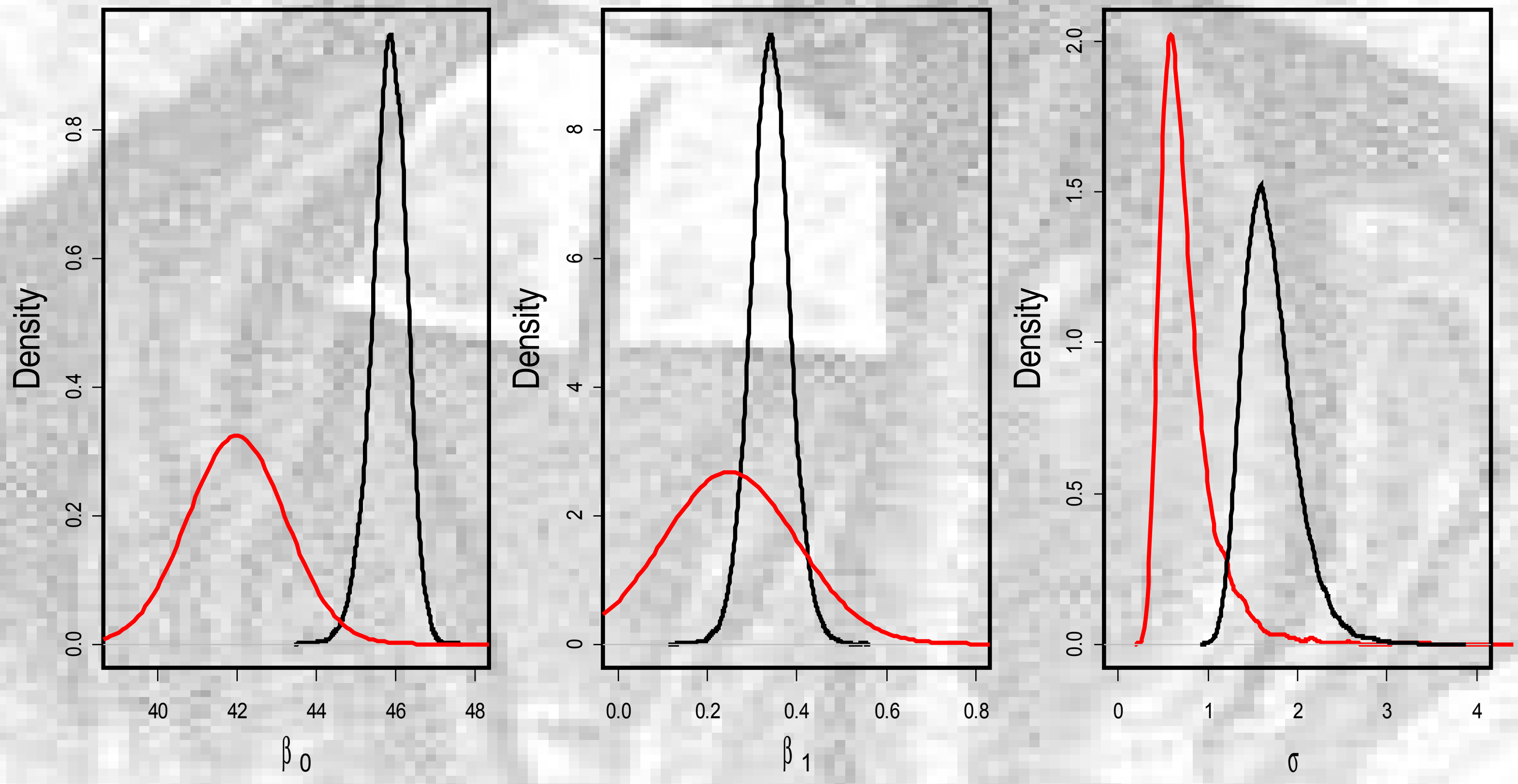
- Run 2 chains
- Take the marginal distribution of β_0 by dropping the other parameters
  - we sum over their variation
- Yes, it is that simple



β0
47.0
46.0
45.0
44.0
0
200
400
600
800
1000
Iteration
β1
0.45
0.35
0.25
0
200
400
600
800
1000
Iteration
σ2
3.0
2.5
2.0
1.5
1.0
0
200
400
600
800
1000
Iteration

Marginal Distributions

- Red lines: prior, black lines: posteriors



Joint Posterior

- β_0 and β_1 only
- No correlation
  - ρ=0.00089