Bayesian statistics – Nanjing Forestry University

# Lecture 4

- Wei Wu, The University of Southern Mississippi
- December, 2016

Acknowledgement: SESYNC Bayesian modeling for socio-environmental data workshop by Hobbs, Collins, and Che-Castaldo

THE UNIVERSITY OF SOUTHERN MISSISSIPPI

GULF COAST RESEARCH LABORATORY

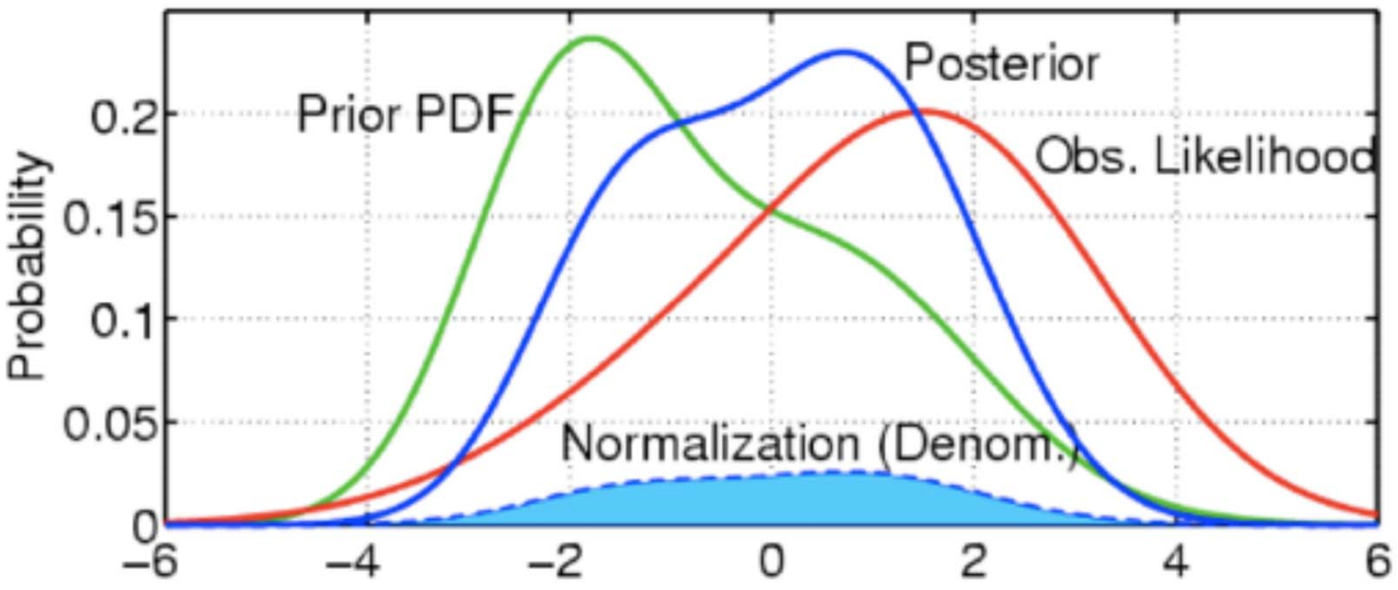
# The MCMC algorithm

- Some intuition
- Accept-reject sampling with Metropolis algorithm
- Introduction to full-conditional distributions
- Gibbs sampling (exercise)
- Metropolis-Hastings algorithm
- Implementing accept-reject sampling

## MCMC learning outcomes

1. Develop a big picture understanding of how MCMC allows us to approximate the marginal posterior distribution for parameters and latent quantities.
2. Understand and be able to code a simple MCMC algorithm.
3. Appreciate the different methods that can be used within MCMC algorithms to make draws from the posterior distribution.
    1. Metropolis
    2. Metropolis-Hastings
    3. Gibbs
4. Understand concepts of burn-in and convergence.
5. Be able to write full-conditional distributions.

# Remember the marginal distribution of data



# We have simple solutions for the posterior for simple models

$$[\phi|y,n] = \text{beta}\left(\phi|\underbrace{\overbrace{\alpha}^{\text{The prior }\alpha}+y}_{\text{The new }\alpha},\underbrace{\overbrace{\beta}^{\text{The prior}\beta}+n-y}_{\text{The new }\beta}\right)$$

# Problems of high dimension do not have simple solutions

$$[\boldsymbol{\theta}_1, \boldsymbol{\theta}_{2,}\boldsymbol{\theta}_3, \boldsymbol{\theta}_4, z_i, y_i, u_i] = \frac{[y_i|\boldsymbol{\theta}_1 z_i][u_i|\boldsymbol{\theta}_2, z_i][z_i|\boldsymbol{\theta}_3, \boldsymbol{\theta}_4][\boldsymbol{\theta}_1][\boldsymbol{\theta}_2][\boldsymbol{\theta}_3][\boldsymbol{\theta}_4]}{\int\int\int\int[y_i|\boldsymbol{\theta}_1 z_i][u_i|\boldsymbol{\theta}_2, z_i][z_i|\boldsymbol{\theta}_3, \boldsymbol{\theta}_4][\boldsymbol{\theta}_1][\boldsymbol{\theta}_2][\boldsymbol{\theta}_3][\boldsymbol{\theta}_4]d\boldsymbol{\theta}_1 d\boldsymbol{\theta}_2 d\boldsymbol{\theta}_3 d\boldsymbol{\theta}_4}$$

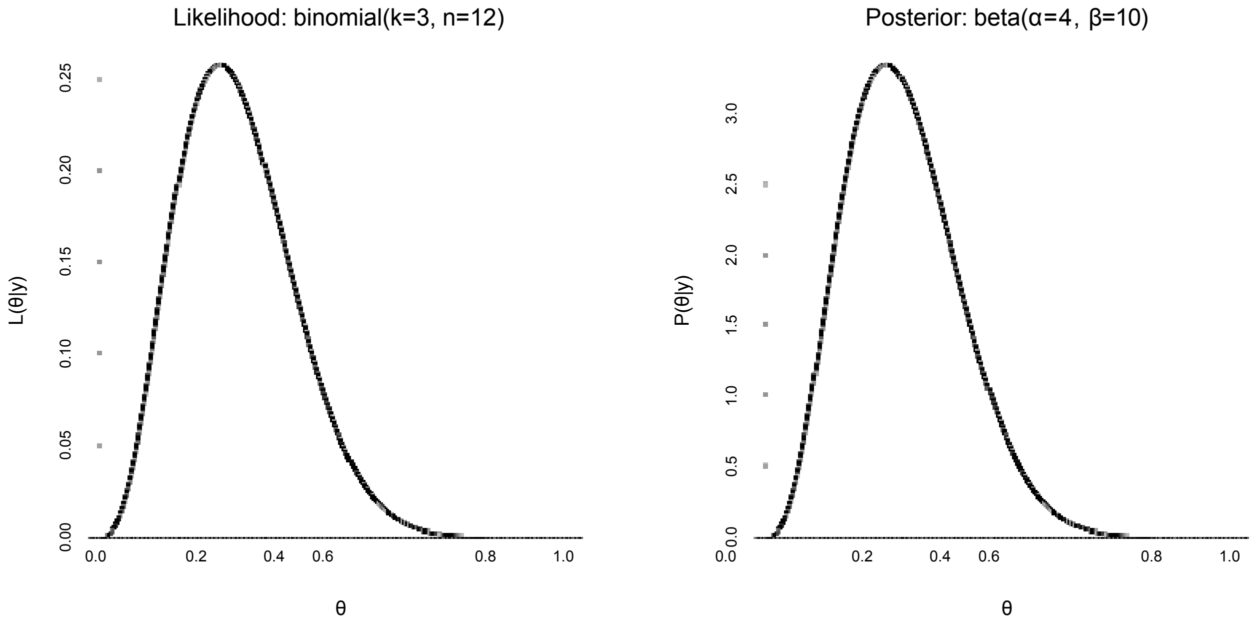
# What we are doing in MCMC?

Recall that the posterior distribution is proportional to the joint:

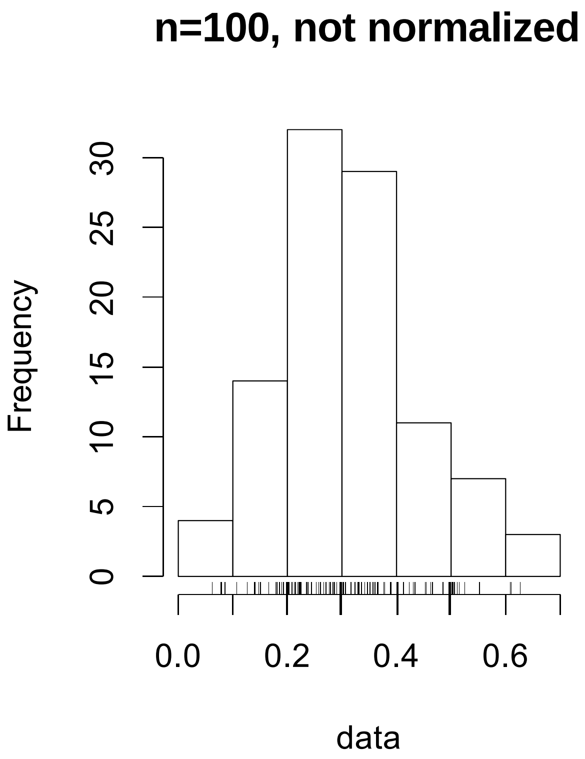
$$[\theta|y] \propto [y|\theta][\theta], \tag{1}$$

because the marginal distribution of the data $\int [y|\theta][\theta]d\theta$ is a constant after the data have been observed.

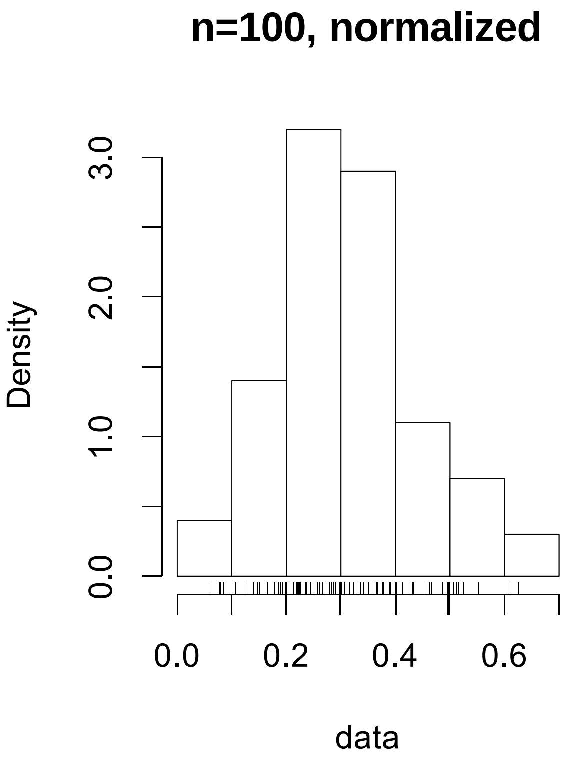
# What we are doing in MCMC?



# What we are doing in MCMC?



n=100, not normalized



n=100, normalized

# The MCMC algorithm

Allows us to find the marginal posterior distribution of each of the unknowns while avoiding any formal integration.

The essential idea of MCMC is that we can learn about the unknowns by making many random draws from their marginal posterior distributions.

We can think of MCMC as an algorithm that normalizes the likelihood profile weighted by prior.

We can also easily obtain posterior distributions of any quantity that is derived from the unknowns we estimate – enormous benefit of MCMC.

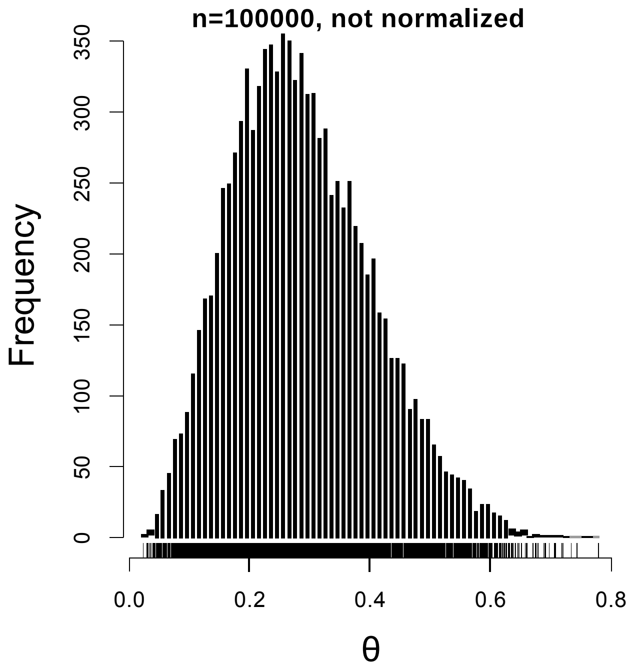
# What are we doing in MCMC?

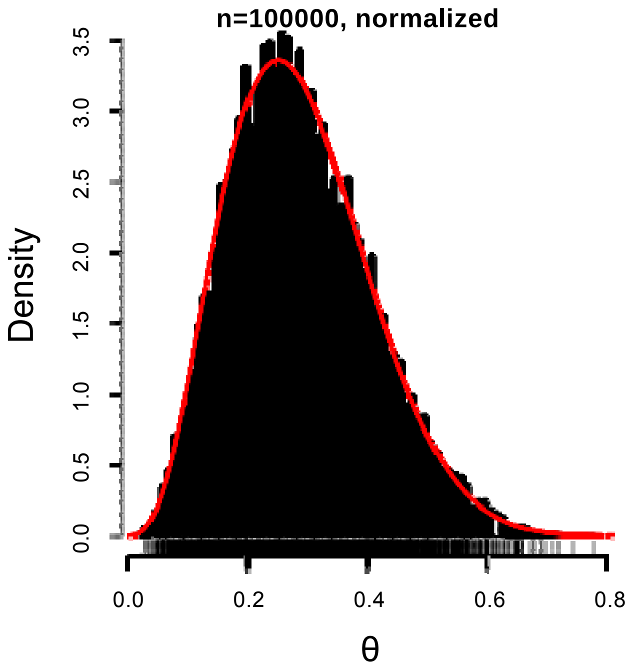
- The posterior distribution is unknown, but the likelihood is known as a likelihood profile and we know the priors.
- We want to accumulate many, many values that represent the random samples in the simulated posterior distribution proportionate to their probability.
- MCMC generates these samples using the likelihood and the priors to decide which samples to keep and which to throw away.
- We can then use these samples to calculate statistics describing the distribution: means, medians, variances, credible intervals etc.

# What are we doing in MCMC?

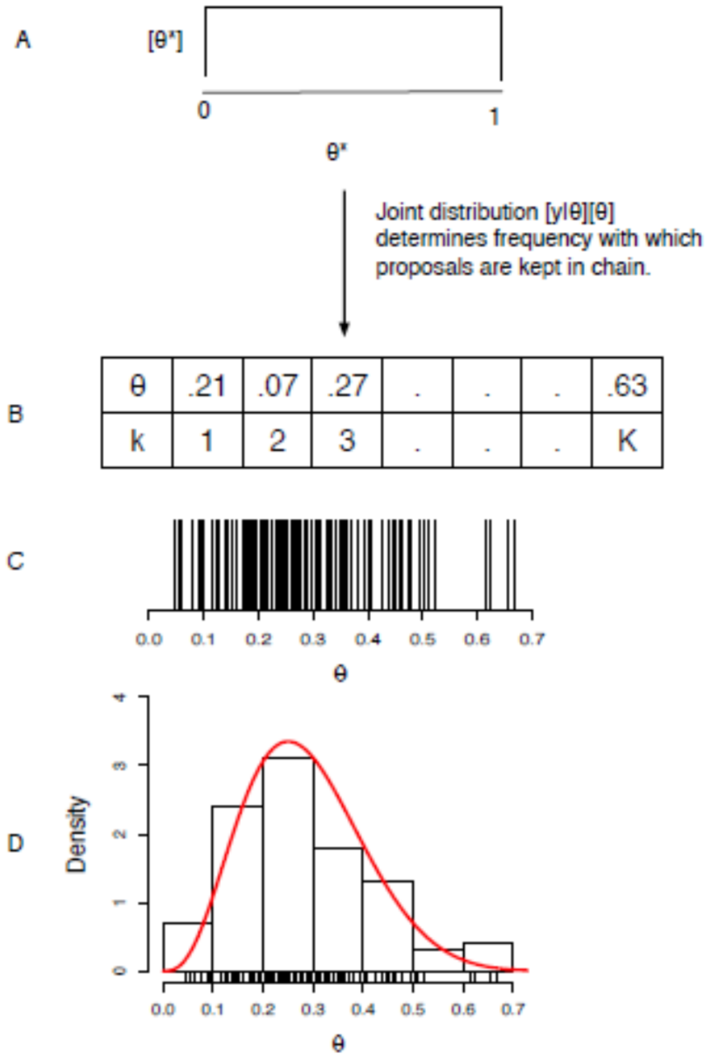
The marginal posterior distribution of each unobserved quantity is approximated by samples accumulated in the chain.

Chain: A sequence of values accumulated from random draws from the posterior distribution.





# What are we doing in MCMC



Joint distribution [y|θ][θ] determines frequency with which proposals are kept in chain.

| θ | .21 | .07 | .27 | . | . | . | .63 |
|---|---|---|---|---|---|---|---|
| k | 1 | 2 | 3 | . | . | . | K |

# Algorithms for drawing samples from the posterior

1. Accept-reject samplers
    1. Metropolis: requires a symmetric proposal distribution (e.g., normal, uniform)
    2. Metropolis-Hastings: allows asymmetric proposal distributions (e.g., beta, gamma, lognormal)
2. Gibbs: accepts all proposals because they come directly from the posterior using conjugates.

# Metropolis updates

We keep the more probable members of the posterior distribution by comparing a proposal with the current value in the chain.

| $k$ | 1 | 2 |
|---|---|---|
| Proposal $\theta^{*k+1}$ | | $\theta^{*2}$ |
| Test | | $P(\theta^{*2}) > P(\theta^{1})$ |
| Chain($\theta^{k}$) | $\theta^{1}$ | $\theta^{2} = \theta^{*2}$ |

# Metropolis updates

We keep the more probable members of the posterior distribution by comparing a proposal with the current value in the chain.

| $k$ | 1 | 2 | 3 |
|---|---|---|---|
| Proposal $\theta^{*k+1}$ | | $\theta^{*2}$ | $\theta^{*3}$ |
| Test | | $P(\theta^{*2}) > P(\theta^{1})$ | $P(\theta^{2}) > P(\theta^{*3})$ |
| Chain($\theta^{k}$) | $\theta^{1}$ | $\theta^{2} = \theta^{*2}$ | $\theta^{3} = \theta^{2}$ |

# Metropolis updates

We keep the more probable members of the posterior distribution by comparing a proposal with the current value in the chain.

| $k$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Proposal $\theta^{*k+1}$ |  | $\theta^{*2}$ | $\theta^{*3}$ | $\theta^{*4}$ |
| Test |  | $P(\theta^{*2}) > P(\theta^{1})$ | $P(\theta^{2}) > P(\theta^{*3})$ | $P(\theta^{3}) > P(\theta^{*4})$ |
| Chain($\theta^{k}$) | $\theta^{1}$ | $\theta^{2} = \theta^{*2}$ | $\theta^{3} = \theta^{2}$ | $\theta_{4} = \theta_{3}$ |

# Metropolis updates

$$[\boldsymbol{\theta}^{*k+1}|y] = \frac{\overbrace{[y|\boldsymbol{\theta}^{*k+1}]}^{\text{likelihood}}\overbrace{[\boldsymbol{\theta}^{*k+1}]}^{\text{prior}}}{\int [y|\boldsymbol{\theta}][\boldsymbol{\theta}]\,d\boldsymbol{\theta}}$$

$$[\boldsymbol{\theta}^{k}|y] = \frac{\overbrace{[y|\boldsymbol{\theta}^{k}]}^{\text{likelihood}}\;\overbrace{[\boldsymbol{\theta}^{k}]}^{\text{prior}}}{\int [y|\boldsymbol{\theta}][\boldsymbol{\theta}]\,d\boldsymbol{\theta}}$$

$$R = \frac{[\boldsymbol{\theta}^{*k+1}|y]}{[\boldsymbol{\theta}^{k}|y]}$$

# When do we keep the proposal?

$$P_R = \min(1, R)$$

Keep $\theta^{*k+1}$ as the next value in the chain with probability $P_R$ and keep $\theta^k$ with probability $1 - P_R$.

## When do we keep the proposal?

1. Calculate $R$ based on likelihoods and priors.
2. Draw a random number, $U$ from uniform distribution 0,1 If $R > U$, we keep the proposal $\theta^{*k+1}$ as the next value in the chain.
3. Otherwise, we retain $\theta^k$ as the next value.

# A simple example for one parameter

Grace is interested in estimating the prevalence of *Chytrid* fungus in a population of frogs.

She is sort of lazy, so she only samples 12 of them, of which 3 have the fungus.

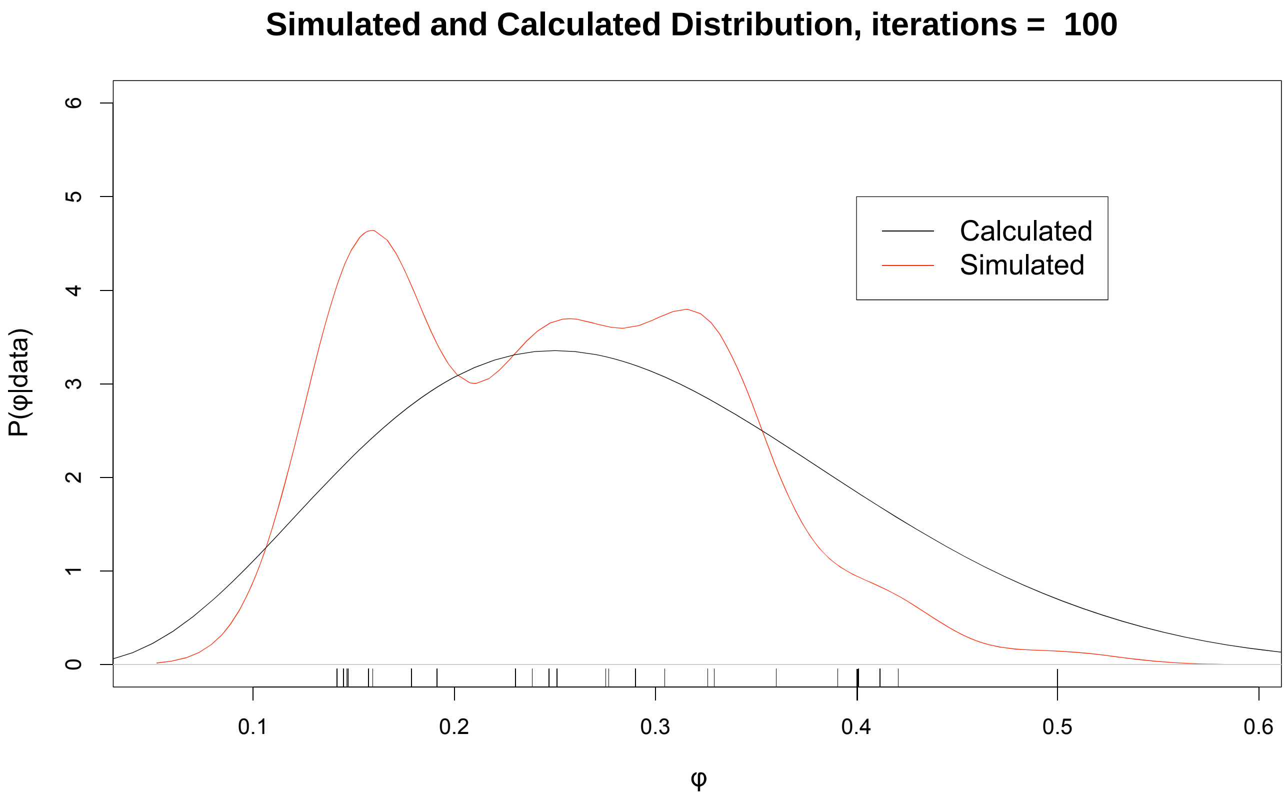
What is her best estimate of prevalence?

How would she calculate the parameters of the posterior on the back of a cocktail napkin?

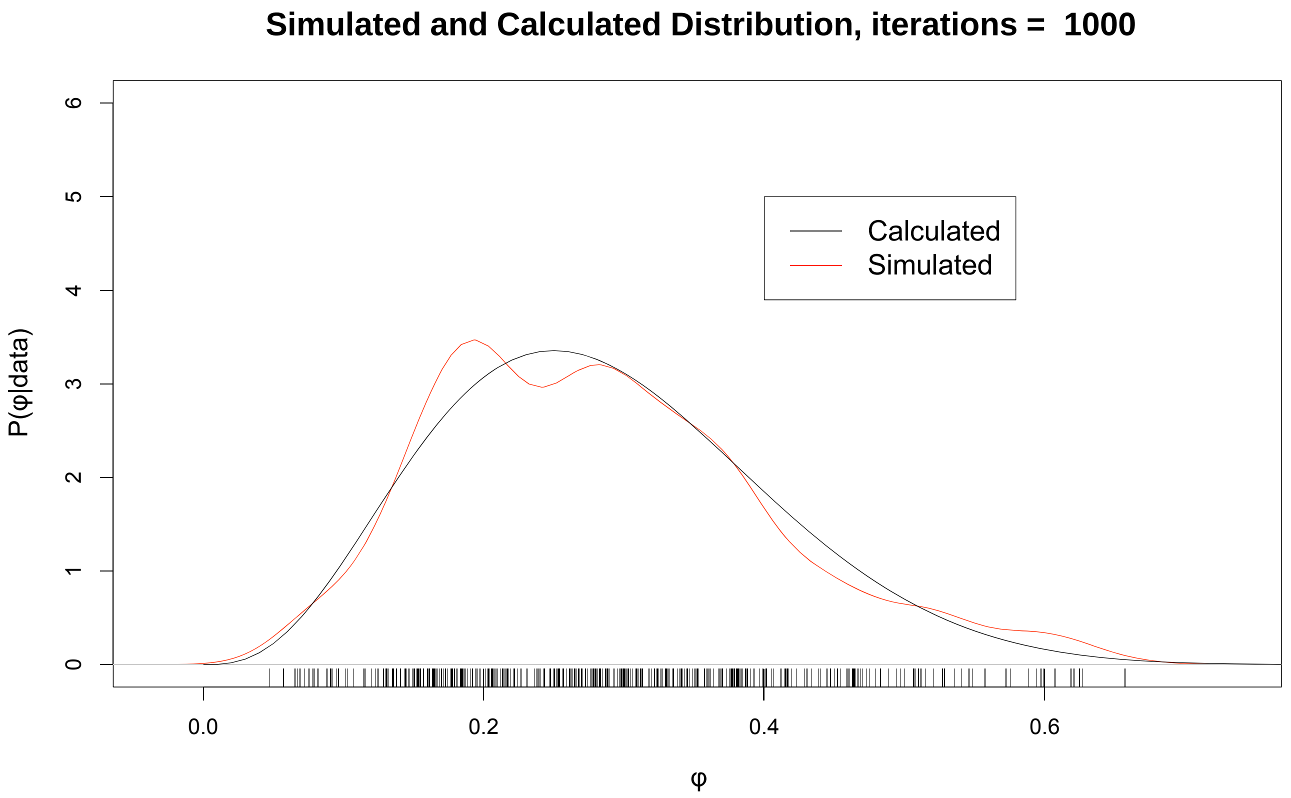
# The model

$$[\varphi|y] \propto \text{binomial}(y|n,\varphi)\text{beta}(\varphi|1,1)$$

# Simulation

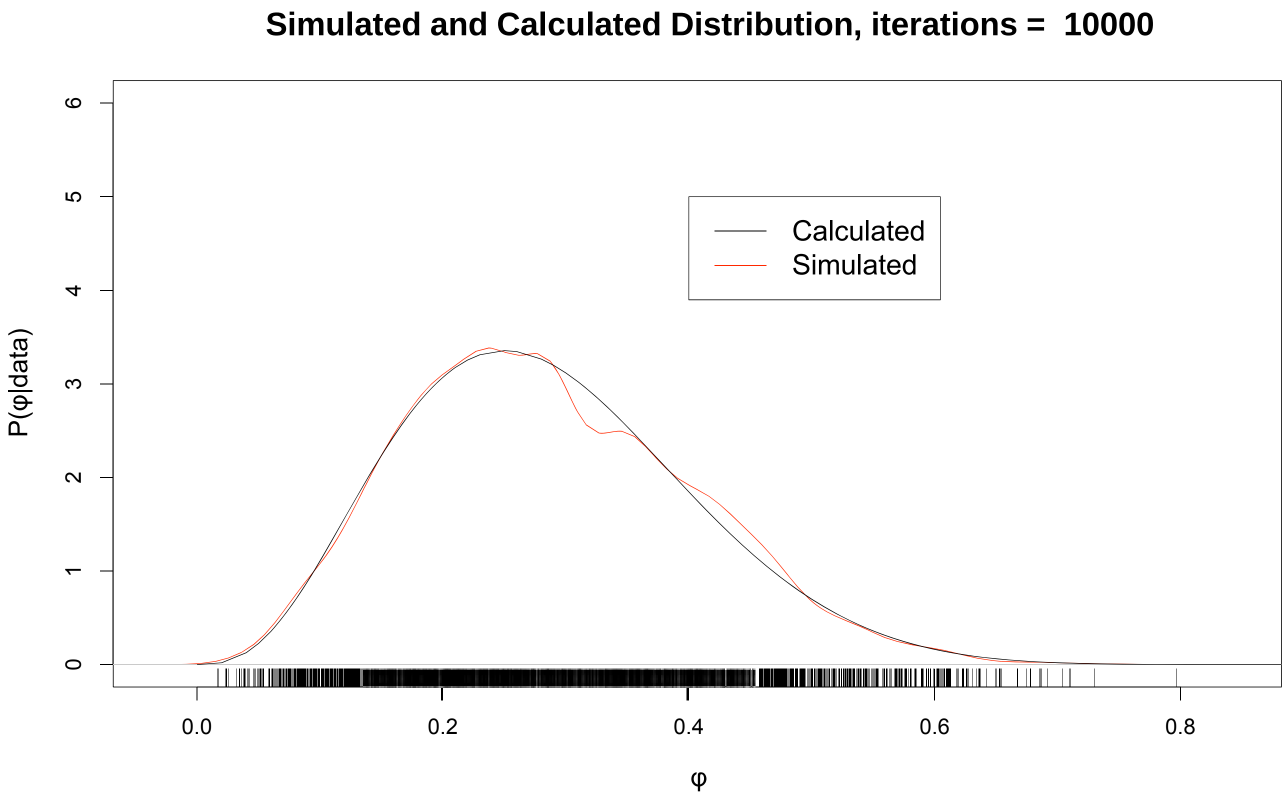

Simulated and Calculated Distribution, iterations = 100
P(φ|data)
0
1
2
3
4
5
6
0.1
0.2
0.3
0.4
0.5
0.6
φ
Calculated
Simulated

# Simulation

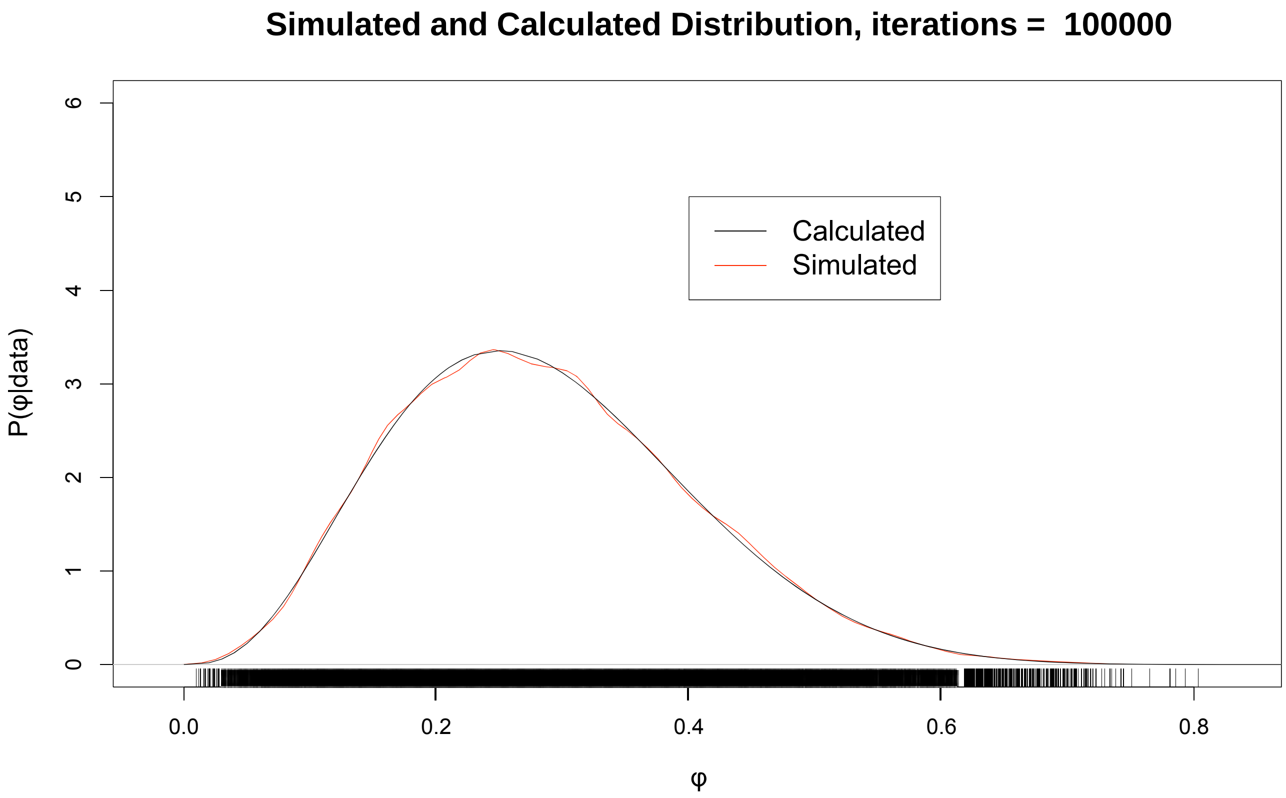


Simulated and Calculated Distribution, iterations =  1000
Calculated
Simulated
P(φ|data)
0
1
2
3
4
5
6
0.0
0.2
0.4
0.6
φ

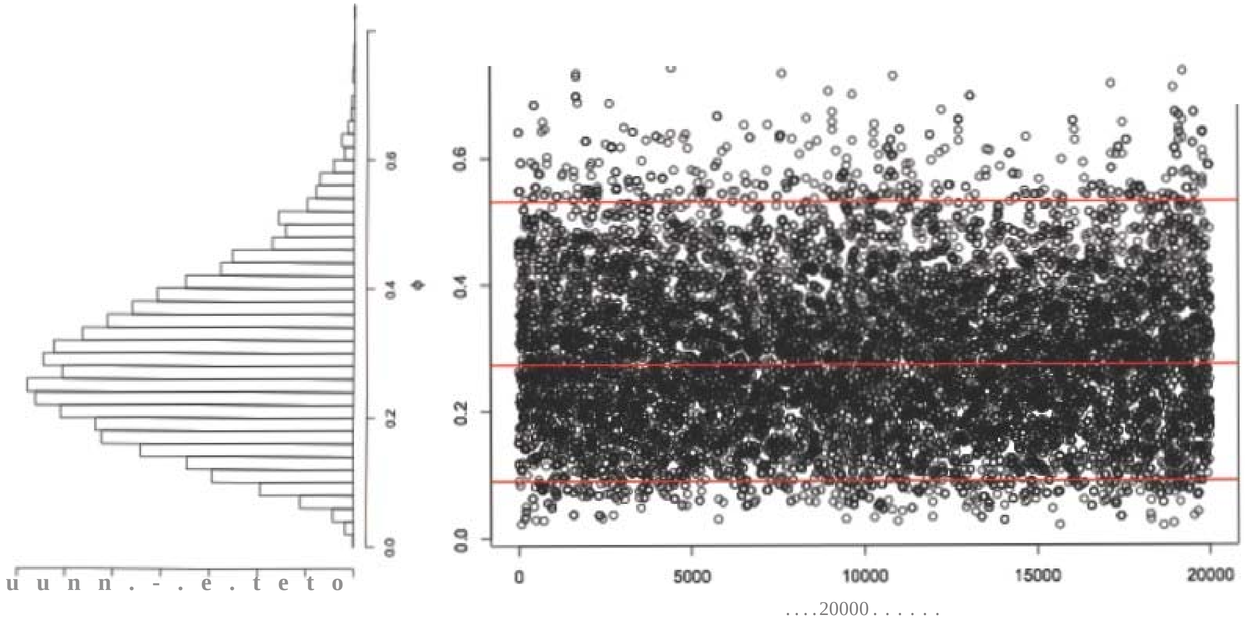
# Simulation


Simulated and Calculated Distribution, iterations = 10000
Calculated
Simulated
P(φ|data)
0
1
2
3
4
5
6
0.0
0.2
0.4
0.6
0.8
φ

# Simulation

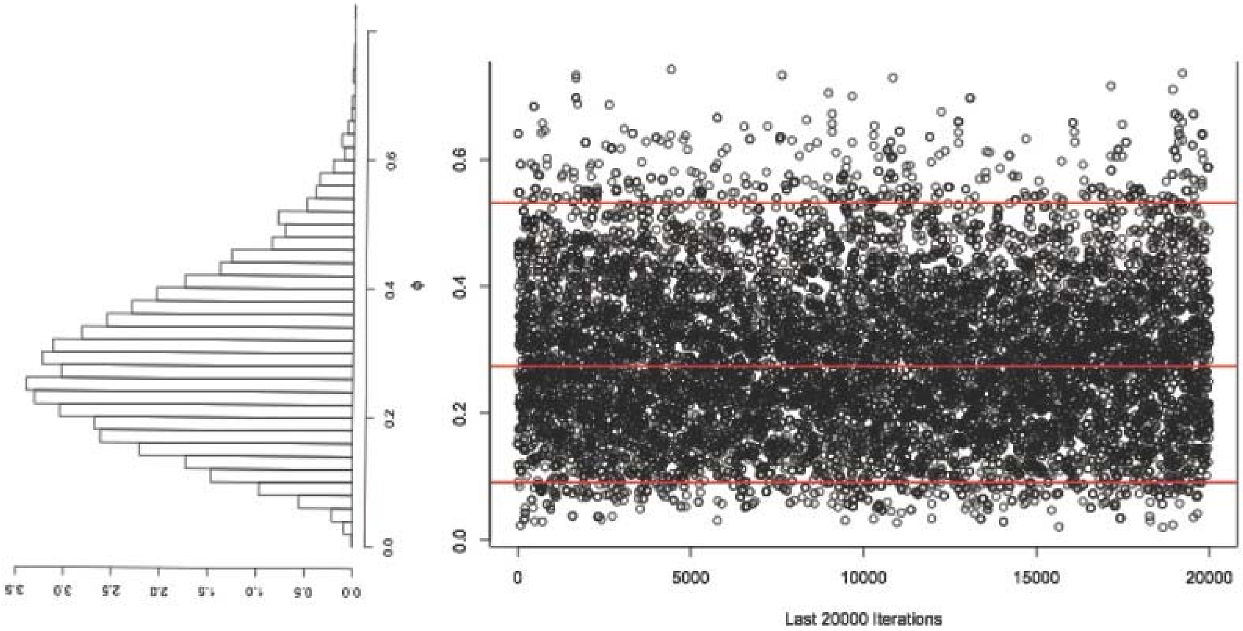

Simulated and Calculated Distribution, iterations = 100000
Calculated
Simulated
P(φ|data)
0
1
2
3
4
5
6
0.0
0.2
0.4
0.6
0.8
φ

# Simulation



0.0
0.2
0.4
0.6
ϕ
0.0
0.2
0.4
0.6
0
5000
10000
15000
20000
u u n n . - . e . t e t o
....20000......

## Simulation



The chain has *converged when* adding more samples does not change the shape of the posterior distribution. We throw away samples that are accumulated before convergence (burn-in).

# Multiple parameters and latent quantities

We write out an expression for the posterior and joint distribution using a DAG as a guide.

We decompose the expression of the multivariate joint distribution into a series of univariate distributions called *full-conditional distributions*. We choose a sampling method for each one.

We then cycle through each unobserved quantity, sampling from the its full-conditional distribution, treating the others as if they were known and constant.

Note that this takes a complex problem and turns it into a series of simple problems that we solve, as in the example above, one at a time.

# Multiple parameters and latent quantities

Let $\theta$ be a vector of length $k$ containing all of the unobserved quantities we seek to understand. Let $\theta_{-j}$ be a vector of length $k-1$ that contains all of the unobserved quantities *except* $\theta_j$. The full-conditional distribution of $\theta_j$ is

$$[\theta_j | y, \theta_{-j}],$$

which we notate as

$$[\theta_j | \cdot].$$

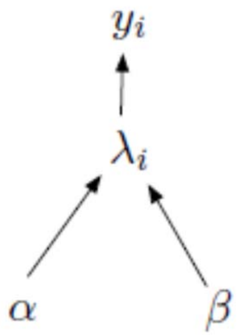
It is the posterior distribution of $\theta_j$ conditional on all of the parameters and the data, which we assume are *known*.

# Example

Clark 2003 considered the problem of modeling fecundity of spotted owls and the implication of individual variation in fecundity for population growth rate.

Data were number of offspring produced by per pair of owls, sample size = 197.

# Example



$$\begin{aligned}[\boldsymbol{\lambda},\boldsymbol{\alpha},\boldsymbol{\beta}|\mathbf{y}] &\propto \prod_{i=1}^{n}\mathsf{Poisson}\left(y_i|\boldsymbol{\lambda}_i\right)\mathsf{gamma}\left(\boldsymbol{\lambda}_i|\boldsymbol{\alpha},\boldsymbol{\beta}\right)\\ &\times \mathsf{gamma}\left(\boldsymbol{\alpha}|.001,.001\right)\mathsf{gamma}\left(\boldsymbol{\beta}|.001,.001\right)\end{aligned}$$

# Example

Posterior and joint:

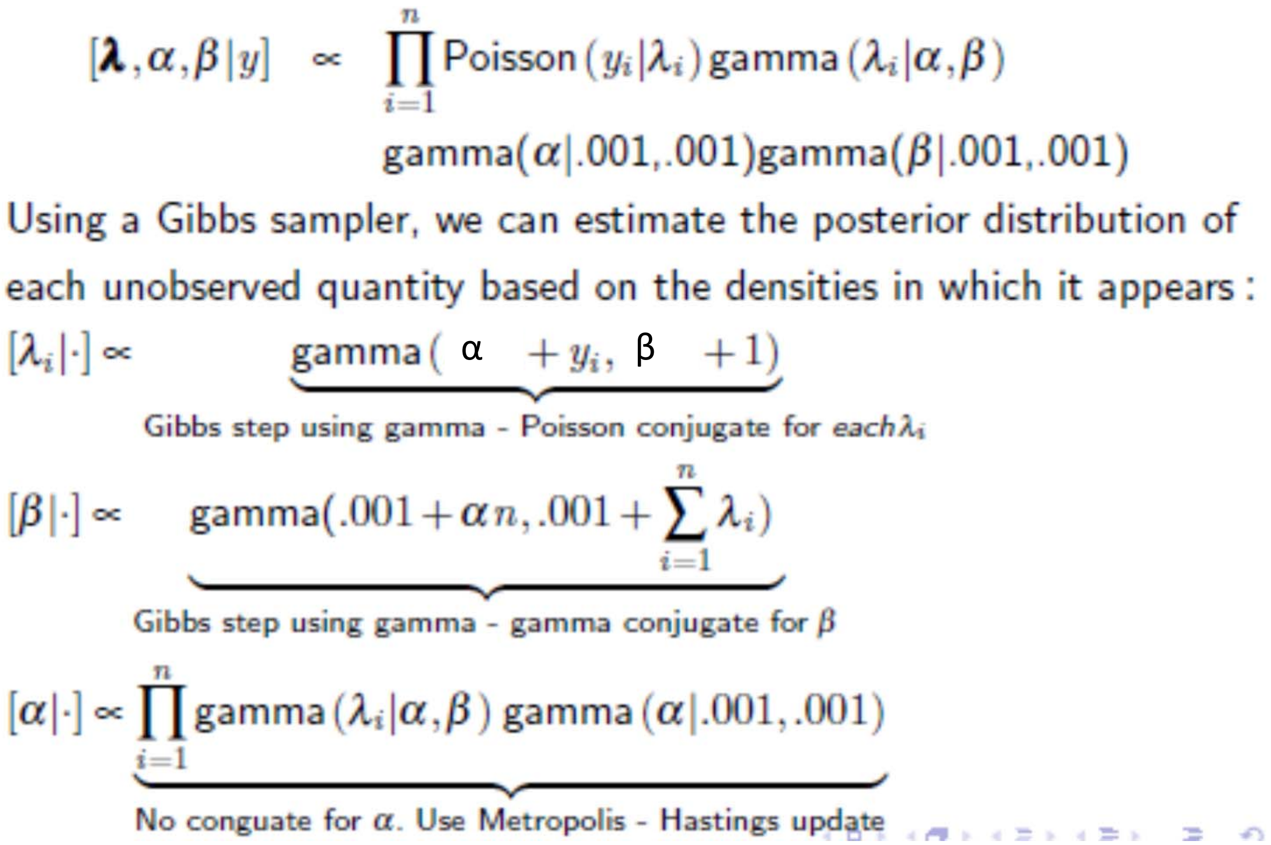
$$
\begin{aligned}
[\lambda, \alpha, \beta \mid \mathbf{y}] &\propto \prod_{i=1}^{n} \text{Poisson}(y_i \mid \lambda_i)\,\text{gamma}(\lambda_i \mid \alpha, \beta) \\
&\times \text{gamma}(\alpha \mid .001, .001)\,\text{gamma}(\beta \mid .001, .001)
\end{aligned}
$$

Full conditionals:

$$[\lambda \mid .] \propto \prod_{i=1}^{n} \text{Poisson}(y_i \mid \lambda_i)\,\text{gamma}(\lambda_i \mid \alpha, \beta)$$

$$[\beta \mid .] \propto \prod_{i=1}^{n} \text{gamma}(\lambda_i \mid \alpha, \beta)\,\text{gamma}(\beta \mid .001, .001)$$

$$[\alpha \mid .] \propto \prod_{i=1}^{n} \text{gamma}(\lambda_i \mid \alpha, \beta)\,\text{gamma}(\alpha \mid .001, .001)$$



$$
\begin{aligned}
[\boldsymbol{\lambda},\alpha,\beta|y] \;\propto\; & \prod_{i=1}^{n}\text{Poisson}(y_i|\lambda_i)\,\text{gamma}(\lambda_i|\alpha,\beta)\\
& \text{gamma}(\alpha|.001,.001)\text{gamma}(\beta|.001,.001)
\end{aligned}
$$

Using a Gibbs sampler, we can estimate the posterior distribution of each unobserved quantity based on the densities in which it appears :

$$[\lambda_i|\cdot] \propto \underbrace{\text{gamma}(\alpha + y_i,\ \beta + 1)}_{\text{Gibbs step using gamma - Poisson conjugate for } each\lambda_i}$$

$$[\beta|\cdot] \propto \underbrace{\text{gamma}(.001+\alpha n, .001+\sum_{i=1}^{n}\lambda_i)}_{\text{Gibbs step using gamma - gamma conjugate for } \beta}$$

$$[\alpha|\cdot] \propto \underbrace{\prod_{i=1}^{n}\text{gamma}(\lambda_i|\alpha,\beta)\,\text{gamma}(\alpha|.001,.001)}_{\text{No conguate for } \alpha\text{. Use Metropolis - Hastings update}}$$

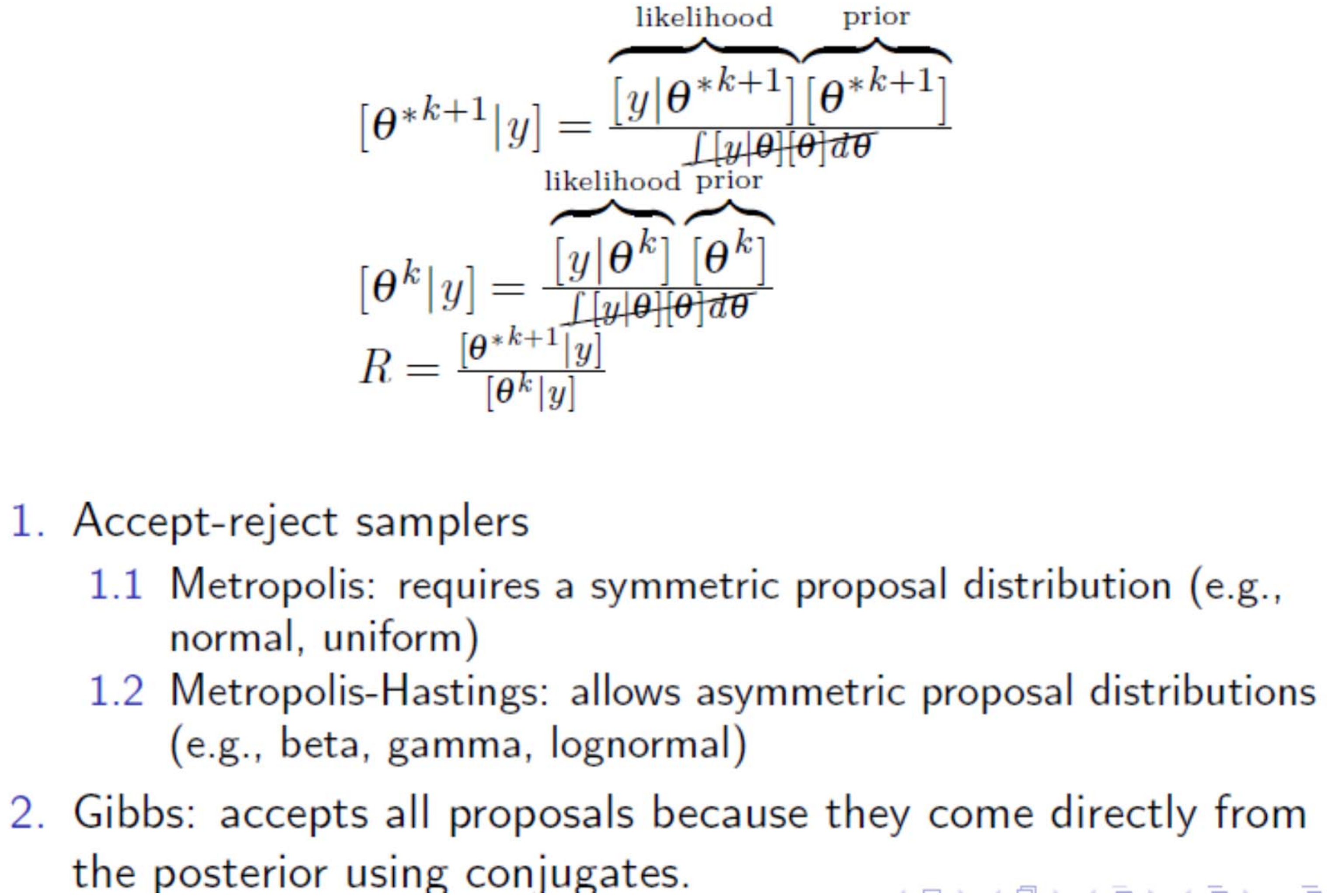
# Metropolis updates

$$[\theta^{*k+1}|y] = \frac{\overbrace{[y|\theta^{*k+1}]}^{\text{likelihood}}\overbrace{[\theta^{*k+1}]}^{\text{prior}}}{\int [y|\theta][\theta]\,d\theta}$$

$$[\theta^{k}|y] = \frac{\overbrace{[y|\theta^{k}]}^{\text{likelihood}}\ \overbrace{[\theta^{k}]}^{\text{prior}}}{\int [y|\theta][\theta]\,d\theta}$$

$$R = \frac{[\theta^{*k+1}|y]}{[\theta^{k}|y]}$$

# Metropolis updates



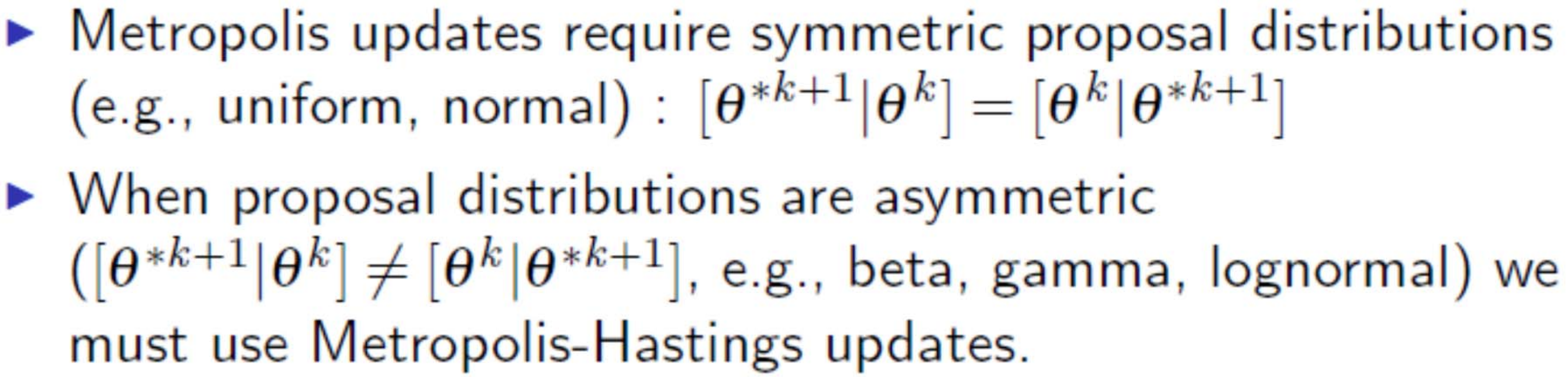
$$[\boldsymbol{\theta}^{*k+1}|y] = \frac{\overbrace{[y|\boldsymbol{\theta}^{*k+1}]}^{\text{likelihood}}\overbrace{[\boldsymbol{\theta}^{*k+1}]}^{\text{prior}}}{\cancel{\int[y|\boldsymbol{\theta}][\boldsymbol{\theta}]d\boldsymbol{\theta}}}$$

$$[\boldsymbol{\theta}^{k}|y] = \frac{\overbrace{[y|\boldsymbol{\theta}^{k}]}^{\text{likelihood}}\overbrace{[\boldsymbol{\theta}^{k}]}^{\text{prior}}}{\cancel{\int[y|\boldsymbol{\theta}][\boldsymbol{\theta}]d\boldsymbol{\theta}}}$$

$$R = \frac{[\boldsymbol{\theta}^{*k+1}|y]}{[\boldsymbol{\theta}^{k}|y]}$$

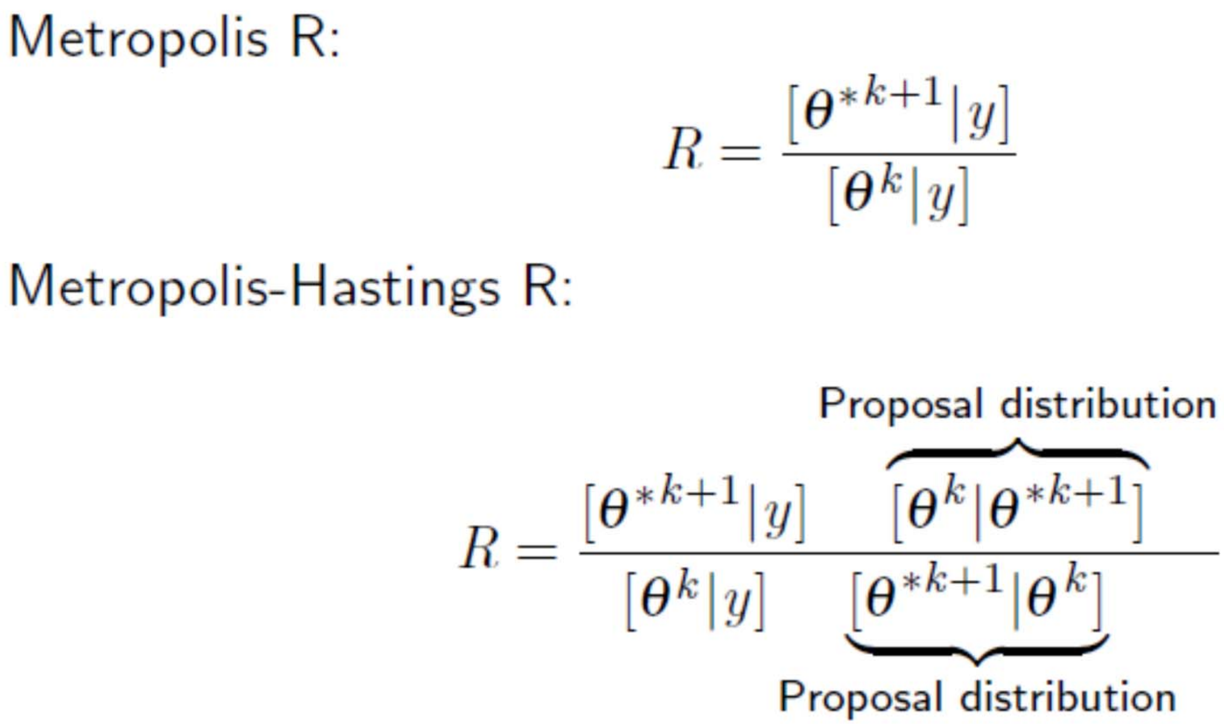
1. Accept-reject samplers
   1.1 Metropolis: requires a symmetric proposal distribution (e.g., normal, uniform)
   1.2 Metropolis-Hastings: allows asymmetric proposal distributions (e.g., beta, gamma, lognormal)
2. Gibbs: accepts all proposals because they come directly from the posterior using conjugates.

# Metropolis-Hasting updates



- Metropolis updates require symmetric proposal distributions (e.g., uniform, normal) : $[\theta^{*k+1}|\theta^{k}] = [\theta^{k}|\theta^{*k+1}]$
- When proposal distributions are asymmetric ($[\theta^{*k+1}|\theta^{k}] \neq [\theta^{k}|\theta^{*k+1}]$, e.g., beta, gamma, lognormal) we must use Metropolis-Hastings updates.

# Metropolis-Hasting updates



Metropolis R:

$$R = \frac{[\theta^{*k+1}|y]}{[\theta^k|y]}$$

Metropolis-Hastings R:

$$R = \frac{[\theta^{*k+1}|y] \quad \overbrace{[\theta^k|\theta^{*k+1}]}^{\text{Proposal distribution}}}{[\theta^k|y] \quad \underbrace{[\theta^{*k+1}|\theta^k]}_{\text{Proposal distribution}}}$$

# Proposal distribution

Metropolis R:

$$R = \frac{[\boldsymbol{\theta}^{*k+1}|y]}{[\boldsymbol{\theta}^{k}|y]}$$

Metropolis-Hastings R:

$$R = \frac{[\boldsymbol{\theta}^{*k+1}|y] \quad \overbrace{[\boldsymbol{\theta}^{k}|\boldsymbol{\theta}^{*k+1}]}^{\text{Proposal distribution}}}{[\boldsymbol{\theta}^{k}|y] \quad \underbrace{[\boldsymbol{\theta}^{*k+1}|\boldsymbol{\theta}^{k}]}_{\text{Proposal distribution}}}$$

# Proposal distribution

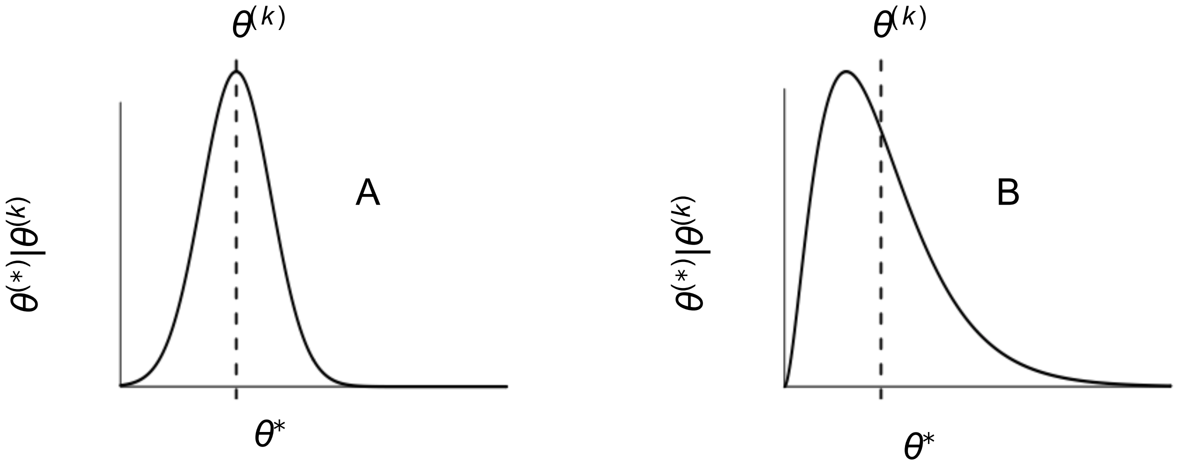
Independent chain have proposal distributions that do not depend on the current value ($\theta^k$) in the chain.

Dependent chains, as you might expect, have proposal distributions that *do* depend on the current value of the chain ($\theta^k$). In this case we draw from

$$[\theta^{*k+1}|\theta^k, \sigma]$$

where $\sigma$ is a tuning parameter that we specify to obtain an acceptance rate of about 40%. Note that my notation and notation of others simplifies this distribution to $[\theta^{*k+1}|\theta^k]$ The sigma is implicit because it is a constant, not a random variable.

# Proposal distribution



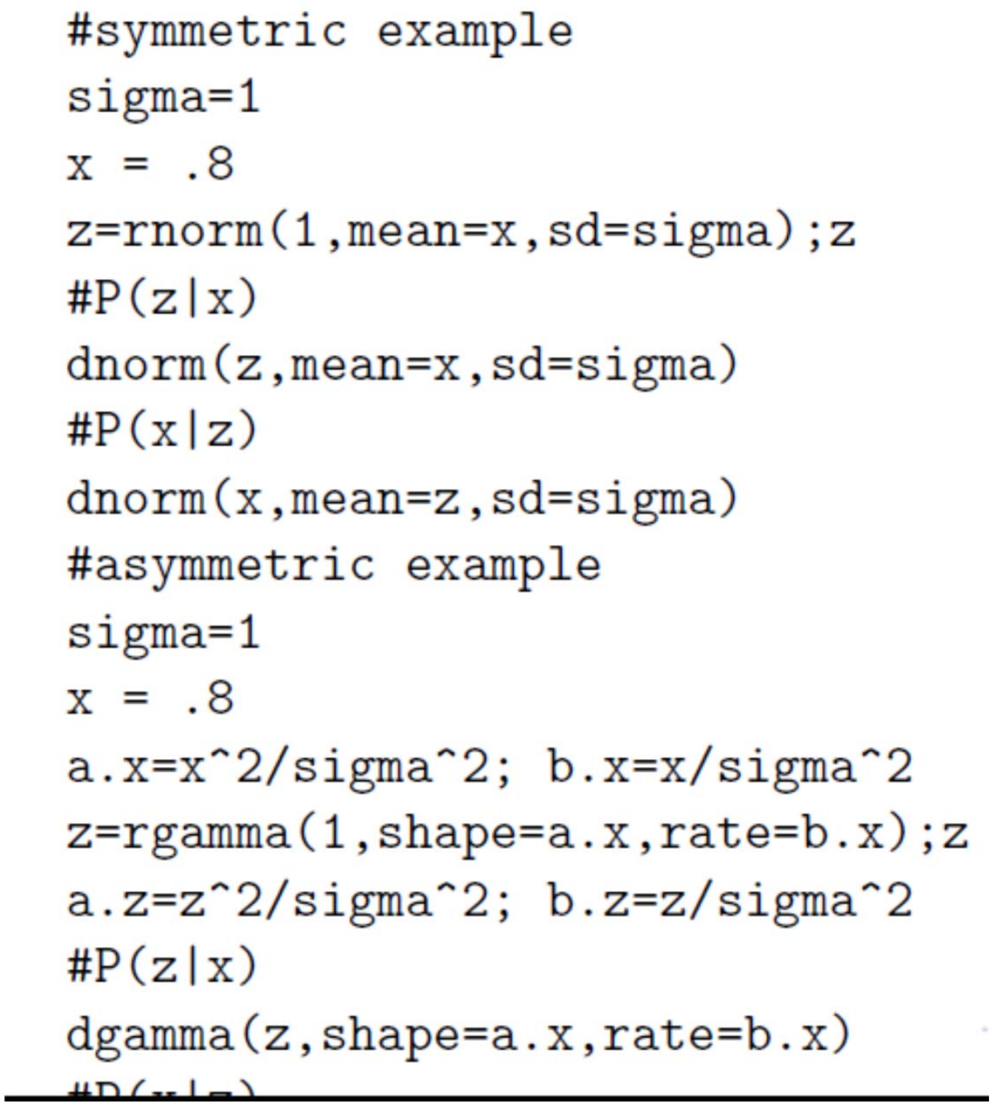
# Definition of symmetry

A proposal distribution is symmetric if and only if

$$[\theta^{*k+1}|\theta^{k}] = [\theta^{k}|\theta^{*k+1}]. \tag{4}$$

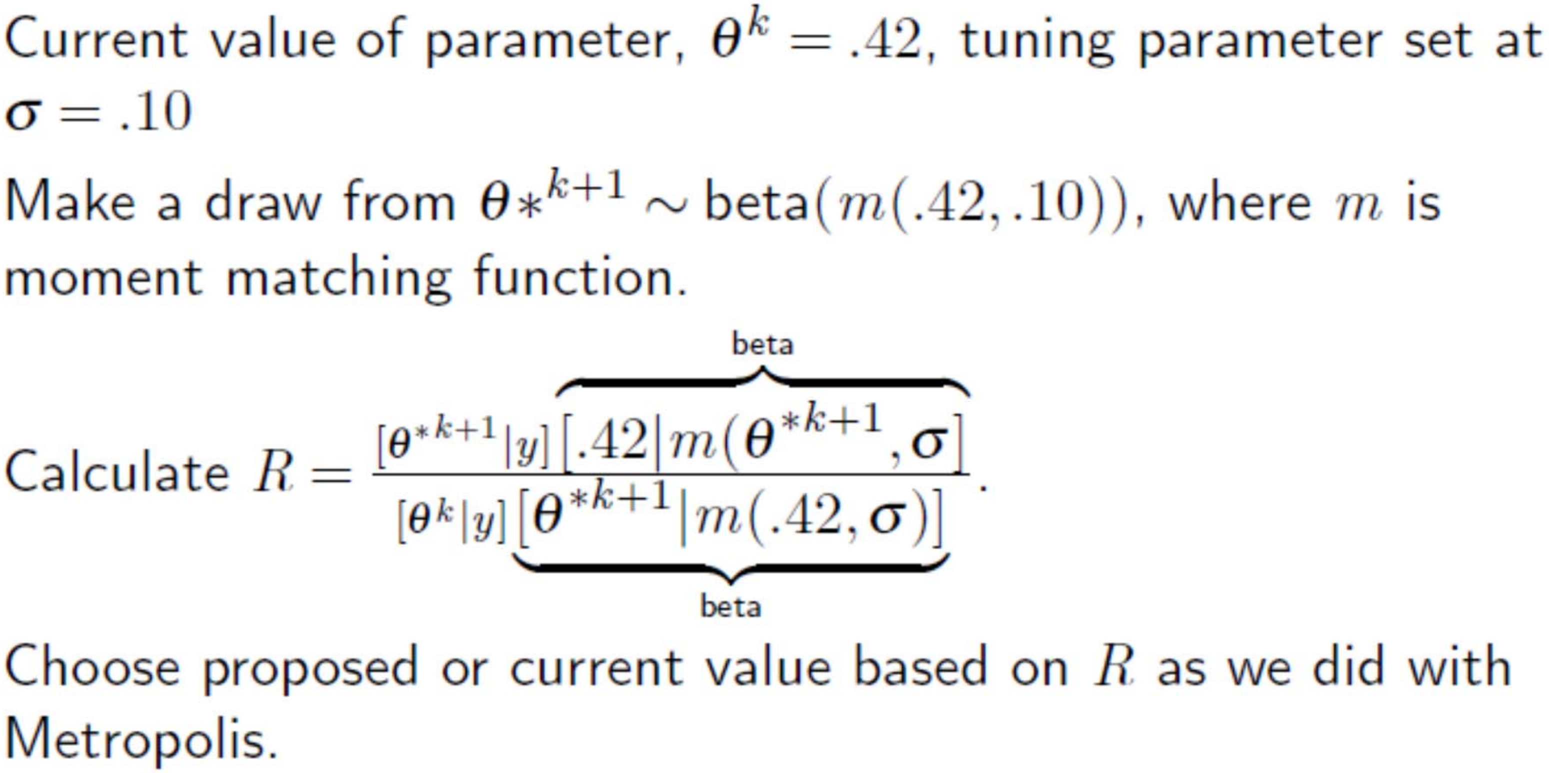
Normal and uniform are symmetric. Gamma, beta, lognormal are not.

## Illustrating with code



```
#symmetric example
sigma=1
x = .8
z=rnorm(1,mean=x,sd=sigma);z
#P(z|x)
dnorm(z,mean=x,sd=sigma)
#P(x|z)
dnorm(x,mean=z,sd=sigma)
#asymmetric example
sigma=1
x = .8
a.x=x^2/sigma^2; b.x=x/sigma^2
z=rgamma(1,shape=a.x,rate=b.x);z
a.z=z^2/sigma^2; b.z=z/sigma^2
#P(z|x)
dgamma(z,shape=a.x,rate=b.x)
#P(x|z)
```

# Example using beta proposal distribution



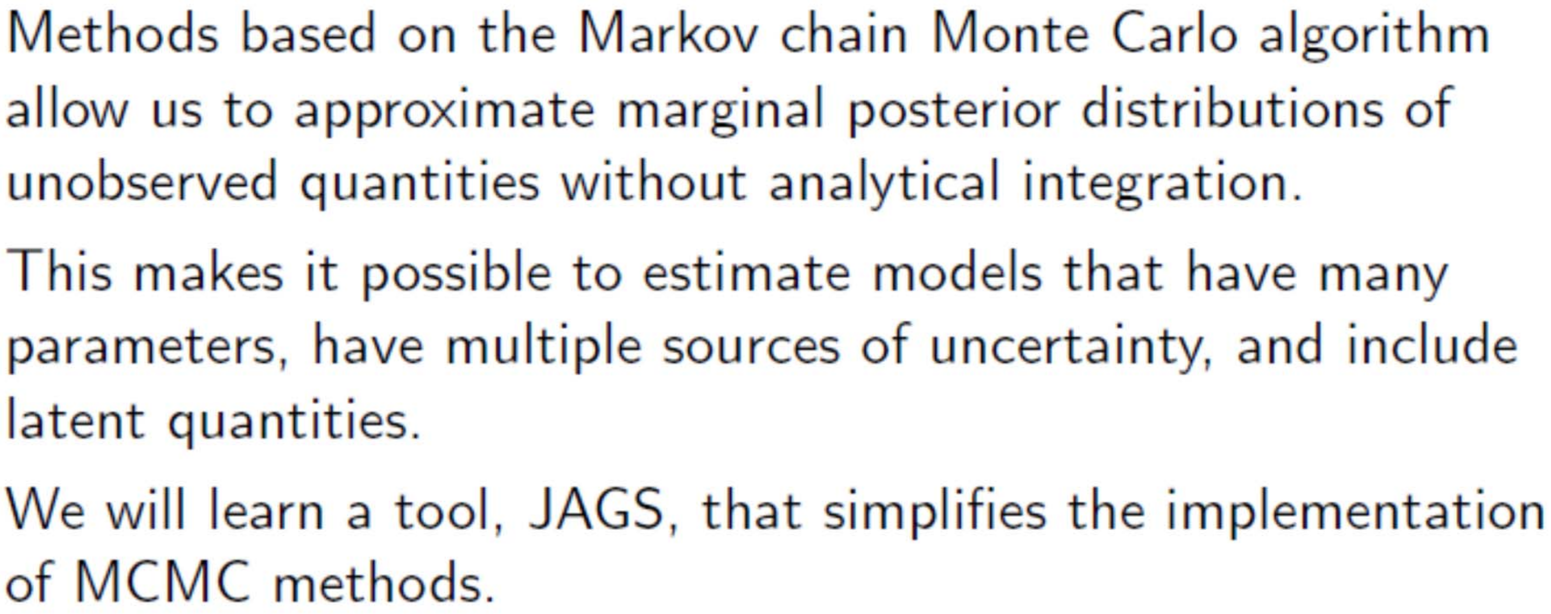
Current value of parameter, $\theta^k = .42$, tuning parameter set at $\sigma = .10$

Make a draw from $\theta{*}^{k+1} \sim \text{beta}(m(.42, .10))$, where $m$ is moment matching function.

Calculate $R = \dfrac{[\theta^{*k+1}|y]\overbrace{[.42|m(\theta^{*k+1}, \sigma]}^{\text{beta}}}{[\theta^k|y]\underbrace{[\theta^{*k+1}|m(.42, \sigma)]}_{\text{beta}}}.$

Choose proposed or current value based on $R$ as we did with Metropolis.

# MCMC



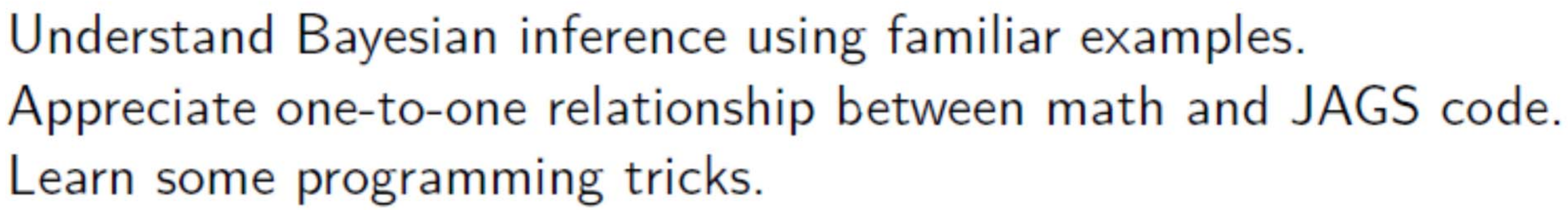
Methods based on the Markov chain Monte Carlo algorithm allow us to approximate marginal posterior distributions of unobserved quantities without analytical integration.

This makes it possible to estimate models that have many parameters, have multiple sources of uncertainty, and include latent quantities.

We will learn a tool, JAGS, that simplifies the implementation of MCMC methods.

# Bayesian Regression

# Outline

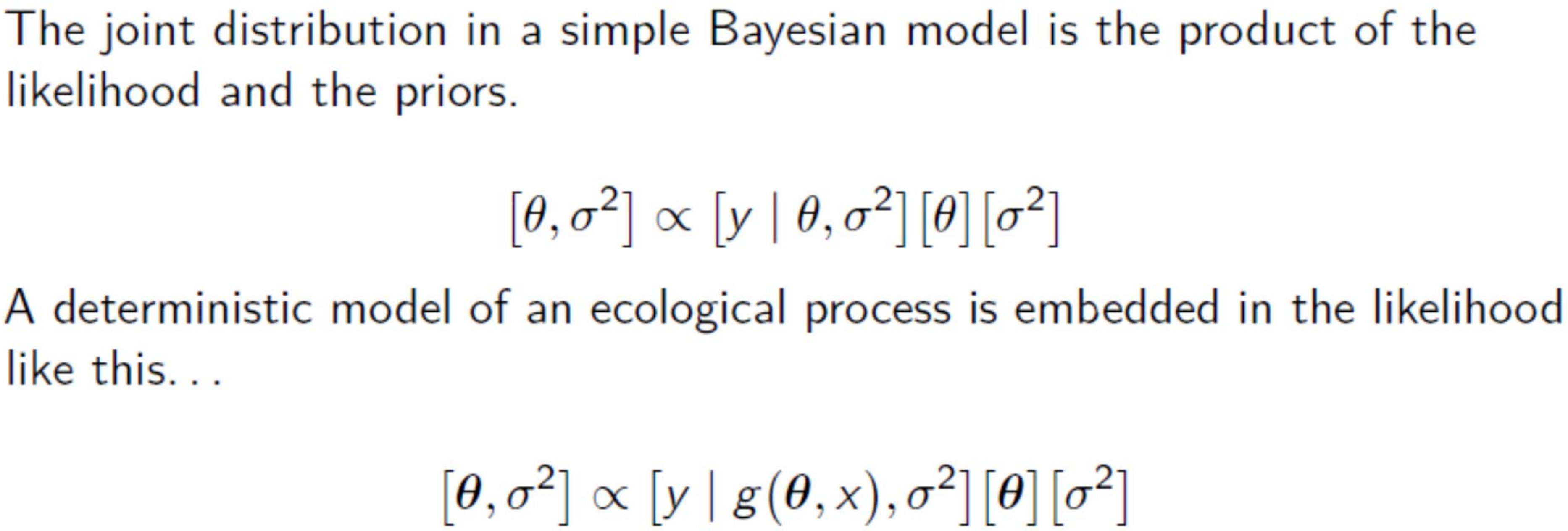


Understand Bayesian inference using familiar examples.
Appreciate one-to-one relationship between math and JAGS code.
Learn some programming tricks.

# Notation

$$\underbrace{g(\boldsymbol{\theta}, x)}_{\text{deterministic model}}$$

# Embedding the deterministic model



The joint distribution in a simple Bayesian model is the product of the likelihood and the priors.

$$[\theta, \sigma^2] \propto [y \mid \theta, \sigma^2]\,[\theta]\,[\sigma^2]$$

A deterministic model of an ecological process is embedded in the likelihood like this. . .

$$[\boldsymbol{\theta}, \sigma^2] \propto [y \mid g(\boldsymbol{\theta}, x), \sigma^2]\,[\boldsymbol{\theta}]\,[\sigma^2]$$

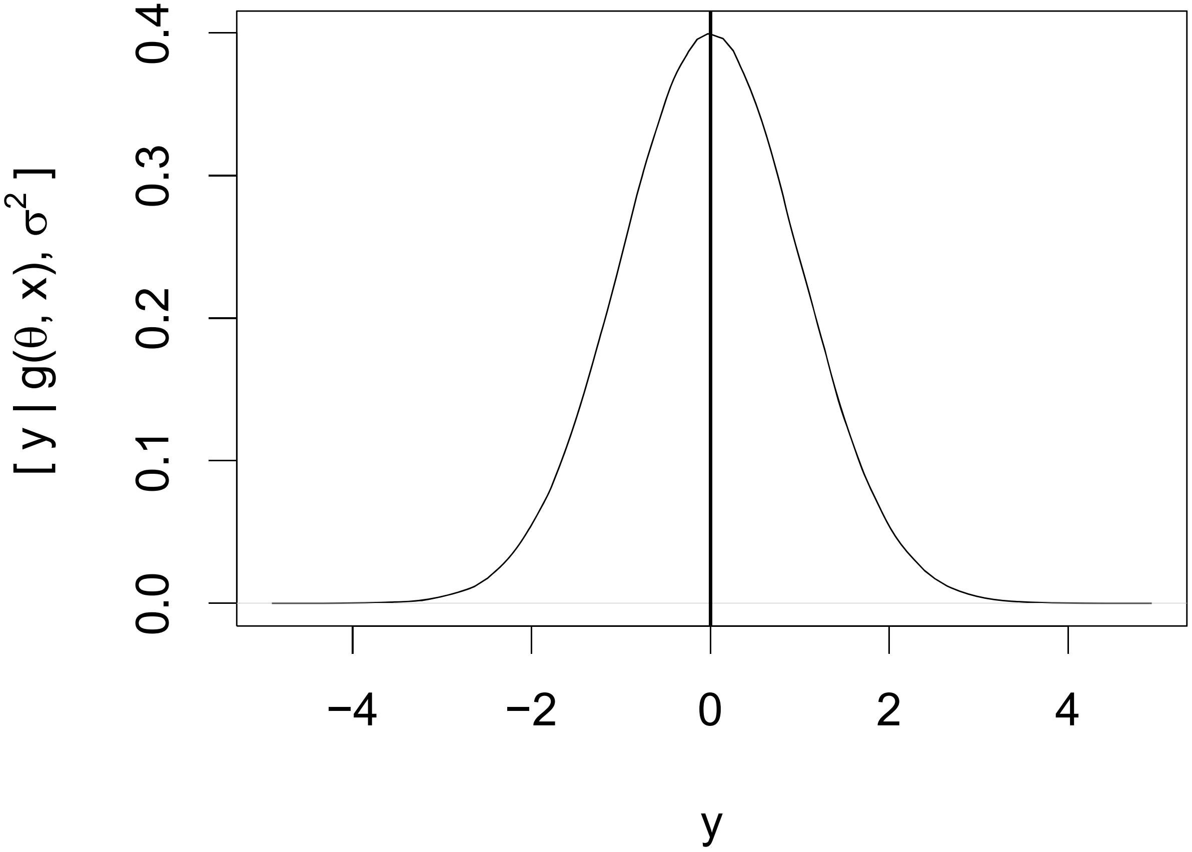
## The general set up for a simple Bayesian regression model

$$\mu_i = \underbrace{g(\boldsymbol{\theta}, x_i)}_{\text{deterministic model}}$$

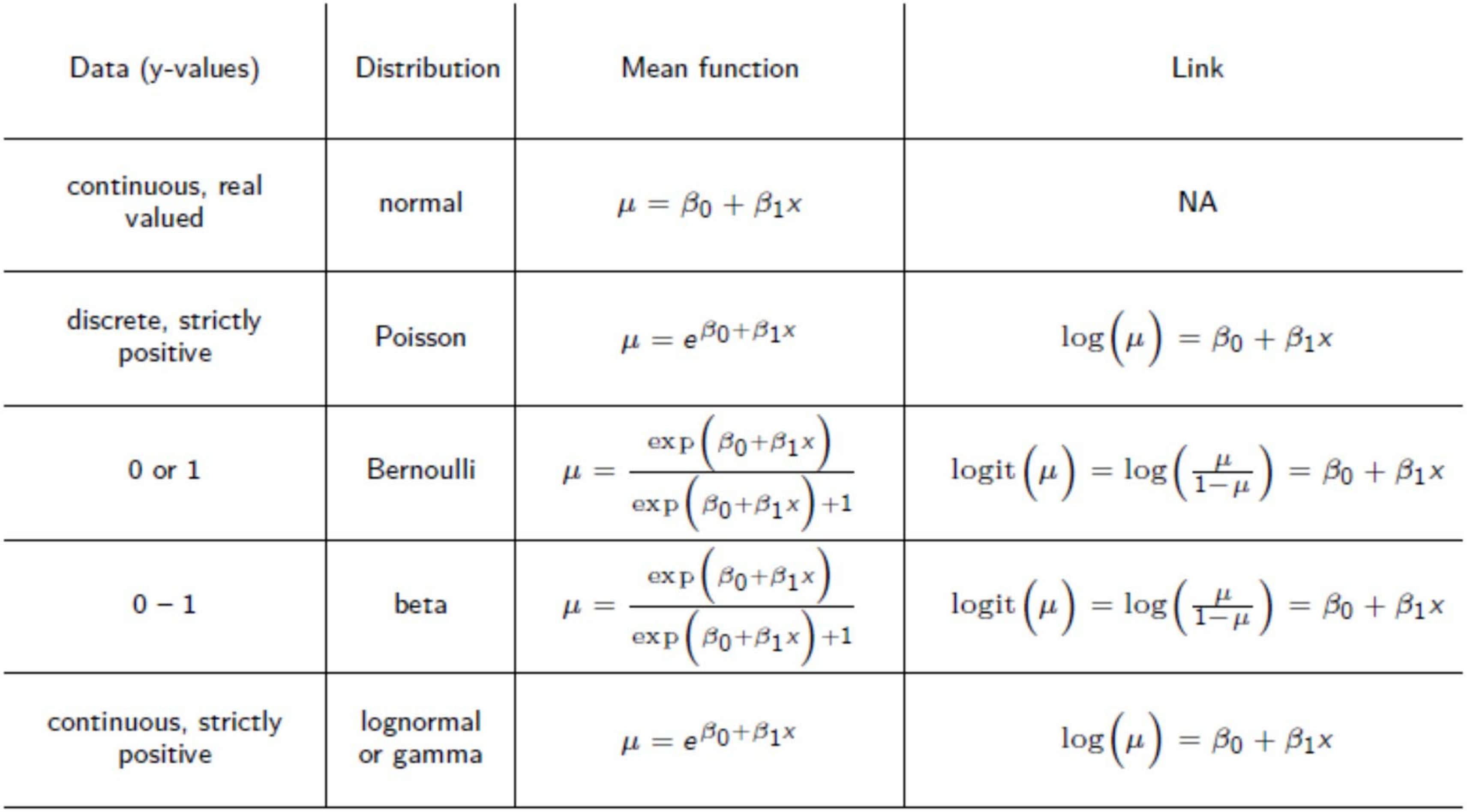
$$y_i \sim \underbrace{f(\mu_i, \sigma^2)}_{\text{stochastic model}}$$

$$[\boldsymbol{\theta}, \sigma^2] \propto [y \mid g(\boldsymbol{\theta}), \sigma^2] [\boldsymbol{\theta}] [\sigma^2]$$

# A graphical representation



# You don’t have to be normal!



| Data (y-values) | Distribution | Mean function | Link |
|---|---|---|---|
| continuous, real valued | normal | $\mu = \beta_0 + \beta_1 x$ | NA |
| discrete, strictly positive | Poisson | $\mu = e^{\beta_0+\beta_1 x}$ | $\log\left(\mu\right) = \beta_0 + \beta_1 x$ |
| 0 or 1 | Bernoulli | $\mu = \frac{\exp\left(\beta_0+\beta_1 x\right)}{\exp\left(\beta_0+\beta_1 x\right)+1}$ | $\operatorname{logit}\left(\mu\right) = \log\left(\frac{\mu}{1-\mu}\right) = \beta_0 + \beta_1 x$ |
| $0 - 1$ | beta | $\mu = \frac{\exp\left(\beta_0+\beta_1 x\right)}{\exp\left(\beta_0+\beta_1 x\right)+1}$ | $\operatorname{logit}\left(\mu\right) = \log\left(\frac{\mu}{1-\mu}\right) = \beta_0 + \beta_1 x$ |
| continuous, strictly positive | lognormal or gamma | $\mu = e^{\beta_0+\beta_1 x}$ | $\log\left(\mu\right) = \beta_0 + \beta_1 x$ |

# Lots of flexibility as a modeler

Continent-wide Adélie penguin population dynamics

$$z_{s,y} \sim \text{lognormal}\left(z_{s,y} \mid g(\beta_1, \beta_2, \beta_3, \beta_4, \epsilon_y, z_{s,y-1}), \sigma^{process}\right)$$

$$g(\theta) = \log\left(z_{s,y-1} e^{\beta_1 + \beta_2 \text{wsic}_{s,y} + \beta_3 \text{ssic}_{s,y} + \beta_4 \text{krill}_{s,y} + \epsilon_y}\right)$$

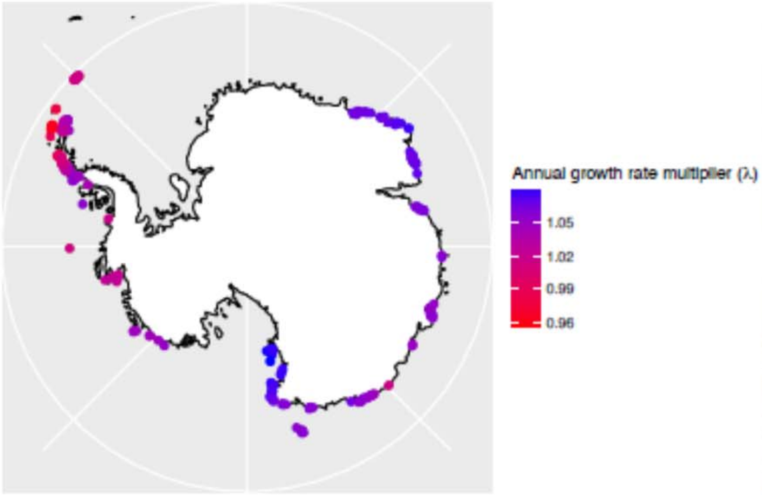
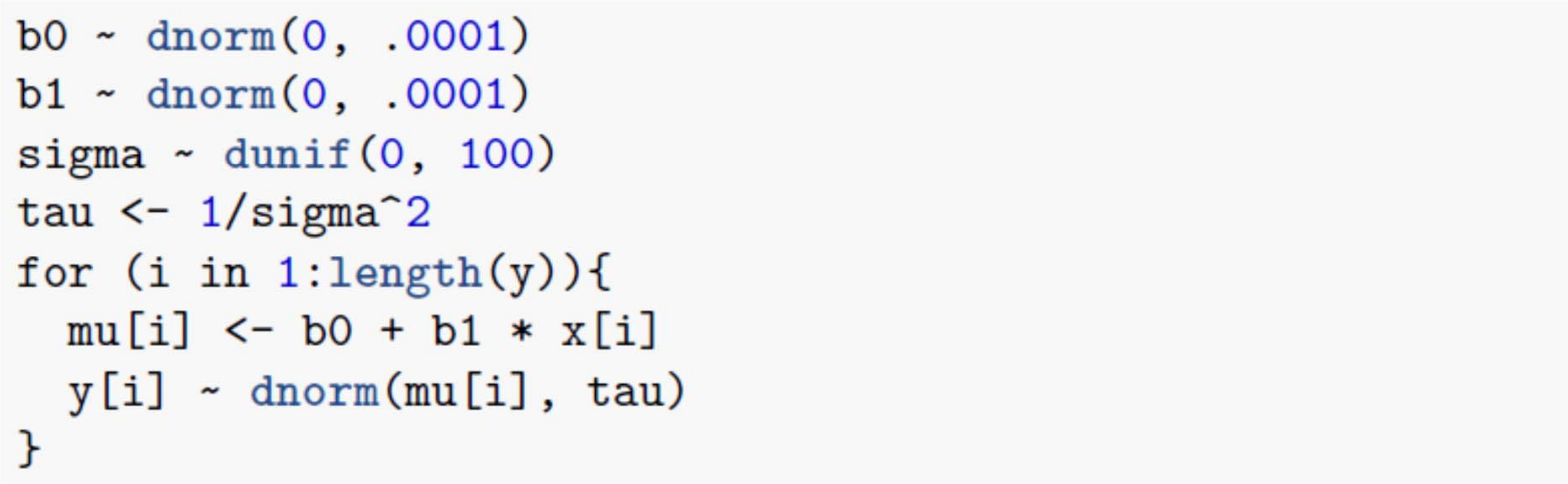


Photo c/o Heather J. Lynch

# Normal data, continuous and real valued

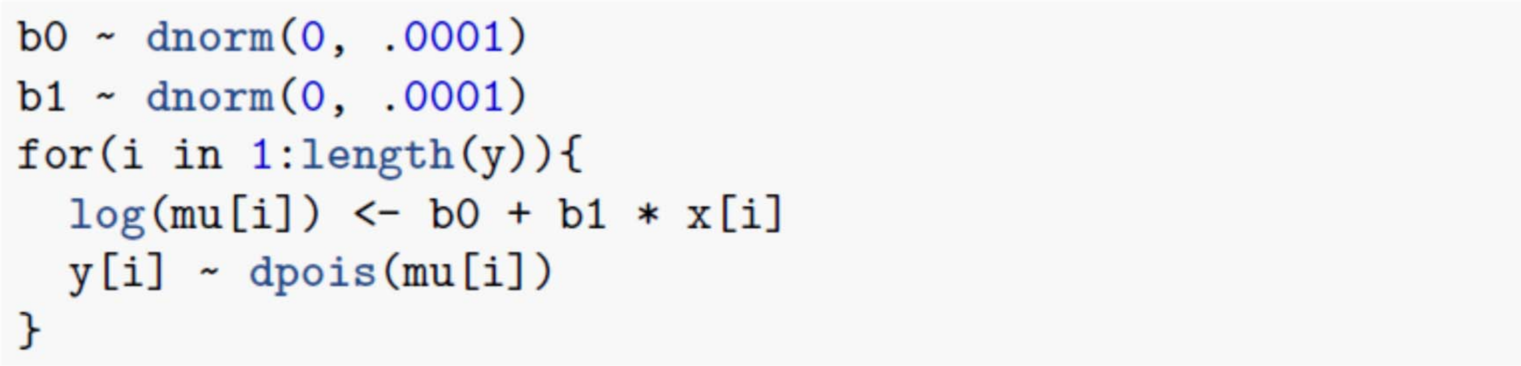
$$
\begin{aligned}
[\beta_0, \beta_1, \sigma \mid \mathbf{y}] \quad &\propto \quad \prod_{i=1}^{N} \text{normal}\big(y_i \mid g\big(\beta_0, \beta_1, x_i\big), \sigma^2\big) \times \\
&\quad\;\; \text{normal}\big(\beta_0 \mid 0, .001\big)\,\text{normal}\big(\beta_1 \mid 0, .001\big) \times \\
&\quad\;\; \text{uniform}\big(\sigma \mid 0, 100\big) \\
g\big(\beta_0, \beta_1, x_i\big) \quad &= \quad \beta_0 + \beta_1 x_i
\end{aligned}
$$



```
b0 ~ dnorm(0, .0001)
b1 ~ dnorm(0, .0001)
sigma ~ dunif(0, 100)
tau <- 1/sigma^2
for (i in 1:length(y)){
  mu[i] <- b0 + b1 * x[i]
  y[i] ~ dnorm(mu[i], tau)
}
```

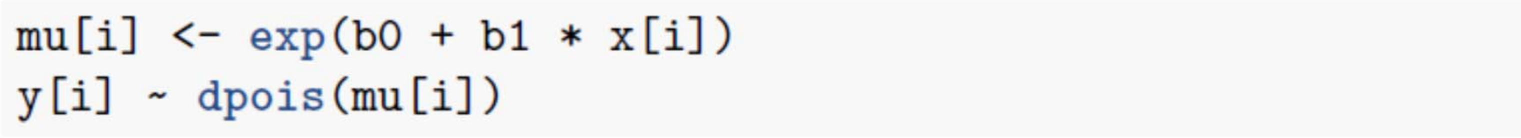
# Poisson, discrete and positive

$$
\begin{aligned}
[\beta_0, \beta_1 \mid \mathbf{y}] &\propto \prod_{i=1}^{N} \text{Poisson}\big(y_i \mid g\big(\beta_0, \beta_1, x_i\big)\big) \times \\
&\quad \text{normal}\big(\beta_0 \mid 0, .001\big)\,\text{normal}\big(\beta_1 \mid 0, .001\big) \\
g\big(\beta_0, \beta_1, x_i\big) &= e^{\beta_0 + \beta_1 x_i}
\end{aligned}
$$



```
b0 ~ dnorm(0, .0001)
b1 ~ dnorm(0, .0001)
for(i in 1:length(y)){
  log(mu[i]) <- b0 + b1 * x[i]
  y[i] ~ dpois(mu[i])
}
```

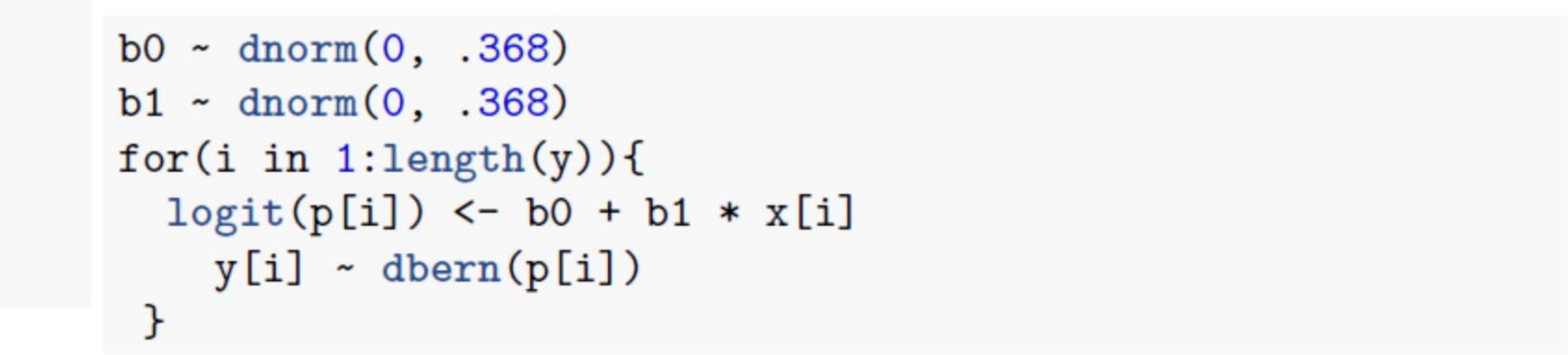
or



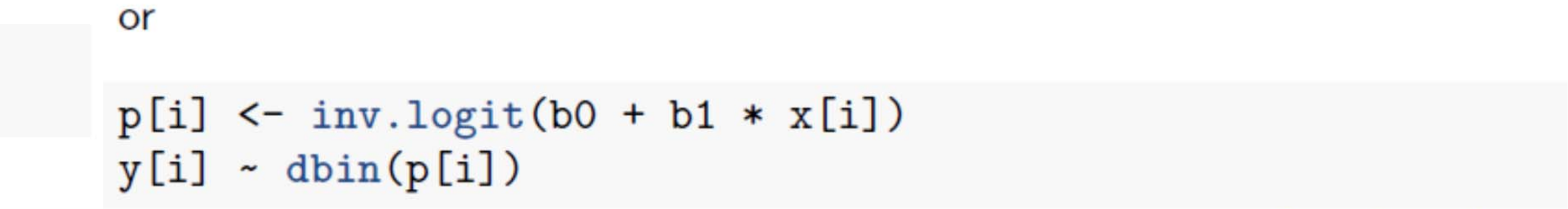
```
mu[i] <- exp(b0 + b1 * x[i])
y[i] ~ dpois(mu[i])
```

# Bernoulli, data 0 or 1 (aka logistic)

$$\begin{aligned}[\beta_0, \beta_1 \mid \mathbf{y}] &\propto \prod_{i=1}^{N} \text{Bernoulli}(y_i \mid g(\beta_0, \beta_1, x_i)) \times \\ &\quad \text{normal}(\beta_0 \mid 0, .001)\, \text{normal}(\beta_1 \mid 0, .001) \\ g(\beta_0, \beta_1, x_i) &= \frac{e^{\beta_0 + \beta_1 x_i}}{e^{\beta_0 + \beta_1 x_i} + 1}\end{aligned}$$



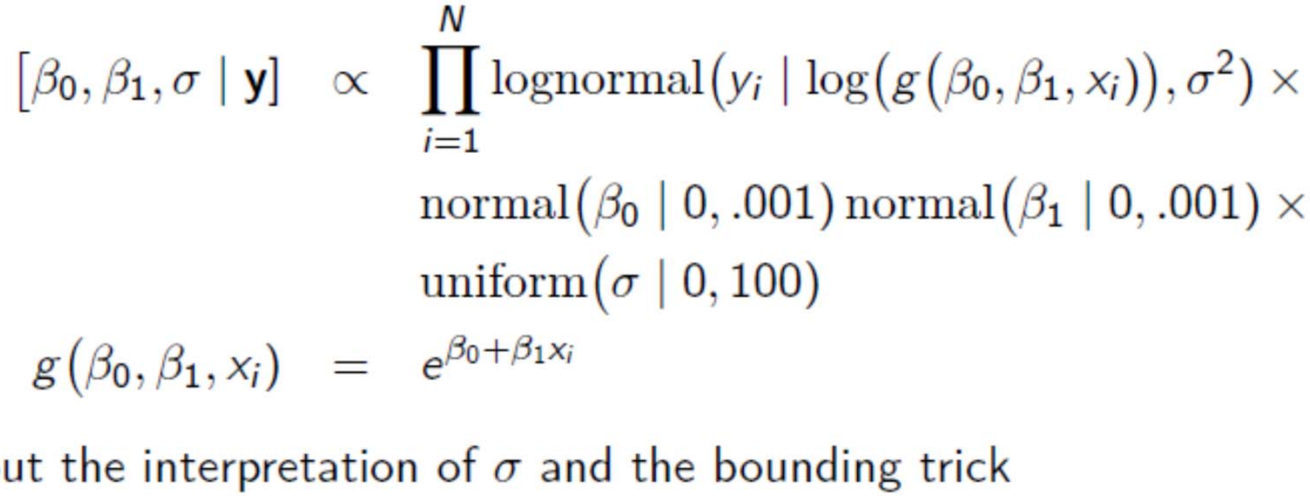
```
b0 ~ dnorm(0, .368)
b1 ~ dnorm(0, .368)
for(i in 1:length(y)){
  logit(p[i]) <- b0 + b1 * x[i]
    y[i] ~ dbern(p[i])
 }
```



or

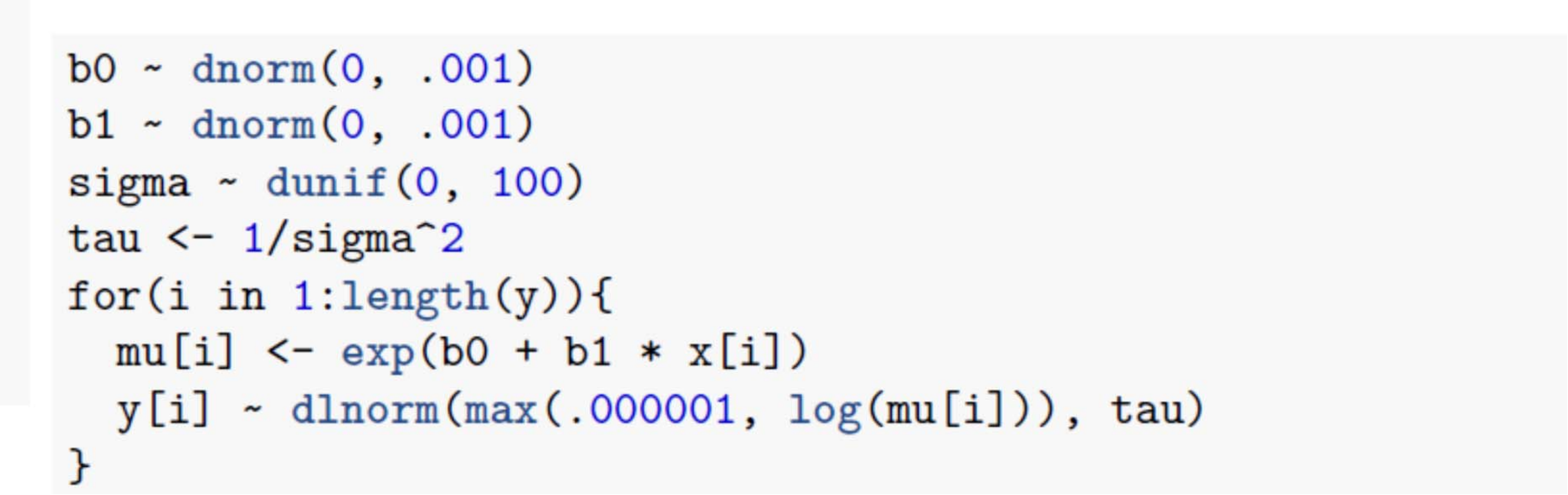
```
p[i] <- inv.logit(b0 + b1 * x[i])
y[i] ~ dbin(p[i])
```

# lognormal, data continuous and > 0



$$
\begin{aligned}
[\beta_0, \beta_1, \sigma \mid \mathbf{y}] &\propto \prod_{i=1}^{N} \text{lognormal}\big(y_i \mid \log\big(g\left(\beta_0, \beta_1, x_i\right)\big), \sigma^2\big) \times \\
&\quad \text{normal}\big(\beta_0 \mid 0, .001\big)\, \text{normal}\big(\beta_1 \mid 0, .001\big) \times \\
&\quad \text{uniform}\big(\sigma \mid 0, 100\big) \\
g\left(\beta_0, \beta_1, x_i\right) &= e^{\beta_0 + \beta_1 x_i}
\end{aligned}
$$

Talk about the interpretation of $\sigma$ and the bounding trick



```
b0 ~ dnorm(0, .001)
b1 ~ dnorm(0, .001)
sigma ~ dunif(0, 100)
tau <- 1/sigma^2
for(i in 1:length(y)){
  mu[i] <- exp(b0 + b1 * x[i])
  y[i] ~ dlnorm(max(.000001, log(mu[i])), tau)
}
```

# Centering and standardizing

The remainder of the slides apply to all of the general linear models, but I will use a simple linear model for normally distributed data as an example.

# Centering predictor data

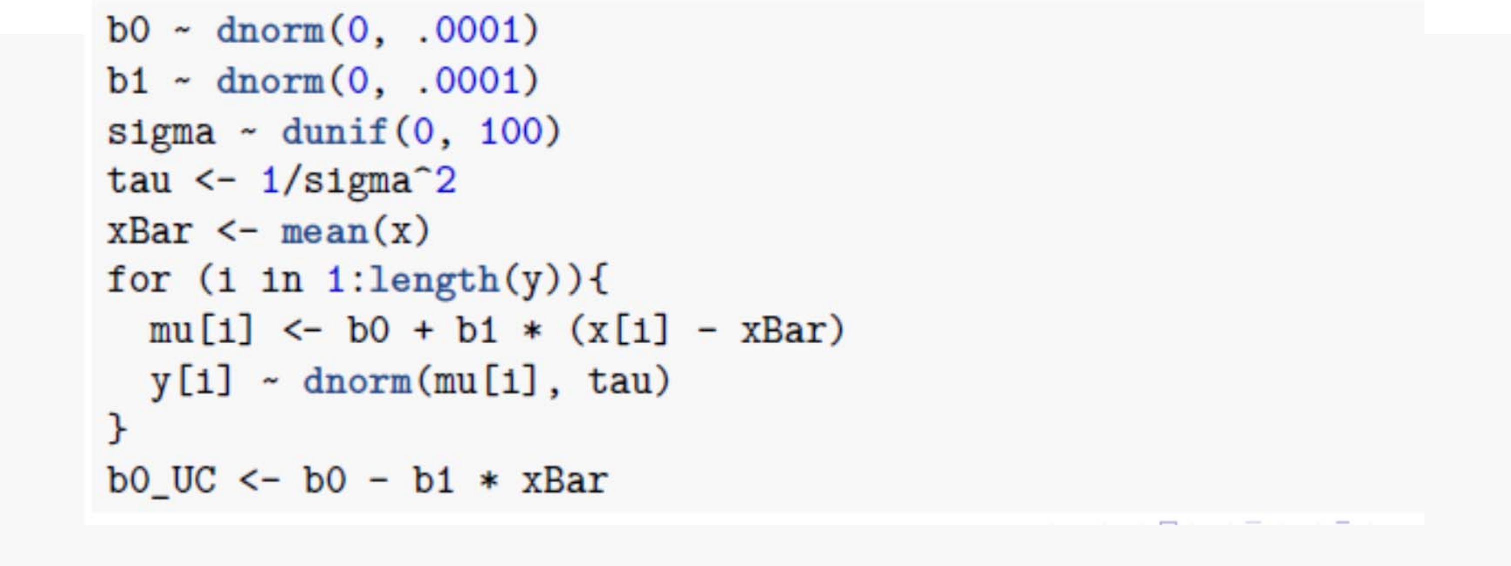
$$y_i = \beta_0 + \beta_1 (x_i - \overline{x})$$

Why complicate things. . . ?

To reduce autocorrelation in MCMC chain and speed convergence.
To make the intercept more easily interpretable.

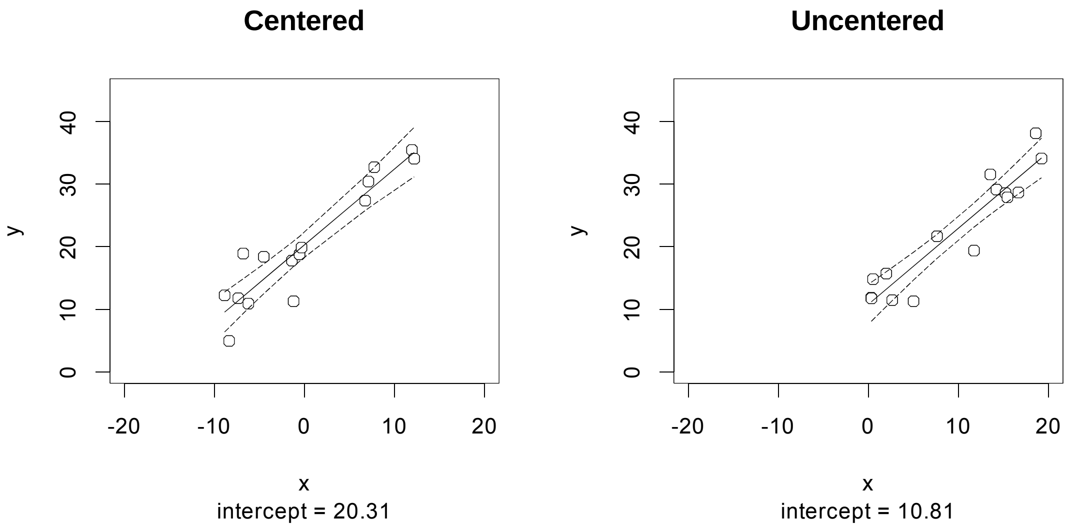
# Centering predictor data

$$
\begin{aligned}
[\beta_0, \beta_1, \sigma \mid \mathbf{y}] &\propto \prod_{i=1}^{N} \text{normal}\big(y_i \mid g(\beta_0, \beta_1, x_i, \bar{x}), \sigma^2\big) \times \\
&\quad \text{normal}(\beta_0 \mid 0, .001)\,\text{normal}(\beta_1 \mid 0, .001) \times \\
&\quad \text{uniform}(\sigma \mid 0, 100) \\
g(\beta_0, \beta_1, x_i) &= \beta_0 + \beta_1 (x_i - \bar{x})
\end{aligned}
$$



```
b0 ~ dnorm(0, .0001)
b1 ~ dnorm(0, .0001)
sigma ~ dunif(0, 100)
tau <- 1/sigma^2
xBar <- mean(x)
for (i in 1:length(y)){
  mu[i] <- b0 + b1 * (x[i] - xBar)
  y[i] ~ dnorm(mu[i], tau)
}
b0_UC <- b0 - b1 * xBar
```

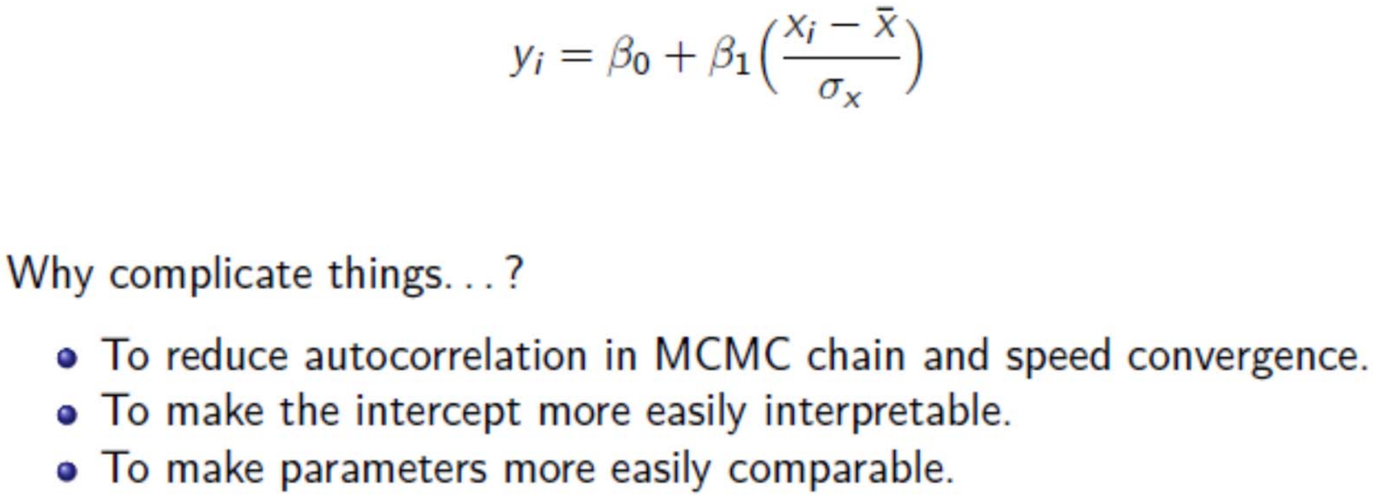
# Recovering uncentered parameters



$$\begin{aligned} B_0 &= \beta_0 - \beta_1 * \overline{x} \\ B_1 &= \beta_1 \end{aligned}$$

- For this to work properly, all of the coefficients in the model must be *added*.
- Slopes will not be the same if there is an interaction term or quadratic. In these cases, back transforming is not simple.

# Standardizing predictor data

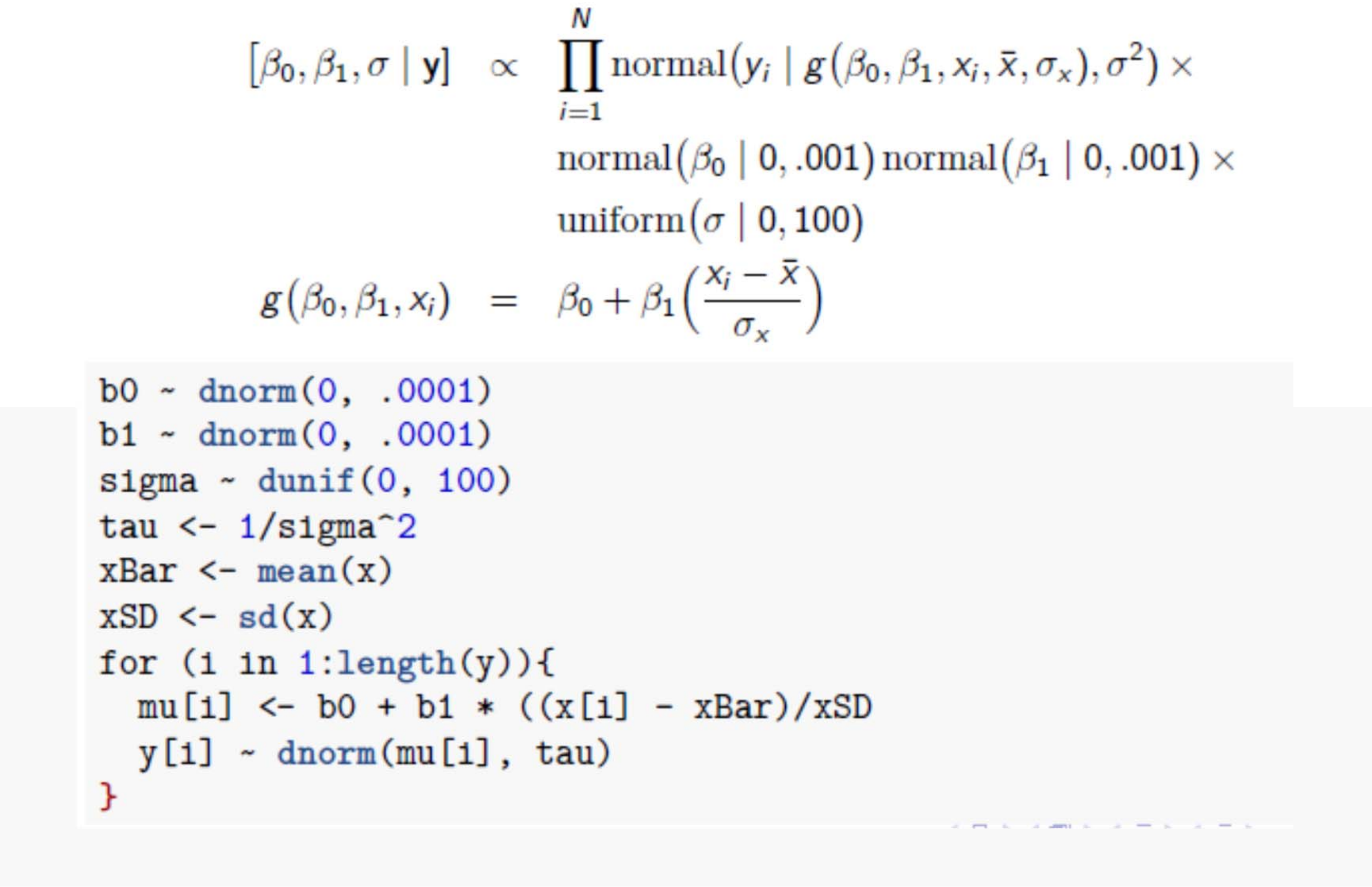


$$y_i = \beta_0 + \beta_1\left(\frac{x_i - \bar{x}}{\sigma_x}\right)$$

Why complicate things. . . ?

- To reduce autocorrelation in MCMC chain and speed convergence.
- To make the intercept more easily interpretable.
- To make parameters more easily comparable.

# Standardizing predictor data



$$\begin{aligned}
[\beta_0, \beta_1, \sigma \mid \mathbf{y}] &\propto \prod_{i=1}^{N} \text{normal}\big(y_i \mid g(\beta_0, \beta_1, x_i, \bar{x}, \sigma_x), \sigma^2\big) \times \\
&\quad \text{normal}(\beta_0 \mid 0, .001)\, \text{normal}(\beta_1 \mid 0, .001) \times \\
&\quad \text{uniform}(\sigma \mid 0, 100) \\
g(\beta_0, \beta_1, x_i) &= \beta_0 + \beta_1 \left( \frac{x_i - \bar{x}}{\sigma_x} \right)
\end{aligned}$$

```
b0 ~ dnorm(0, .0001)
b1 ~ dnorm(0, .0001)
sigma ~ dunif(0, 100)
tau <- 1/sigma^2
xBar <- mean(x)
xSD <- sd(x)
for (i in 1:length(y)){
  mu[i] <- b0 + b1 * ((x[i] - xBar)/xSD
  y[i] ~ dnorm(mu[i], tau)
}
```

# Recovering unstandardized parameters

$$\begin{aligned}
y_i &= \beta_0 + \beta_1\left(\frac{x_i - \bar{x}}{\sigma_x}\right) \\
B_0 &= \beta_0 - \frac{\beta_1 \bar{x}}{\sigma_x} \\
B_1 &= \frac{\beta_1}{\sigma_x}
\end{aligned}$$

This only works if there are not squared values or interactions.