Bayesian data analysis in ecology: A brief introduction

Kai Zhu

UC Santa Cruz
https://zhulab.ucsc.edu

CForBio 2020

Probability theory of Bayesian statistics

Basic probability

Consider an event, A,

$\Pr(A)$: probability of event A.

Also written as $[A]$,

$$\Pr(A) = [A] \in [0, 1].$$

Probability must be between 0 and 1.

Conditional probability

Consider two events, A and B,

$[A]$: probability of event A,

$[B]$: probability of event B.

Joint probability,

$[A \cap B]$: probability of both events occurring together.

Conditional probability theory (and intuition) tells us that $[A \cap B] = [B] \times [A|B]$.

It must also be true that $[B \cap A] = [A] \times [B|A]$.

Bayes' theorem

Then we have

- $[A \cap B] = [B] \times [A|B]$,
- $[B \cap A] = [A] \times [B|A]$.

Thomas Bayes (1701?-1761) noticed that $[A \cap B] = [B \cap A]$, therefore it must be true that $[B] \times [A|B] = [A] \times [B|A]$.

Division of both sides by $[B]$ gives Bayes' theorem.

$$[A|B] = \frac{[A] \times [B|A]}{[B]}.$$

Application: Fungal infection and seedling survival

Hersh et al. (2012, *Ecology*) planted seedlings of six species in 60 plots. They observed survival and assayed fungal infection on cultures and with DNA sequencing.

Suppose

- fungal infection rate is θ—overall infection in the environment,
- seedling survival rate if *not* infected is s_0—"health" seedlings, natural mortality,
- seedling survival rate if infected is s_1—"unhealth" seedlings, natural mortality + infection.

Q: What is the infection rate if the seedling survives?

Application: Fungal infection and seedling survival

A: Let $I = 1$ be infection event ($I = 0$ not infected), and $S = 1$ be survival event ($S = 0$ if dead). Then

- fungal infection rate is θ: $[I = 1] = \theta$,
- seedling survival rate if *not* infected is s_0: $[S = 1|I = 0] = s_0$,
- seedling survival rate if infected is s_1: $[S = 1|I = 1] = s_1$.

Application: Fungal infection and seedling survival

Use Bayes' theorem,

$$\begin{aligned}
[I=1|S=1] &= \frac{[S=1|I=1][I=1]}{[S=1]}, \\
&= \frac{[S=1|I=1][I=1]}{[S=1|I=1][I=1]+[S=1|I=0][I=0]}, \\
&= \frac{s_1\theta}{s_1\theta+s_0(1-\theta)}.
\end{aligned}$$

Note that $[I=1]=\theta$, so $[I=1] \neq [I=1|S=1]$.

Why? Because we updated our belief in fungal infection by observing seedling survival.

Application: Fungal infection and seedling survival

Revisit

$$[I = 1|S = 1] = \frac{[S = 1|I = 1][I = 1]}{[S = 1|I = 1][I = 1] + [S = 1|I = 0][I = 0]}.$$

- $[I = 1]$ is our prior belief,
- $[S = 1|I = 1]$ and $[S = 1|I = 0]$ are our experiment,
- $[I = 1|S = 1]$ is our posterior estimate, after we update our prior belief using data collected from experiment. Our posterior estimate is different from our prior belief, because we gained knowledge through the experiment.

Very simple example, but the idea carries over all Bayesian statistics.

Frequentist approach to linear models

Frequentist vs. Bayesian

Frequentist:

- Data are a *repeatable* random sample—there is a frequency.
- Underlying parameters remain constant during this repeatable process.
- *Parameters* are fixed.

Bayesian:

- Data are observed from the realized sample.
- Parameters are unknown and described probabilistically.
- *Data* are fixed.

Linear model as an example

Linear models are the basic building blocks for almost all statistical models, for example,

- linear regression,
- analysis of variance (ANOVA).

Simple linear regression

For the ith response variable y_i and predictor x_i,

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, \cdots, n.$$

Assume the random error is distributed as *normal* (or *Gaussian*),

$$\epsilon_i \sim N(0, \sigma^2).$$

Expand equations

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, \cdots, n.$$

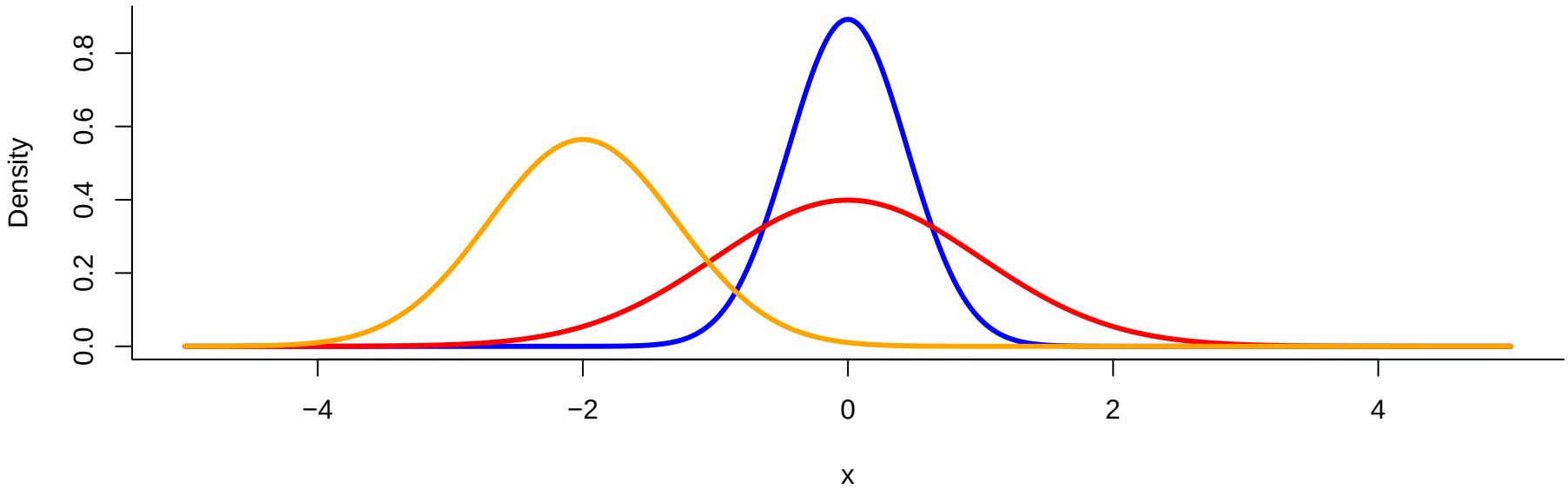
becomes

$$\begin{aligned} y_1 &= \beta_0 + \beta_1 x_1 + \epsilon_1 \\ y_2 &= \beta_0 + \beta_1 x_2 + \epsilon_2 \\ \vdots & \qquad \vdots \\ y_n &= \beta_0 + \beta_1 x_n + \epsilon_n \end{aligned}$$

Note that there is only a fixed β_0 and a fixed β_1—also called fixed effects.

Normal distribution in theory

$$x_1 \sim N(\mu = 0, \sigma^2 = 0.2),$$
$$x_2 \sim N(\mu = 0, \sigma^2 = 1),$$
$$x_3 \sim N(\mu = -2, \sigma^2 = 0.5).$$



Normal distribution in practice

Q: How to simulate (or "generate using computer") random numbers from normal distribution?

A: Use `rnorm` function. `r` means random number generator.

```
x <- rnorm(n = 100, mean = 0, sd = 1)
```

- `n`: how many samples to simulate,
- `mean`: μ parameter,
- `sd`: σ (**not** σ^2) parameter.

Shortcut:

```
x <- rnorm(100, 0, 1)
```

Normal distribution in practice

Q: How to check if simulated numbers are *really* normal?

```
x[1:4] # display first 4 values
```

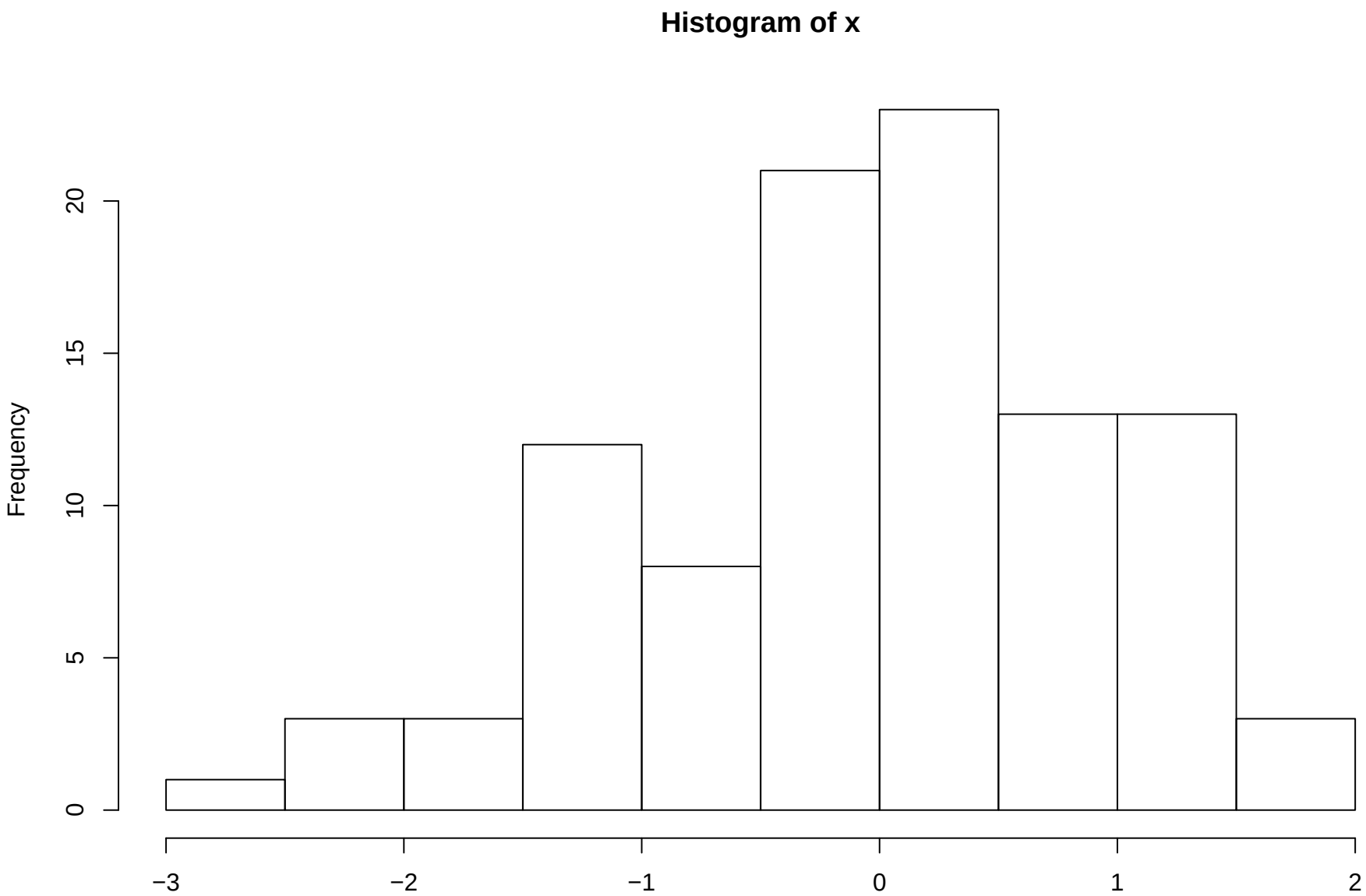
```
## [1]  1.09000346 -2.63493707  0.06108367 -1.15532931
```

You can see them, but still can't tell...

A: Use histogram.

```
hist(x)
```

Normal distribution in R



The problem

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2).$$

What are data (known) and what are parameters (unknown)?

- Knowns are
  - y_i: response—e.g., tree growth,
  - x_i: covariate—e.g., temperature.
- Unknowns are
  - β_0, β_1: regression coefficent, intercept and slope—we want to estimate: how much do trees respond to temperature?
  - σ^2: variance parameter—sometimes we want to know: how accurate are our observations?

Frequentist solution

Maximum likelihood estimate (MLE):

- Likelihood is the probability of observing data given parameter,

$$[y_i, x_i \mid \beta_0, \beta_1, \sigma].$$

- Maximum likelihood estimate (MLE) method finds the parameters that give the maximum value of likelihood,

$$\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma} \leftarrow \max[y_i, x_i \mid \beta_0, \beta_1, \sigma].$$

Frequentist solution

MLE solution for linear regression:

$$
\begin{aligned}
\hat{\beta}_1 &= \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2} \\
\hat{\beta}_0 &= \bar{y} - \hat{\beta}_1\bar{x} \\
\hat{\sigma} &= \cdots
\end{aligned}
$$

Application: Simulated tree growth

Simulation is an important tool to check your understanding. It starts with *prescribed* parameters to generate *artifial* data. Then you only use the data to estimate parameters (pretending you don't know the *prescribed* parameters you started with). See if your *estimated* parameter is close to your *prescribed* (true) parameters.

Suppose you went to 100 sites to measure tree growth rate. The annual temperature is 22 °C on average, and 5 °C of standard deviation. Now, suppose trees grow 5 mm/yr at 0 °C, and 1 mm/yr with every 1 °C increase in temperature. Your sampling error is 1 mm/yr of standard deviation.

Q: What would tree growth rate be?

Application: Simulated tree growth

Translate to math

- $i = 1, \cdots, 100,$
- $x_i \sim N(\mu = 22, \sigma = 5),$
- $\beta_0 = 5,$
- $\beta_1 = 1,$
- $\sigma = 1.$

Application: Simulated tree growth

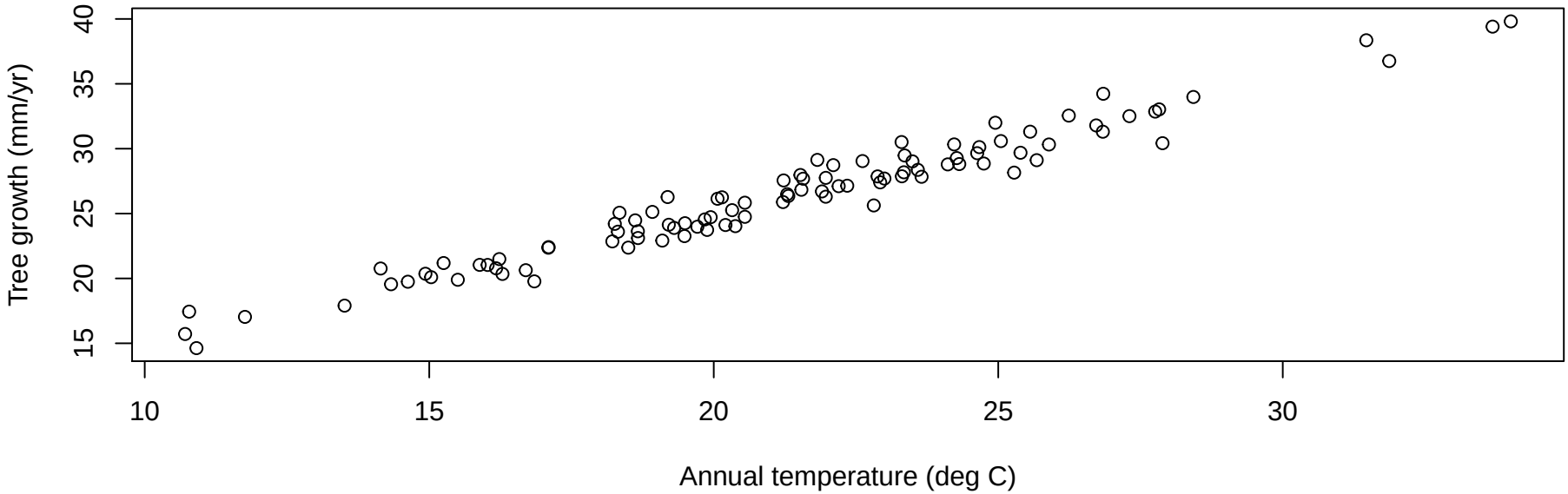
Translate to R

```
n.tree <- 100
ann.temp <- rnorm(n = n.tree, mean = 22, sd = 5)

sim.tree.growth <- function(n, x, b0, b1, s) {
  e <- rnorm(n, 0, s)
  y <- b0 + b1 * x + e
}

tree.grow <- sim.tree.growth(
  n = n.tree, x = ann.temp,
  b0 = 5, b1 = 1, s = 1
)
```

Application: Simulated tree growth

Check your simulation

```
plot(ann.temp, tree.grow,
   xlab = "Annual temperature (deg C)",
   ylab = "Tree growth (mm/yr)"
)
```



Application: Simulated tree growth

Estimate parameters, using the MLE solution,

$$\begin{aligned} \hat{\beta}_1 &= \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2} \\ \hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x} \end{aligned} .$$

```
esti.tree.growth <- function(x, y) {
  x.bar <- mean(x)
  y.bar <- mean(y)
  b1.hat <- sum((x - x.bar) * (y - y.bar)) /
    sum((x - x.bar)^2)
  b0.hat <- y.bar - b1.hat * x.bar
  c(b0.hat, b1.hat) # concatenate output
}
```

Application: Simulated tree growth

Estimate parameters, using the MLE solution,

```
esti.tree.growth(ann.temp, tree.grow)
```

```
## [1] 4.770324 1.015446
```

What are the *true* values?

$$b_0 = 5, b_1 = 1.$$

We are pretty close to the true values.

Application: Simulated tree growth

An easy way to do in R

```
lm(tree.grow ~ ann.temp)
```

```
##
## Call:
## lm(formula = tree.grow ~ ann.temp)
##
## Coefficients:
## (Intercept)     ann.temp
##       4.770        1.015
```

Identical to our function! This is precisely how R implements linear model.

Bayesian approach to linear models

Bayesian solution

Bayesian statistics assumes a **prior** distribution for parameters (usually vague, meaning no information), uses Bayes' theorem to update the prior distribution by integrating **likelihood** from data, and eventally obtains **posterior** distribution for parameters.

In our case, the likelihood is $[y_i, x_i \mid \beta_0, \beta_1, \sigma]$.

The prior is $[\beta_0, \beta_1, \sigma]$.

The posterior $[\beta_0, \beta_1, \sigma \mid y_i, x_i] = ?$

Bayesian solution

According to Bayes' theorem,

$$[\beta_0, \beta_1, \sigma \mid y_i, x_i] = \frac{[y_i, x_i \mid \beta_0, \beta_1, \sigma][\beta_0, \beta_1, \sigma]}{[y_i, x_i]}.$$

We don't know $[y_i, x_i]$, but we know any distribution, including the posterior, has to sum to one. So we can ignore denominator, making the equation proportional.

$$[\beta_0, \beta_1, \sigma \mid y_i, x_i] \propto [y_i, x_i \mid \beta_0, \beta_1, \sigma][\beta_0, \beta_1, \sigma].$$

Bayesian solution

We use non-informative priors because we have no knowledge,

- $\beta_0 \sim N(0, \tau_0 = 1/\sigma_0^2)$: usually τ_0 is very small (σ_0 is very large),
- $\beta_1 \sim N(0, \tau_1 = 1/\sigma_1^2)$: usually τ_1 is very small (σ_1 is very large),
- $\frac{1}{\sigma^2} \sim G(a, b)$[1]: usually a, b are very small.

The posterior solution is too complicated; it involves numerical simulation.

Fortunately, we have numerical methods such as Markov chain Monte Carlo (MCMC). We will not get into details; we will use software JAGS.

[1] G stands for Gamma distribution.

JAGS

Of the many MCMC methods, we focus on Gibbs sampler, implemented as BUGS (Bayesian inference Using Gibbs Sampling) language.

JAGS (Just Another Gibbs Sampler) is a *dialect* of BUGS language, which

- works closely with R, similar syntax,
- is very flexible, allowing users to write their own models.

Other BUGS *dialects* are WinBUGS, OpenBUGS.

Other MCMC software includes Stan, INLA.

JAGS and R

JAGS can be downloaded at http://mcmc-jags.sourceforge.net.

Also install packages in R, so that R can "talk to" JAGS.

```
install.packages("R2jags")
install.packages("mcmcplots")
```

We will be using JAGS within R.

Model recap

The model is

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

which can be rewritten as (**likelihood**)

$$\begin{aligned} y_i &\sim N(\mu_i, \sigma^2 = 1/\tau), \\ \mu_i &= \beta_0 + \beta_1 x_i. \end{aligned}$$

We will use these **priors**

$$\begin{aligned} \beta_0 &\sim N(0, \tau' = 10^{-3}), \\ \beta_1 &\sim N(0, \tau' = 10^{-3}), \\ \tau &\sim G(10^{-3}, 10^{-3}). \end{aligned}$$

JAGS

Model

```
sim.mod <- function() {

  # likelihood
  for (i in 1:n) {
    y[i] ~ dnorm(mu[i], tau) # tau = 1/sigma2
    mu[i] <- b0 + b1 * x[i]
  }

  # priors
  b0 ~ dnorm(0, 1.0E-3)
  b1 ~ dnorm(0, 1.0E-3)
  tau ~ dgamma(1.0E-3, 1.0E-3)
}
```

JAGS

Data (known) are

```
sim.dat <- list()
sim.dat$n <- n.tree
sim.dat$y <- tree.grow
sim.dat$x <- ann.temp
```

Parameters (unknown)—we assumed vague **priors** and want to know **posteriors**,

```
sim.par <- c("b0", "b1", "tau")
```

JAGS

Fit the model

```
library(R2jags)
sim.fit <-
  jags(
    data = sim.dat,
    model.file = sim.mod,
    parameters.to.save = sim.par,
    n.chains = 1, n.iter = 1e3
  )
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 100
##    Unobserved stochastic nodes: 3
```

JAGS

Summarize the results

```
sim.fit
```

```
##                mu.vect sd.vect
## b0               4.774   0.468
## b1               1.015   0.022
## tau              0.977   0.147
## deviance       287.812   2.450
```

MCMC diagnostics

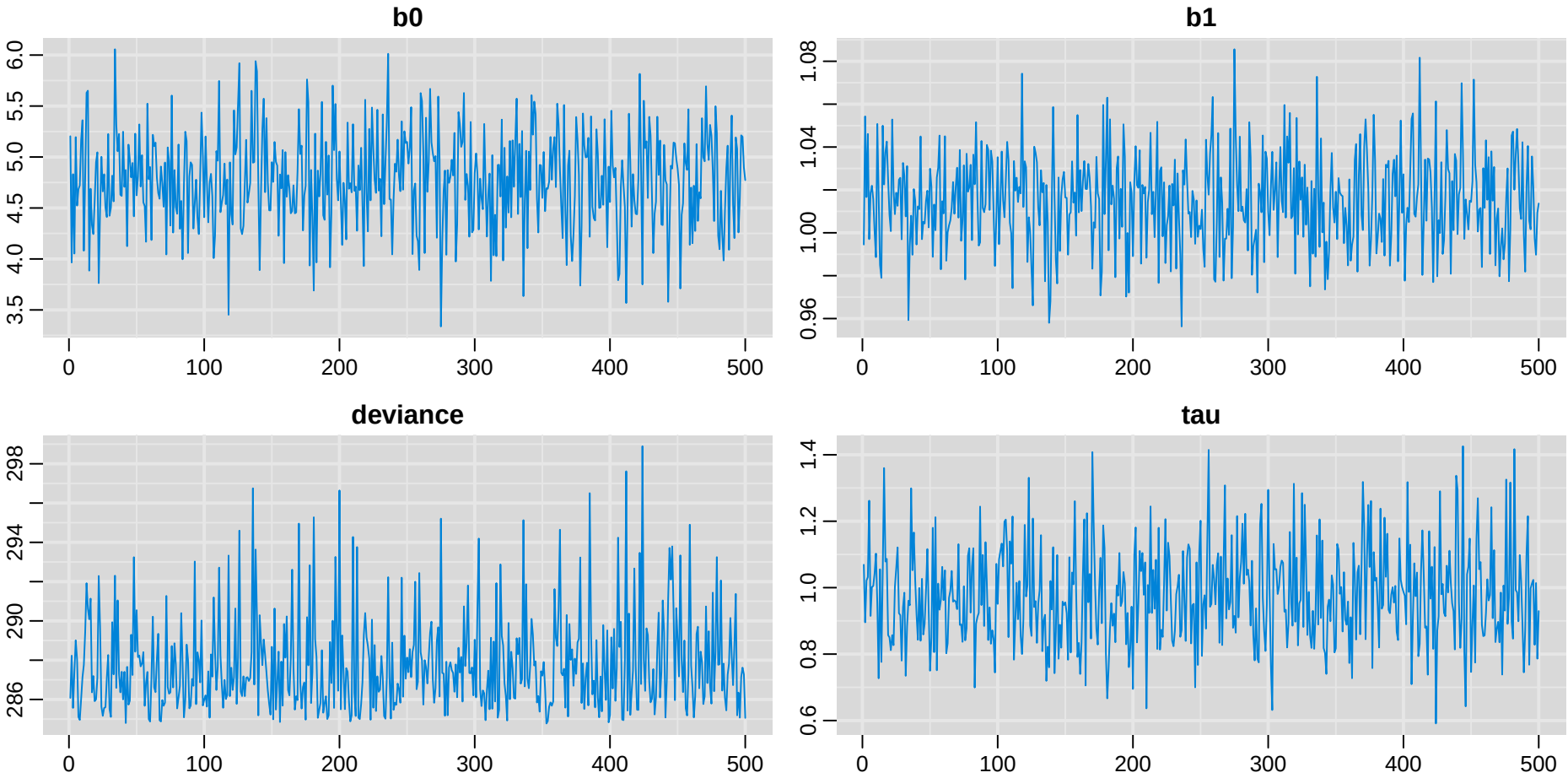
Check covergence

```
mcmcplot(sim.fit)
```

MCMC diagnostics

Trace plot

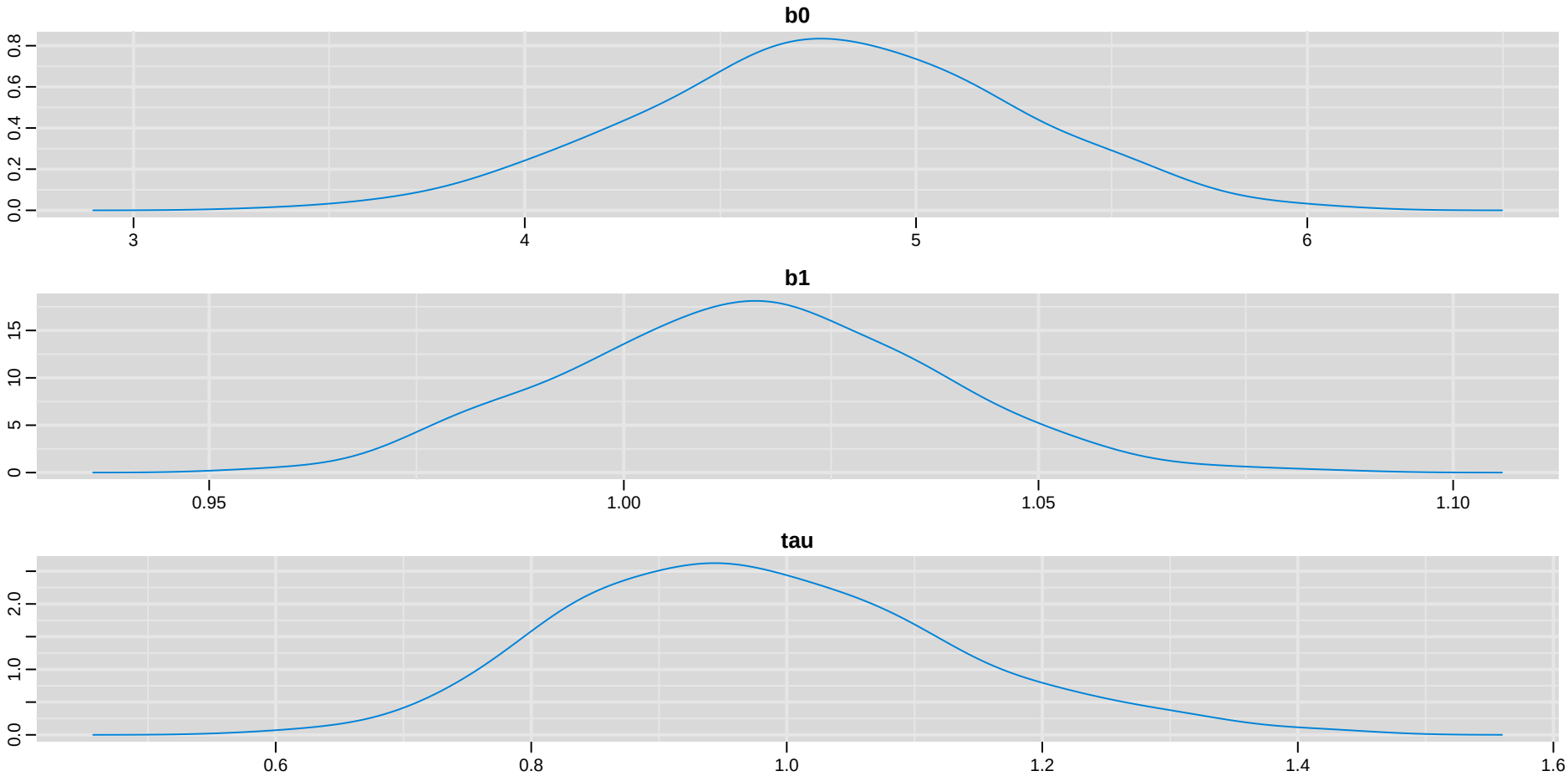
```
traplot(sim.fit)
```



MCMC diagnostics

Density plot

```
denplot(sim.fit)
```



Compare results

Simulation (true values):

$$\beta_0 = 5, \beta_1 = 1.$$

Frequentist (MLE) approach:

```
##     b0     b1
## 4.770 1.015
```

Bayesian (JAGS) approach:

```
##     b0     b1
## 4.774 1.015
```

Application: Seedling recruitment along climate gradient

Zhu et al. (2014, *Global Change Biology*) examined the relationship between tree seedling abundance and climate in eastern US forests. Here we use a subset of the data, focusing on a species, loblolly pine (*Pinus taeda*).

Read data

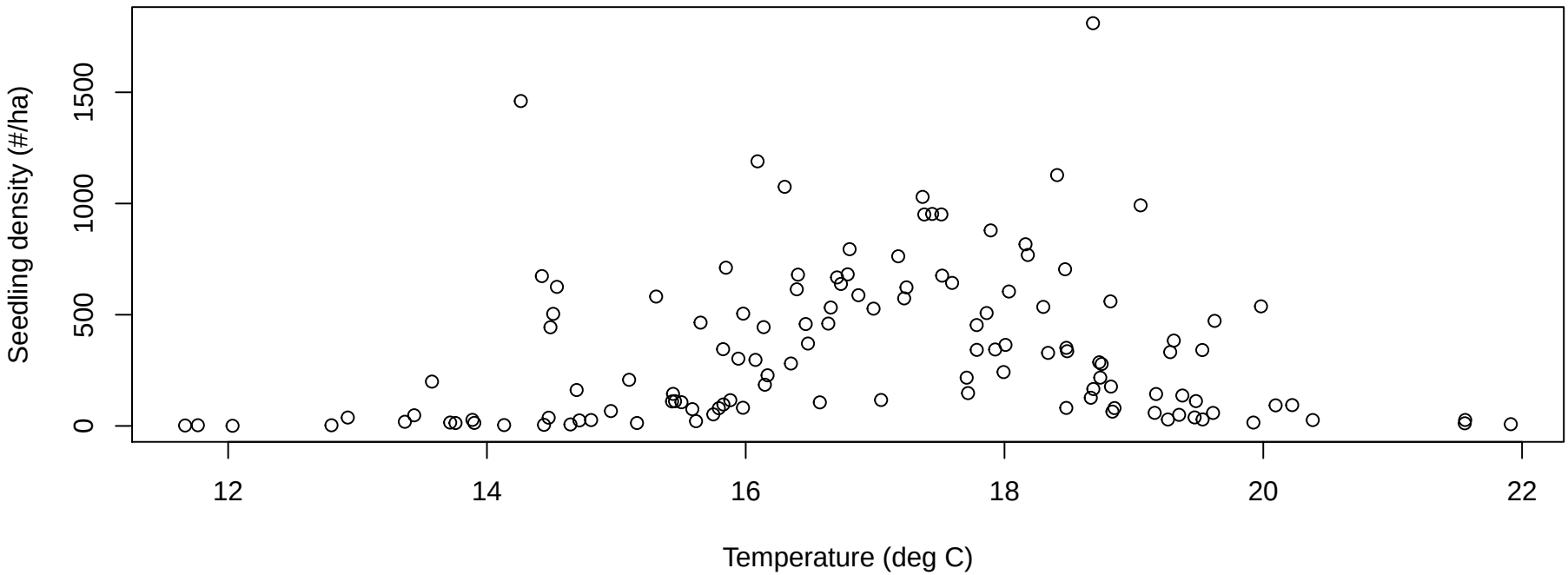
```
raw.dat <- read.csv("data/seedling.csv")
```

- `seed`: seedling density (#/ha),
- `tmp`: annual temperature (°C).

Application: Seedling recruitment along climate gradient

Explore data

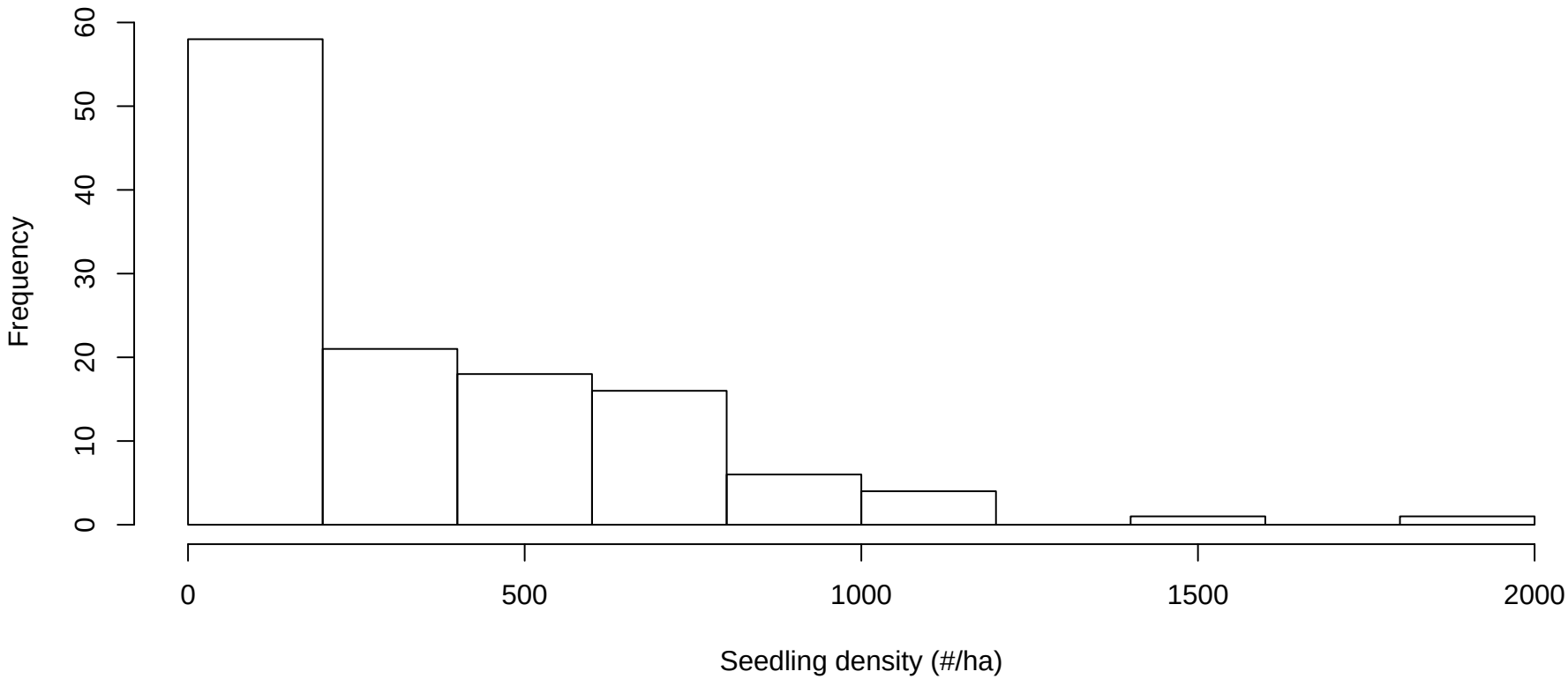
```
plot(raw.dat$tmp, raw.dat$seed,
  ylab = "Seedling density (#/ha)",
  xlab = "Temperature (deg C)")
```



Application: Seedling recruitment along climate gradient

Explore data

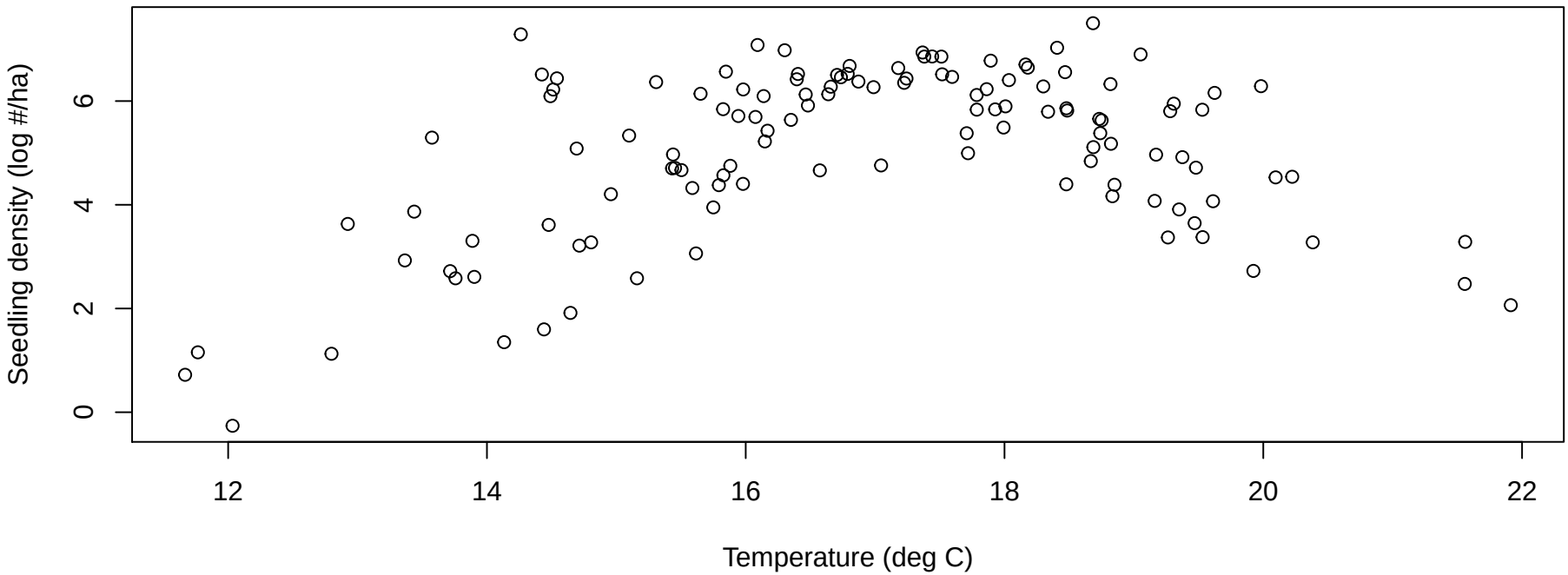
```
hist(raw.dat$seed, main = "",
  xlab = "Seedling density (#/ha)")
```



Application: Seedling recruitment along climate gradient

Transform data (log)

```
plot(raw.dat$tmp, log(raw.dat$seed),
  ylab = "Seedling density (log #/ha)",
  xlab = "Temperature (deg C)")
```



Application: Seedling recruitment along climate gradient

Here we use a simplified model,

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

where

- y_i: log-transformed seedling density (#/ha) in region i,
- x_i: temperature (°C) in region i.

Q: How to fit data to this model in JAGS and R?

Application: Seedling recruitment along climate gradient

The model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

can be rewritten as (**likelihood**)

$$\begin{aligned} y_i \sim & \quad N(\mu_i, \sigma^2 = 1/\tau), \\ \mu_i = & \quad \beta_0 + \beta_1 x_i + \beta_2 x_i^2, \end{aligned}$$

and **priors**

$$\begin{aligned} \beta_0, \beta_1, \beta_2 \sim & \quad N(0, \tau' = 10^{-3}), \\ \tau \sim & \quad G(10^{-3}, 10^{-3}). \end{aligned}$$

Application: Seedling recruitment along climate gradient

The model

```
seed.mod <- function() {

  # likelihood
  for (i in 1:n) {
    y[i] ~ dnorm(mu[i], tau)
    mu[i] <- b0 + b1 * x[i] + b2 * x[i]^2
  }

  # priors
  b0 ~ dnorm(0, 1.0E-3)
  b1 ~ dnorm(0, 1.0E-3)
  b2 ~ dnorm(0, 1.0E-3)
  tau ~ dgamma(1.0E-3, 1.0E-3)
}
```

Application: Seedling recruitment along climate gradient

Data

```
seed.dat <- list()
seed.dat$y <- log(raw.dat$seed)
seed.dat$x <- raw.dat$tmp
seed.dat$n <- nrow(raw.dat)
```

Parameters

```
seed.par <- c("b0", "b1", "b2", "tau")
```

Application: Seedling recruitment along climate gradient

Fit the model

```
seed.fit <-
  jags(
    data = seed.dat,
    model.file = seed.mod,
    parameters.to.save = seed.par,
    n.chains = 1, n.iter = 1e3
  )
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 125
##    Unobserved stochastic nodes: 4
##    Total graph size: 758
```

Application: Seedling recruitment along climate gradient

Check the model

```
mcmcplot(seed.fit)
traplot(seed.fit)
denplot(seed.fit)
```

Extending models

Frequentist vs. Bayesian inference

Recap

X = data (known), θ = parameter (unkonwn).

Likelihood is the probability of observing data given parameter,

$$[X|\theta].$$

Frequentist inference uses maximum likelihood estimate (MLE),

$$\hat{\theta} \leftarrow \max[X|\theta]$$

Here the "true" θ is a fixed, unknown value. $\hat{\theta}$ is our "best" guess.

Frequentist vs. Bayesian inference

X = data (known), θ = parameter (unkonwn).

Likelihood is the probability of observing data given parameter,

$$[X|\theta].$$

Bayesian inference assumes θ is a random variable, with some *prior* distribution ($[\theta]$), and after we observe the data given parameter—*likelihood* ($[X|\theta]$), we want to find the *posterior* distribution of parameter given data $[\theta|X]$.

Frequentist vs. Bayesian inference

This process is made possible by Bayes' theorem,

$$[\theta|X] = \frac{[\theta] \times [X|\theta]}{[X]}.$$

Because probability always sums to 1, we can simplify to

$$[\theta|X] \propto [X|\theta][\theta]$$

Computation: Markov chain Monte Carlo (MCMC). Usually difficult. Here we use software (JAGS). Other options are WinBUGS, OpenBUGS, STAN, INLA, LibBi, etc.

More complicated models

- Poisson regression
- Logistic regression
- Nonlinear hierarchical model

Poisson regression

Poisson distribution for count data. The model is

$$y_i \sim Po(\lambda_i), \quad \lambda_i > 0.$$

Link function

$$\log(\lambda_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p.$$

Link function is deterministic. Stochasticity comes from Poisson distribution.

Poisson regression

Zhu et al. (2015, *Ecology*) examined tree recruitment in the eastern United States. They found that newly recruited saplings (recruitment) have a negative relationship with conspecific neighboring trees (conspecific negative density dependence, CNDD).

Likelihood

$$\begin{aligned} y_i \sim & \quad Po(\lambda_i), \\ \log(\lambda_i) = & \quad \beta_0 + \beta_1 x_i. \end{aligned}$$

Priors

$$\beta_0, \beta_1 \sim N(0, \tau = 10^{-3}).$$

Poisson regression

JAGS code

```
# likelihood
for (i in 1:n) {
  y[i] ~ dpois(lam[i])
  log(lam[i]) <- b0 + b1 * x[i]
}
# priors
b0 ~ dnorm(0, 1.0E-3)
b1 ~ dnorm(0, 1.0E-3)
```

Logistic regression

Bernoulli distribution for binary (e.g., presence/absence) data. The model is

$$y_i \sim Bi(\theta_i), \quad 0 < \theta_i < 1.$$

Link function

$$\text{logit}(\theta_i) = \log\left(\frac{\theta_i}{1-\theta_i}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p.$$

Link function is deterministic. Stochasticity comes from Bernoulli distribution.

$\frac{\theta_i}{1-\theta_i}$ is also called *odds ratio*.

Logistic regression

Zhu et al. (2012, *Global Change Biology*) analyzed tree species occurrence (presence/absence) across latitudes to understand climate change impact on forest biogeography.

Model

$$\begin{aligned} y_i \sim & \quad Bi(\theta_i), \\ \text{logit}(\theta_i) = & \quad \beta_0 + \beta_1 x_i + \beta_2 x_i^2. \end{aligned}$$

where

- y_i is tree occurrence,
- x_i is relative latitude.

Logistic regression

Likelihood

$$\begin{aligned} y_i \sim & \quad Bi(\theta_i), \\ \text{logit}(\theta_i) = & \quad \beta_0 + \beta_1 x_i + \beta_2 x_i^2. \end{aligned}$$

Priors

$$\beta_0, \beta_1, \beta_2 \sim N(0, \tau = 10^{-3}).$$

Logistic regression

JAGS code

```
# likelihood
for (i in 1:n) {
  y[i] ~ dbern(theta[i])
  logit(theta[i]) <- b0 + b1 * x[i] + b2 * x[i]^2
}
# priors
b0 ~ dnorm(0, 1.0E-3)
b1 ~ dnorm(0, 1.0E-3)
b2 ~ dnorm(0, 1.0E-3)
```

Nonlinear hierarchical model

Widely recognized as a significant carbon sink, North American forests have experienced a history of recovery and are facing an uncertain future. Zhu et al. (2018, *Nature Communications*) combined the US and Canada forest inventory observations to integrate two key mechanisms: the natural process of forest growth and regeneration, and climate change that is likely to alter the growth process. They found that climate change effectively modifies the forest recovery trajectory, but the larger factor is that overall forest growth is limited.

The essense of the analysis is a nonlinear hierarchical growth model.

Nonlinear hierarchical model

For forest biomass y with stand age x, the growth is modeled as a Monod function (1st hierarchy),

$$y = \mu \frac{x}{k + x} + \epsilon, \quad \epsilon \sim N(0, \sigma^2).$$

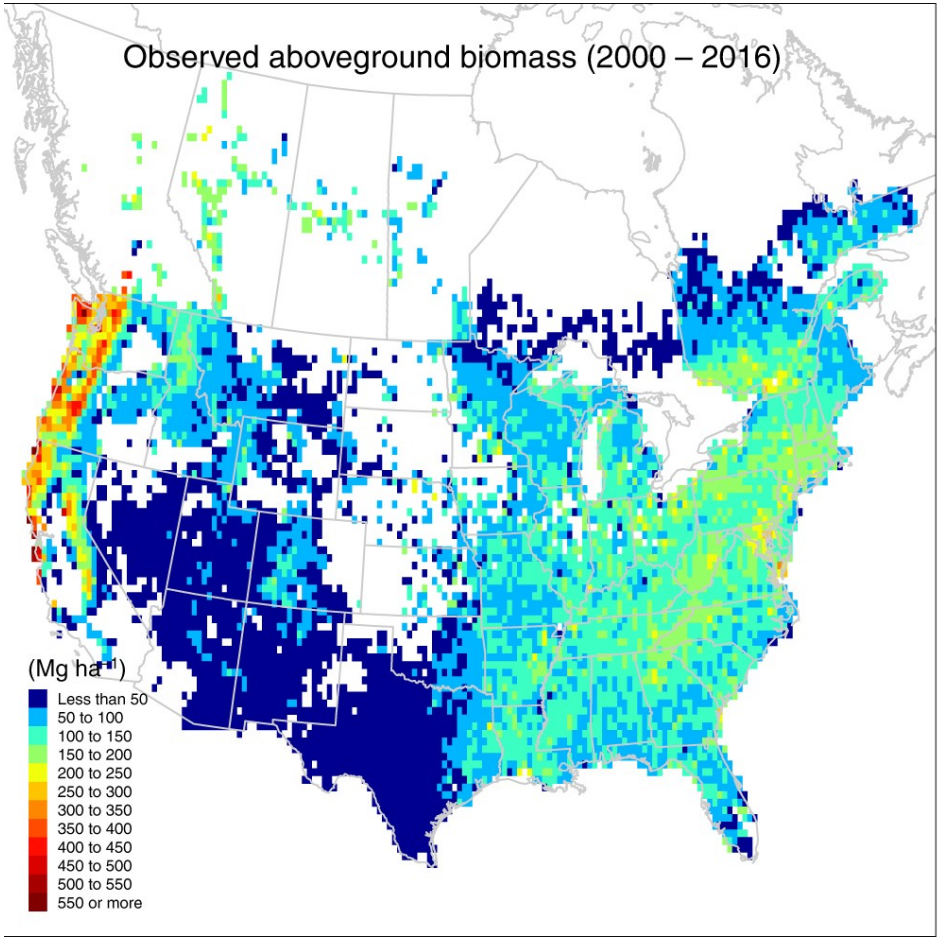
Parameters of the Monod growth then are modeled as functions of temperature T and precipitation P (2nd hierarchy),

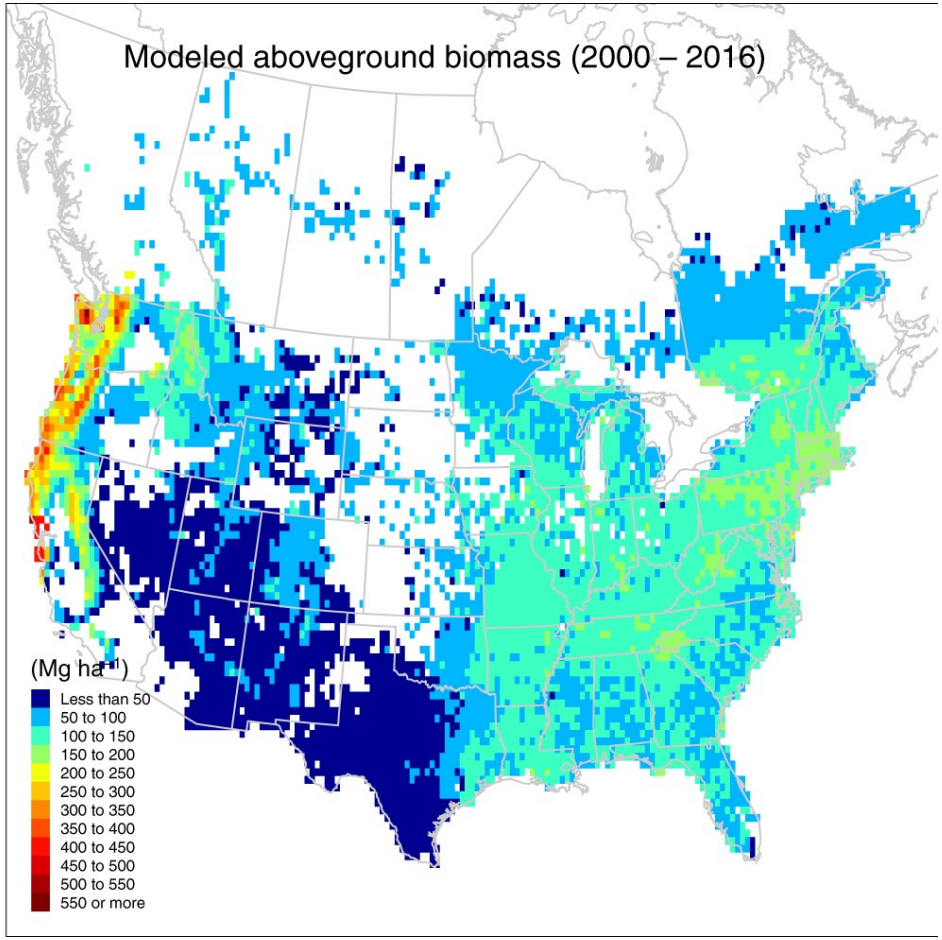
$$\mu = \beta_0 + \beta_1 T + \beta_2 P,$$
$$k = \gamma_0 + \gamma_1 T + \gamma_2 P.$$

Nonlinear hierarchical model

```
for (i in 1:n.plt) { # likelihood
  y[i]   ~  dnorm(eta[i], tau[ft[i]])
  eta[i] <- mu[i] * x[i] / (k[i] + x[i])
  mu[i]  <- z[i, ] %*% beta[, ft[i]]
  k[i]   <- z[i, ] %*% gamma[, ft[i]]
}
for (j in 1:n.ft) { # priors
  tau[j]       ~ dgamma(1E-3, 1E-3)
  beta[1, j]   ~ dunif(0, 1 / 1E-3)
  beta[2, j]   ~ dunif(-1 / 1E-3, 1 / 1E-3)
  beta[3, j]   ~ dunif(-1 / 1E-3, 1 / 1E-3)
  gamma[1, j] ~ dunif(0, 1 / 1E-3)
  gamma[2, j] ~ dunif(-1 / 1E-3, 1 / 1E-3)
  gamma[3, j] ~ dunif(-1 / 1E-3, 1 / 1E-3)
}
```

Nonlinear hierarchical model





Nonlinear hierarchical model

This idea has been further explained in Zhu et al. (2019, *PNAS*) and developed in Zhu (2020, *New Phytologist*), to propose an integrated view of forest regrowth and modified growth due to environmental change.