


Introduction to Applied Bayesian Analysis in Wildlife Ecology

Jeffrey W. Doser

May 11, 2024



Bayesian Basics



Bayes' Theorem

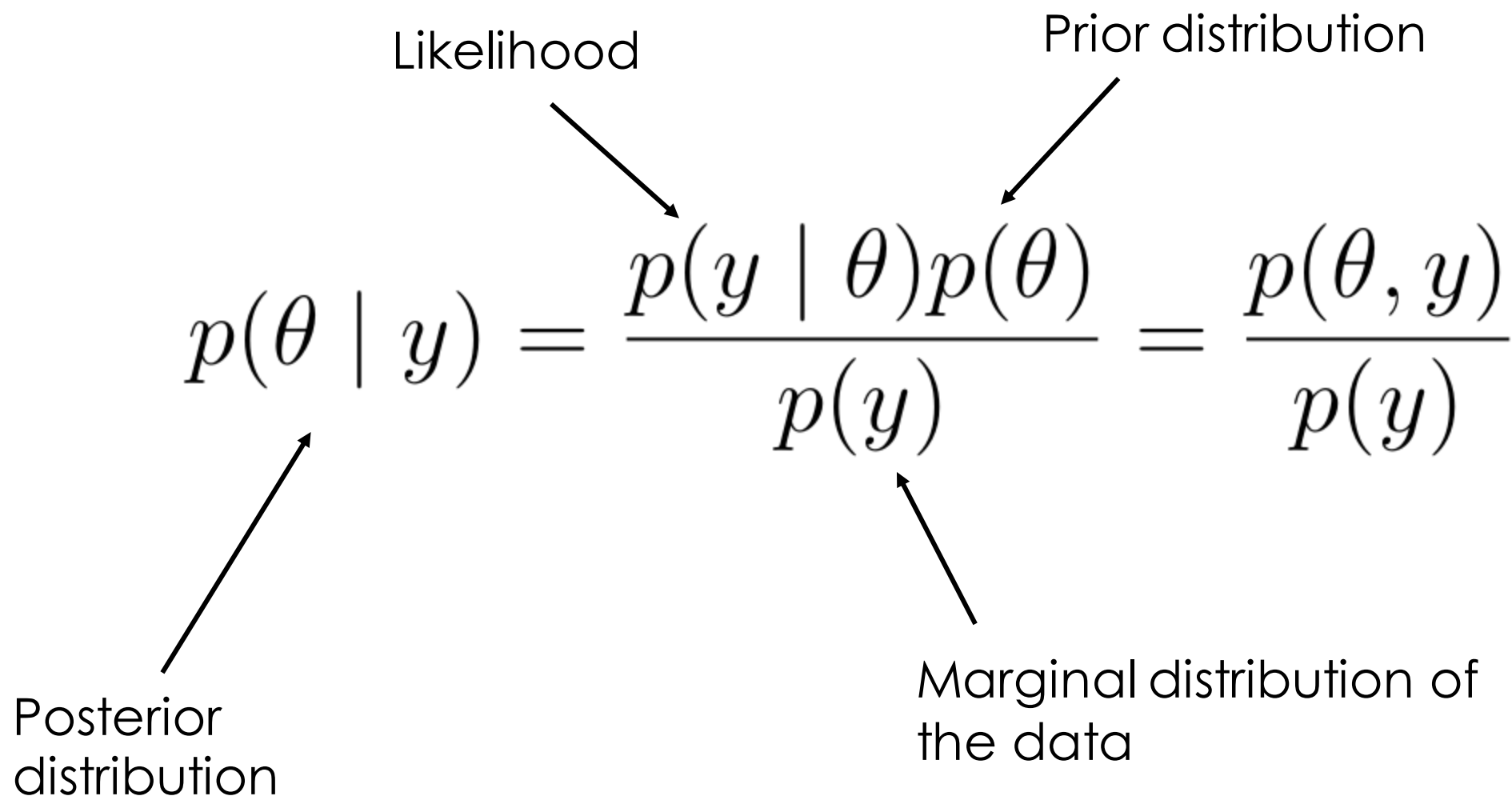


Diagram illustrating Bayes' Theorem with labels and arrows:

- Likelihood** points to $p(y | \theta)$ in the numerator.
- Prior distribution** points to $p(\theta)$ in the numerator.
- Posterior distribution** points to $p(\theta | y)$ on the left side of the equation.
- Marginal distribution of the data** points to $p(y)$ in the denominator.

$$p(\theta | y) = \frac{p(y | \theta)p(\theta)}{p(y)} = \frac{p(\theta, y)}{p(y)}$$

Bayes' Theorem

The diagram shows the equation for Bayes' Theorem with three labels and arrows pointing to specific parts of the formula:

- Likelihood**: An arrow points from this label to the term $p(y | \theta)$ in the numerator.
- Prior distribution**: An arrow points from this label to the term $p(\theta)$ in the numerator.
- Marginal distribution of the data**: An arrow points from this label to the term $p(y)$ in the denominator.

$$p(\theta | y) = \frac{p(y | \theta)p(\theta)}{p(y)} = \frac{p(\theta, y)}{p(y)}$$

How do we actually estimate the posterior distribution?

Posterior
distribution

Marginal distribution of
the data

Example: a simple linear model

What is the relationship between wingspan and body length?

We sample nine individuals, measure wingspan (y_i) and body length (x_i) for each individual.



Blue-eyed hooktail

Example: a simple linear model



Blue-eyed hooktail

What is the relationship between wingspan and body length?

We sample nine individuals, measure wingspan (y_i) and body length (x_i) for each individual.

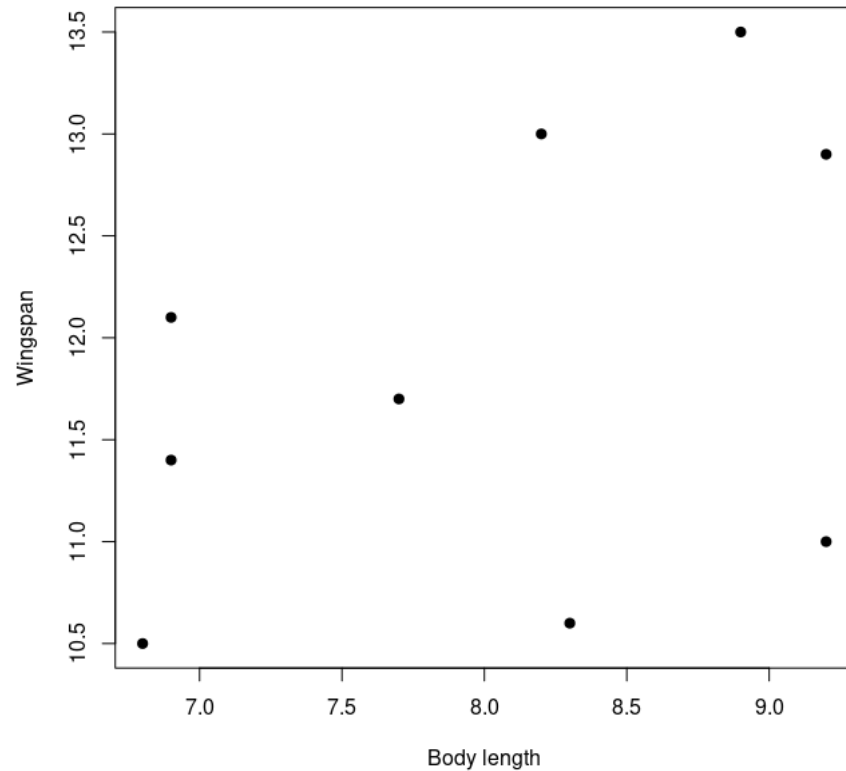
How do we specify a simple linear model in a Bayesian framework?

Three key steps in specifying a Bayesian model

Step 1: Determine the deterministic relationships you want to assess

Three key steps in specifying a Bayesian model

Step 1: Determine the deterministic relationships you want to assess



We want to quantify the linear relationship between body mass and wing length

Three key steps in specifying a Bayesian model

Step 2: Specify the likelihood (i.e., the distribution of the data)

Three key steps in specifying a Bayesian model

Step 2: Specify the likelihood (i.e., the distribution of the data)

$$y_i \sim \text{Normal}(\beta_0 + \beta_1 \cdot x_i, \sigma^2)$$

Body mass is a continuous metric, so we will use a normal distribution

Three key steps in specifying a Bayesian model

Step 2: Specify the likelihood (i.e., the distribution of the data)

$$y_i \sim \text{Normal}(\beta_0 + \beta_1 \cdot x_i, \sigma^2)$$

Body mass is a continuous metric, so we will use a normal distribution

We will talk about other distributions when we get to GLMs

Three key steps in specifying a Bayesian model

Step 3: Specify the prior distributions

Three key steps in specifying a Bayesian model

Step 3: Specify the prior distributions

$$\beta_0 \sim \text{Normal}(0, 1000)$$

$$\beta_1 \sim \text{Normal}(0, 1000)$$

$$\sigma \sim \text{Inverse-Gamma}(0.001, 0.001)$$

Three key steps in specifying a Bayesian model

Step 1: Determine the deterministic relationships you want to assess

Step 2: Specify the likelihood


Step 3: Specify the prior distributions

Three key steps in specifying a Bayesian model

Step 1: Determine the deterministic relationships you want to assess

Step 2: Specify the likelihood

Step 3: Specify the prior distributions



Only this third step is unique to Bayesian analysis. The other two you should be familiar with from maximum likelihood!

How to specify a prior distribution?

How to specify a prior distribution?

- Need a prior distribution for **every** parameter in the model
 - For our hooktail example, we have three parameters: the intercept, the slope, and the residual variance

How to specify a prior distribution?

- Need a prior distribution for **every** parameter in the model
 - For our hooktail example, we have three parameters: the intercept, the slope, and the residual variance (or standard deviation)
- **We should choose a distribution with possible values that make sense for the given parameter**

How to specify a prior distribution?

- Need a prior distribution for **every** parameter in the model
 - For our hooktail example, we have three parameters: the intercept, the slope, and the residual variance (or standard deviation)
- **We should choose a distribution with possible values that make sense for the given parameter**

Why is the normal distribution a bad prior distribution for a variance or standard deviation parameter?

How to specify a prior distribution?

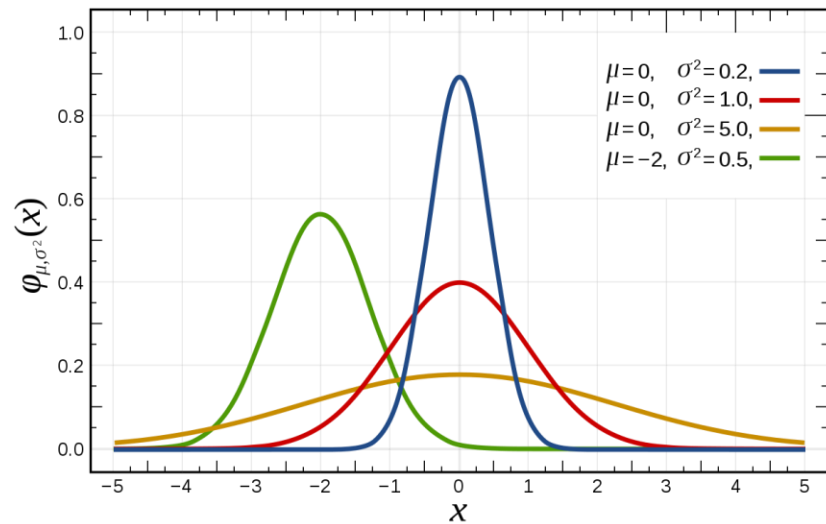
- The possible values a distribution takes is called the **support** of the distribution

How to specify a prior distribution?

- The possible values a distribution takes is called the **support** of the distribution
- Different distributions are used for different types of variables:
 - Positive only, positive or negative
 - Continuous or discrete
 - Continuous, but contained within an interval (e.g., 0-1)

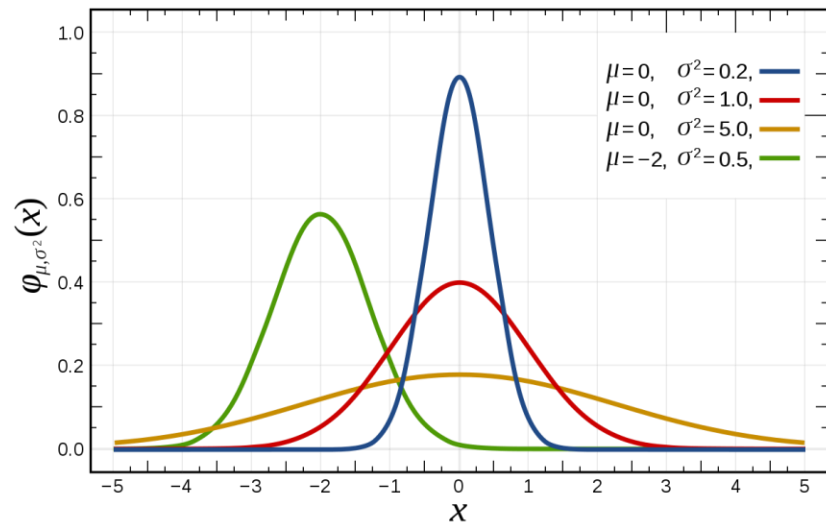
Continuous distributions for positive and negative numbers

Normal (Gaussian) distribution

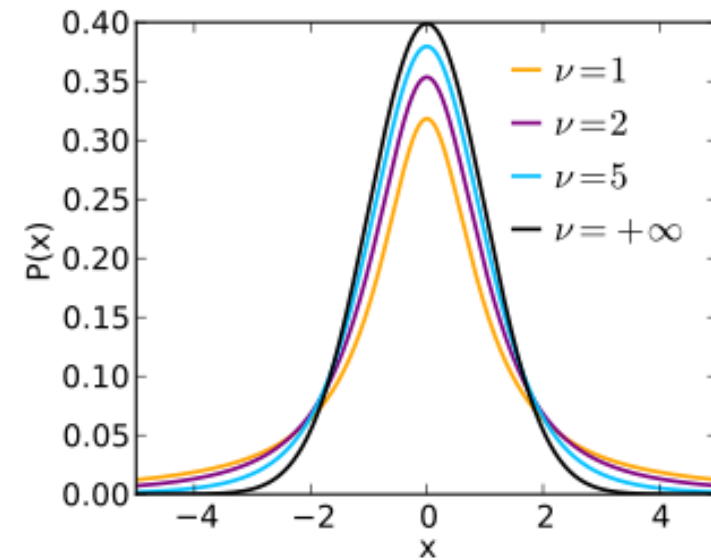


Continuous distributions for positive and negative numbers

Normal (Gaussian) distribution

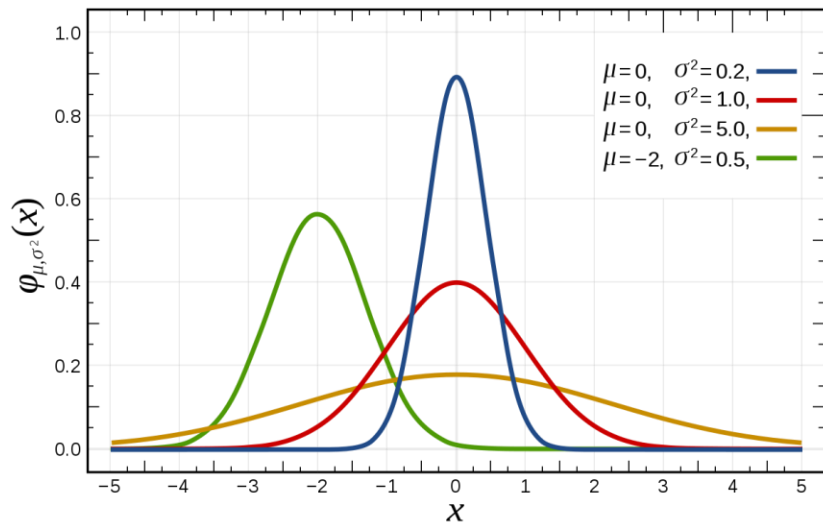


Student's t distribution

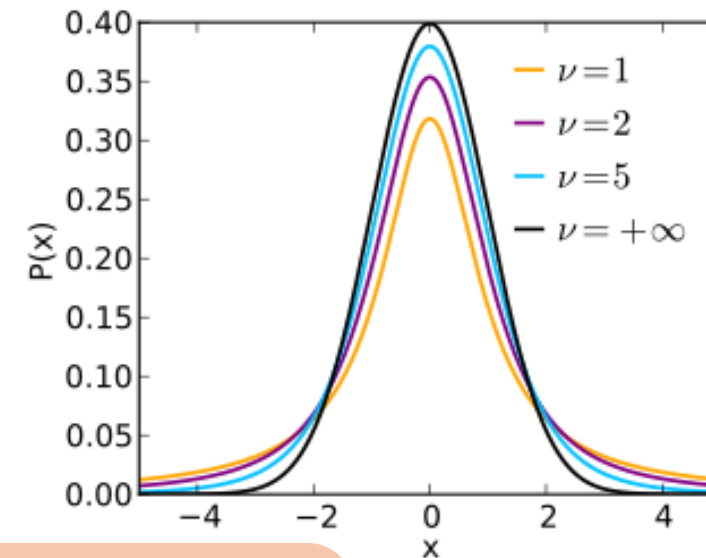


Continuous distributions for positive and negative numbers

Normal (Gaussian) distribution



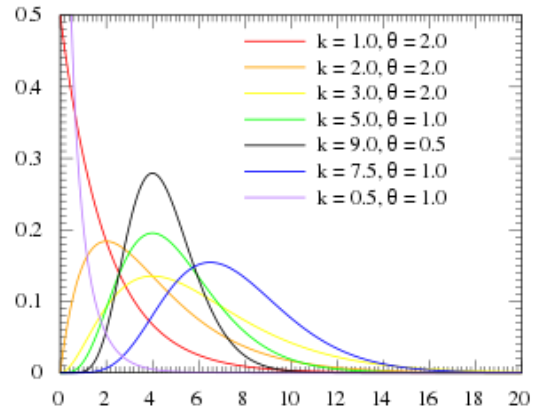
Student's t distribution



Often used for intercept/slope parameters that can be positive and/or negative

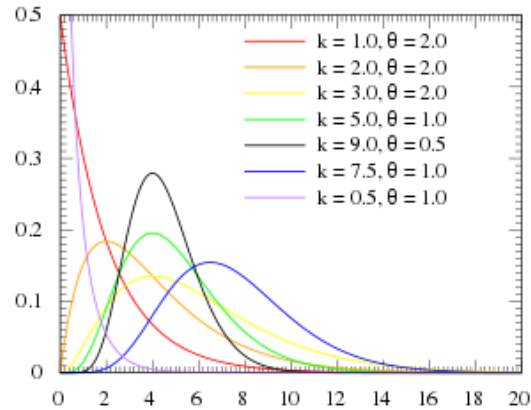
Distributions for positive only, continuous values

Gamma

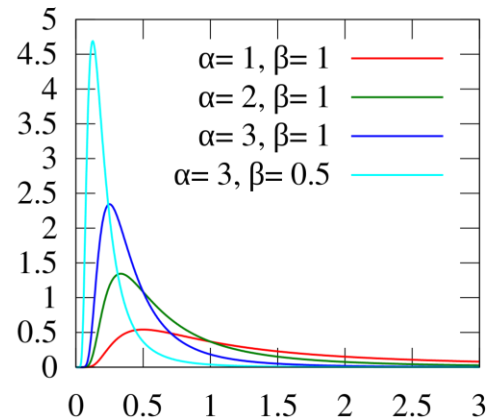


Distributions for positive only, continuous values

Gamma

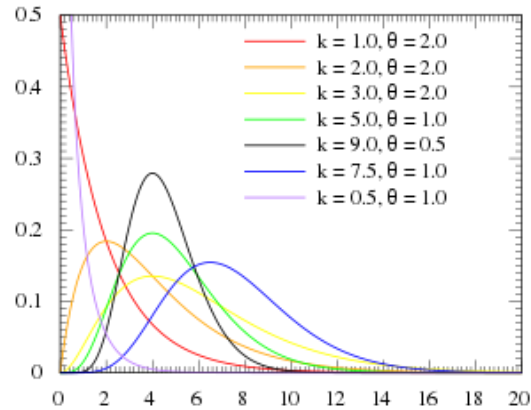


Inverse-Gamma

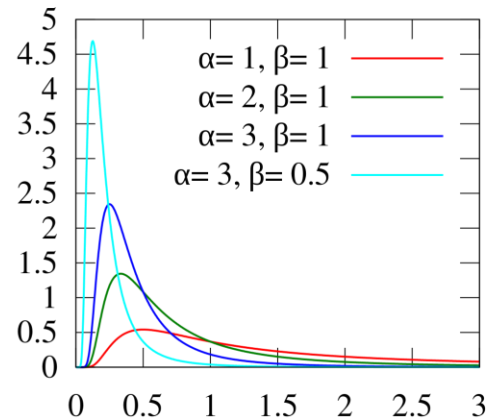


Distributions for positive only, continuous values

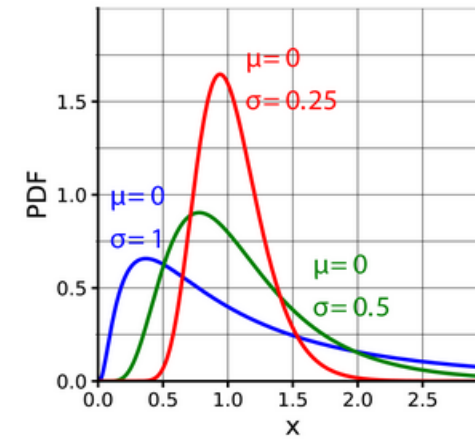
Gamma



Inverse-Gamma

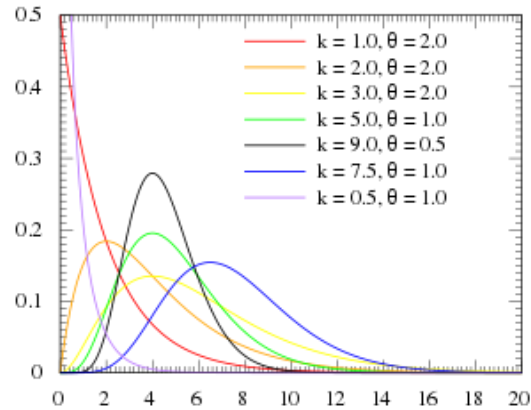


Log-normal

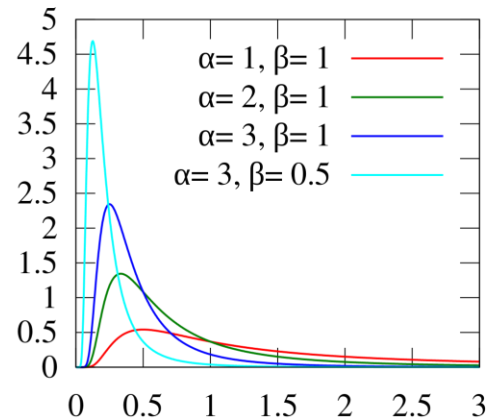


Distributions for positive only, continuous values

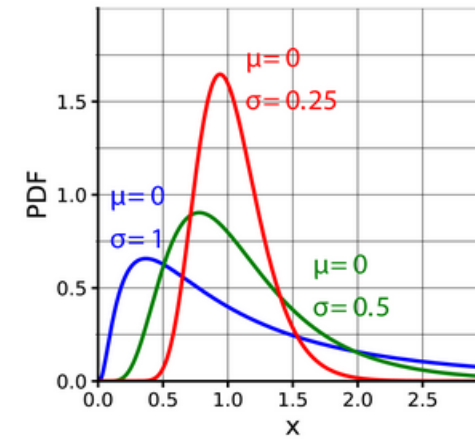
Gamma



Inverse-Gamma



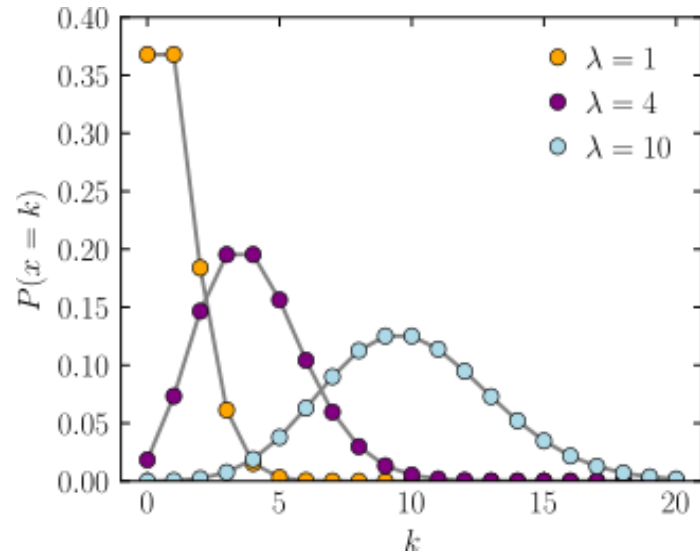
Log-normal



Useful for variance/standard deviation parameters

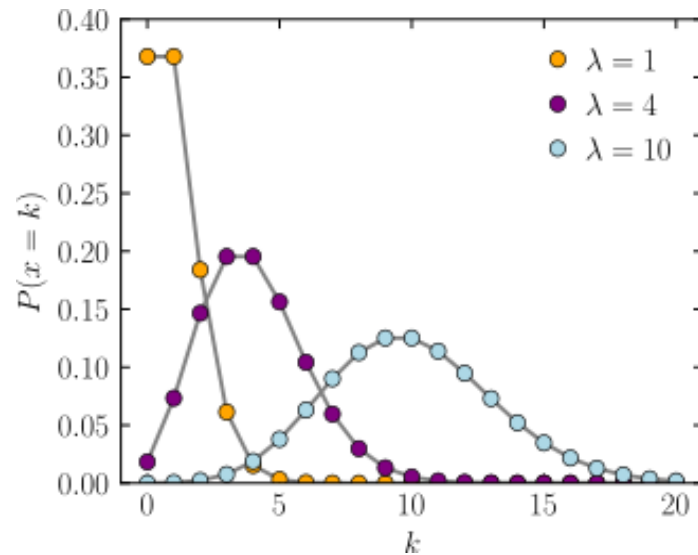
Distributions for positive only, discrete values

Poisson

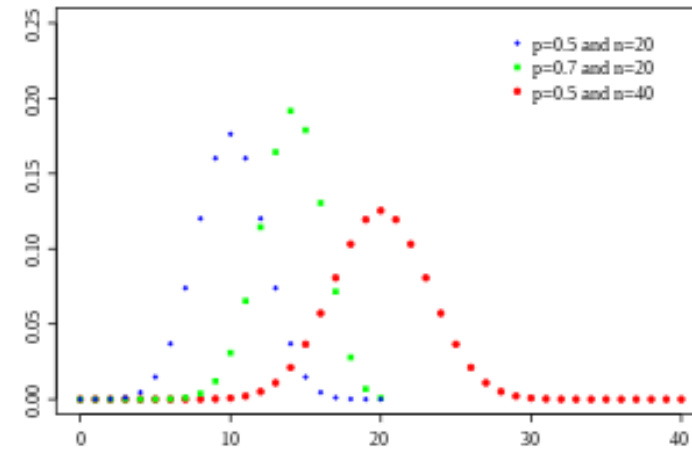


Distributions for positive only, discrete values

Poisson

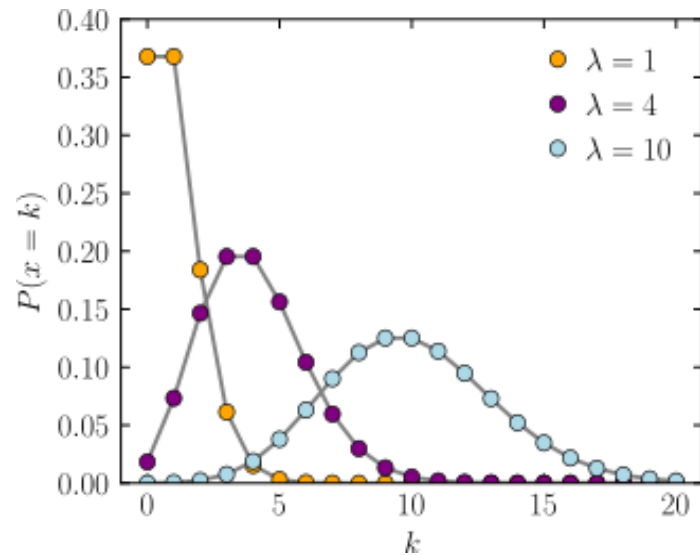


Binomial

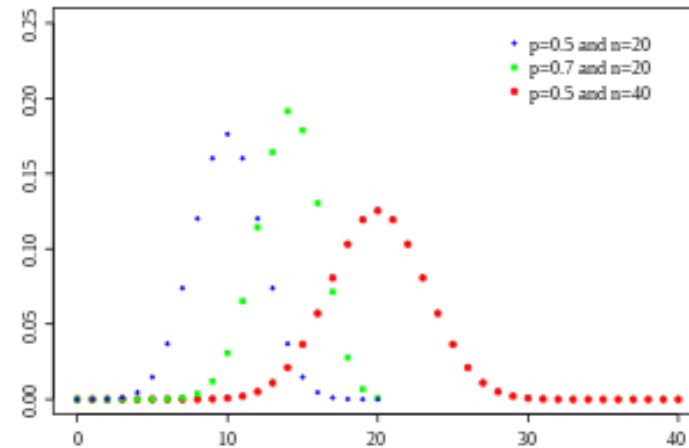


Distributions for positive only, discrete values

Poisson



Binomial



Very commonly used in wildlife ecology

So many distributions!

<https://www.math.wm.edu/~leemis/chart/UDR/UDR.html>

Choosing the prior distribution

- It is up to you to choose the prior distribution



Choosing the prior distribution

- It is up to you to choose the prior distribution
- Prior distributions can be either:
 - **Informative:** you have a lot of previous knowledge on the system that you want this to influence your conclusions



Choosing the prior distribution

- It is up to you to choose the prior distribution
- Prior distributions can be either:
 - **Informative**: you have a lot of previous knowledge on the system that you want this to influence your conclusions
 - **Weakly informative/vague**: you don't have a lot of previous knowledge on the system and/or you want the conclusions to be drawn primarily from the current data set



Choosing the prior distribution

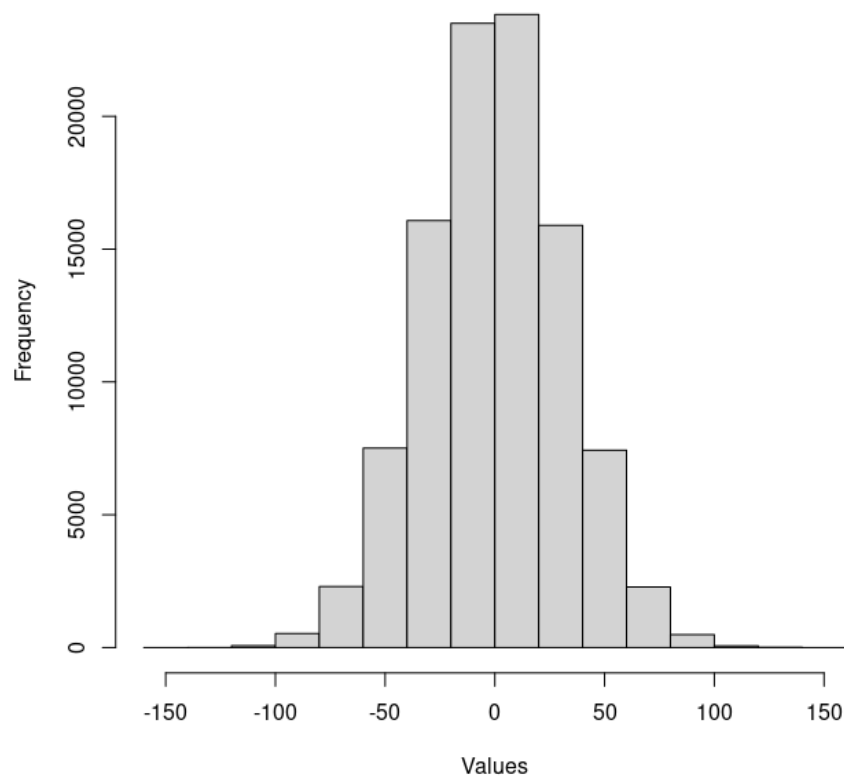
- It is up to you to choose the prior distribution
- Prior distributions can be either:
 - **Informative**: you have a lot of previous knowledge on the system that you want this to influence your conclusions
 - **Weakly informative/vague**: you don't have a lot of previous knowledge on the system and/or you want the conclusions to be drawn primarily from the current data set
- In ecology, vague prior distributions are the most common.



Example: priors in our simple linear model

$$\beta_0 \sim \text{Normal}(0, 1000)$$

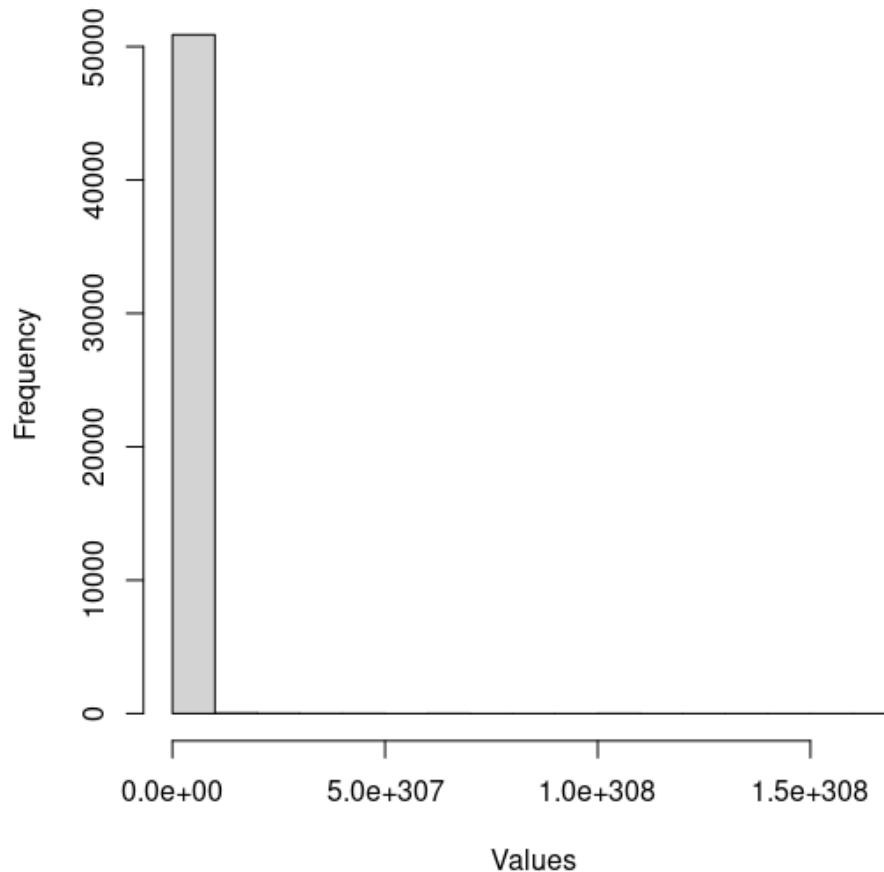
$$\beta_1 \sim \text{Normal}(0, 1000)$$



Blue-eyed hooktail

Example: priors in our simple linear model

$$\sigma \sim \text{Inverse-Gamma}(0.001, 0.001)$$



Blue-eyed hooktail

Choosing priors in practice

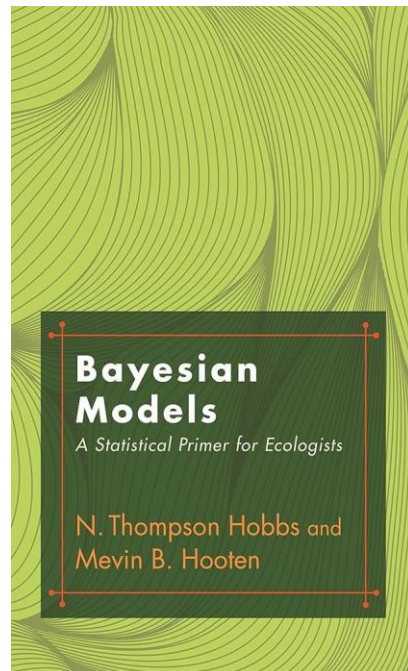
- Certain priors lead to faster models (called **conjugate priors**)

Choosing priors in practice

- Certain priors lead to faster models (called **conjugate priors**)
- Many modern R packages for Bayesian analysis (e.g., `brms`) choose vague priors by default

Choosing priors in practice

- Certain priors lead to faster models (called **conjugate priors**)
- Many modern R packages for Bayesian analysis (e.g., `brms`) choose vague priors by default
- Many good standard recommendations for priors in wildlife ecology



Methods in Ecology and Evolution



COMMENTARY | [Open Access](#) |

The use of Bayesian priors in Ecology: The good, the bad and the not great

Katharine M. Banner Kathryn M. Irvine, Thomas J. Rodhouse

Bayesian estimation

$$p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{p(y)} = \frac{p(\theta, y)}{p(y)}$$

- Depending on our choice of likelihood and prior distributions, the above can sometimes turn into just plugging in numbers into equations.
- Other times, we need a sophisticated algorithm to estimate the posterior distribution
- In general, we use an approach called **Markov Chain Monte Carlo**

Markov chain Monte Carlo (MCMC)

- An algorithm used to approximate the posterior distribution
- Algorithms developed in 1950s-1980s, but popularized in 1990s by Gelfand and Smith (1990).
- **Key idea: repeatedly sample values from a mathematical expression to generate a distribution of values for each parameter. This distribution will eventually approximate the posterior distribution for each parameter.**

Markov chain Monte Carlo (MCMC)

- An algorithm used to approximate the posterior distribution
- Algorithms developed in 1950s-1980s, but popularized in 1990s by Gelfand and Smith (1990).
- **Key idea: repeatedly sample values from a mathematical expression to generate a distribution of values for each parameter. This distribution will eventually approximate the posterior distribution for each parameter.**



Monte

MCMC Steps

1. Choose initial values for all parameters

MCMC Steps

1. Choose initial values for all parameters
2. Propose a new value for each parameter, one at a time

MCMC Steps

1. Choose initial values for all parameters
2. Propose a new value for each parameter, one at a time
3. Repeat step 2 a whole bunch of times

MCMC Steps

1. Choose initial values for all parameters
2. Propose a new value for each parameter, one at a time
3. Repeat step 2 a whole bunch of times
4. Summarize

MCMC Step 1: set initial values

- Generally, the initial (starting) values are not too important

MCMC Step 1: set initial values

- Generally, the initial (starting) values are not too important
- Needs to be a valid number for the given parameter (e.g., no negative number for a variance parameter)

MCMC Step 1: set initial values

- Generally, the initial (starting) values are not too important
- Needs to be a valid number for the given parameter (e.g., no negative number for a variance parameter)
- Can be important for more complicated models that are difficult to estimate

MCMC Step 1: set initial values

- Generally, the initial (starting) values are not too important
- Needs to be a valid number for the given parameter (e.g., no negative number for a variance parameter)
- Can be important for more complicated models that are difficult to estimate
- Generally unimportant when working with `brms` or other formula-based R packages

MCMC Step 2: Propose new value

- Propose a new value for each parameter one at a time based on a statistical algorithm.

MCMC Step 2: Propose new value

- Propose a new value for each parameter one at a time based on a statistical algorithm.
- For some parameters, we always accept the proposed value because our algorithm is very efficient.

MCMC Step 2: Propose new value

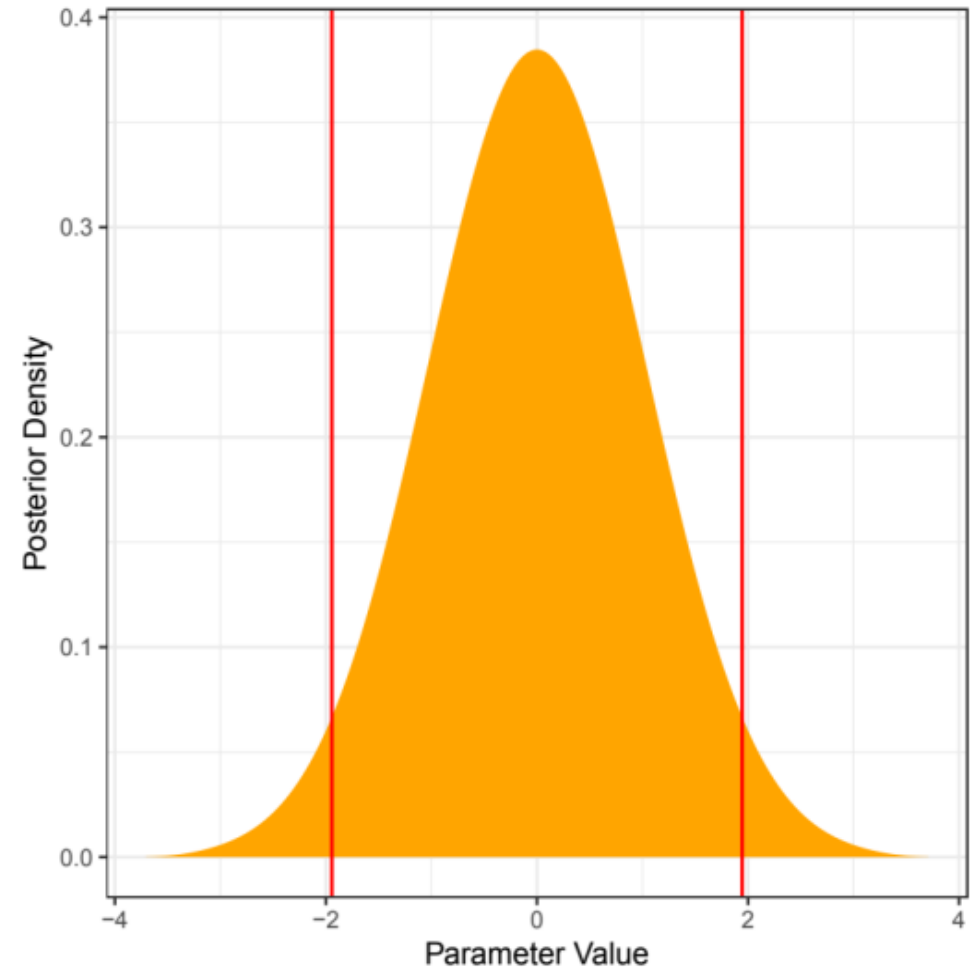
- Propose a new value for each parameter one at a time based on a statistical algorithm.
- For some parameters, we always accept the proposed value because our algorithm is very efficient.
- For parameters with less efficient algorithms, we will accept the new value with some probability p .

MCMC Step 3: Repeat

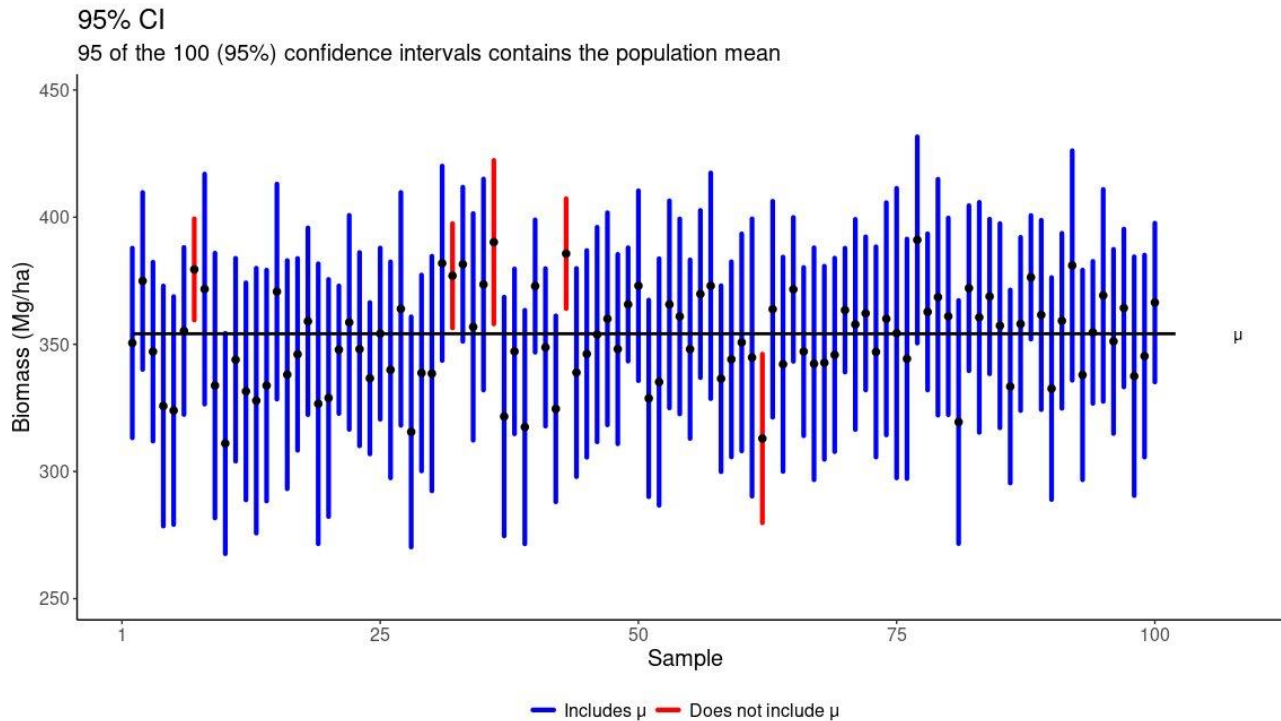
- Repeat Step 2 "many" times to generate a set of samples from the posterior distribution for each parameter.

MCMC Step 4: Summarize

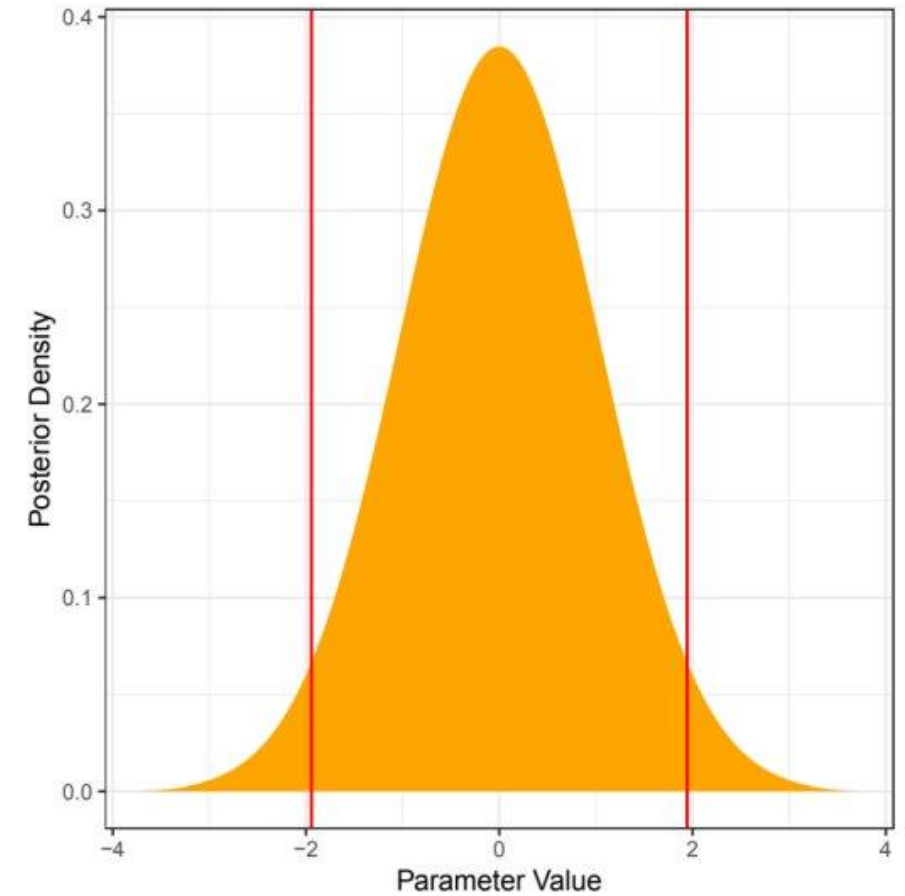
- Point estimate: mean, median, mode
- Uncertainty: standard deviation, or 95% credible interval
- Credible intervals are simply the quantiles of the set of values (e.g., a 95% credible interval is formed from the 2.5% quantile and the 97.5% quantile)



Confidence interval vs. Credible interval



Confidence interval



Credible interval

Main difficulty with MCMC

- Recall from our definition of MCMC: the distribution of values generated from MCMC will **eventually** approximate the posterior distribution for each parameter

Main difficulty with MCMC

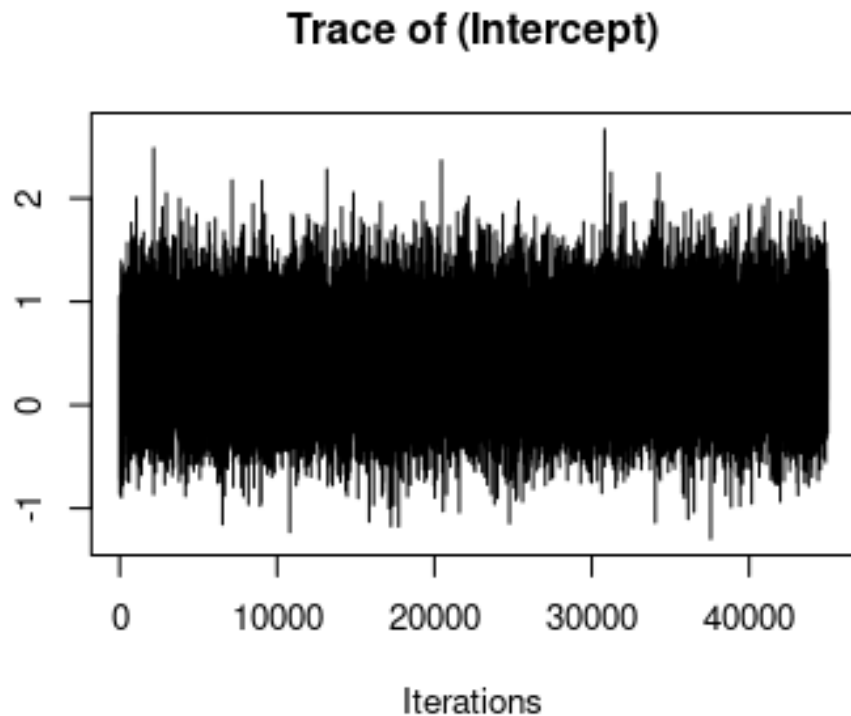
- Recall from our definition of MCMC: the distribution of values generated from MCMC will **eventually** approximate the posterior distribution for each parameter
- How do we know when the values we have are what we want (the posterior)?

Main difficulty with MCMC

- Recall from our definition of MCMC: the distribution of values generated from MCMC will **eventually** approximate the posterior distribution for each parameter
- How do we know when the values we have are what we want (the posterior)?
- **Convergence assessments**
- Main approach:
 - Run more than one MCMC "chains", each with different starting values
 - Compare the values from the different chains. If they are similar, we are good to go. If not, need to run longer.

Convergence assessment

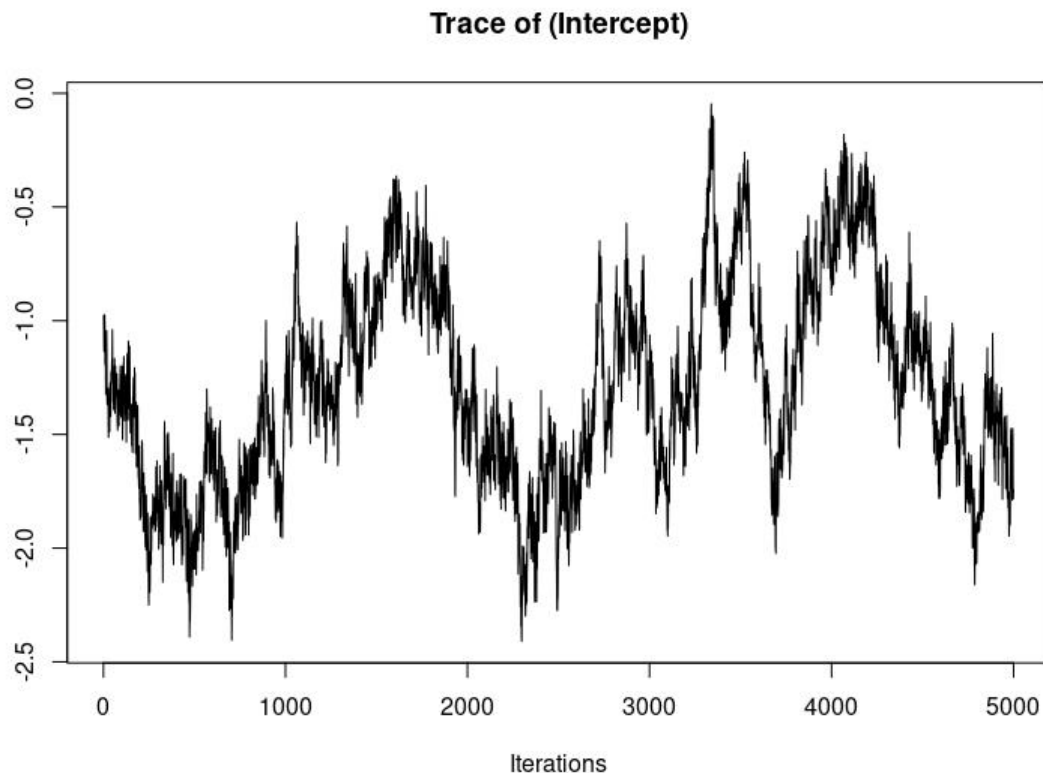
Visual assessment of traceplots



- A good traceplot is one that looks like a cross-section of grass
- Values are centered around a single value across all iterations
- If all traceplots look like the one on the right for all parameters, we say the model has "converged" and we can start interpreting things

Convergence assessment

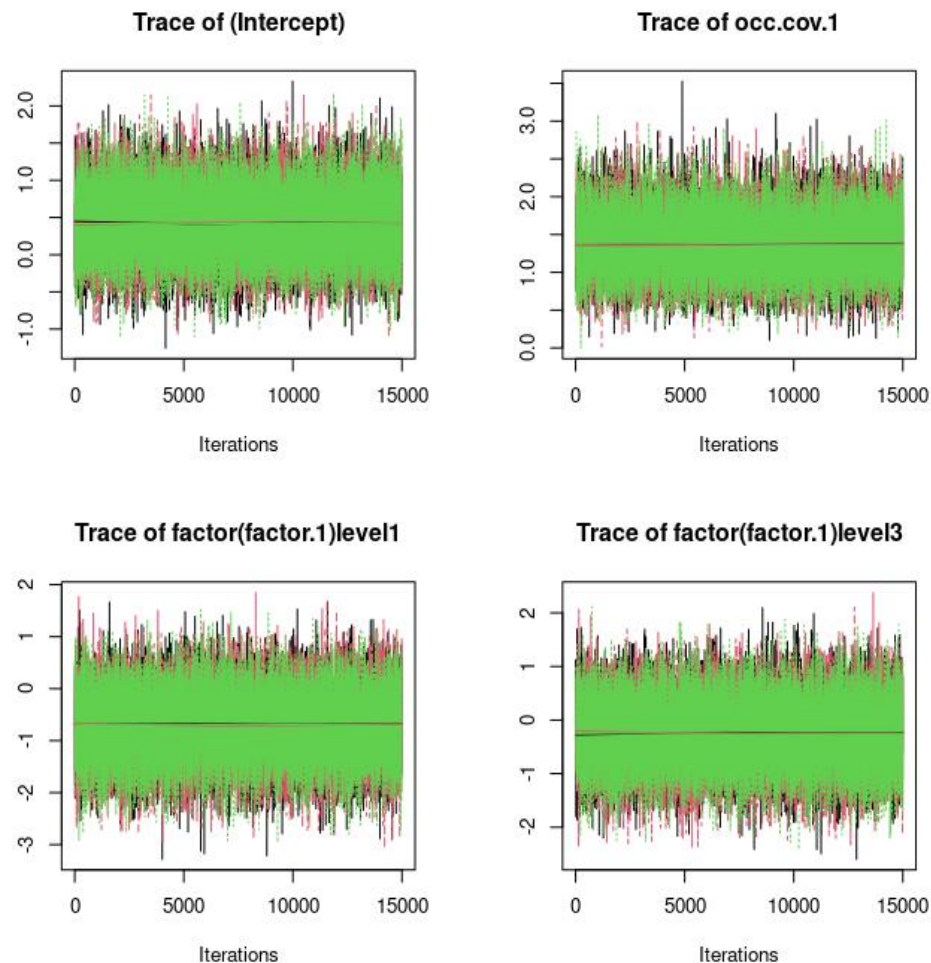
Visual assessment of traceplots



- A bad traceplot is not centered around a single value
- A bad traceplot shows high autocorrelation (i.e., it gets stuck in places and moves slowly). A traceplot with high autocorrelation has bad **mixing**
- The traceplot on the left does not show convergence

Convergence assessment

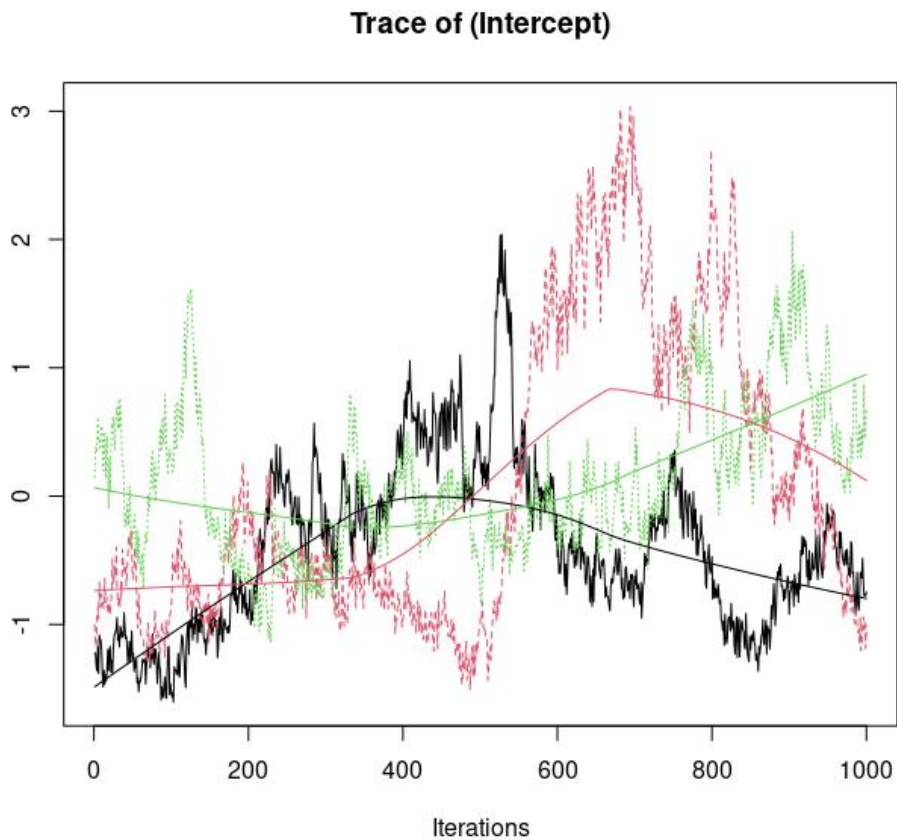
Gelman-Rubin diagnostic (Rhat)



- Run multiple chains
- Compare the amount of variation across chains to amount of variation within chains.
- Values less than 1.1 suggest convergence
- Helpful to plot traceplots of chains overlayed
- On left, all $R_{hat} < 1.001$

Convergence assessment

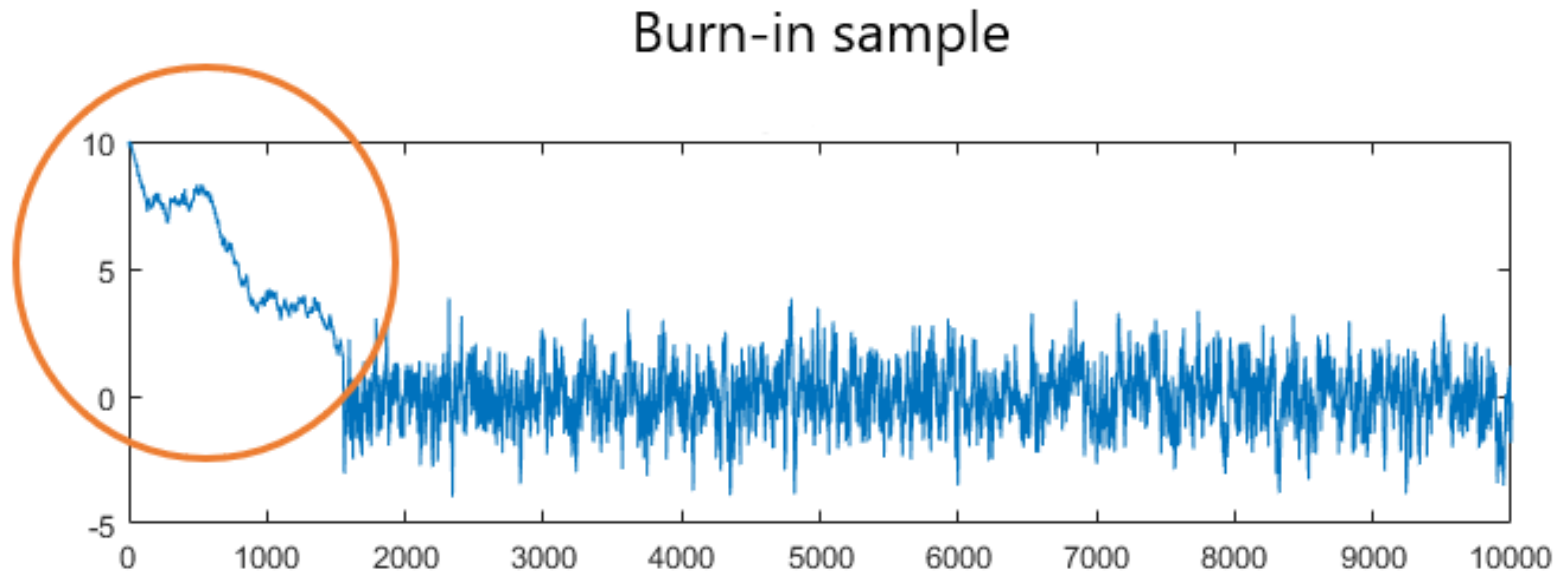
Gelman-Rubin diagnostic (Rhat)



- The chains on the left have not converged
- $R_{\text{hat}} = 1.43$

Convergence assessment: burn-in

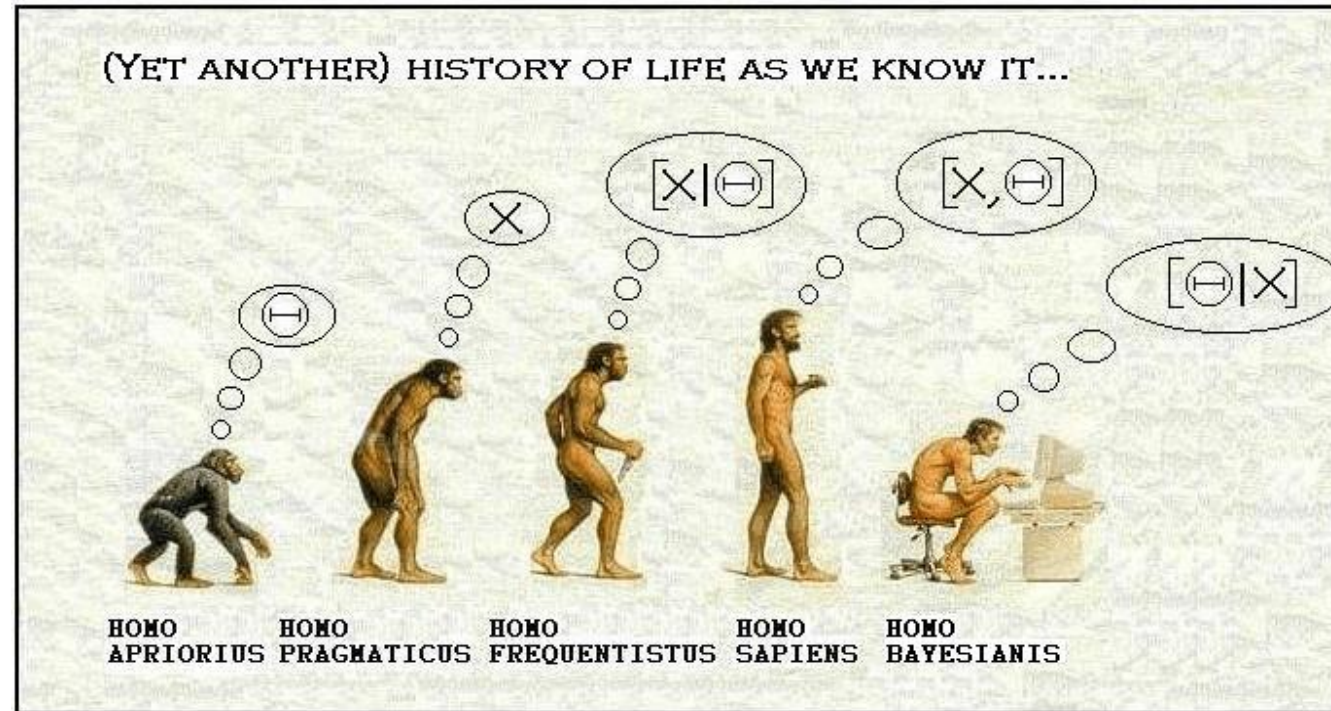
- The initial part of the chain may be sensitive to what initial value you choose
- **Burn-in** is the set of samples at the beginning of the chain that you throw away.
- Sometimes also called **warm-up**



MCMC: what do you need to specify?

- Prior distributions (software may do this by default)
- Initial values (software may do this by default)
- Number of samples/iterations
- Burn-in
- Thinning rate: how often do you want to save an MCMC sample

Bayesian analysis



Questions?

Bayesian Software



How to fit a Bayesian model?

- Options over the last few years have exploded. TONS of development to make Bayesian modeling more accessible.
- Three options for how to do this, each with pros and cons
 1. Write your own MCMC algorithm (sampler)
 2. Use specialized Bayesian programming languages
 3. Use formula-based R packages

1. Write your own MCMC algorithm

1. Write your own MCMC algorithm

- The most time-consuming approach, in terms of learning and writing the code
- Requires in-depth knowledge of how MCMC works
- Not necessary for well-developed models
- Not practical for most wildlife researchers/professionals

1. Write your own MCMC algorithm

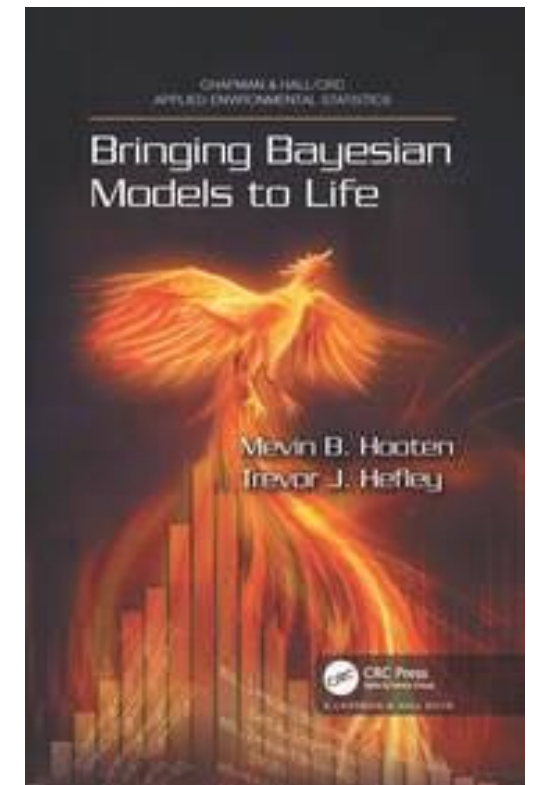
- The most time-consuming approach, in terms of learning and writing the code
- Requires in-depth knowledge of how MCMC works
- Not necessary for well-developed models
- Not practical for most wildlife researchers/professionals

- Gives you the most flexibility in customizing your model
- Ideal for complicated and/or slow models
- Can be rewarding if you really like stats/math

1. Write your own MCMC algorithm

- The most time-consuming approach, in terms of learning and writing the code
- Requires in-depth knowledge of how MCMC works
- Not necessary for well-developed models
- Not practical for most wildlife researchers/professionals

- Gives you the most flexibility in customizing your model
- Ideal for complicated and/or slow models
- Can be rewarding if you really like stats/math



2. Specialized Bayesian programming languages

2. Specialized Bayesian programming languages

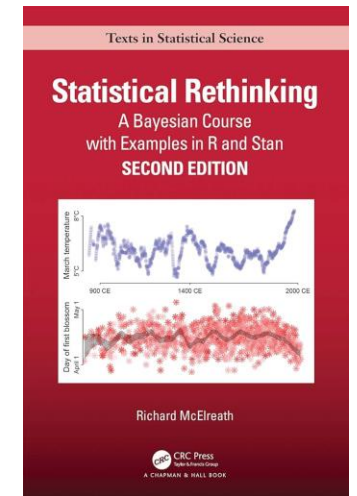
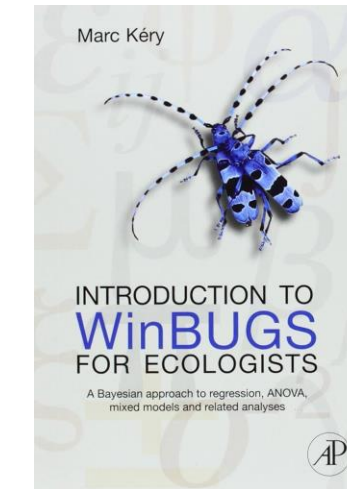
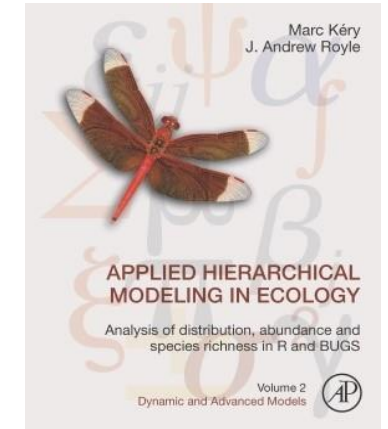
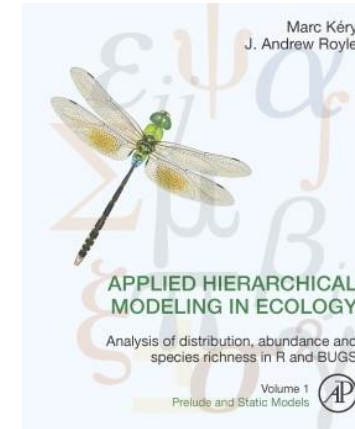
- Requires learning specialized programming languages
- Can be time-consuming to learn
- Very slow for complex models

2. Specialized Bayesian programming languages

- Requires learning specialized programming languages
 - Can be time-consuming to learn
 - Very slow for complex models
-
- Gives you a lot of flexibility to write complex models, without needing to get too in-depth on how MCMC works
 - Tons of learning resources available
 - Syntax is very helpful for thinking critically about your model
 - Many wildlife ecologists use this approach

2. Specialized Bayesian programming languages

- Requires learning specialized programming languages
- Can be time-consuming to learn
- Very slow for complex models



- Gives you a lot of flexibility to write complex models, without needing to get too in-depth on how MCMC works
- Tons of learning resources available
- Syntax is very helpful for thinking critically about your model
- Many wildlife ecologists use this approach

Bayesian programming languages

- Three most common languages are JAGS (going out of style), NIMBLE, and Stan
- Probabilistic programming languages



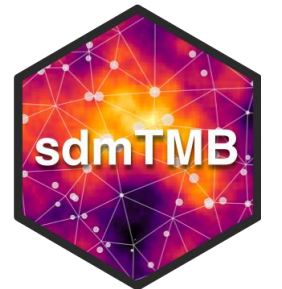
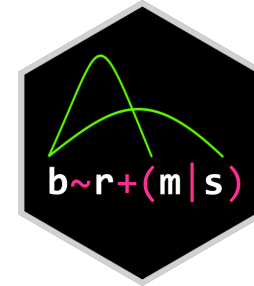
NIMBLE



Stan

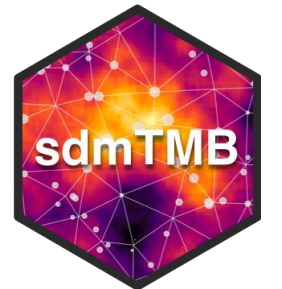
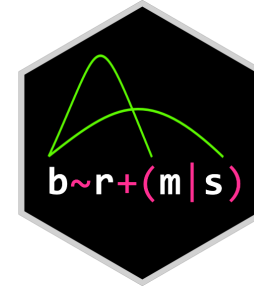
We won't focus much on NIMBLE or Stan today. However, the GitHub page has all exercises we do also performed in NIMBLE if you want to compare later.

3. Formula-based R packages



3. Formula-based R packages

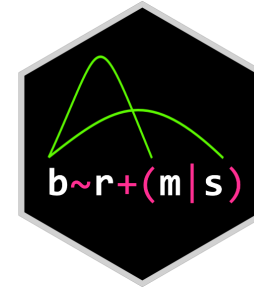
- Can only fit models included in the package functionality (i.e., less flexibility)
- Differing syntax across packages could make more time spent learning



3. Formula-based R packages

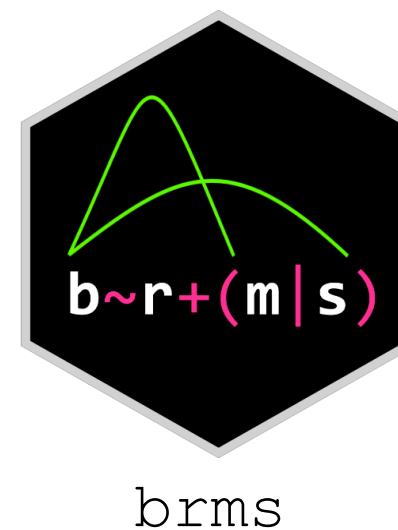
- Can only fit models included in the package functionality (i.e., less flexibility)
- Differing syntax across packages could make more time spent learning

- Often specialized for a specific model type, which means they may run faster than NIMBLE/Stan
- MUCH less coding. Use familiar syntax to `lm()`, `glm()`, `lmer()`, etc.
- Tons of development over the last few years, which means more options to fit complex models with less coding
- More reproducible



Formula-based R packages

- We will focus on using the `brms` package (Bürkner 2017)
- **B**ayesian **R**egression **M**odels using **S**tan
- Very flexible for fitting a lot of different models
- Extremely similar syntax to `lme4` (Bates et al. 2015)
- `brms` (and `Stan`) use a special type of MCMC called Hamiltonian Monte Carlo (Hamilton is another good cat name...)



Questions?

Exercise



2a-intro-to-brms.R

