

Introduction to Bayesian Analysis for Seabird Research

Pacific Seabird Group 2023 Annual Meeting
14 February 2023, 3:00-5:00 pm PT

Workshop Goals

1. Recognize **value of Bayesian framework** for seabird research.
2. Understand **fundamental differences** between Frequentist and Bayesian frameworks.
3. Learn **basic components** of a Bayesian analysis.
4. Learn how to **interpret results** from a Bayesian analysis.
5. Build **community and share resources** for future learning.



Amelia DuVall
(She/Her)

Graduate Research Assistant
School of Aquatic & Fishery Sciences
University of Washington



Dr Tim Jones
(He/Him)

Research Scientist
School of Aquatic & Fishery Sciences
University of Washington



Michelle Kissling
(She/Her)

PhD Student
University of Montana
Wildlife Biology Program



Molly McDevitt
(She/Her)

PhD Student
University of Montana
Wildlife Biology Program

Introductions

1. Name
2. Title/Affiliation
3. Pronouns
4. What brought you to this workshop?
5. On a scale of 1 (*not at all! I want to learn!*) to 5 (*pretty comfortable, I have run a few models on my own*), how familiar are you with Bayesian analyses?

Seabird data is messy.

Some of the challenges include:

- Difficult to get to colonies
- Hard to find individuals and/or hard to count them all
- Individuals are inaccessible during portions of life history

Which can lead to small sample sizes, gaps, and uncertainty in our datasets.

Good news, many of these issues can be addressed using Bayesian methods!

Overview

1. Welcome, Goals & Introductions
2. Bayes: An Updated Way of Thinking
3. Bayesian Data Analysis
4. Interpreting Bayesian Models
5. Building Your Own Models
6. Tips/Tricks and Further Information
7. Workshop Wrap-up & Optional Happy Hour

Bayes:

An Updated Way of Thinking

What is your belief?

I am going to show a video of some penguins attempting to 'board' an iceberg.

What is your initial expectation of the probability that an individual attempt is successful?

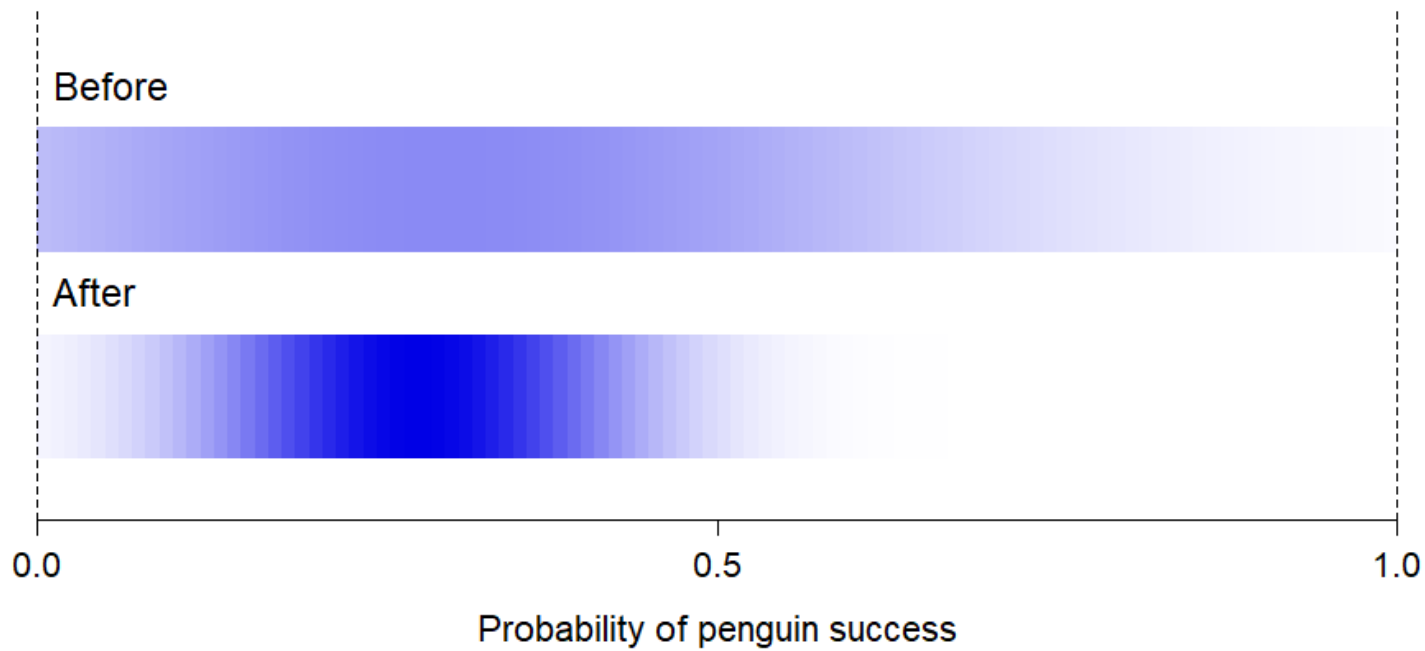


What is your belief?

I am going to show a video of some penguins attempting to 'board' an iceberg.

What is your initial expectation of the probability that an individual attempt is successful?

What do you think the probability is now that you have seen some attempts?



Thinking like a Bayesian

Bayesian thinking is a means of updating your certainty in something dependent on **data** and your **prior knowledge**.

It provides a natural way to incorporate existing knowledge into the estimation of processes and uncertainty.

It also provides a more natural way of thinking about uncertainty (we'll get to that...).

Schools of Thought:

Bayesian vs. Frequentist

DID THE SUN JUST EXPLODE?
(IT'S NIGHT, SO WE'RE NOT SURE.)

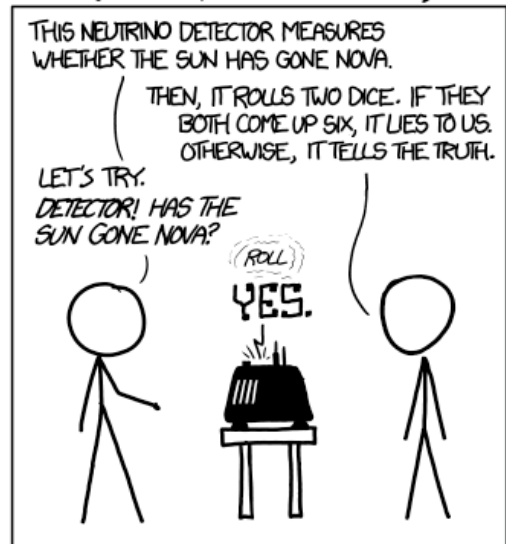
THIS NEUTRINO DETECTOR MEASURES
WHETHER THE SUN HAS GONE NOVA.

THEN, IT ROLLS TWO DICE. IF THEY
BOTH COME UP SIX, IT LIES TO US.
OTHERWISE, IT TELLS THE TRUTH.

LET'S TRY.

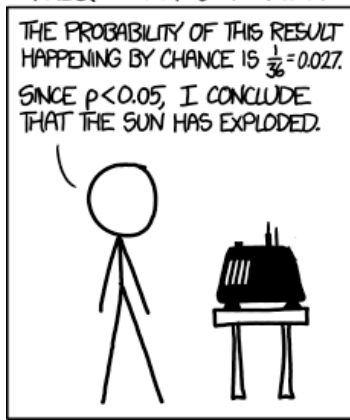
DETECTOR! HAS THE
SUN GONE NOVA?

(ROLL)
YES.



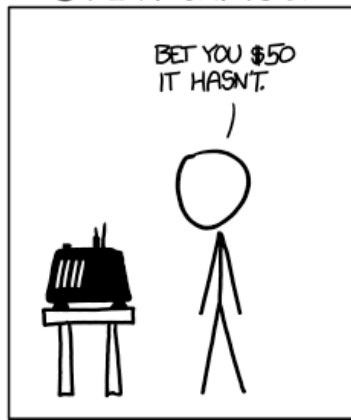
FREQUENTIST STATISTICIAN:

THE PROBABILITY OF THIS RESULT
HAPPENING BY CHANCE IS $\frac{1}{36} = 0.027$.
SINCE $p < 0.05$, I CONCLUDE
THAT THE SUN HAS EXPLODED.



BAYESIAN STATISTICIAN:

BET YOU \$50
IT HASN'T.



Bayesian

Frequentist

Data	Fixed	Repeatable random samples
Probability	Degrees of belief	Based on long-run frequency
Parameters	Random variables	Fixed but unknown
Calculates	$P(\text{hypothesis} \text{data})$	$P(\text{data} \text{hypothesis})$
Source of information	data/observations + prior belief	data/observations
Hypothesis testing	Bayes Factor (re-expression of Bayes' theorem)	p-values
Interval interpretation	Credible intervals	Confidence intervals
Point estimate	Mean, Median, Mode of posterior probability distribution	Maximum likelihood estimation (MLE), least squares estimate

Using a **frequentist** approach, a solution can be found for many (often linear) problems using either a closed form equation or a relatively simple algorithm.

For **Bayesian** analyses, even simple problems usually do not have an analytical solution and require modern computational techniques to find the solution(s).

In addition to debates about whether science should be subjective or objective (i.e. priors and belief??), this led to the dominance of **frequentist** thinking.

There's no such thing as a Bayesian (only) model.

A Bayesian model is not a separate class of model.

- You can fit a simple linear regression using Bayesian techniques
- You can fit an age-structured population model using frequentist methods

The two approaches are more based on the philosophical differences between the two and what that means for interpretation of the model and it's parameters.

Bayes Theorem

“Bayes’ theorem allows us to take our **beliefs** about the world, combine them with **data**, and then transform this combination into an estimate of the strength of our beliefs given the evidence we’ve observed.”

- *Bayesian Statistics The Fun Way* (Kurt 2019)

Notation Refresher

Before going any further, some terminology:

$p(\mathbf{A})$: probability of \mathbf{A}

- \mathbf{A} can be a value (i.e. a rate, or parameter), or occurrence

$p(\mathbf{A} \mid \mathbf{B})$: probability of \mathbf{A} given that \mathbf{B} is true

- This is a conditional probability

We will more often think of these in terms of probability distributions

- $p(\mathbf{A} = \mathbf{a})$: probability that \mathbf{A} has a specific value \mathbf{a}

Bayes Theorem

$$p(\textcolor{blue}{A} | \textcolor{red}{B}) =$$

The probability of A given B is equal to...

Bayes Theorem

$$p(\textcolor{blue}{A}|\textcolor{red}{B}) = p(\textcolor{blue}{A})$$

The probability of $\textcolor{blue}{A}$ given $\textcolor{red}{B}$ is equal to the probability of $\textcolor{blue}{A}$...

Bayes Theorem

$$p(\textcolor{blue}{A}|\textcolor{red}{B}) = p(\textcolor{blue}{A}) \cdot p(\textcolor{red}{B}|\textcolor{blue}{A})$$

The probability of **A** given **B** is equal to the probability of **A** multiplied by the probability of **B** given **A**...

Bayes Theorem

$$p(\textcolor{blue}{A}|\textcolor{red}{B}) = \frac{p(\textcolor{blue}{A}) \cdot p(\textcolor{red}{B}|\textcolor{blue}{A})}{p(\textcolor{red}{B})}$$

The probability of **A** given **B** is equal to the probability of **A** multiplied by the probability of **B** given **A** divided by the probability of **B**.

Bayes Theorem: Penguin example

We want to find the probability of different **success rates** (our parameter of interest) based on our observations of **data**.

$$p(\textcolor{blue}{A} | \textcolor{red}{B}) = \frac{p(\textcolor{blue}{A}) \cdot p(\textcolor{red}{B} | \textcolor{blue}{A})}{p(\textcolor{red}{B})}$$

Bayes Theorem: Penguin example

We want to find the probability of different **success rates** (our parameter of interest) based on our observations of **data**.

$$p(\text{success rate} | \text{data}) = \frac{p(\text{success rate}) \cdot p(\text{data} | \text{success rate})}{p(\text{data})}$$

probability of success rate, or initial knowledge about the success rate (e.g., not 0 or 1)

probability distribution of success rate given our data

probability of our data

For a given success rate, we can calculate the probability of our data (e.g., if success rate were 0.5, what is the probability of observing 5 out of 18 successes?)

Questions?

Bayesian Data Analysis: The pieces of the model

Overview

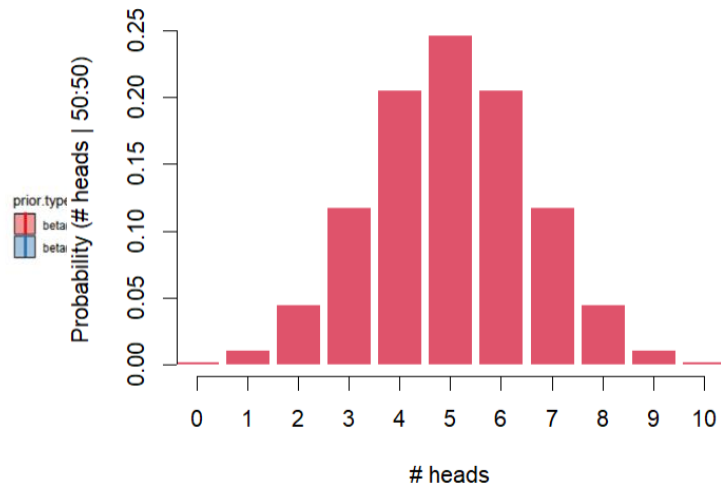
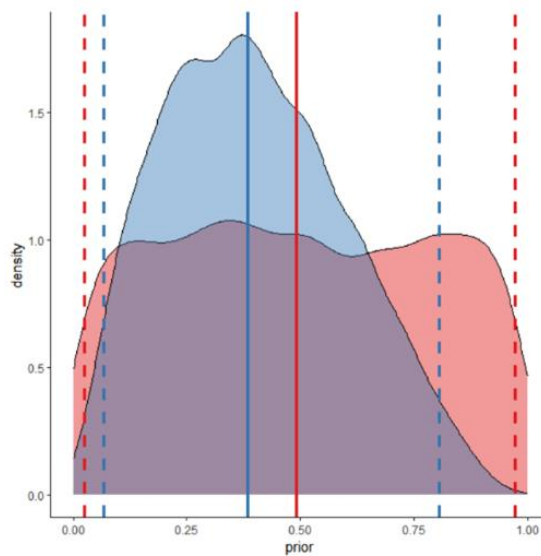
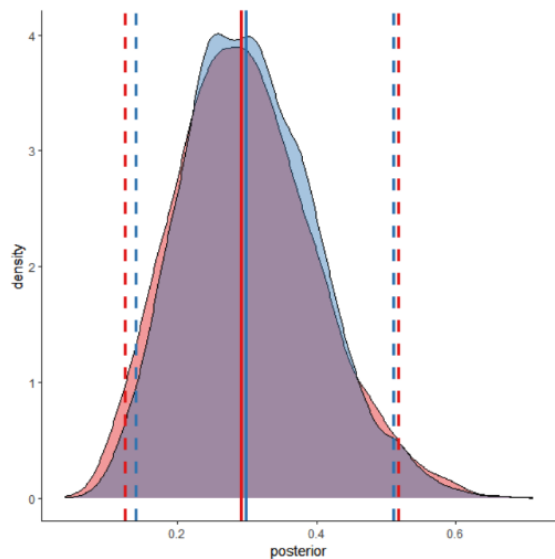
1. Welcome, Goals & Introductions
2. Bayes: An Updated Way of Thinking
- 3. Bayesian Data Analysis**
4. Interpreting Bayesian Models
5. Building Your Own Models
6. Tips/Tricks and Further Information
7. Workshop Wrap-up & Optional Happy Hour

Key Takeaways

$p(\text{model} \mid \text{data}) :$
posterior distribution

\propto $p(\text{model}) :$
prior distributions

■ $p(\text{data} \mid \text{model}) :$
model likelihood



Components of any Bayesian model:

Inputs



Data ('I cannot make bricks without clay': Sherlock Holmes, The Adventure of the Copper Beeches, 1892)

Model structure ('You know my methods, Watson': Sherlock Holmes, The Crooked Man, 1893)

- This should encapsulate how you think the **parameters** talk to each other, and the **data**
 - Deterministic part (output is entirely determined by **parameters** - i.e. intercept/slope)
 - Stochastic part (given an input the outputs are random - aka the data generating component)

Priors ('Education never ends, Watson. It is a series of lessons, with the greatest for the last': Sherlock Holmes, His Last Bow, 1917)

- Probability distributions of **model parameters** based on a-priori knowledge

Components of any Bayesian model:

Inputs & outputs

$$p(model|data) = \frac{p(model)p(data|model)}{p(data)}$$

$p(data)$: probability of the data

- is unknown, but importantly constant, so is usually omitted

$$p(model|data) \propto p(model)p(data|model)$$

 is proportional to

$p(model)$: **prior** distributions (think of model as the collective structure and parameters)

$p(data | model)$: **model likelihood** (probability of data given model structure and parameters)

$p(model | data)$: **posterior** distribution (our updated belief in model parameters given the data)

Components of any Bayesian model: Inputs & output for penguins

$$p(\text{success rate} | \text{data}) = \frac{p(\text{success rate}) \cdot p(\text{data} | \text{success rate})}{p(\text{data})}$$

$p(\text{data})$: probability of the data

- this is unknown, but importantly constant, so is usually omitted

$$p(\text{success rate} | \text{data}) \propto p(\text{success rate})p(\text{data} | \text{success rate})$$

$p(\text{success rate})$: **prior** distributions : $S \sim \text{Beta}(2,3)$

$p(\text{data} | \text{model})$: **model likelihood** : $\text{data} \sim \text{Binomial}(n = 18, p = S)$

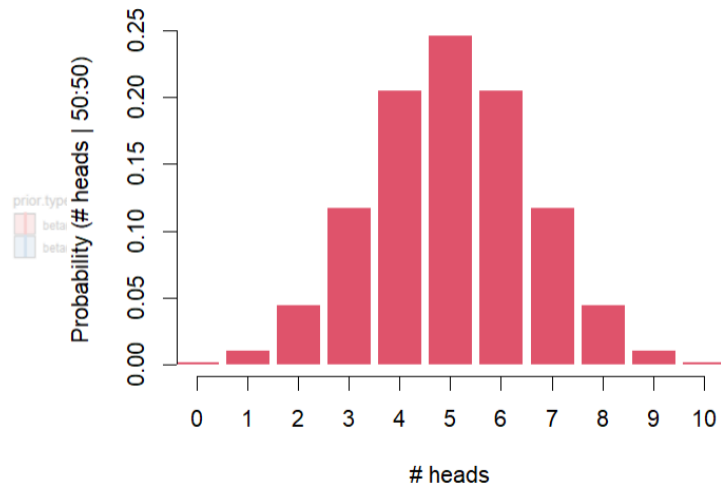
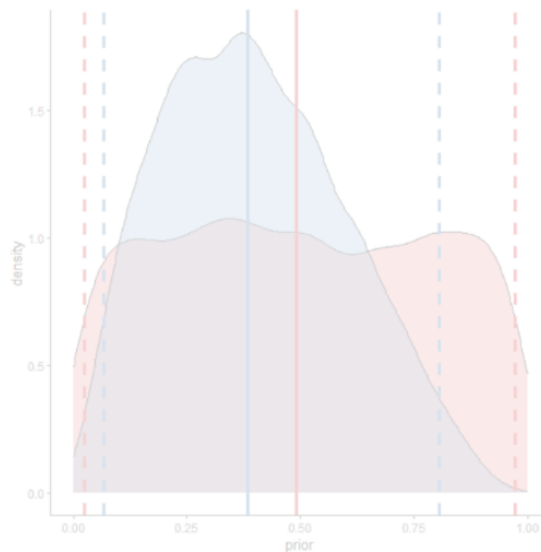
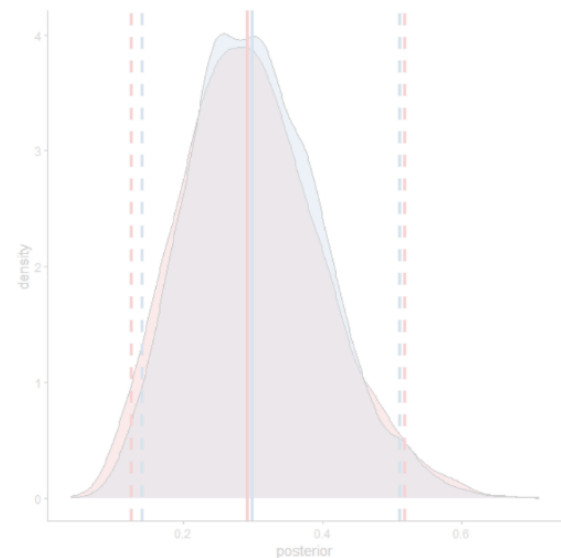
$p(\text{success rate} | \text{data})$: **posterior** distribution : $p(S | \text{data})$

Key Takeaways

$p(\text{model} \mid \text{data}) :$
posterior distribution

\propto $p(\text{model}) :$
prior distributions

■ $p(\text{data} \mid \text{model}) :$
model likelihood

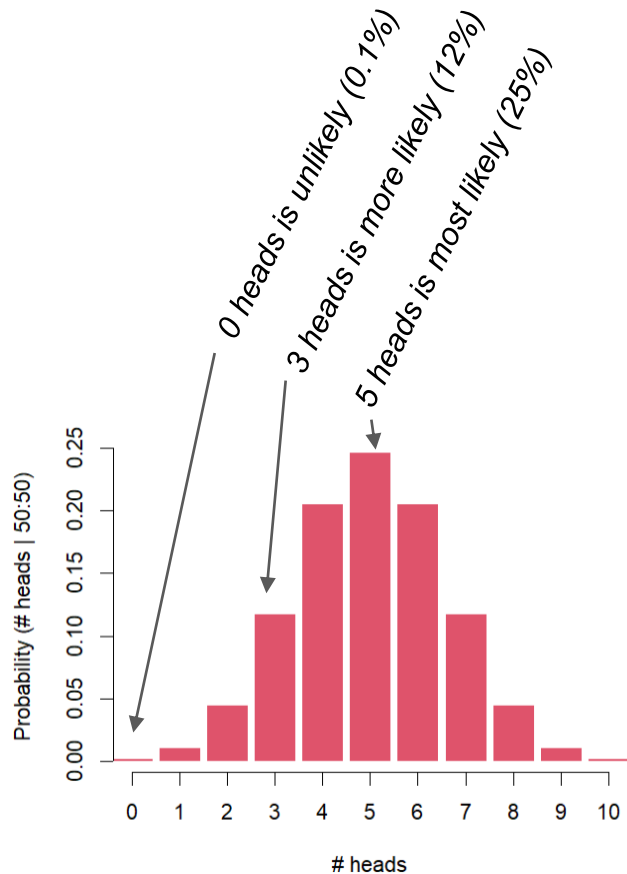


What is the likelihood?

The likelihood is the probability of the **data** given a particular **model**

Think of tossing a fair coin

- By assuming the coin is fair we are **fixing** the probability of heads at 0.5 - this is the **parameter** or **model**
- We toss the coin ten times and count the number of heads
- The number of heads we get is our **data**



Likelihood and likelihood function

Likelihood function

- How probability is shared across all possible observations

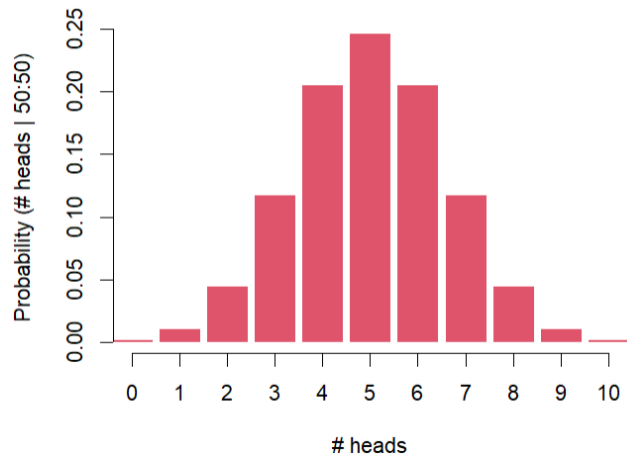
Quantifies how much a data point agrees with the fixed model

- E.g. 5 heads agrees with a $p = 0.5$ model more than an observation of 1 heads
- Used to judge among competing parameter values

Model can take any form, so long as it has parameters that relate to the data via the likelihood function

Likelihood function : binomial

(One of a set of probability distributions)



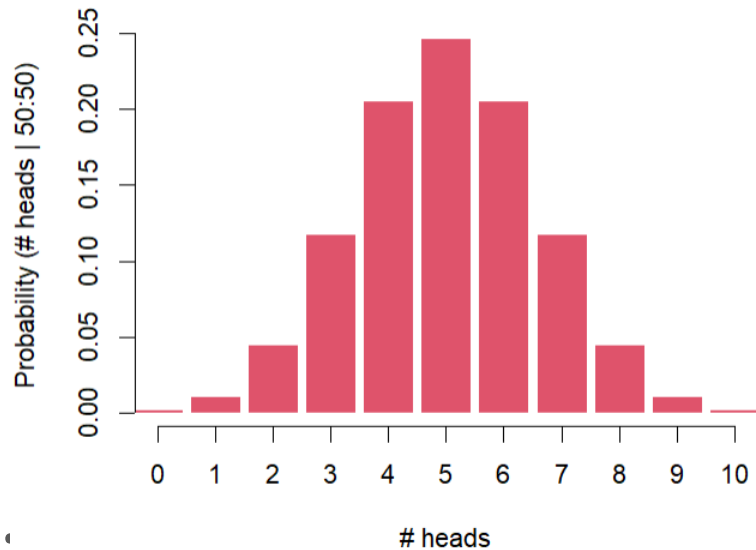
Closing likelihood statements

Likelihood = $p(\text{data} \mid \text{model})$

Likelihood is simulated data as defined by the model

Likelihood can be calculated using:

- Trial and error (NOT efficient!)
- Compute the maximum of a function by hand (rarely doable in practice)
- **Iterative optimization algorithm.**



How do we estimate $p(\text{model} \mid \text{data})$?

We have defined **data**, **priors**, model structure, how do we estimate posterior parameter estimates

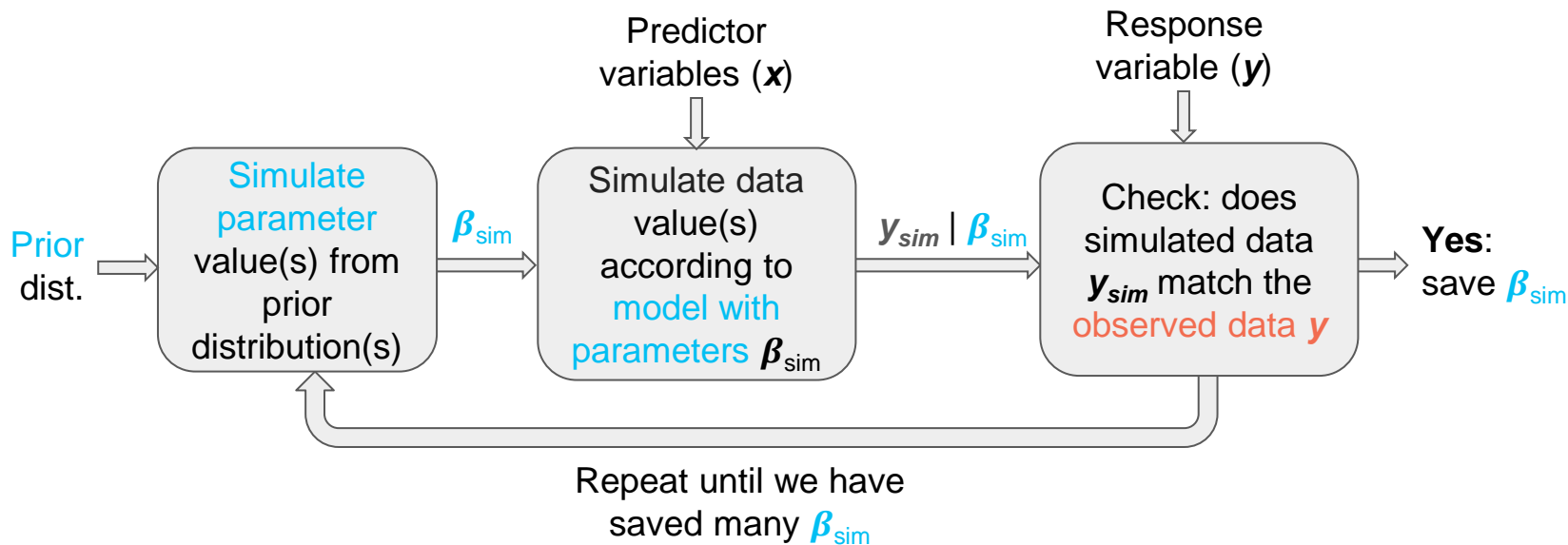
I am Bayesian Estimating
Machine (B.E.M.)

I am [insert prior] to
meet you



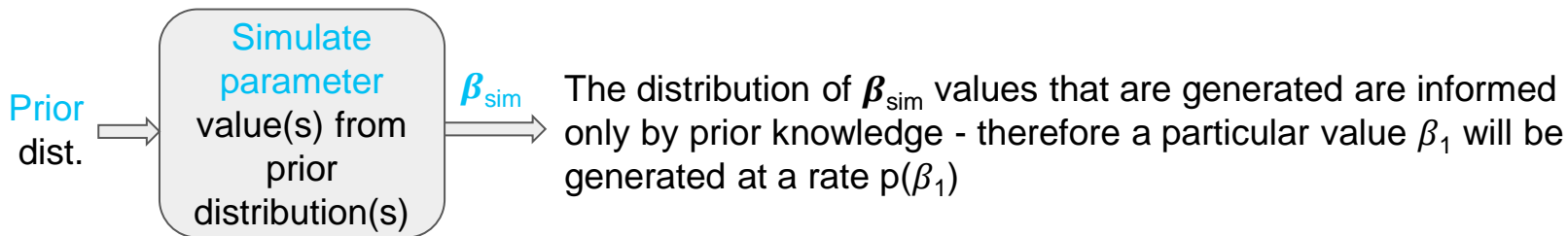
B.E.M.

B.E.M is a silly, but illustrative, and very hard working machine for estimating posterior distributions



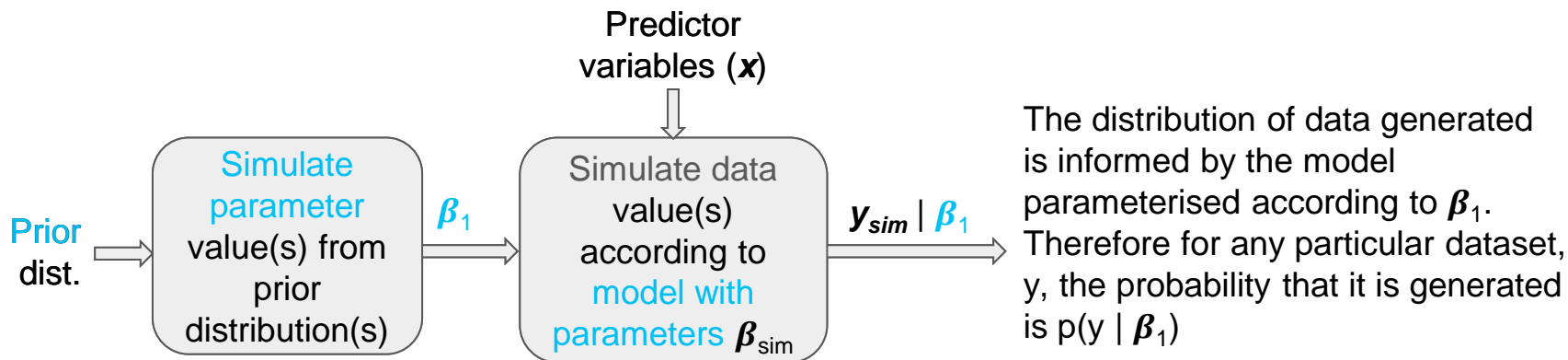
B.E.M.

B.E.M is a silly, but illustrative, and very hard working machine for estimating posterior distributions



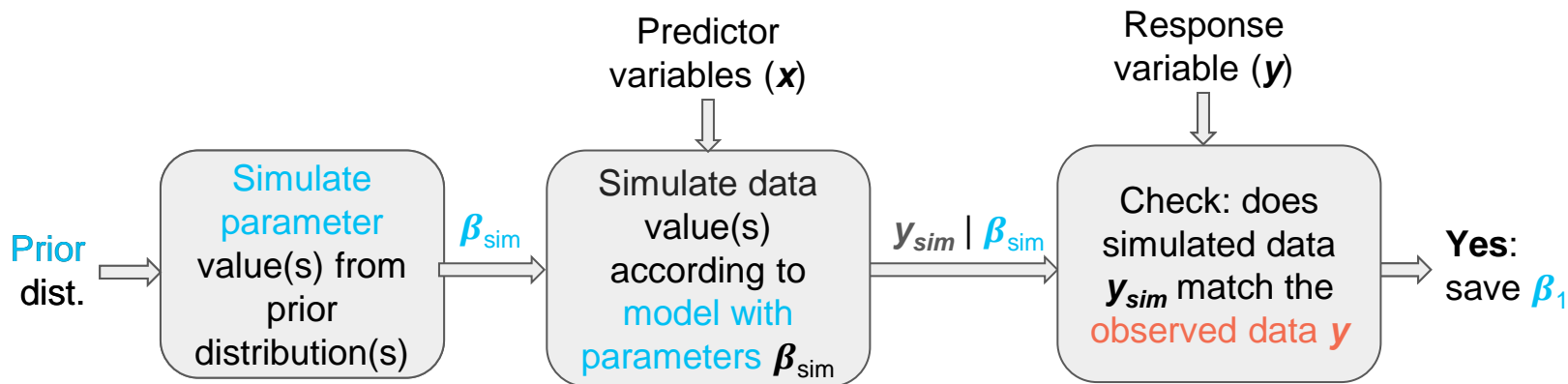
B.E.M.

B.E.M is a silly, but illustrative, and very hard working machine for estimating posterior distributions



B.E.M.

B.E.M is a silly, but illustrative, and very hard working machine for estimating posterior distributions

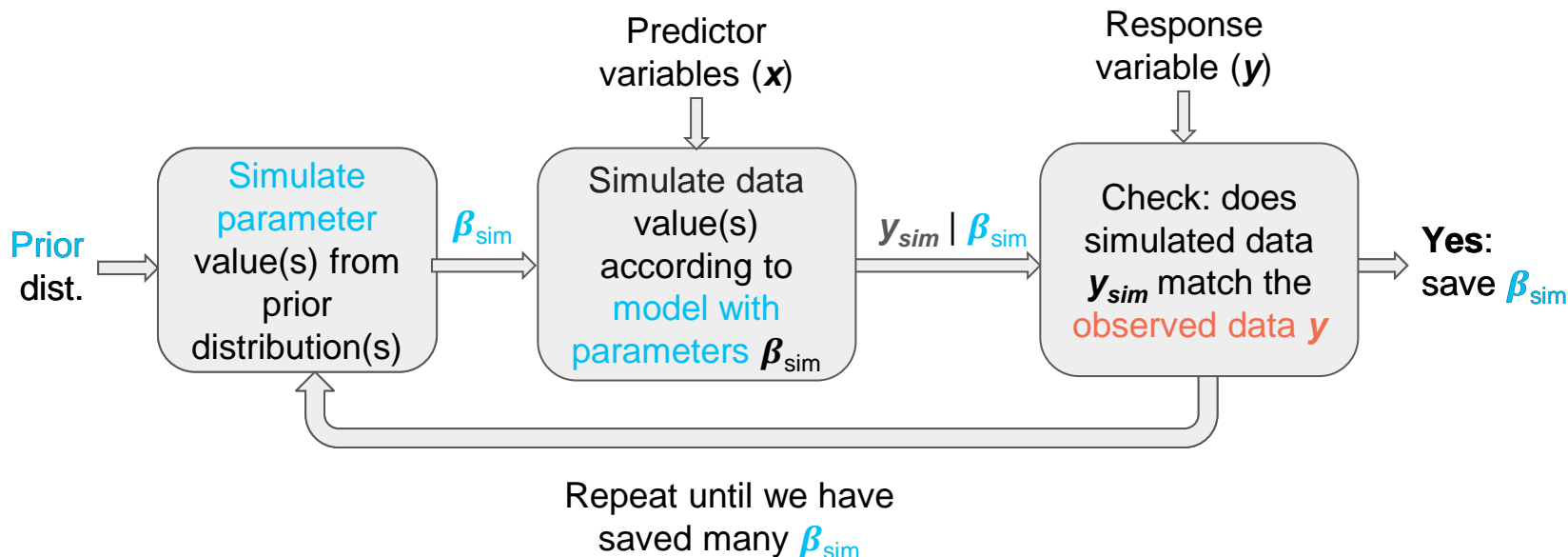


The number of times β_1 is saved is proportional to the rate it is generated $p(\beta_1)$ multiplied by the rate it generates the data $p(y | \beta_1)$

$$p(\beta_1 | y) \propto p(\beta_1) p(y | \beta_1)$$

B.E.M.

B.E.M is a silly, but illustrative, and very hard working machine for estimating posterior distributions



Applying B.E.M. to the penguin data

For the penguins we need to define a model for success rate

- We observed n attempts, of which y attempts were successful

$$y \sim \text{Binomial}(n, S)$$

- Where S is the success rate parameter we wish to estimate
- This defines our likelihood as $p(\text{data} \mid \text{model}) = \text{binomial}(y \mid n, S)$

Prior

- We were initially somewhat uninformed, but we know S must be between 0 and 1, and potentially skewed towards lower values

Bayesian Data Analysis:

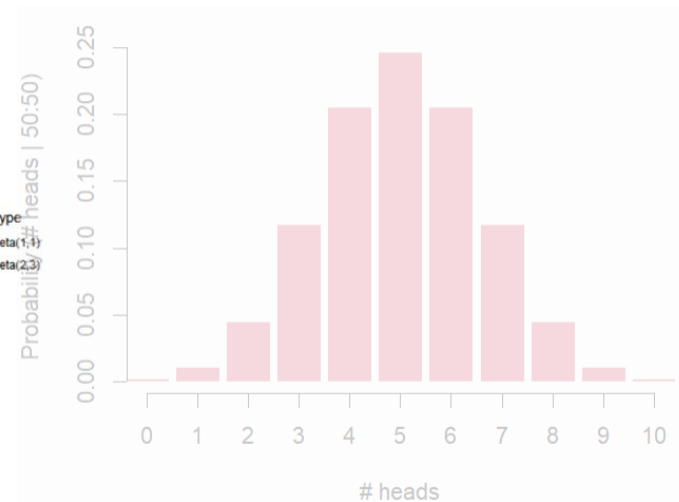
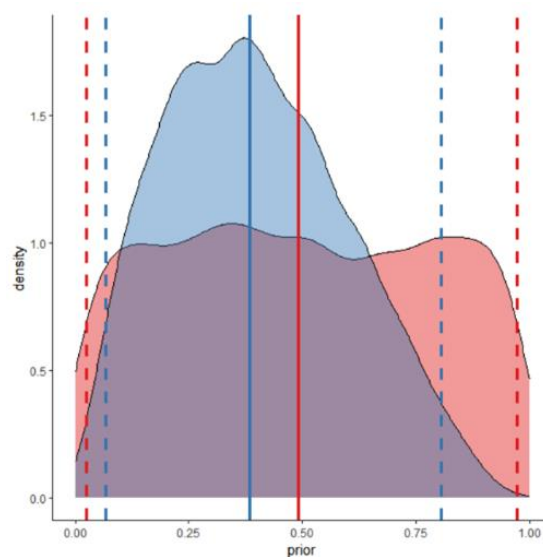
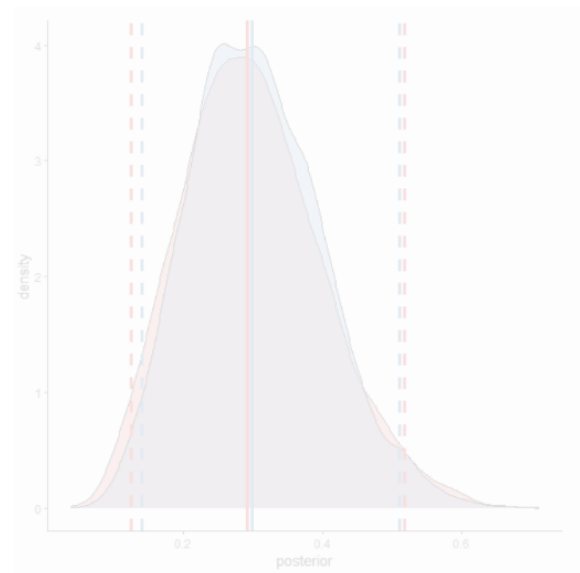
Defining priors

Key Takeaways

$p(\text{model} \mid \text{data}) :$
posterior distribution

$\propto p(\text{model}) :$
prior distributions

■ $p(\text{data} \mid \text{model}) :$
model likelihood



Defining priors

Priors can, with sparse data, enable inference on key parameters that might otherwise be impossible (cool!)

Priors are defined based on a probability distribution (i.e. normal...)

- Distribution parameters should reflect prior knowledge (i.e. distribution mean) and associated certainty (i.e. distribution width, shape)
 - Note that the way the model interprets your prior may not match your intention

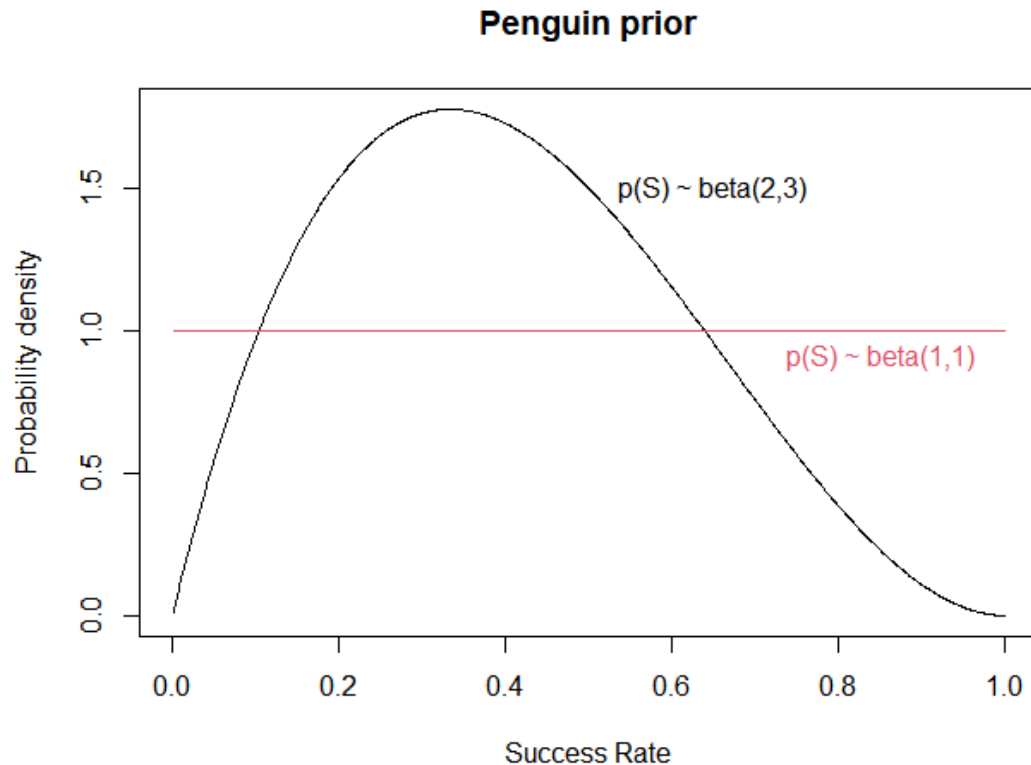
Priors could be informed by results from another study

- If not, we often know a realistic range (i.e. demographic rates)

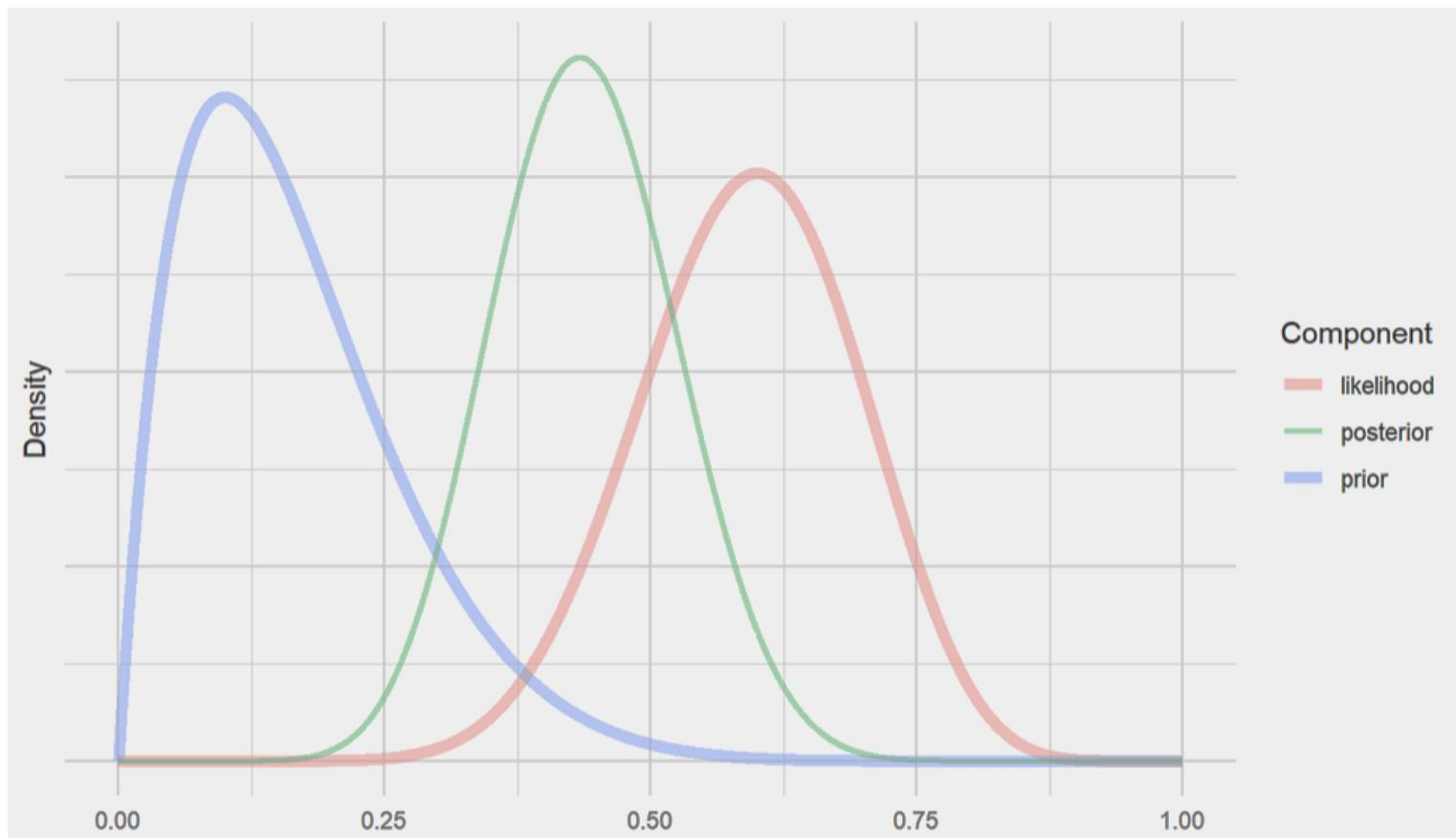
Priors are less influential and quickly overwhelmed by **informative data**

- If your main results depend on priors, then there either needs to be solid justification for them and/or suitable sensitivity analyses to show results are invariant to prior definition

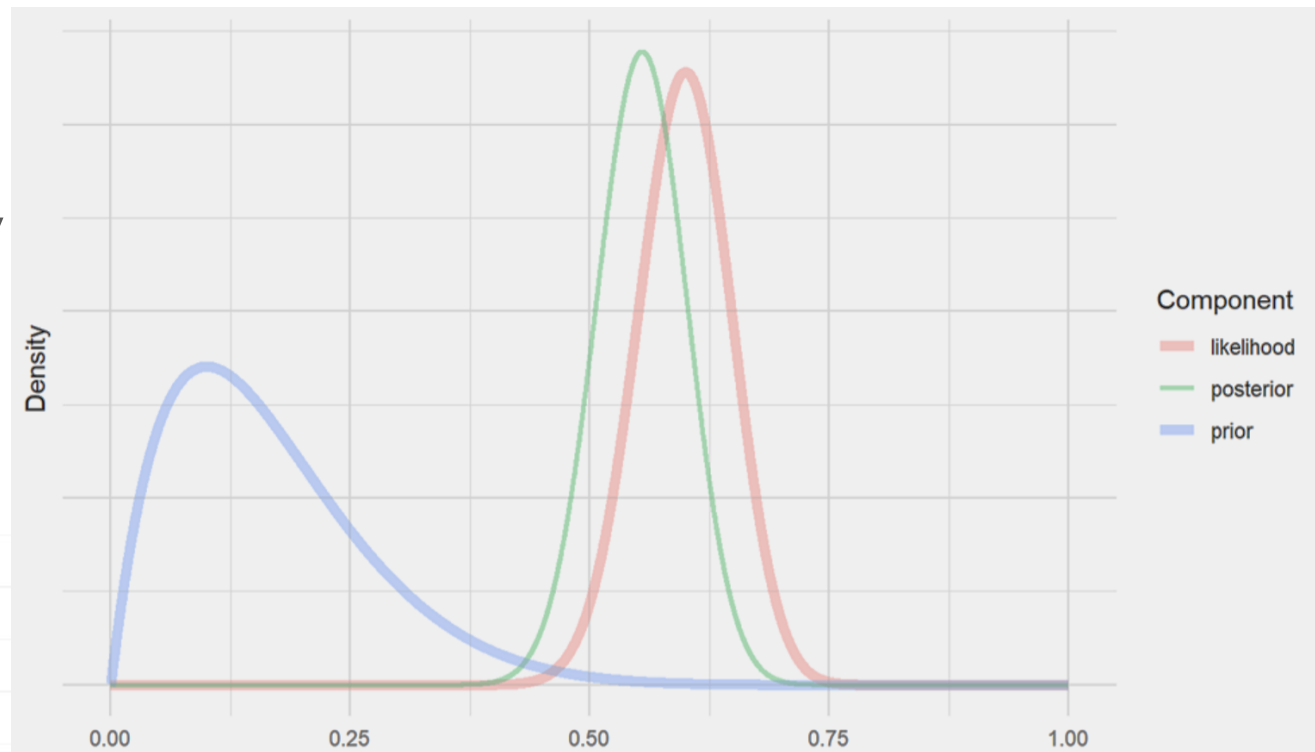
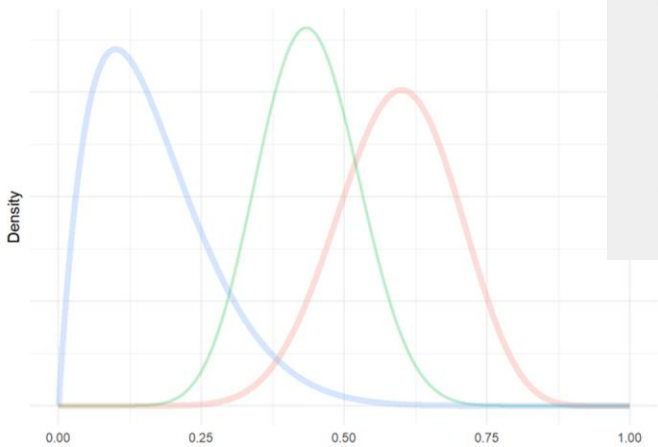
Priors for penguin success rate



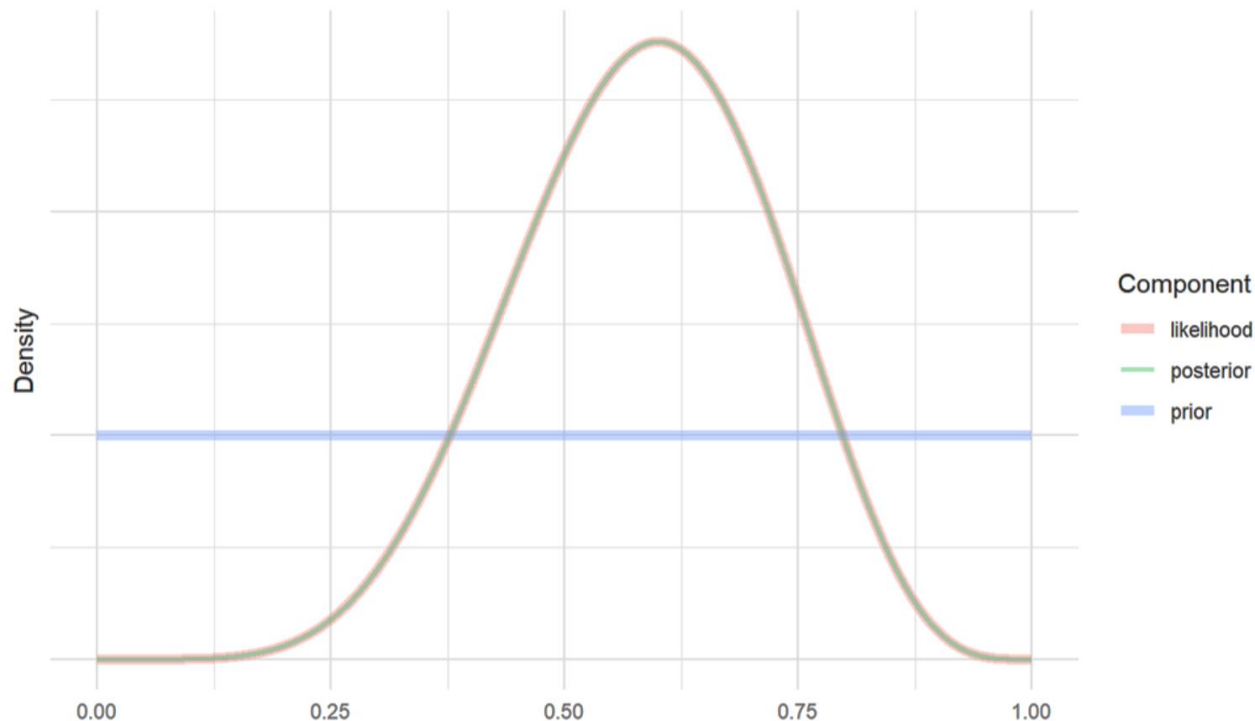
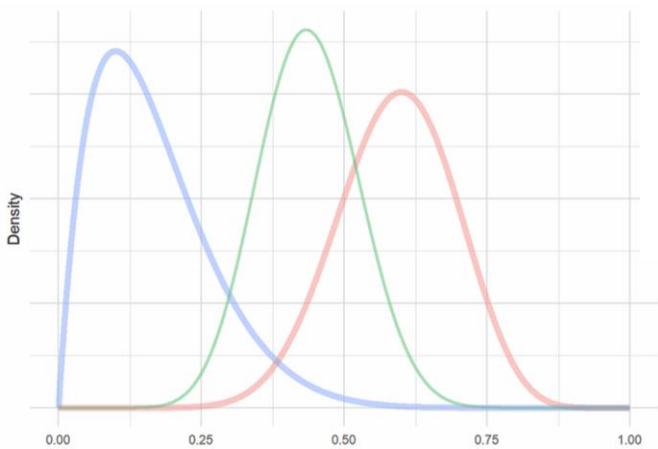
Priors, likelihood & posterior interactions



Priors are **less**
influential and
quickly
overwhelmed by
informative
data



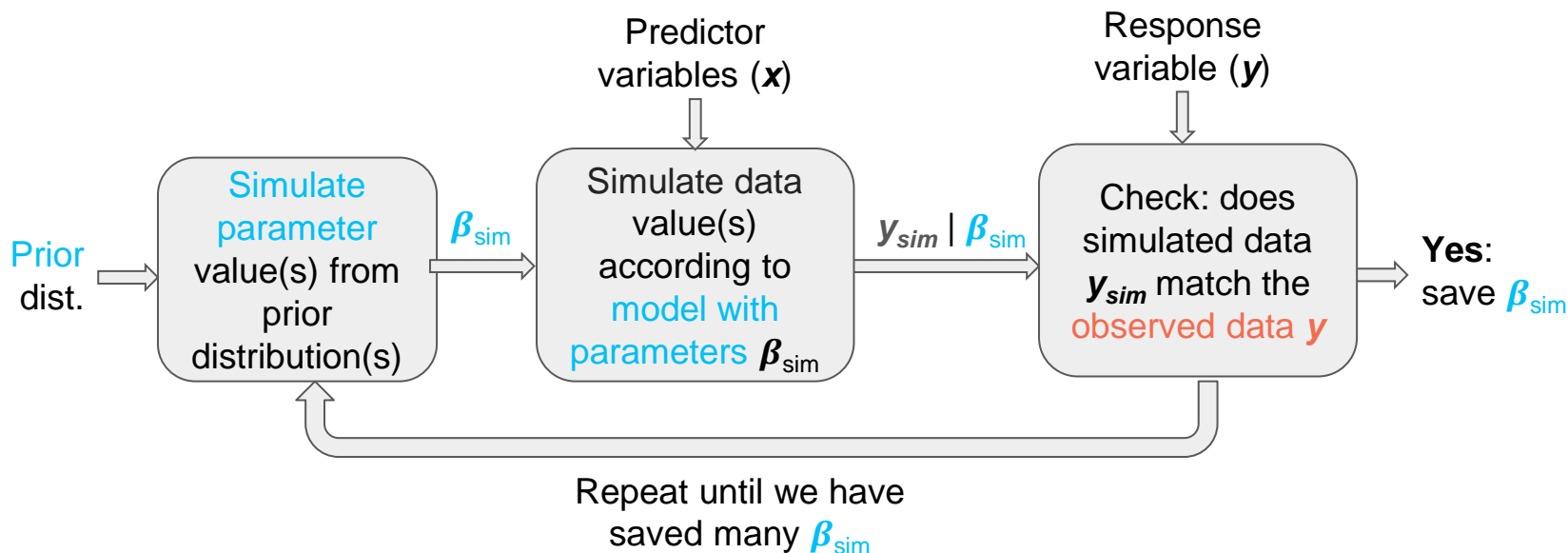
Likelihood has
even more
influence with
uninformative
prior



Bayesian Data Analysis: Model fitting/MCMC

B.E.M.

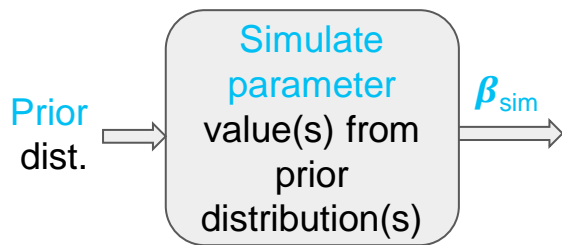
B.E.M is a silly, but illustrative, and very hard working machine for estimating posterior distributions



Applying B.E.M. to the penguin data

Simulated **Success rates** (according to prior $p(\mathbf{S}) \sim \text{beta}(2,3)$)

0.08 0.11 0.17 0.18 0.19 0.29 0.31 0.32 0.33 0.43 0.46 0.46 0.51 0.52 0.53 0.55 0.64 0.67 0.68 0.84



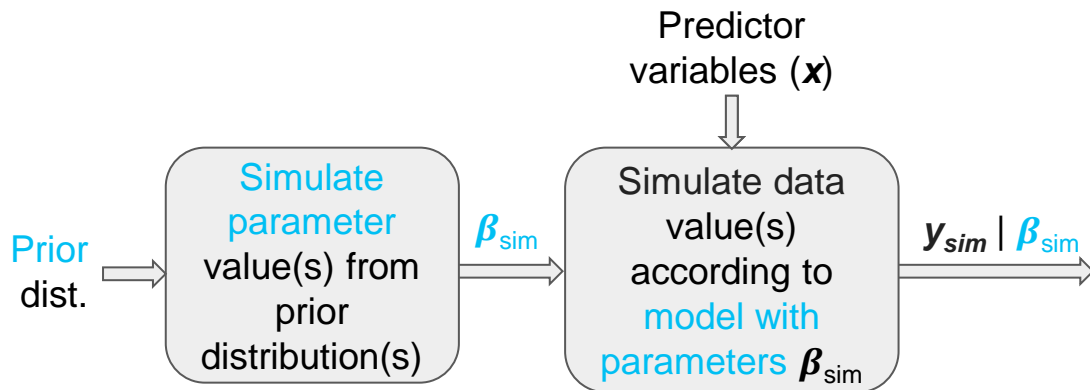
Applying B.E.M. to the penguin data

Simulated Success rates (according to prior $p(S) \sim \text{beta}(2,3)$)

0.08 0.11 0.17 0.18 0.19 0.29 0.31 0.32 0.33 0.43 0.46 0.46 0.51 0.52 0.53 0.55 0.64 0.67 0.68 0.84

Simulated data (according to data $\sim \text{binomial}(n = 18, p = S)$)

5 3 3 3 5 7 5 2 5 11 6 7 9 12 10 8 12 10 10 17



Applying B.E.M. to the penguin data

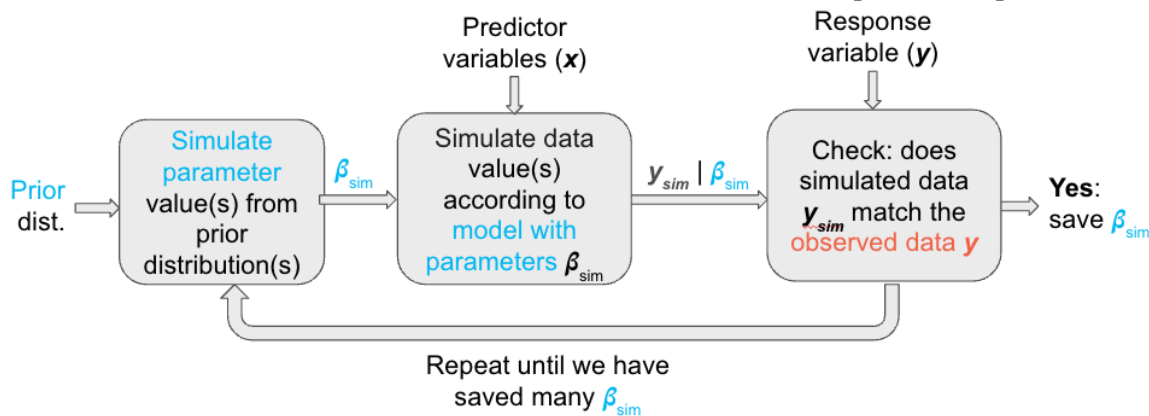
Simulated **Success rates** (according to prior $p(\mathbf{S}) \sim \text{beta}(2,3)$)

0.08 0.11 0.17 0.18 **0.19** 0.29 **0.31** 0.32 **0.33** 0.43 0.46 0.46 0.51 0.52 0.53 0.55 0.64 0.67 0.68 0.84

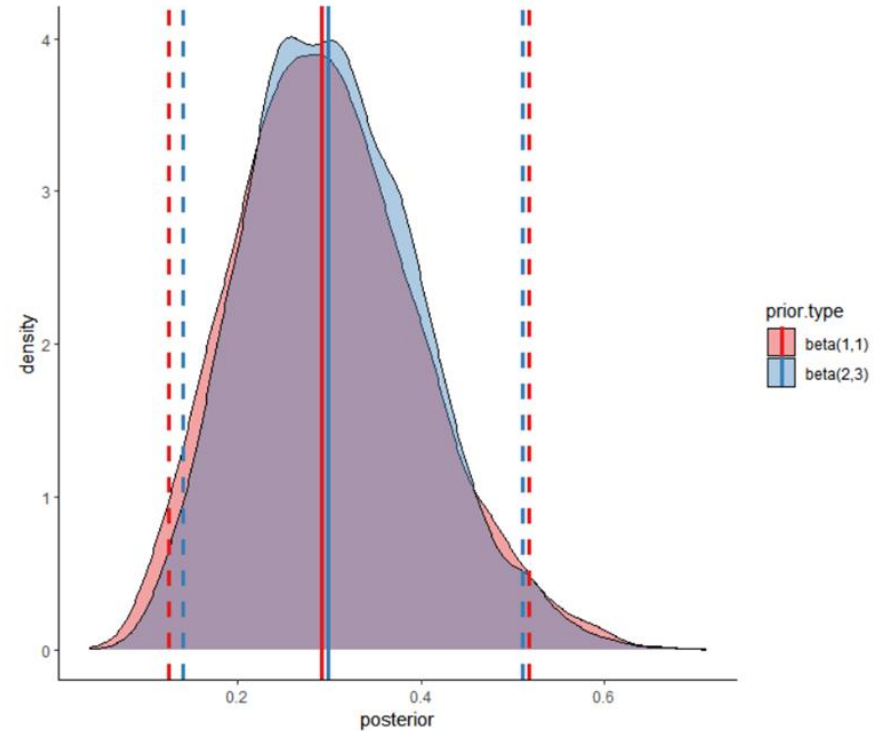
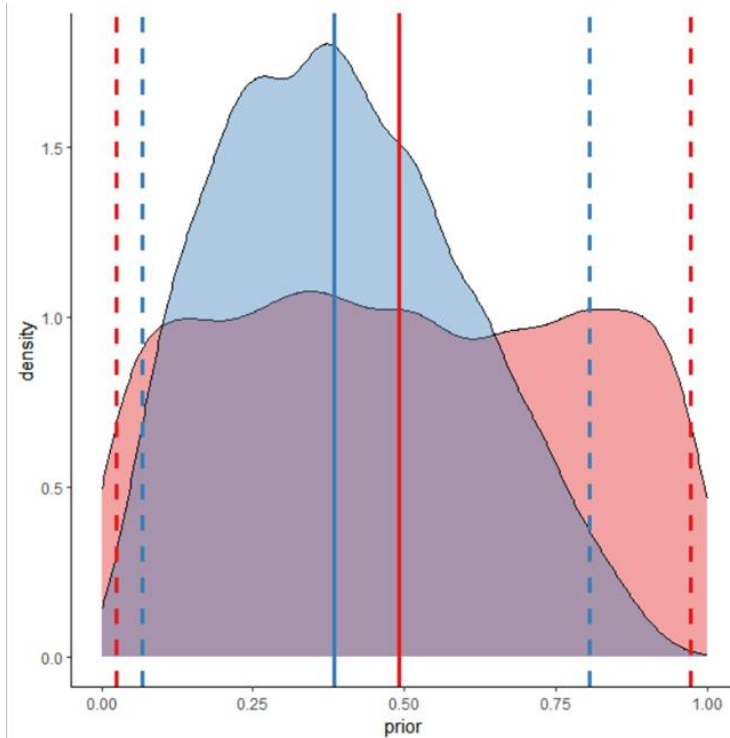
Simulated data (according to **data** $\sim \text{binomial}(n = 18, p = \mathbf{S})$)

5 3 3 3 **5** 7 **5** 2 **5** 11 6 7 9 12 10 8 12 10 10 17

Only keep the values for \mathbf{S} that are consistent with our **data**
The resulting distribution of values for \mathbf{S} make up the posterior $p(\mathbf{S} \mid \text{data})$



Applying B.E.M. to the penguin data



How do we estimate $p(\text{model} \mid \text{data})$?

Some may have noticed that B.E.M isn't particularly efficient

- E.g. Even with one data point it required 50,000-100,000 simulations to get 5,000 estimates of success rate

So, we require a method that is smarter about choosing the parameter values to explore

Markov Chain Monte Carlo (MCMC)

- Monte Carlo - simulation of many random numbers
- Markov Chain - an elegant way of choosing those numbers

B.E.M. now believes that
they are pleased to meet
you {based on 87% probability that
the pleased score is >8}

Go.od..b..y. . . .
[please recharge]



Markov Chain Monte Carlo

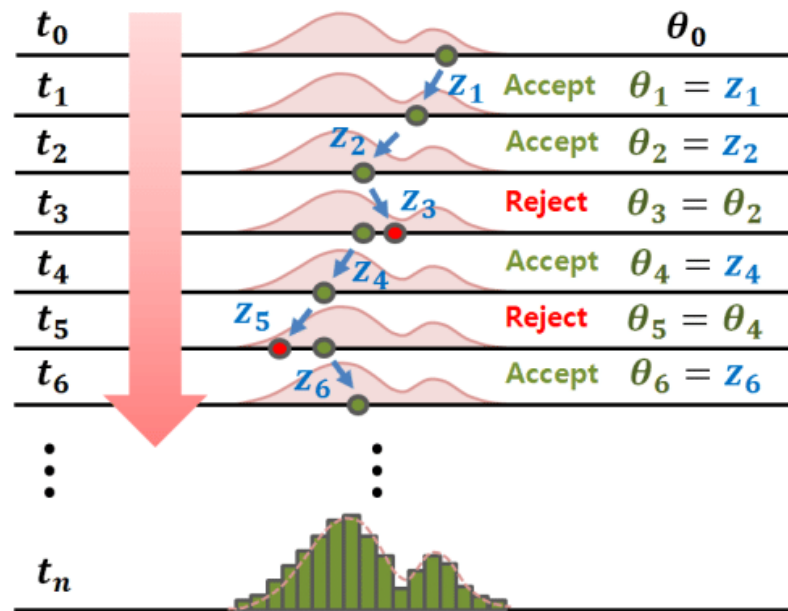
Iterative process of parameter exploration

Monte Carlo

- I randomly guess some potential next steps in my exploration

Markov Chain

- The step I choose to take next is governed by where I am currently, informed by the landscape around me



30 Second Intro to Markov Chain Monte Carlo

Propose a parameter value(s)

Evaluate likelihood at proposal

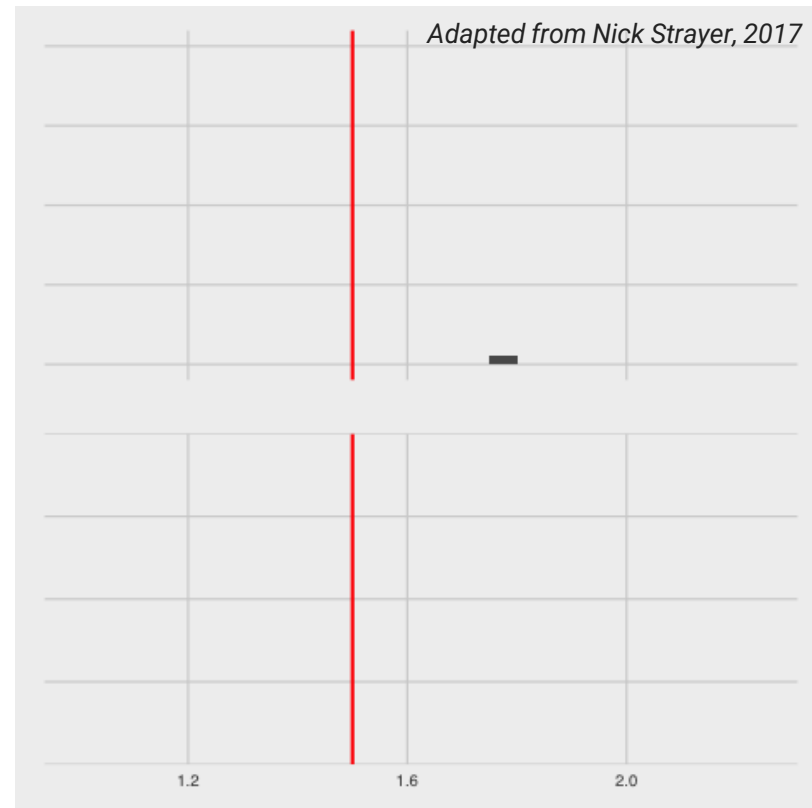
Propose an alternative parameter value(s)

Evaluate likelihood at alt proposal

Compare likelihoods

Save a parameter value

Repeat for a loooooooooooooooooong time



Highly recommend

<http://chi-feng.github.io/mcmc-demo/app.html?algorithm=H2MC&target=banana>

<https://livefreeordichotomize.com/posts/2017-10-14-mcmc-explainer/>

Pros/Cons of MCMC

Pros

- Once MCMC finds and settles into the target set the resulting distribution of values is the desired distribution of parameter estimates
- Can be used to estimate complex non-linear models

Cons

- Can take a long time to converge
- Can be finicky and sensitive to model parameterisation in some instances
- Requires a more hands-on approach for checking the model 'worked'

Bayesian model-fitting

Difference in model estimation

- Models fitted via maximum likelihood (frequentist) - find a **single value for each parameter**
- Bayesian models - find a **set of values for each parameter** consistent with the **data** and **prior** expectation
- Bayesian models are concerned with finding and exploring a **target set** (which makes up the posterior)

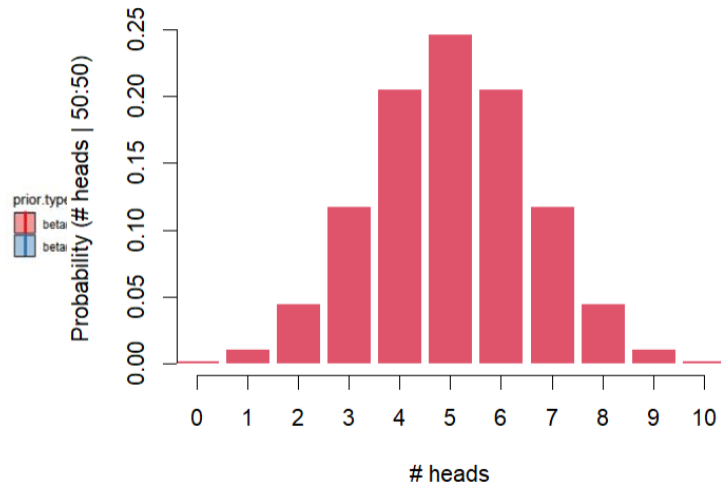
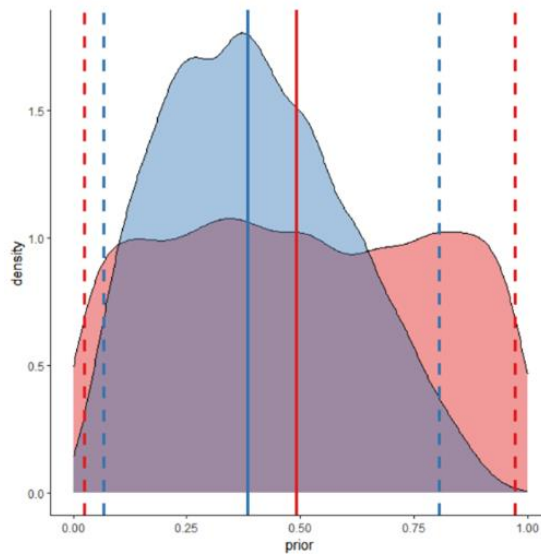
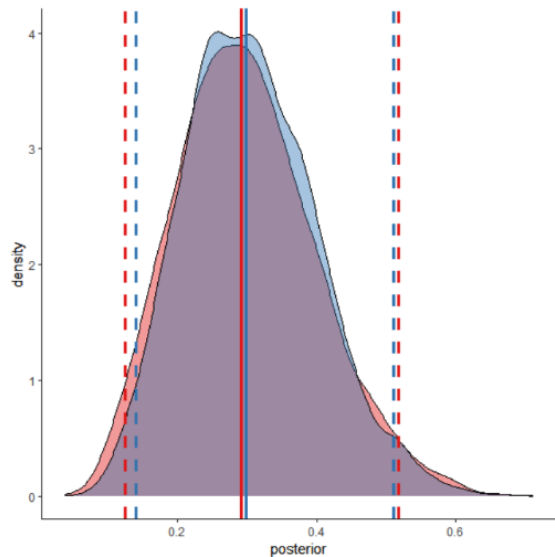
However, we can evaluate probabilities for a specific location and use that to guide exploration - enter MCMC

Key Takeaways

$p(\text{model} \mid \text{data}) :$
posterior distribution

\propto $p(\text{model}) :$
prior distributions

■ $p(\text{data} \mid \text{model}) :$
model likelihood



Questions?

Interpreting Bayesian Models

Overview

1. Welcome, Goals & Introductions
2. Bayes: An Updated Way of Thinking
3. Bayesian Data Analysis
- 4. Interpreting Bayesian Models**
5. Building Your Own Models
6. Tips/Tricks and Further Information
7. Workshop Wrap-up & Optional Happy Hour

Data analysis

Estimating things of interest

Quantifying uncertainty

Using models to make decisions/test hypotheses

Using models to make predictions

Model comparison

What you won't see

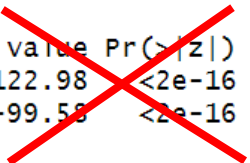
Test statistics and p-values

- These are grounded in null hypothesis (usually 0) significance testing

Model selection (AIC) or fit (R^2) criteria

- However, equivalent measures can be calculated

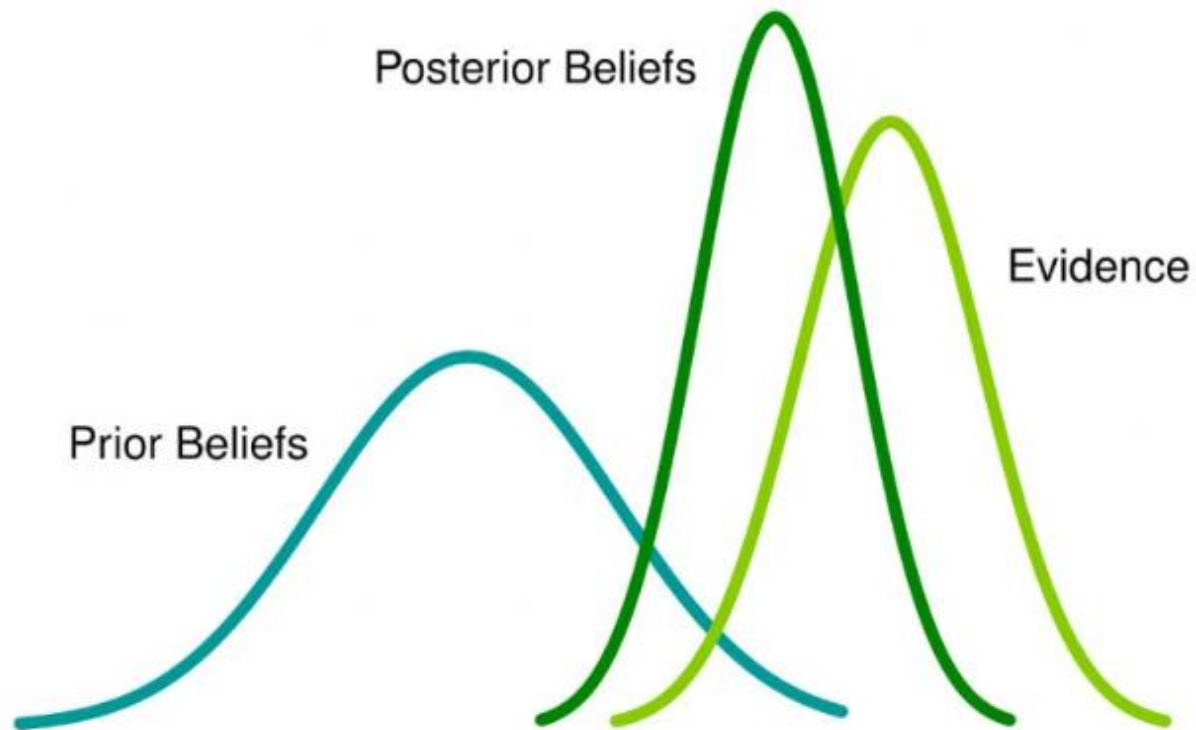
```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  48.6190414  0.3953319  122.98 <2e-16 ***
year        -0.0196054  0.0001969  -99.58 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



What do we use?

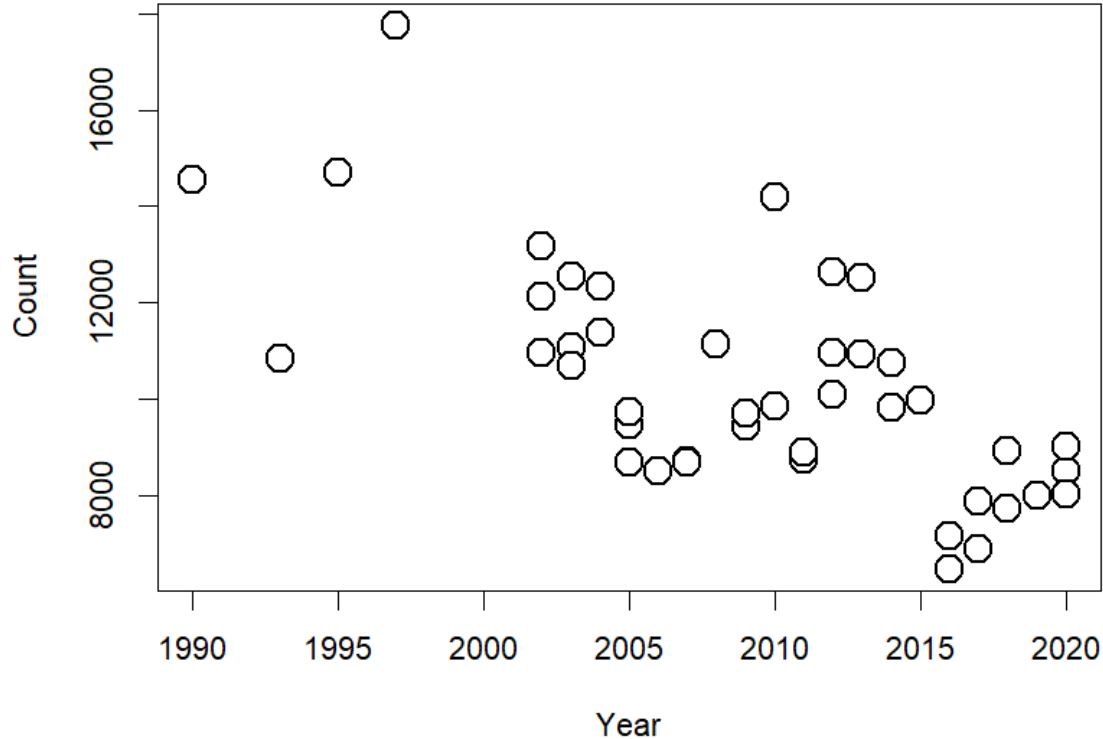
```
(Dispersion parameter for poisson family taken to be 1)
```

```
Null deviance: 21187  on 42  degrees of freedom
Residual deviance: 11484  on 41  degrees of freedom
AIC: 11963
```



Interpreting Bayesian Models ...with murrees!

Example: Estimating trend in murre abundance



Example: Estimating trend in murre abundance

Need a model for Counts $\sim f(\text{time})$

Bayesian version of a poisson GLM

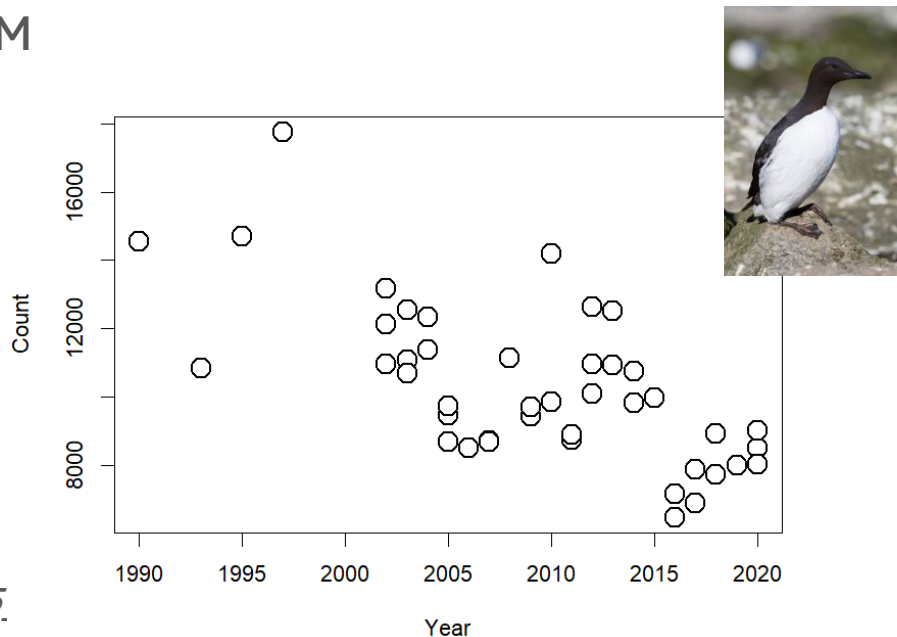
Functional part

$$\log(\lambda_t) = \beta_0 + \beta_1 t$$

Observation part

$$C_t \sim \text{poisson}(\lambda_t)$$

[For argument's sake, we'll set $t = 0$ @ 2005.]



Example: Estimating trend in murre abundance

Need a model for Counts $\sim f(\text{time})$

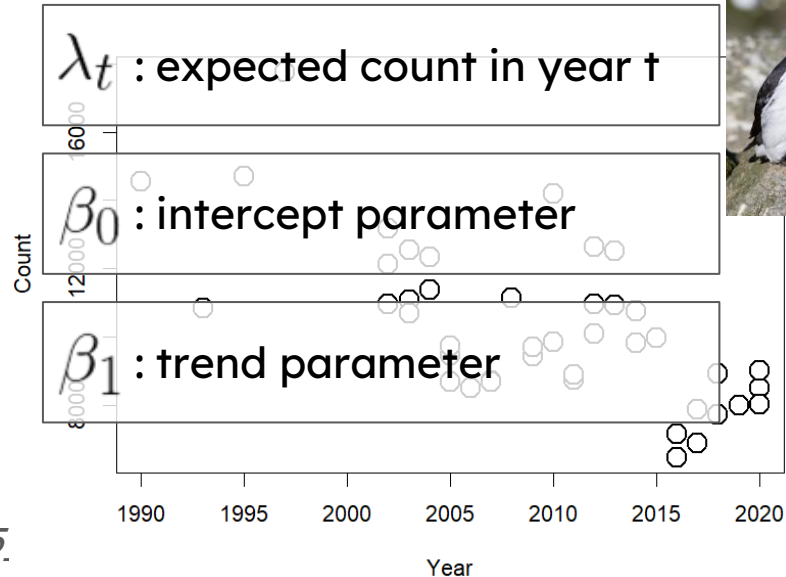
Bayesian version of a poisson GLM

Functional part

$$\log(\lambda_t) = \beta_0 + \beta_1 t$$

Observation part

$$C_t \sim \text{poisson}(\lambda_t)$$



[For argument's sake, we'll set $t = 0$ @ 2005.]

Example: Estimating trend in murre abundance

Need a model for Counts $\sim f(\text{time})$

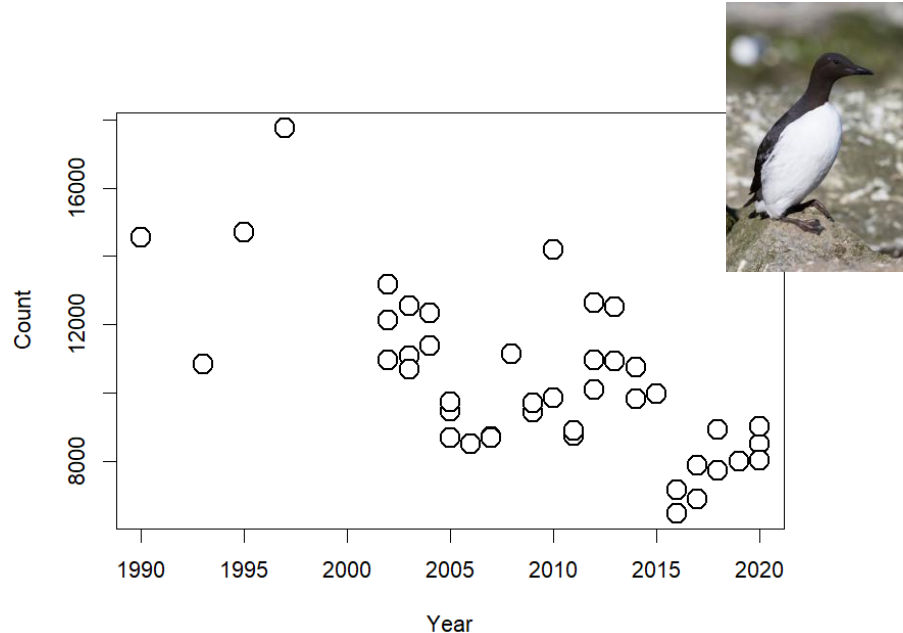
Model has two parameters

β_0 - Intercept (on log scale)

β_1 - Trend (on log scale)

Need to define priors for both

- What are our initial beliefs



Priors

Imagine you hadn't seen the data, but know the context

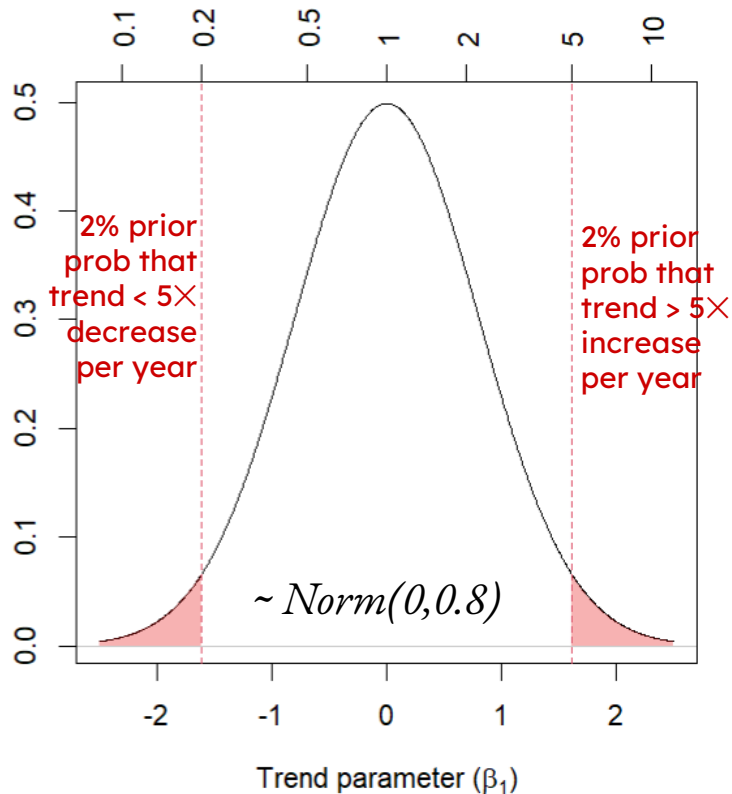
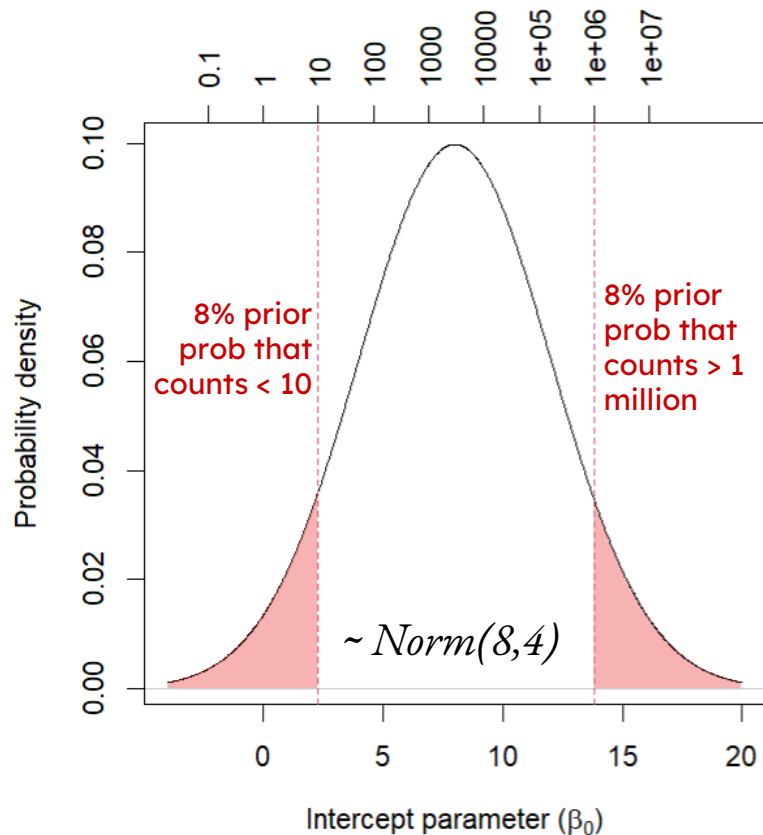
- Murre colony monitoring
- Estimation of trend through time

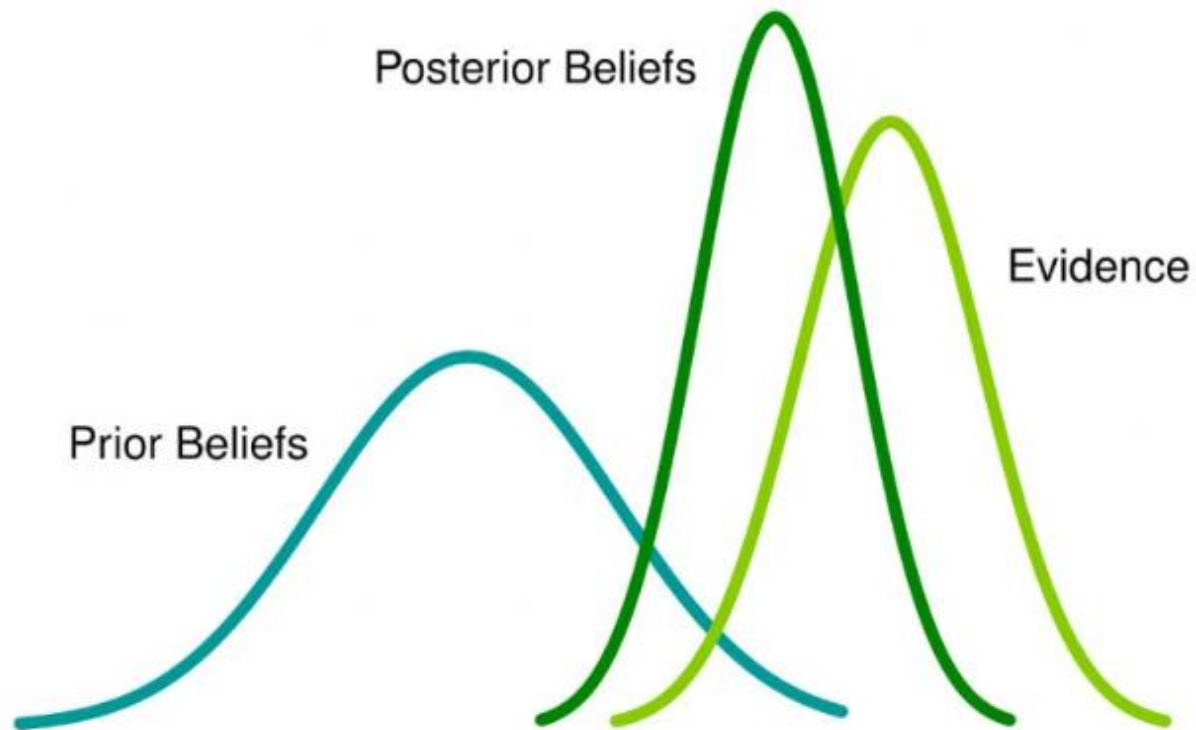
Our priors should encode some information we have initially

- Murre colonies: range in size from 10's to 100,000's of individuals
 - Our initial belief in the intercept should reflect this
- Trends: $\exp(\beta_1)$ represents per annum (relative) growth rate
 - A consistent and sustained 5-fold per annum change in either direction seems unlikely

Within a Bayesian model we convert these to a probability distribution

Priors

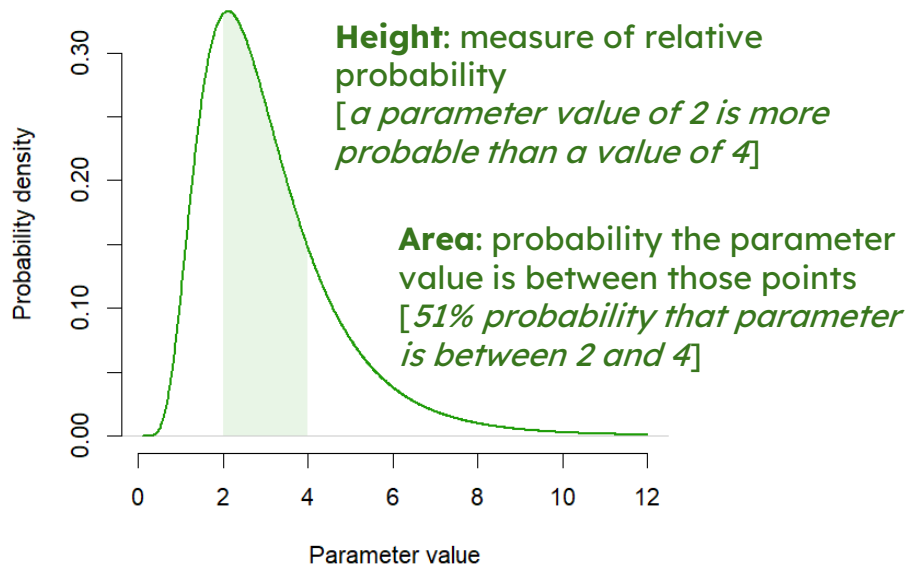




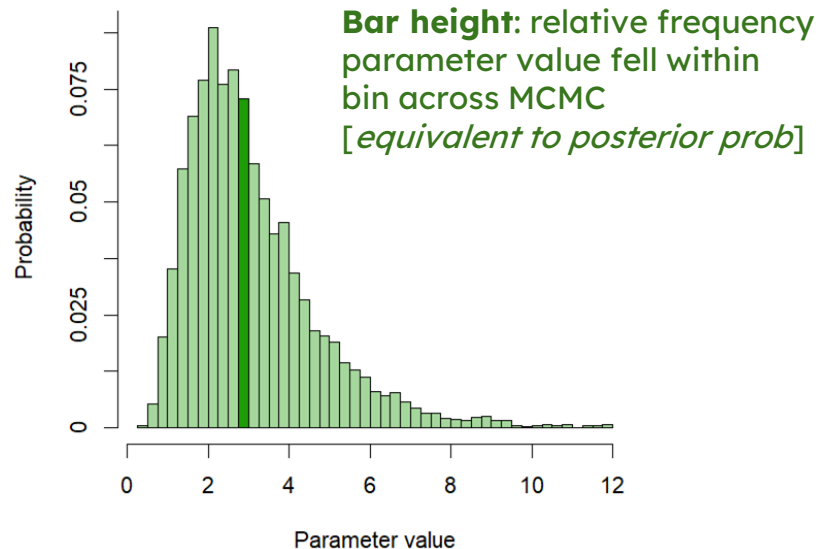
What is a posterior distribution?

Posterior distribution - probability distribution representing our updated (prior + data) belief in a parameters value for a particular model

What we are trying to estimate



What we get from MCMC



Data analysis

Estimating things of interest

Quantifying uncertainty

Using models to make decisions/test hypotheses

Using models to make predictions

Model comparison

Posterior distribution

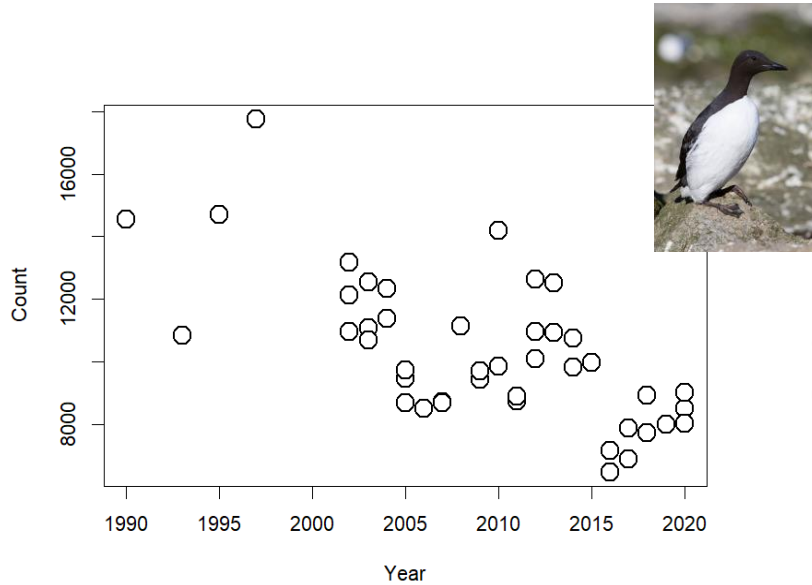
When interpreting a parameter estimated via a Bayesian model we use the posterior distribution

Important to note that the Bayesian interpretation of a parameter **IS the posterior distribution** - the probability distribution among different values

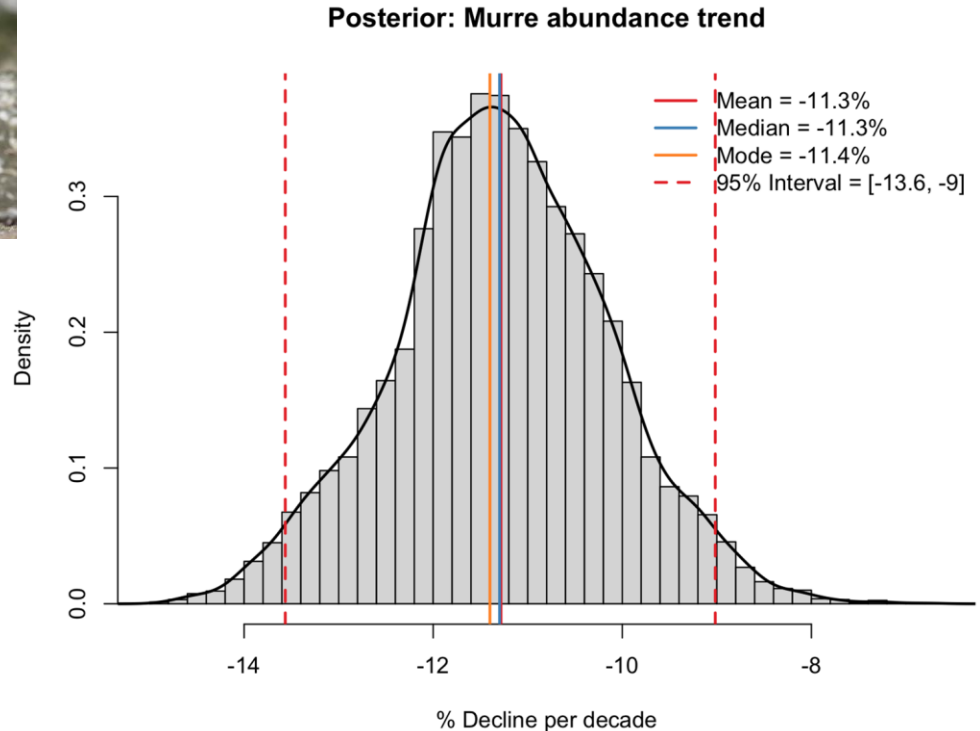
- Reduction to single point-value and/or interval will lose information
- But often it is a necessary step

Should always evaluate the posterior distribution shape as part of analyses

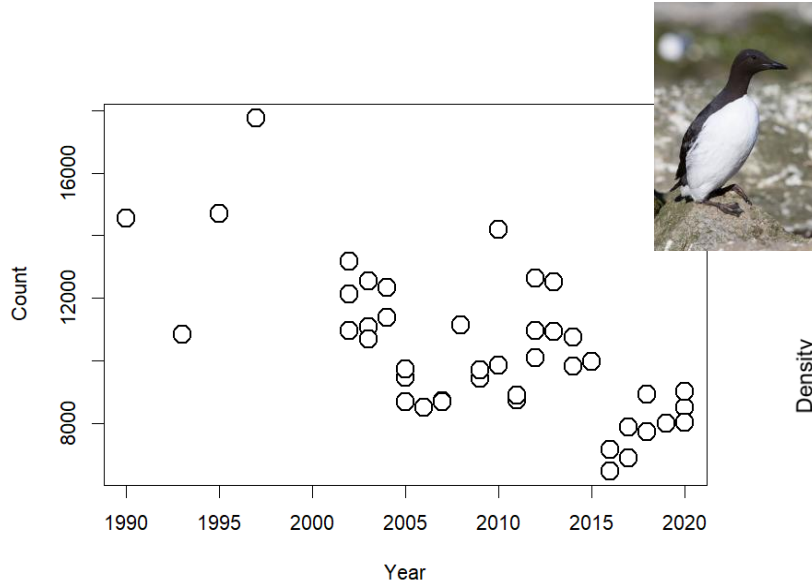
Example: Estimating trend in murre abundance



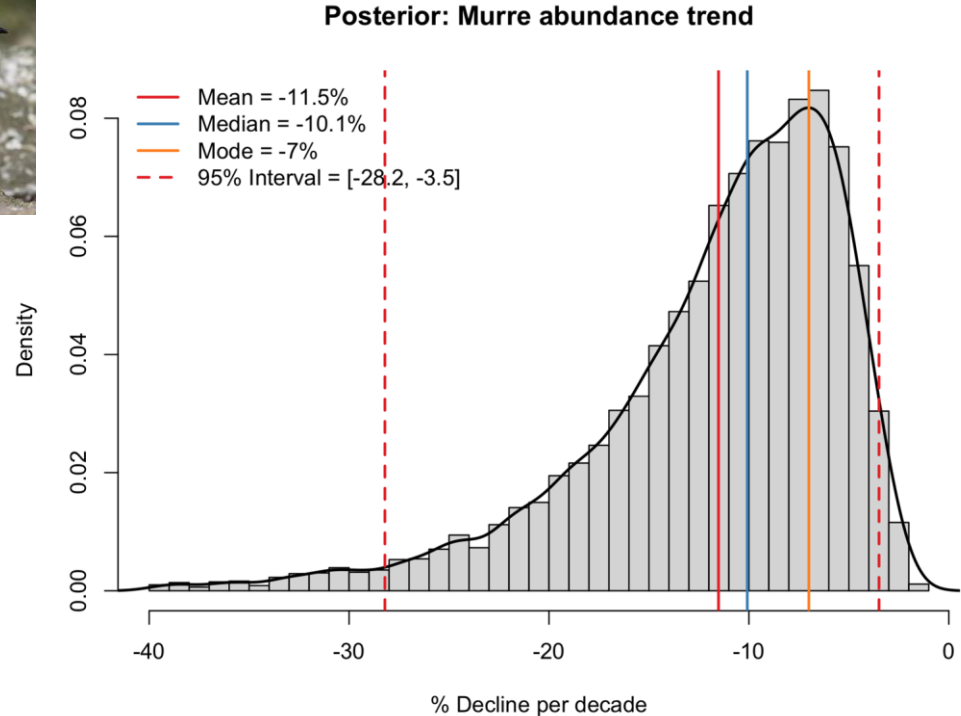
Point estimates: mean, median, mode



Example: Estimating trend in murre abundance



Point estimates: mean, median, mode



Data analysis

Estimating things of interest

Quantifying uncertainty

Using models to make decisions/test hypotheses

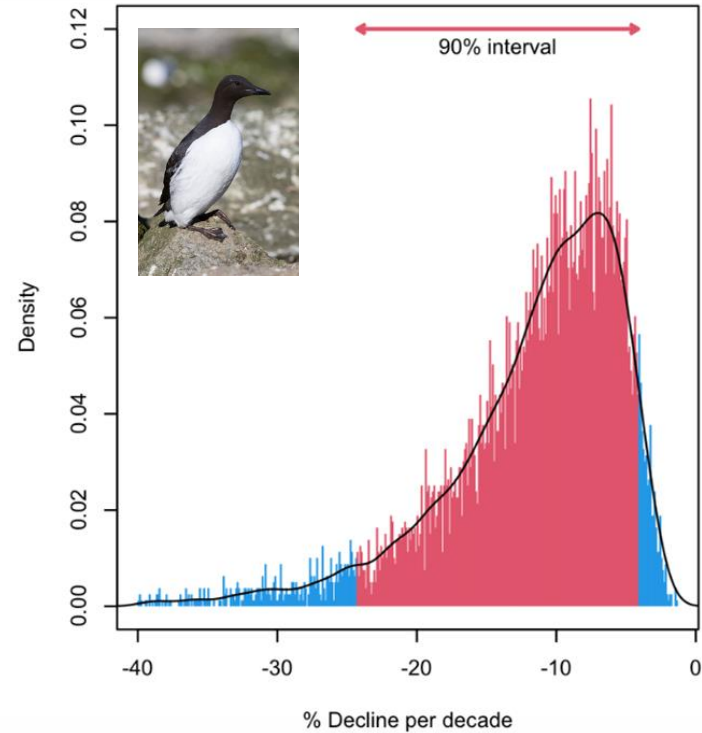
Using models to make predictions

Model comparison

Bayesian credible intervals

Find the range of parameter values corresponding to certain cumulative probability

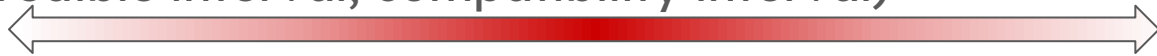
- 90% probability that trend is between 4% and 24% decline per decade
- 50% probability that trend is between 6% and 15% decline per decade etc



Confidence intervals and Bayesian intervals

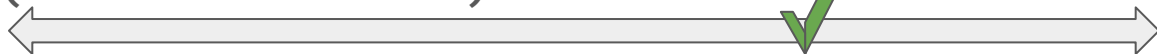
Suppose a 95% interval

Bayesian (credible interval, compatibility interval)



Interpretation: there is 95% probability that the parameter lies in this range (given the model and data)

Frequentist (confidence interval)



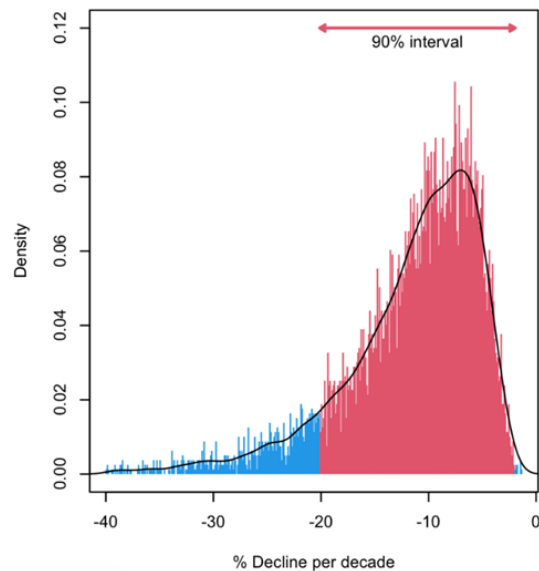
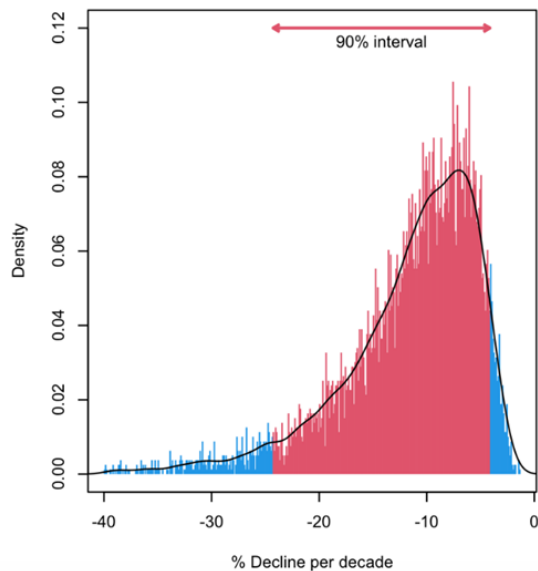
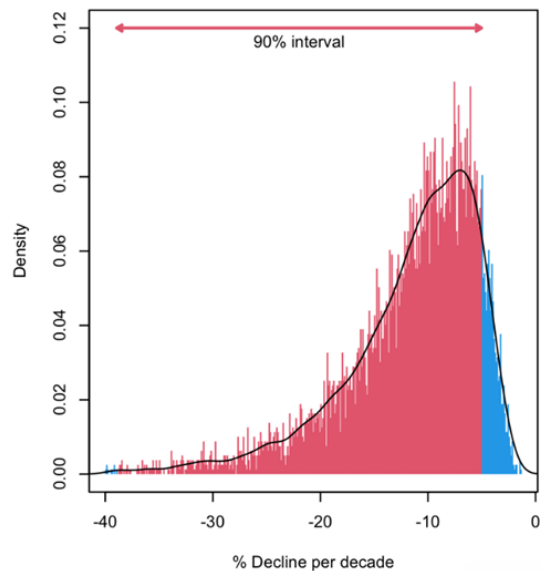
Interpretation: in 95% of data collection repetitions the interval calculated from the data overlaps the 'true' value.

95% confidence does not mean there is a 95% probability that the parameter lies within the interval. Once an interval is calculated, it either covers the parameter value or it does not; it is no longer a matter of probability.

Bayesian credible intervals

There are many different possible credible intervals for any one distribution

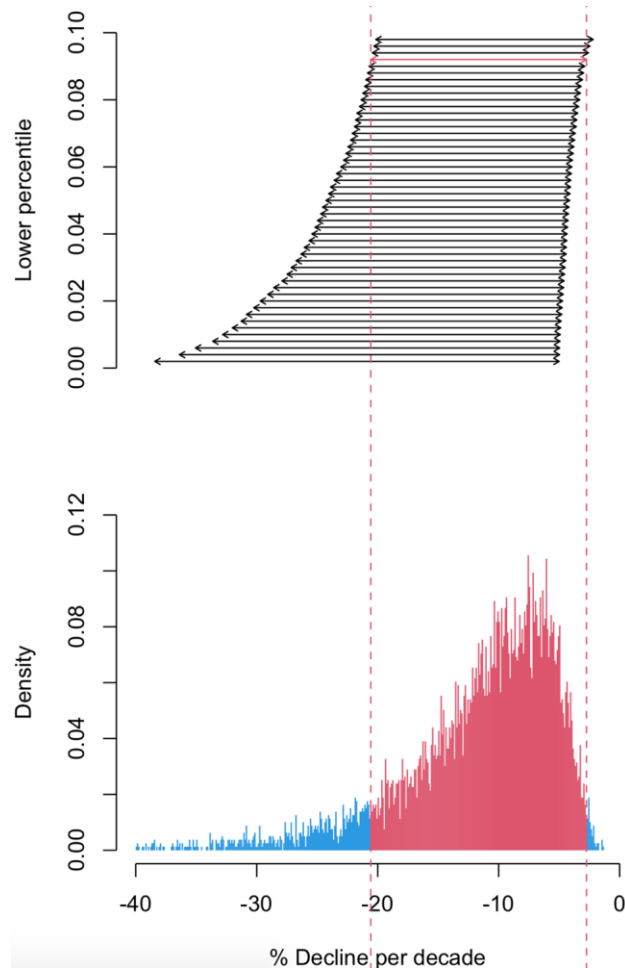
Want interval that best represents the parameter values consistent with



Bayesian credible intervals

Highest density posterior interval (HDPI)

- Narrowest range of values for a given probability interval
- Captures the 'densest' region of parameter values



Data analysis

Estimating things of interest

Quantifying uncertainty

Using models to make decisions/test hypotheses

Using models to make predictions

Model comparison

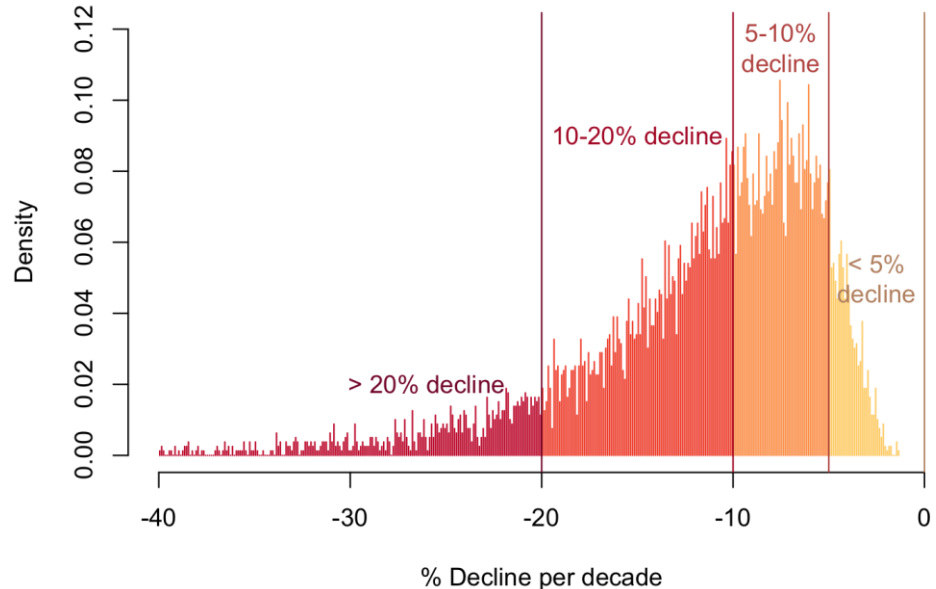
Measuring support for alternative rates

One advantage of Bayesian modelling comes in the ability to measure support for different ranges of parameter values

- i.e., what's the probability of decline $> 10\%$

Allows for informed decision making based on probability

Can also be used for meaningful hypothesis testing



Data analysis

Estimating things of interest

Quantifying uncertainty

Using models to make decisions/test hypotheses

Using models to make predictions

Model comparison

Posterior prediction

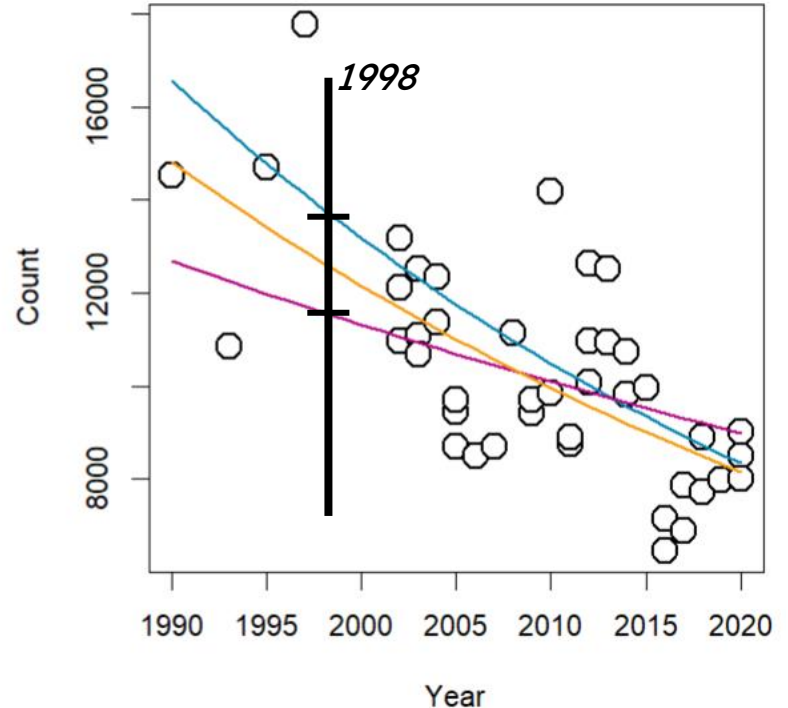
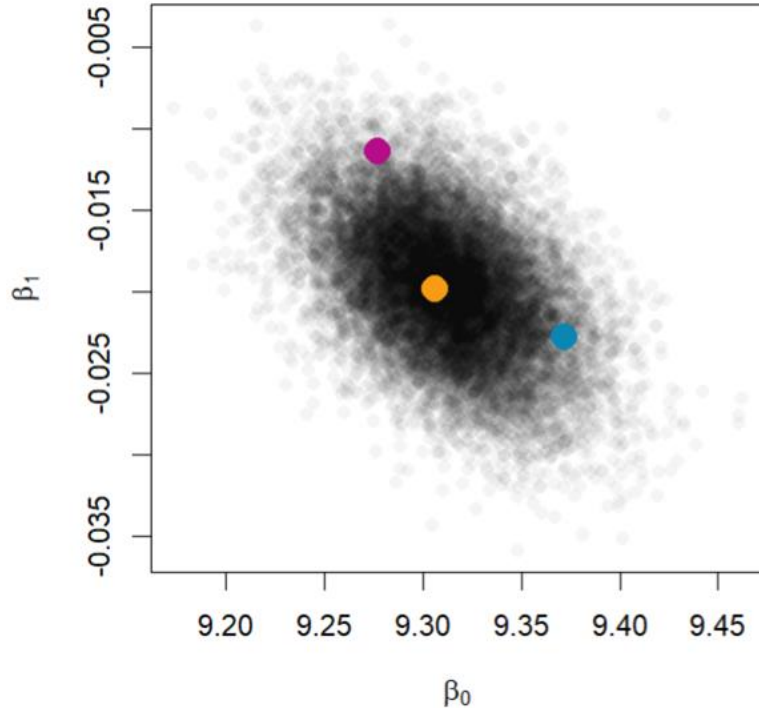
If you can fit a Bayesian model, you can generate values from it

Posterior prediction

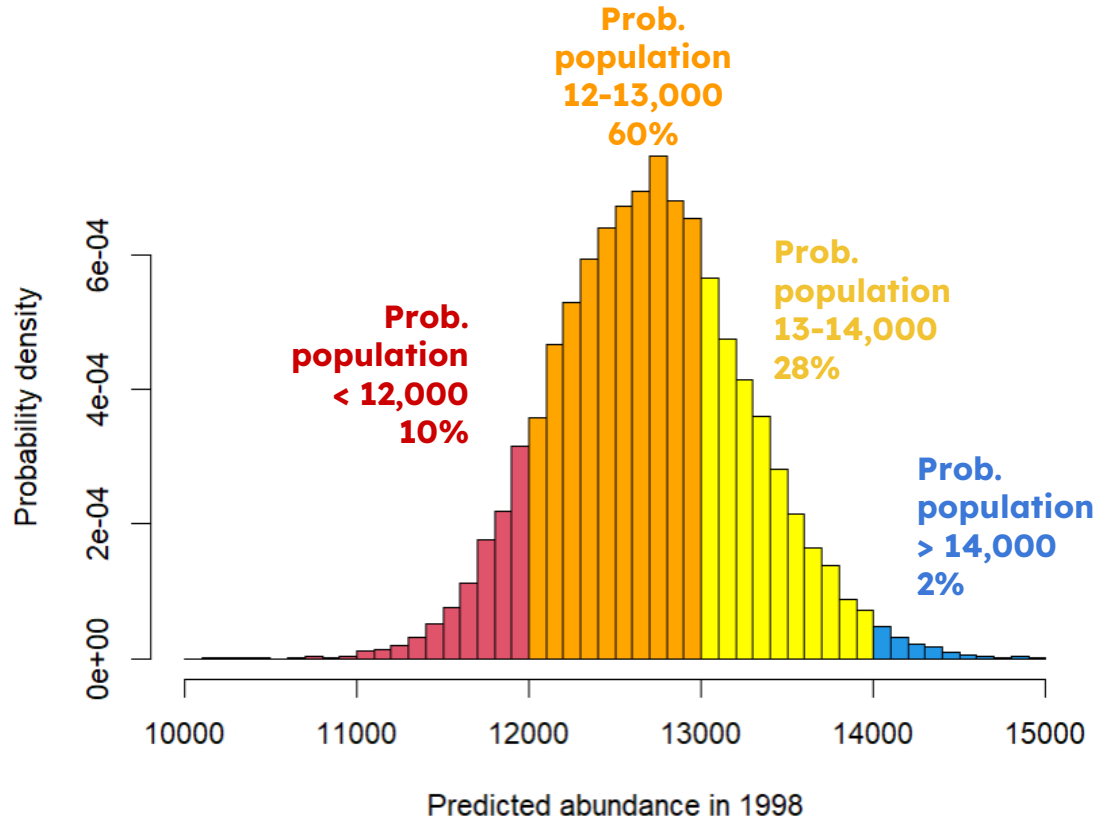
- Take parameter values from the posterior distribution and use those to generate values for a given scenario according to the model
- **Propagates uncertainty** of model **parameters** into predictions
 - i.e. fisheries model forecasts are built on this
- Posterior predictions can be treated as a **posterior distribution**
 - What is our posterior belief regarding a certain prediction
 - Allows identification of probability that predictions are above/below certain threshold

Posterior prediction

What was the expected population size in 1998?



Posterior prediction



Posterior prediction - assessing model fit

Posterior predictive distribution can be used to assess

- Adequacy of model fit (functional form)
- Assumptions of error distributions (how is variation accounted for)
- unusual observations not accounted for by the model

Data analysis

Estimating things of interest

Quantifying uncertainty

Using models to make decisions/test hypotheses

Using models to make predictions

Model comparison

Model comparison

Model comparison criteria choice should be driven by modelling purpose

- **Predictive ability**: cross-validation
- **Model parsimony**: information criterion

Both are possible using Bayesian models, and tools exist for each

Criteria have different names and are calculated differently, but the principle is identical (model fit penalised by complexity)

- Deviance information criterion
- Widely applicable Bayesian information criterion (WBIC or WAIC)

Sample size

What about sample size?

Bayesian models are **valid across all sample sizes**

Obviously more data is better, but the model doesn't break at small n

- Compare this to frequentist stats where model/statistic behavior is based on asymptotic behavior at large n , but break down for small n

At small n the posterior will tend to be more reminiscent of the prior

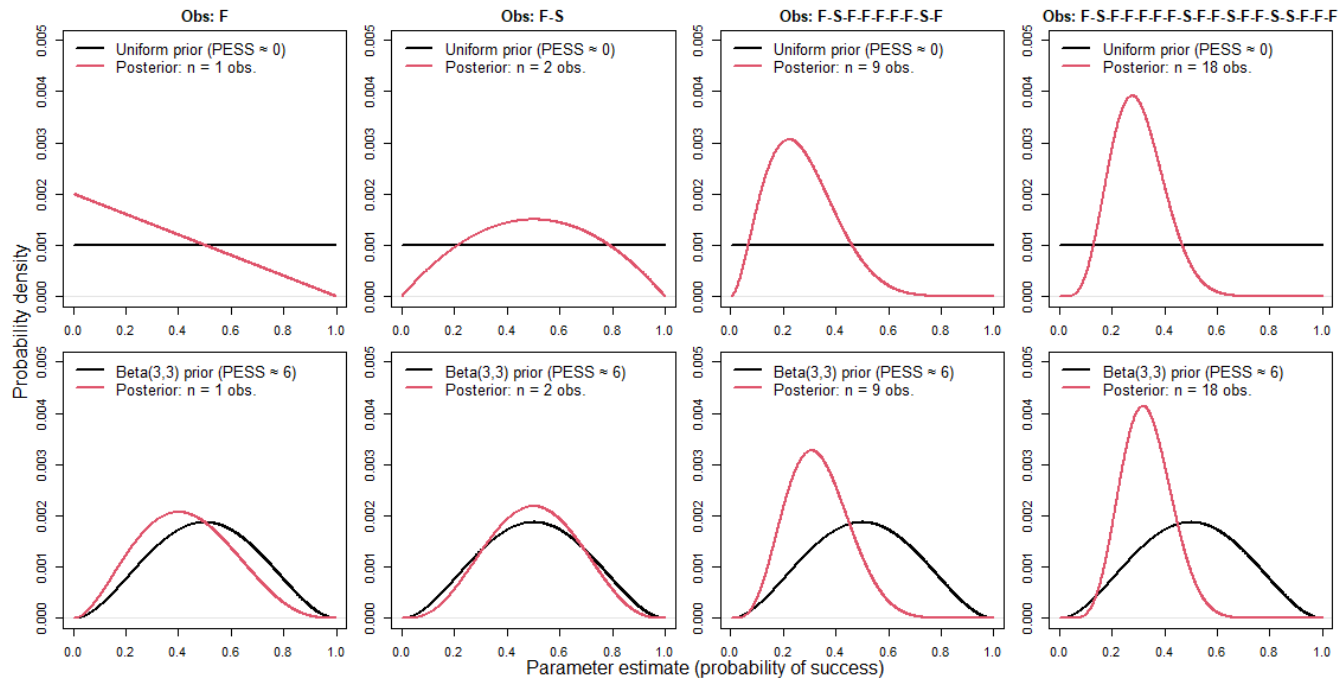
- Prior effective sample size - can be a useful measure for determining how much parameter estimates is solely governed by priors

Sample size

Model produces estimates for all n

Informative prior (Beta) equivalent to having 6 observations beforehand

- 3S 3F



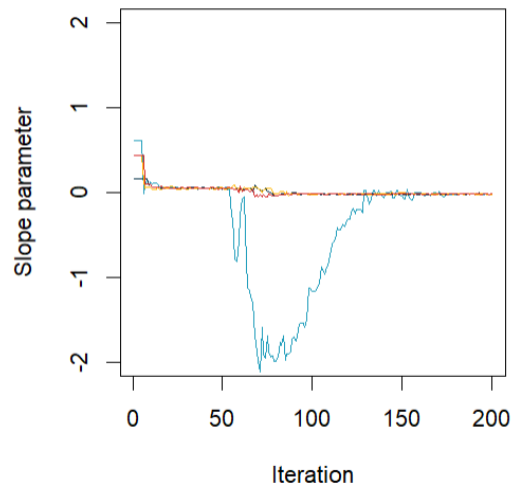
Did my model work?

Fitting Bayesian models typically requires more interaction with the 'machinery' of model fitting

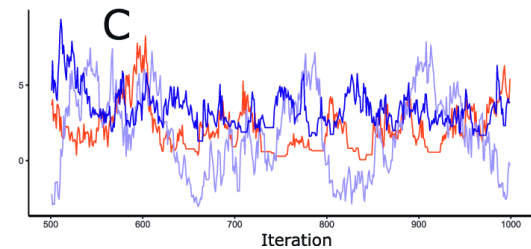
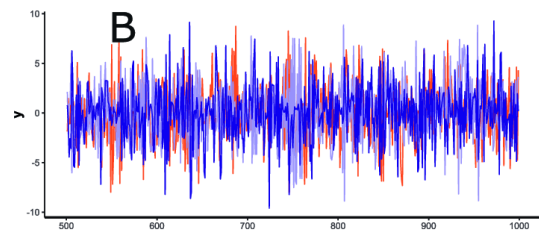
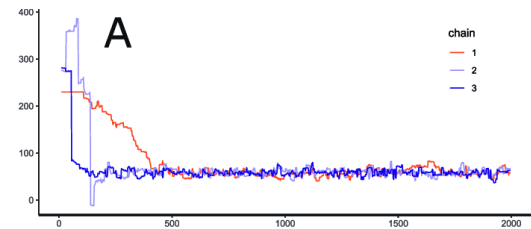
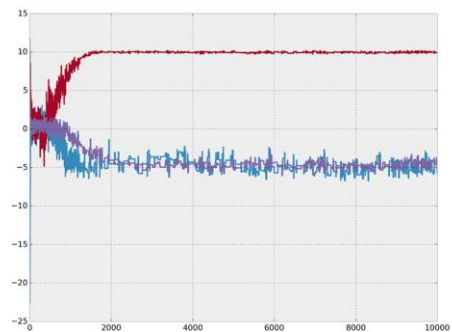
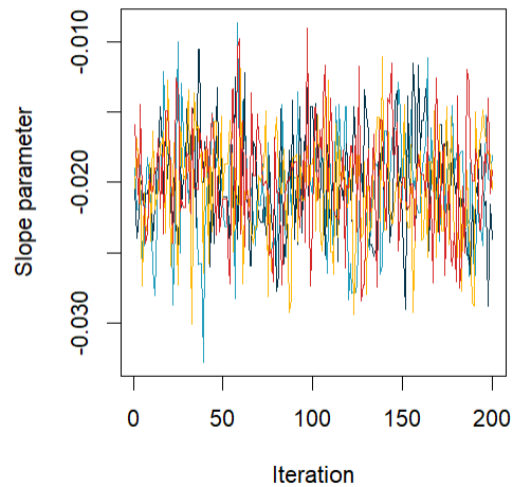
- With GLMs, LMs we often take for granted that the model simply worked and found an appropriate solution
- This is not always the case with Bayesian models fitted via MCMC

Primarily we are concerned with convergence

First 200 iterations



Iterations 201-400



Troubleshooting and model evaluation

This is a vast topic - we will just cover the basics

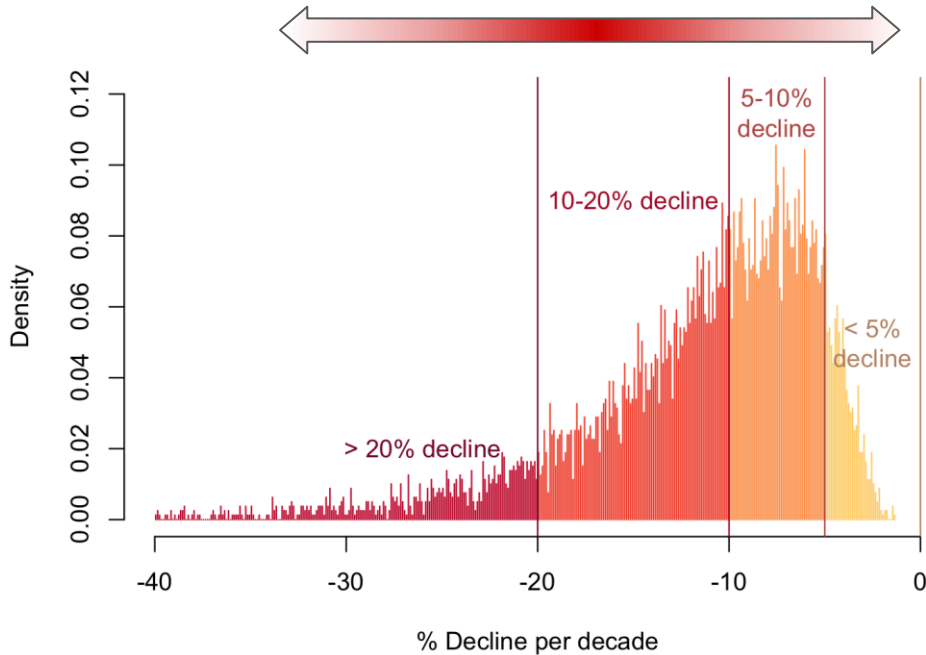
Model convergence

- Start models with multiple parallel MCMC chains (usually 4 or more)
- Examine mixing of chains (traceplot)
- If all of the chains end up in the same spot then we are more confident that the model found the solutions
- Can also calculate measures of within/between chain parameter estimates: R-hat statistic (ideally as close to 1 as possible, or < 1.05)

Key takeaways

Posterior distribution: probabilities across different parameter values

- **Point estimates:** mode = most probable, median = 50:50% split
- **Credible intervals:** more natural way to interpret uncertainty
- **Support for alternate values:** allows more/better options for decision making
- **Posterior predictive distribution:** propagation of uncertainty
- **Model comparison:** comparable tools exist, and follow similar approach



Questions?

Building Your Own Models

Overview

1. Welcome, Goals & Introductions
2. Bayes: An Updated Way of Thinking
3. Bayesian Data Analysis
4. Interpreting Bayesian Models
- 5. Building Your Own Models**
6. Tips/Tricks and Further Information
7. Workshop Wrap-up & Optional Happy Hour

Building the murre model

We will show you the pieces of the murre model written in Stan

Why Stan?

- Quick and efficient
- Gives plenty of warnings when models are poorly specified
- Uses a fancy MCMC algorithm that can explore parameter space more completely
- Can handle classes of models that would take other software prohibitively long run-times



Building the murre model

We will show you the pieces of the murre model written in Stan

But..

- Stan is much pickier about definitions than R
 - This is true for most Bayesian software
- Any full-fat Bayesian software requires more coding know-how than base R

Building the murre model

Fitting a Bayesian GLM is almost as simple as using `glm()`

`rstanarm`

- Contains Bayesian equivalents of typical LMs, GLMs, GLMERs, GAMs
- Uses the same syntax, with the additions of defining prior distributions
- This might be all you need to get started

```
# The simple way
murre_stan <- stan_glm(count ~ year, data = murre.df,
                      family = poisson(link="log"),
                      prior_intercept = normal(8,4),
                      prior = normal(0,0.8),
                      iter = 2000,
                      warmup = 1000,
                      chains = 4)
```

Building the murre model

Some unfamiliar components

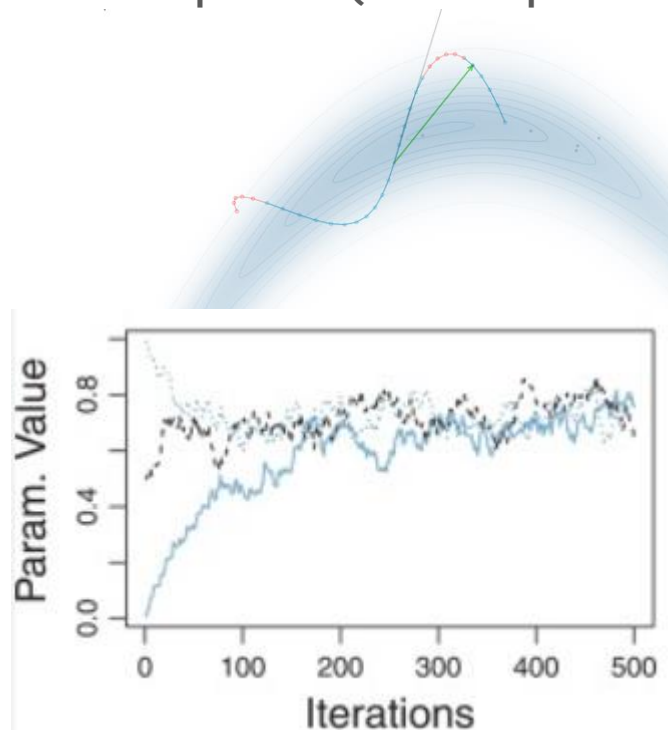
- `iter`: total number of iterations (including warm-up)
- `warmup`: number of warmup iterations
- `chains`: number of MCMC chains

```
# The simple way
murre_stan <- stan_glm(count ~ year, data = murre.df,
  family = poisson(link="log"),
  prior_intercept = normal(8,4),
  prior = normal(0,0.8),
  iter = 2000,
  warmup = 1000,
  chains = 4)
```

Warmup vs Sampling

In Stan (or JAGS, BUGS), MCMC is split into an initial phase (warmup or burn-in) and a sampling phase

- We only use values from sampling phase
- Warm-up used to tune how the model samples (Stan) and/or 'find' the target
- Typically
 - Set the first half of iterations as warm up, and the rest as sampling
 - # of iterations governed by iterations to convergence/explore parameters and run-time - typically thousands of iterations



Model summary and diagnostics

```
> # examine coefficients  
> print(murre_stan, digits=4)
```

```
stan_glm  
family:      poisson [log]  
formula:     count ~ year.off  
observations: 43  
predictors:   2
```

```
-----  
              Median  MAD_SD  
(Intercept)  9.3102  0.0015  
year.off      -0.0196  0.0002
```

Median (point estimate)

median posterior parameter estimate

MAD_SD (measure of uncertainty)

Median absolute deviation (MAD) from the median (think of this as a more robust measure of standard deviation for variation in MCMC parameter value)

```
-----  
  
> # create 95% credible interval  
> posterior_interval(murre_stan, prob=0.95)
```

```
              2.5%      97.5%  
(Intercept)  9.30703001  9.31310158  
year.off      -0.01998104 -0.01921781
```

Posterior interval (credible interval)

95% credible intervals of parameter values

Model summary and diagnostics

```
> # examine coefficients
> print(murre_stan, digits=4)
stan_glm
family:      poisson [log]
formula:     count ~ year.off
observations: 43
predictors:  2
```

```
-----
              Median  MAD_SD
(Intercept)  9.3102  0.0015
year.off     -0.0196  0.0002
-----
```

```
> # create 95% credible interval
> posterior_interval(murre_stan, prob=0.95)
              2.5%      97.5%
(Intercept)  9.30703001  9.31310158
year.off     -0.01998104 -0.01921781
```

```
> # View model estimates and MCMC diagnostics
> launch_shinystan(murre_stan)
```



Use your mouse to highlight areas in the traceplot to zoom into. Double-click to reset. The number in the small black box in the bottom left corner controls the roll period. If you specify a roll period of N the resulting graph will be a moving average, with each plotted point representing the average of N points in the data.

But, what if my model isn't simple?

Even though `rstanarm` can do most of the workhorse basic models, it is worth looking inside the machine to how the model is actually setup

Pulling back the curtain

- Next time....

We don't want to overwhelm you, there is enough here to digest

We will provide a code-file of the murre model written in base Stan, which you are welcome to explore



Key Takeaways

Now more than ever it is easy to start in Bayes

- **Model specification:** if you are comfortable with GLMs, then `rstanarm` is only a teeny-tiny step up
- **Model types and evaluation:** has all the benefits we discussed, with very few drawbacks - could easily replace your GLM/GLMER/GAM workflow
- **Getting comfortable:** allows for a gentle step into Bayes to build confidence
- **So much more:** once you feel comfortable, diving into a Bayesian model built from scratch allows for more complex models, and is incredibly rewarding



```
murre_stan <- stan_glm(count ~ year, data = murre.df,  
  family = poisson(link="log"),  
  prior_intercept = normal(8,4),  
  prior = normal(0,0.8),  
  iter = 2000,  
  warmup = 1000,  
  chains = 4)
```

Model language

Models are typically described in a common language involving

- Data - response, predictors
- Parameters
- How the parameters and predictors interact
- The likelihood function of the response data

When creating the model we need to define each in a specific way

- What *type*, how is it *contained*, and how *many*
- What are the mathematical relationships
- What are the probability distributions

Pulling back the curtain

Stan model layout (can be created and edited in RStudio)

```
// What follows is the different blocks of a Stan model|

data {
  // Declare the types of data going in
  // how they are stored (unit, vector, matrix)
  // dimension (how many)
}

parameters {
  // Define model parameters to be estimated
  // how they are stored and dimensions
}

transformed parameters {
  // If we want to transform parameters we can
  // do so here
}

model {
  // Here is where we define the statistical parts
  // prior definitions
  // likelihood
}
```



Murre model - Data

N_obs: number of observations

- Declared as type **int**

Year: numeric year values

- Vector of reals (decimal)
- Distinction between reals & integers is important
- Year used to calculate lambda & want lambda to be vector of reals
- In equations, type on LHS must match type on the RHS
- Declared as type **real** with dimension N_obs in []

Murres: murre counts

- Vector of integers (count)
- Counts ~ Poisson, only defined for integer data
- Declared as type **int** with dimension N_obs in []

```
data {  
  // Data declarations  
  int N_obs;           // number of observations  
  real Year[N_obs];     // vector of reals predictor values  
  int Murres[N_obs];    // vector of integer counts  
}  
  
parameters {  
  // Model parameter declarations  
  real beta0;           // Defining intercept parameter as a real  
  real beta1;           // Defining slope parameter as a real  
}  
  
transformed parameters {  
  // Parameter transformation  
}  
  
model {  
  // prior definitions  
  beta0 ~ normal(8, 4);   // Prior for intercept  
  beta1 ~ normal(0, 0.8); // Prior for slope  
  
  // deterministic (i.e. the maths part)  
  real lambda[N_obs];    // vector of means  
  
  for(i in 1:N_obs){      // loop over observations  
    // For each Count response, calculate lambda  
    lambda[i] = exp(beta0 + (beta1 * Year[i]));  
  }  
  
  // likelihood  
  Murres ~ poisson(lambda);  
}
```

Murre model - Parameters

beta0: intercept parameter

beta1: slope parameter

- Declared as type **real**

There are multiple equivalent ways to define things

Alternate

- **real** beta[2];
- Establishes beta as a vector of reals with
 - beta[1] : intercept
 - beta[2] : slope

```
data {  
  // Data declarations  
  int N_obs;           // number of observations  
  real Year[N_obs];    // vector of reals predictor values  
  int Murres[N_obs];   // vector of integer counts  
}  
  
parameters {  
  // Model parameter declarations  
  real beta0;          // Defining intercept parameter as a real  
  real beta1;          // Defining slope parameter as a real  
}  
  
transformed parameters {  
  // Parameter transformation  
}  
  
model {  
  // prior definitions  
  beta0 ~ normal(8, 4); // Prior for intercept  
  beta1 ~ normal(0, 0.8); // Prior for slope  
  
  // deterministic (i.e. the maths part)  
  real lambda[N_obs]; // vector of means  
  
  for(i in 1:N_obs){ // loop over observations  
    // For each Count response, calculate lambda  
    lambda[i] = exp(beta0 + (beta1 * Year[i]));  
  }  
  
  // likelihood  
  Murres ~ poisson(lambda);  
}
```

Murre model - Transformed parameters

Left empty, however ...

The deterministic part of the model

$$\log(\lambda_t) = \beta_0 + \beta_1 t$$

Could be placed within this block

Placing model components in the transformed parameters block has speed advantages

```
data {  
  // Data declarations  
  int N_obs;           // number of observations  
  real Year[N_obs];    // vector of reals predictor values  
  int Murres[N_obs];   // vector of integer counts  
}  
  
parameters {  
  // Model parameter declarations  
  real beta0;          // Defining intercept parameter as a real  
  real beta1;          // Defining slope parameter as a real  
}  
  
transformed parameters {  
  // Parameter transformation  
}  
  
model {  
  // prior definitions  
  beta0 ~ normal(8, 4);    // Prior for intercept  
  beta1 ~ normal(0, 0.8);  // Prior for slope  
  
  // deterministic (i.e. the maths part)  
  real lambda[N_obs];     // vector of means  
  
  for(i in 1:N_obs){      // loop over observations  
    // For each Count response, calculate lambda  
    lambda[i] = exp(beta0 + (beta1 * Year[i]));  
  }  
  
  // likelihood  
  Murres ~ poisson(lambda);  
}
```

Murre model - Model priors

```
beta0 ~ normal(8, 4);
```

- Uses standard statistical lingo
- Should be read

“the prior probability distribution for beta0 is normal distribution mean = 8, and standard deviation = 4”

- Similar argument applies for beta1

Stan supports many probability distributions

- Normal, cauchy, beta, t, exponential, dirichlet
- Check documentation for usage/definition

```
data {  
  // Data declarations  
  int N_obs;           // number of observations  
  real Year[N_obs];    // vector of reals predictor values  
  int Murres[N_obs];   // vector of integer counts  
}  
  
parameters {  
  // Model parameter declarations  
  real beta0;          // Defining intercept parameter as a real  
  real beta1;          // Defining slope parameter as a real  
}  
  
transformed parameters {  
  // Parameter transformation  
}  
  
model {  
  // prior definitions  
  beta0 ~ normal(8, 4);    // Prior for intercept  
  beta1 ~ normal(0, 0.8);  // Prior for slope  
  
  // deterministic (i.e. the maths part)  
  real lambda[N_obs];     // vector of means  
  
  for(i in 1:N_obs){      // loop over observations  
    // For each Count response, calculate lambda  
    lambda[i] = exp(beta0 + (beta1 * Year[i]));  
  }  
  
  // likelihood  
  Murres ~ poisson(lambda);  
}
```

Murre model - Deterministic model

lambda: vector of real mean values of length **N_obs**

- We then loop through all observations to calculate **lambda** for each observation the dataset

```
data {  
  // Data declarations  
  int N_obs;           // number of observations  
  real Year[N_obs];    // vector of reals predictor values  
  int Murres[N_obs];   // vector of integer counts  
}  
  
parameters {  
  // Model parameter declarations  
  real beta0;          // Defining intercept parameter as a real  
  real beta1;          // Defining slope parameter as a real  
}  
  
transformed parameters {  
  // Parameter transformation  
}  
  
model {  
  // prior definitions  
  beta0 ~ normal(8, 4); // Prior for intercept  
  beta1 ~ normal(0, 0.8); // Prior for slope  
  
  // deterministic (i.e. the maths part)  
  real lambda[N_obs]; // vector of means  
  
  for(i in 1:N_obs){ // loop over observations  
    // For each Count response, calculate lambda  
    lambda[i] = exp(beta0 + (beta1 * Year[i]));  
  }  
  
  // likelihood  
  Murres ~ poisson(lambda);  
}
```


Murre model - Model likelihood

Murres ~ poisson(lambda)

- Uses the same statistical definition as before, and should be read as

“The probability distribution of Murre counts is poisson with mean lambda”

Note: we define the collective Murres variable ~ poisson(lambda), which implicitly calculates the likelihood across all counts for all lambda

An alternate would be to define

Murres[i] ~ poisson(lambda[i]);

Within the loop - the two approaches are equivalent

```
data {  
  // Data declarations  
  int N_obs;           // number of observations  
  real Year[N_obs];    // vector of reals predictor values  
  int Murres[N_obs];   // vector of integer counts  
}  
  
parameters {  
  // Model parameter declarations  
  real beta0;          // Defining intercept parameter as a real  
  real beta1;          // Defining slope parameter as a real  
}  
  
transformed parameters {  
  // Parameter transformation  
}  
  
model {  
  // prior definitions  
  beta0 ~ normal(8, 4); // Prior for intercept  
  beta1 ~ normal(0, 0.8); // Prior for slope  
  
  // deterministic (i.e. the maths part)  
  real lambda[N_obs]; // vector of means  
  
  for(i in 1:N_obs){ // loop over observations  
    // For each Count response, calculate lambda  
    lambda[i] = exp(beta0 + (beta1 * Year[i]));  
  }  
  
  // likelihood  
  Murres ~ poisson(lambda);  
}
```

Questions?

Tips/Tricks and Further Information

Troubleshooting and model evaluation

This is a vast topic - we will just cover the basics!

Model convergence

- When estimating a model we need to ensure the MCMC algorithm has converged on the posterior distribution
- Possible problems arise due to local maxima ‘trapping’ the exploration process
- Local maxima: <https://chi-feng.github.io/mcmc-demo/app.html?algorithm=H2MC&target=multimodal>

Troubleshooting and model evaluation

Model convergence

- Always start models with multiple parallel MCMC chains (usually 4 or more)
- Examine mixing of chains (e.g., with trace plots)
- If all of the chains end up in the same spot, then we are more confident that the model found the solutions
- Can also calculate measures of within/between chain parameter estimates: R-hat statistic (ideally as close to 1 as possible, or < 1.05)

Troubleshooting and model evaluation

Capturing the extent of possible solutions

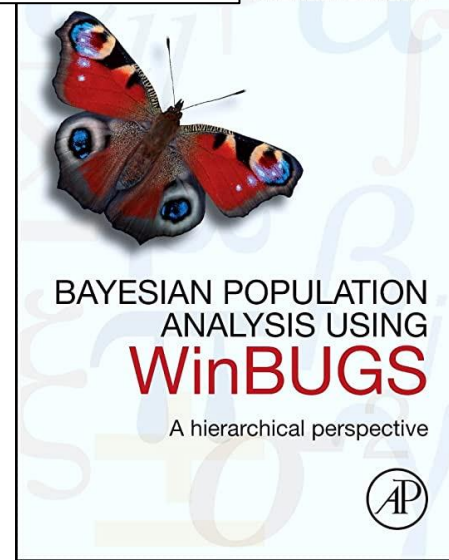
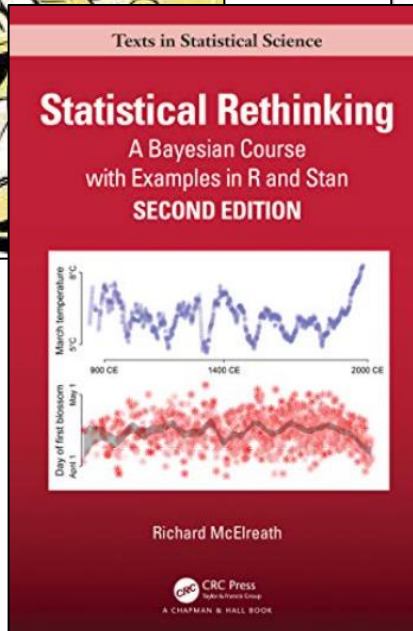
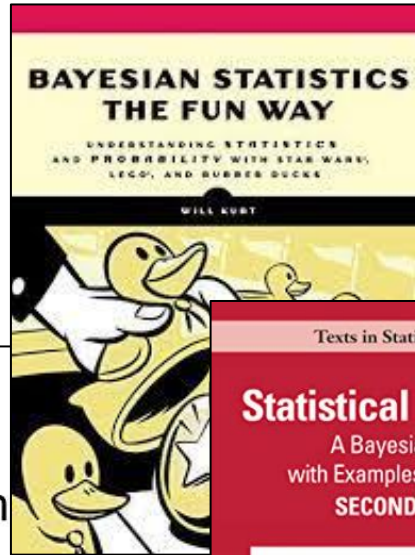
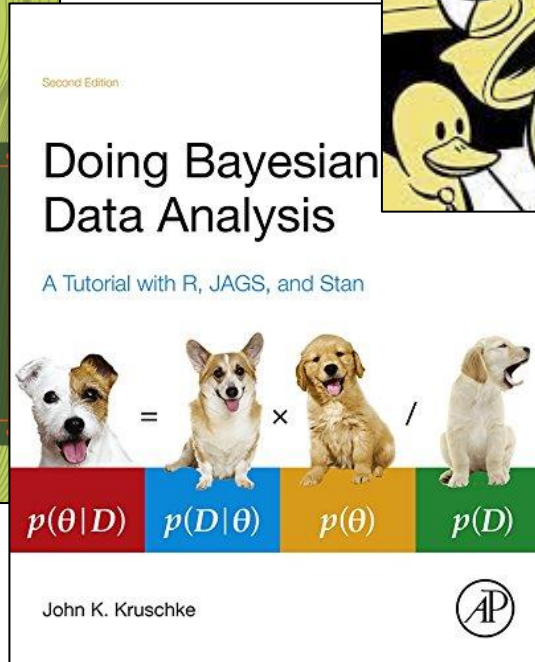
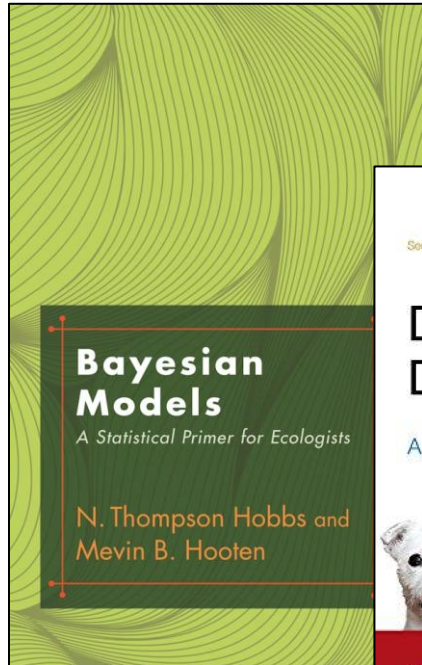
- Running a model for too few iterations can mean that there hasn't been enough time to explore the full range of solutions
 - We want to capture the most probable parameter values, but also the tails in order to estimate uncertainty
- Some software will complain about effective sample size (or tail effective sample size) - the number of estimates for a given parameter
- We ideally want this to be high
- Solution: run the model a little while longer

Troubleshooting and model evaluation

Model identifiability and re-parameterisation

- Sometimes a model is so complex that parameters cannot be estimated effectively
- Often this can be remedied via a re-parameterisation (i.e. modelling $\log(B)$ rather than B), or re-scaling/standardisation of predictors/response

Resources



Other online resources/examples

[Bayesian statistics with R](#)

[Bayesian Inference 2019](#)

[MCMC and the case of the spilled seeds](#)

[A Zero-Math Introduction to Markov Chain Monte Carlo Methods](#)

Further reading in the seabird field

Papers in our field that use Bayesian stats

- Bertram, D. F., M. C. Drever, M. K. McAllister, B. K. Schroeder, D. J. Lindsay, and D. A. Faust. 2015. Estimation of coast-wide population trends of Marbled Murrelets in Canada using a Bayesian hierarchical model. *PLOS ONE* 10:e0134891.
- Bertram, D. F., L. Wilson, K. Charleton, A. Hedd, G. J. Robertson, J. L. Smith, K. H. Morgan, and X. J. Song. 2021. Modelling entanglement rates to estimate mortality of marine birds in British Columbia commercial salmon gillnet fisheries. *Marine Environmental Research* 166:105268.
- Good, T. P., J. E. Jannot, K. A. Somers, and E. J. Ward. 2022. Using Bayesian time series models to estimate bycatch of an endangered albatross. *Fisheries Research* 256:106492.
- Horswill, C., M. J. Wood, and A. Manica. 2022. Temporal change in the contribution of immigration to population growth in a wild seabird experiencing rapid population decline. *Ecography* 2022:e05846.
- Lahoz-Monfort, J. J., B. J. T. Morgan, M. P. Harris, F. Daunt, S. Wanless, and S. N. Freeman. 2013. Breeding together: modeling synchrony in productivity in a seabird community. *Ecology* 94:3–10.

Further reading in the seabird field

- Richard, Y., L. Perriman, C. Lalas, and E. R. Abraham. 2015. Demographic rates of northern royal albatross at Taiaroa Head, New Zealand. *PeerJ* 3:e906.
- Sarzo, B., C. Armero, D. Conesa, J. Hentati-Sundberg, and O. Olsson. 2019. Bayesian immature survival analysis of the largest colony of Common Murre (*Uria aalge*) in the Baltic Sea. *Waterbirds* 42:304.
- Sherley, R. B., R. J. M. Crawford, A. D. de Blocq, B. M. Dyer, D. Geldenhuys, C. Hagen, J. Kemper, A. B. Makhado, L. Pichegru, D. Tom, L. Upfold, J. Visagie, L. J. Waller, and H. Winker. 2020. The conservation status and population decline of the African penguin deconstructed in space and time. *Ecology and Evolution* 10:8506–8516.
- Stantial, M. L., J. B. Cohen, A. J. Darrah, S. Farrell, and B. Maslo. 2021. Habitat-specific behavior, growth rate, and survival of piping plover chicks in New Jersey, USA. *Ecosphere* 12:e03782.
- Tinker, M. T., K. M. Zilliacus, D. Ruiz, B. R. Tershy, and D. A. Croll. 2022. Seabird meta-Population Viability Model (mPVA) methods. *MethodsX* 9:101599.

Further reading in the seabird field

Tucker, A. M., C. P. McGowan, J. E. Lyons, A. DeRose-Wilson, and N. A. Clark. 2021. Species-specific demographic and behavioral responses to food availability during migratory stopover. *Population Ecology* 1438–390X.12094.

van de Schoot, R., S. Depaoli, R. King, B. Kramer, K. Märtens, M. G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen, and C. Yau. 2021. Bayesian statistics and modelling. *Nature Reviews Methods Primers* 1:1–26.

...and many more!

Workshop Wrap-Up

Seabird data is messy.

Some of the challenges include:

- Difficult to get to colonies
- Hard to find individuals and/or hard to count them all
- Individuals are inaccessible during portions of life history

Which can lead to small sample sizes, gaps, and uncertainty in our datasets.

Good news, many of these issues can be addressed using Bayesian methods!

Workshop Goals

1. Recognize **value of Bayesian framework** for seabird research.
2. Understand **fundamental differences** between Frequentist and Bayesian frameworks.
3. Learn **basic components** of a Bayesian analysis.
4. Learn how to **interpret results** from a Bayesian analysis.
5. Build **community and share resources** for future learning.

Where to from here?

- Additional materials to be shared soon (extended version of the workshop presentation, Stan installation information, R script to run the murre model)
- Opt-in email listserv to ask questions, share resources, etc. - let us know if you are interested!
- Gauging interest in establishment of PSG Analytical Technical Committee
- Last but not least...

Optional Happy Hour

Shore Rider Bar

2168 Avenida De La Playa

La Jolla, CA 92037