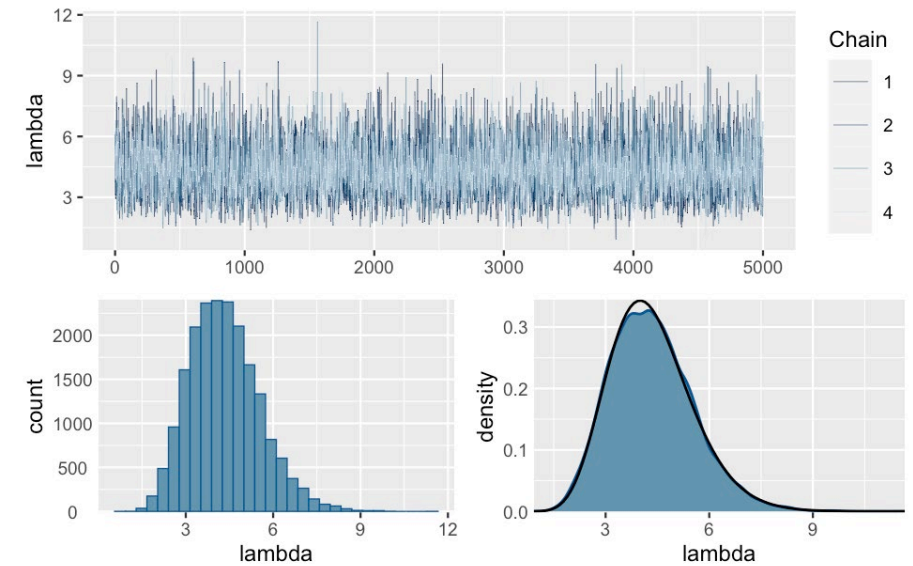
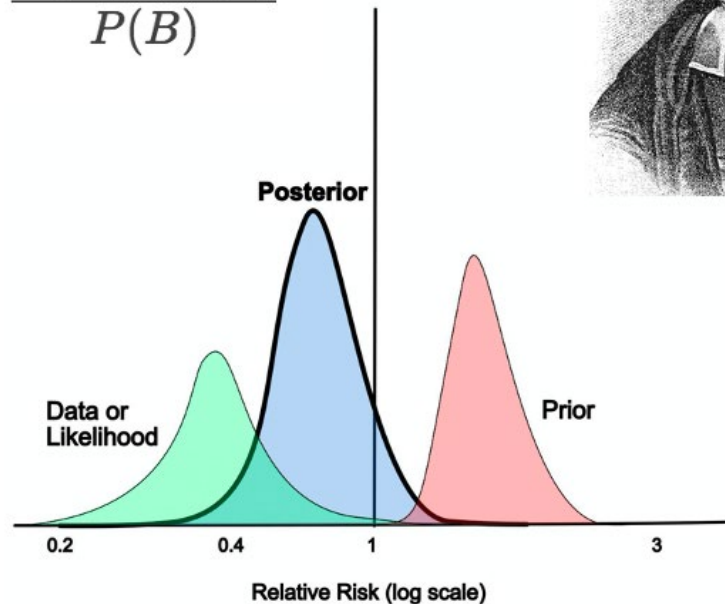


Bayesian Analysis Workshop: Going Bayesian!

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$



Useful R Packages for Bayesian Analyses

- **`brms`** – Provides an interface to fit Bayesian generalized multivariate (non-) linear multilevel models using Stan
- **`jagsUI`** – A Wrapper Around 'rjags' to Streamline 'JAGS' Analyses
- **`bayesplot`** – Provides variety of ggplot2-based plotting functions for Bayesian models
- **`bayestestR`** – Provides a comprehensive and consistent set of functions to analyze and describe posterior distributions generated by a variety of models objects
- **`ubms`** – Bayesian Models for Data from Unmarked Animals using 'Stan'
- **`spOccupancy`** – Fits single-species, multi-species, and integrated non-spatial and spatial occupancy models using Markov Chain Monte Carlo (MCMC)

Useful R Packages for Bayesian Analyses

- **`tidybayes`** - useful packages for wrangling and visualizing (w/ ggplot2) posterior draws and predictions from brms models.
- **`modelr`** - often used in tandem with **tidybayes** for creating complex prediction grids and scenarios
- **`rethinking`** - R package based on *Statistical Rethinking* that uses Stan (i.e., cmdstanr) in background; useful for linear models, GLMMs, gaussian processes, differential equations, and directed network/block models
- **`marginaleffects`** - R package for visualizing marginal effects of fitted models (handles frequentist or bayesian GLMs)
- **`mice`** - package for conducting multiple imputation on missing data before model fitting
- **`bergm`** - Bayesian version of exponential random graph models

Overview

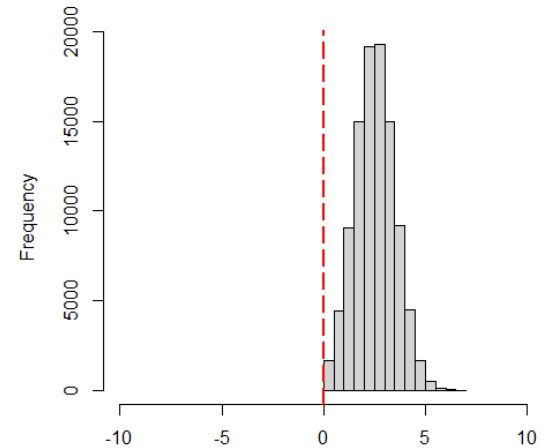
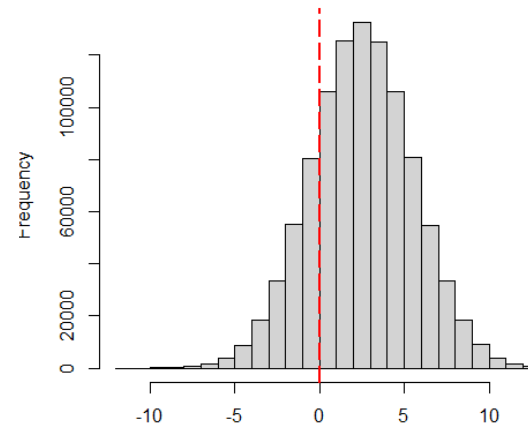
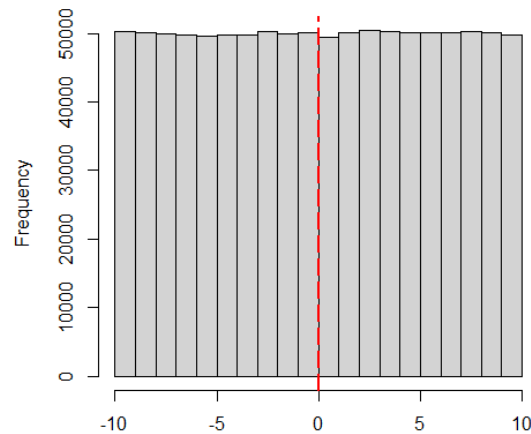
- Steps to Bayesian data analysis
- Specifying Priors (or not)
- MCMC
- Model checks & Diagnostics
- Posterior distributions
- Model Summaries
- Model Warnings
- Applications in the wild

Steps to Bayesian Data Analysis

- Data: Identify variables to be predicted and variables that act as predictors
- Define a descriptive model
- Specify a prior distribution on the parameters
- Use Bayesian inference to re-allocate credibility across parameter values.
- Check that posterior predictions mimic the data with reasonable accuracy (i.e., posterior predictive check)

Priors

- Defined as a probability distribution
- Can be uninformed, weakly informed / vague, informed



Priors

- Defined as a probability distribution
- Can be uninformed, weakly informed / vague, informed
- Common prior distributions
 - Uniform
 - Normal
 - Student-t (brms)
 - Exponential
 - Gamma
 - Beta
 - Many more!

Priors

- Priors in JAGS

- $\sim \text{dnorm}\left(\mu, \frac{1}{\text{variance}}\right)$
 - Precision = $1/\text{variance} \rightarrow 1/\text{sqrt}(\text{sd})$

- Priors in Stan (`brms`)

- 'Traditional' mean, variance specification

- Must be aware of how/if priors are being transformed

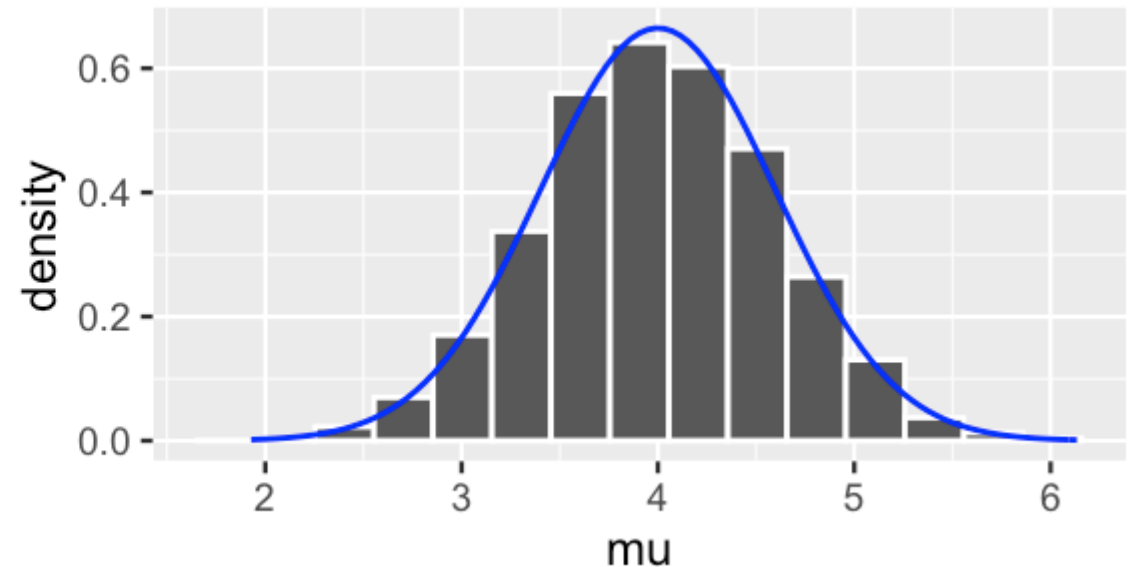
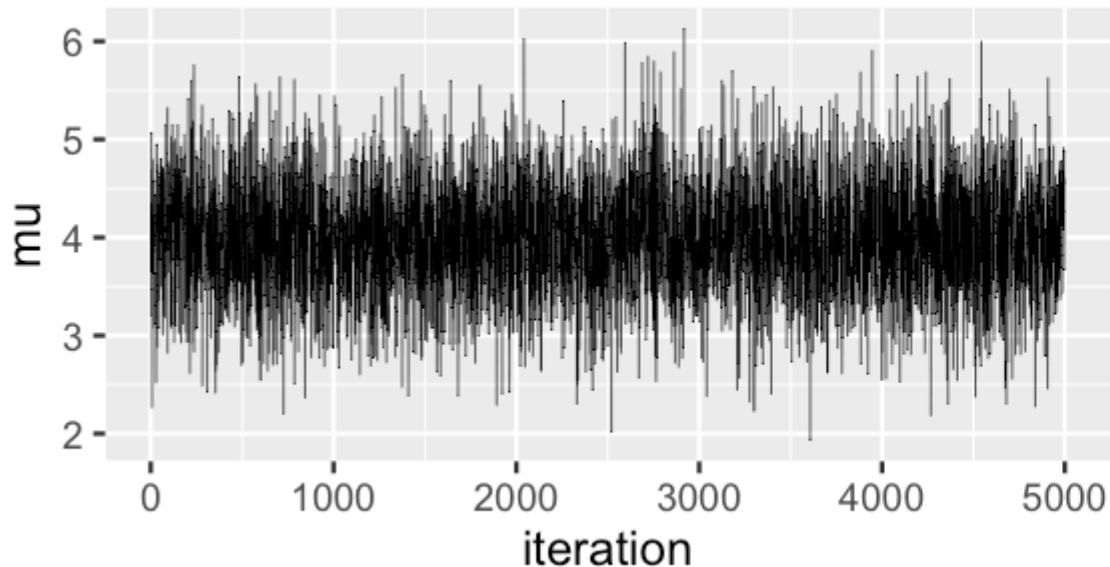
- Uninformed priors can become informative if link function applied

Priors x Data

- <https://micl.shinyapps.io/prior2post/>

Estimating posterior distributions

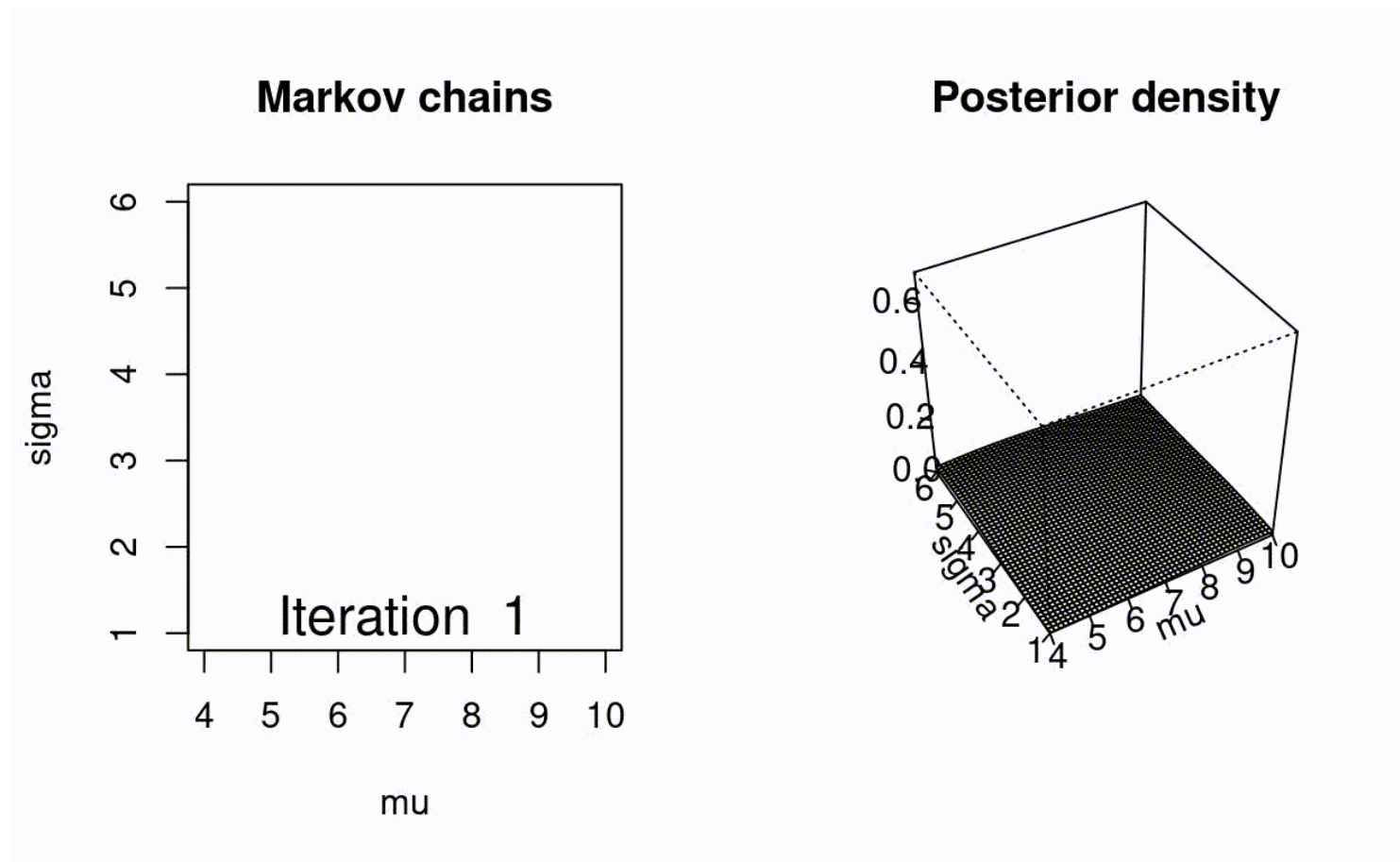
- Closed form solutions rarely exist
- Possible to estimate through Markov chain Monte Carlo methods
 - Following thousands of samples, posterior distribution(s) effectively estimated



MCMC in Action

Parameters

```
mu <- 7  
sigma <- 3
```



This MCMC stuff is NUTS

- NUTS = No U-Turn Sampler
- An efficient extension of the Hamiltonian Monte Carlo algorithm
- Visualize NUTS algorithm:
 - <https://chi-feng.github.io/mcmc-demo/app.html?algorithm=NaiveNUTS&target=banana>

Markov Chains

- How many chains to run?
- Burn-in / warm-up length?
- How many samples are needed?
- Should I thin?
- Model diagnostics of MCMC
 - Visual inspection – mixing, stability
 - R-hat
 - Autocorrelation

Markov Chains

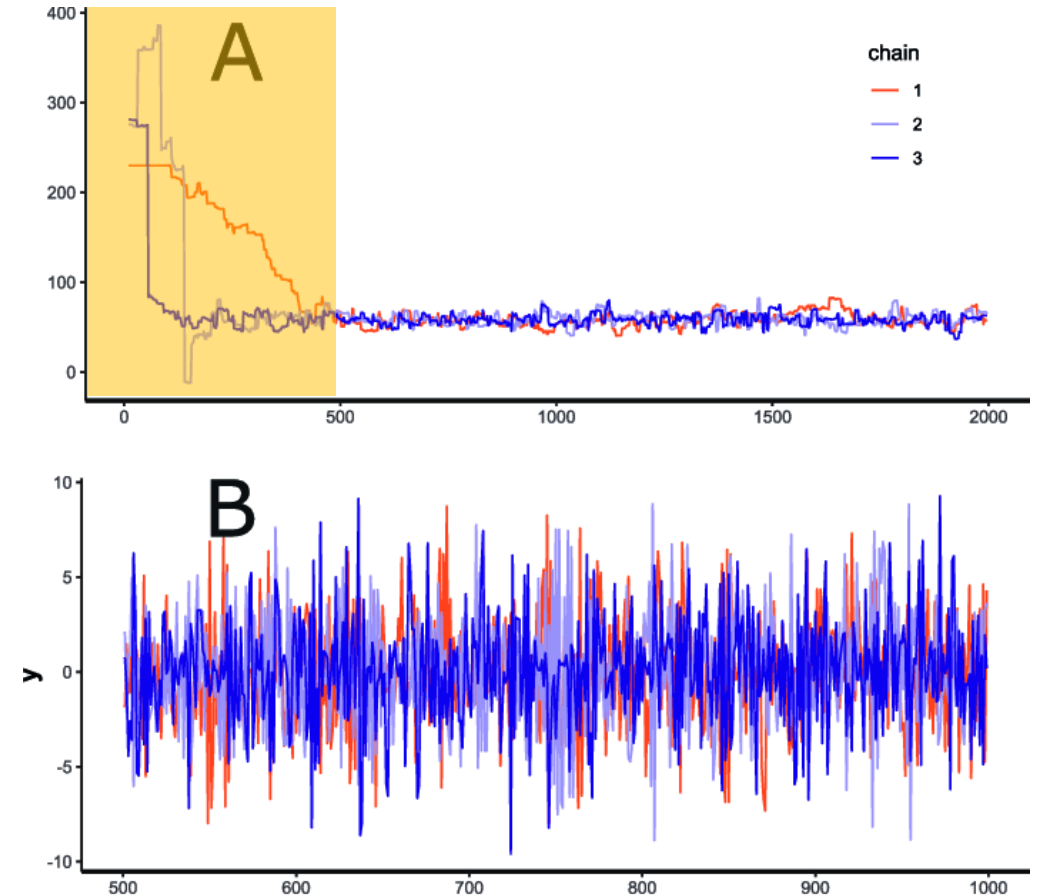
- How many chains to run?
 - At least 3, the more the merrier
 - More chains = greater posterior sample with fewer MCMC samples
 - 1,000 samples from 2 chains = 2k samples
 - 500 samples from 4 chains = 2k samples

```
brm(weight ~ scale(height),  
     data = dat,  
     family = gaussian(),  
     chains = 4,  
     iter = 2000,  
     warmup = 1000,  
     thin = 1,  
     cores = 4)
```

Markov Chains

- Burn-in / warm-up length?
 - Need sufficient warmup so that MCMC chains are sampling the posterior
 - `brms` defaults to first half of MCMC iterations for warmup
 - Typically, 1–5k iterations sufficient

```
brm(weight ~ scale(height),  
     data = dat,  
     family = gaussian(),  
     chains = 4,  
     iter = 2000,  
     warmup = 1000,  
     thin = 1,  
     cores = 4)
```



Markov Chains

- How many samples are needed?
 - iter – warmup = posterior samples
 - Ideally > 100 **effective samples** per parameter, per chain
 - If calculating highest density intervals (HDI), $\geq 40,000$ samples suggested

```
brm(weight ~ scale(height),  
    data = dat,  
    family = gaussian(),  
    chains = 4,  
    iter = 2000,  
    warmup = 1000,  
    thin = 1,  
    cores = 4)
```



```

> summary(m3)
Family: gaussian
Links: mu = identity; sigma = identity
Formula: weight ~ scale(height)
Data: d (Number of observations: 15)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
       total post-warmup draws = 4000

```

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	63.28	2.69	58.14	68.60	1.00	2005	1766
scaleheight	1.72	1.21	0.06	4.51	1.00	1273	904

Family Specific Parameters:

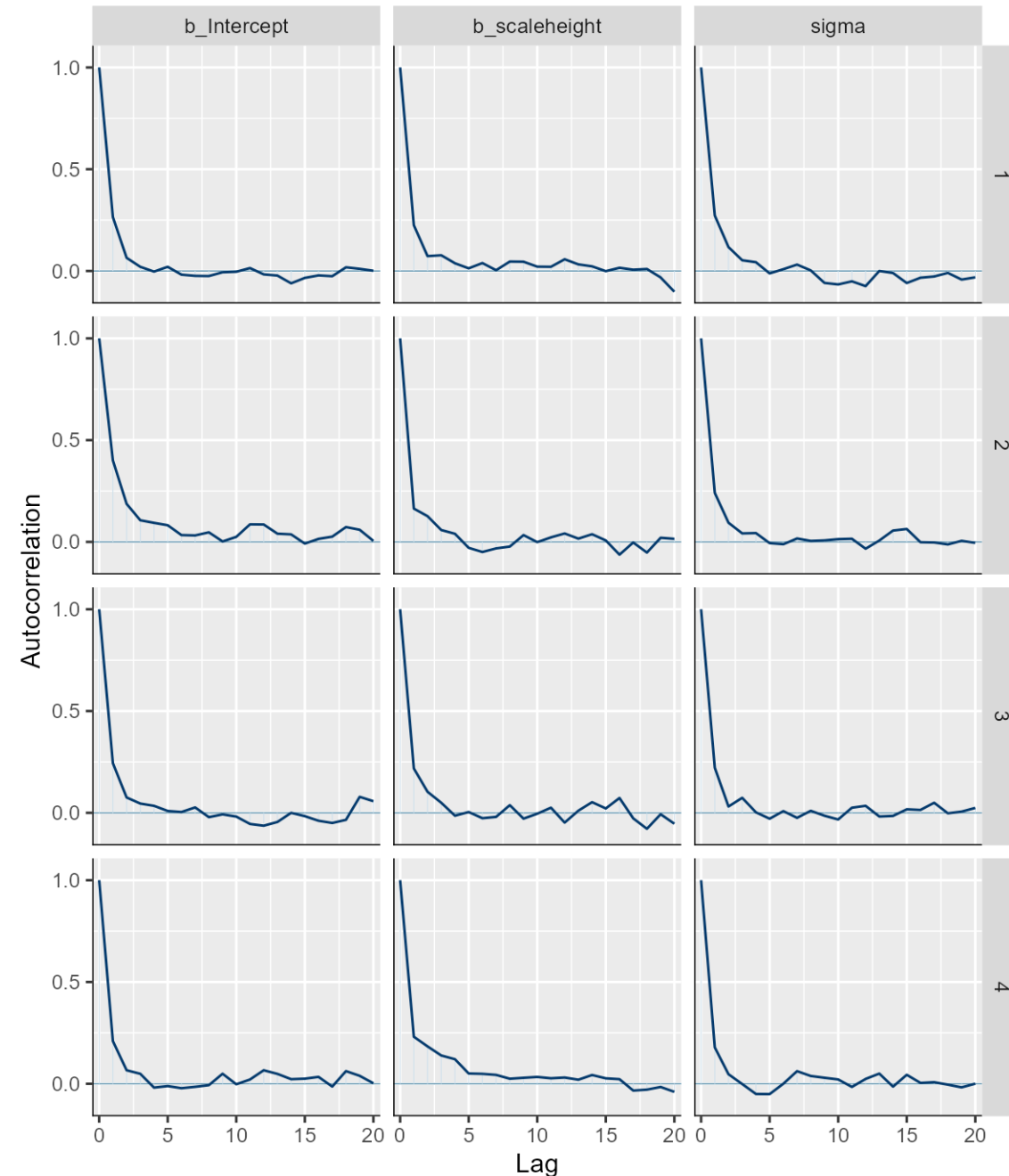
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	10.68	1.48	8.30	13.93	1.00	2418	2315

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS and Tail_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

Markov Chains

- Should I thin?
 - Thinning reduces posterior sample size
 - Can reduce / eliminate correlation in MCMC

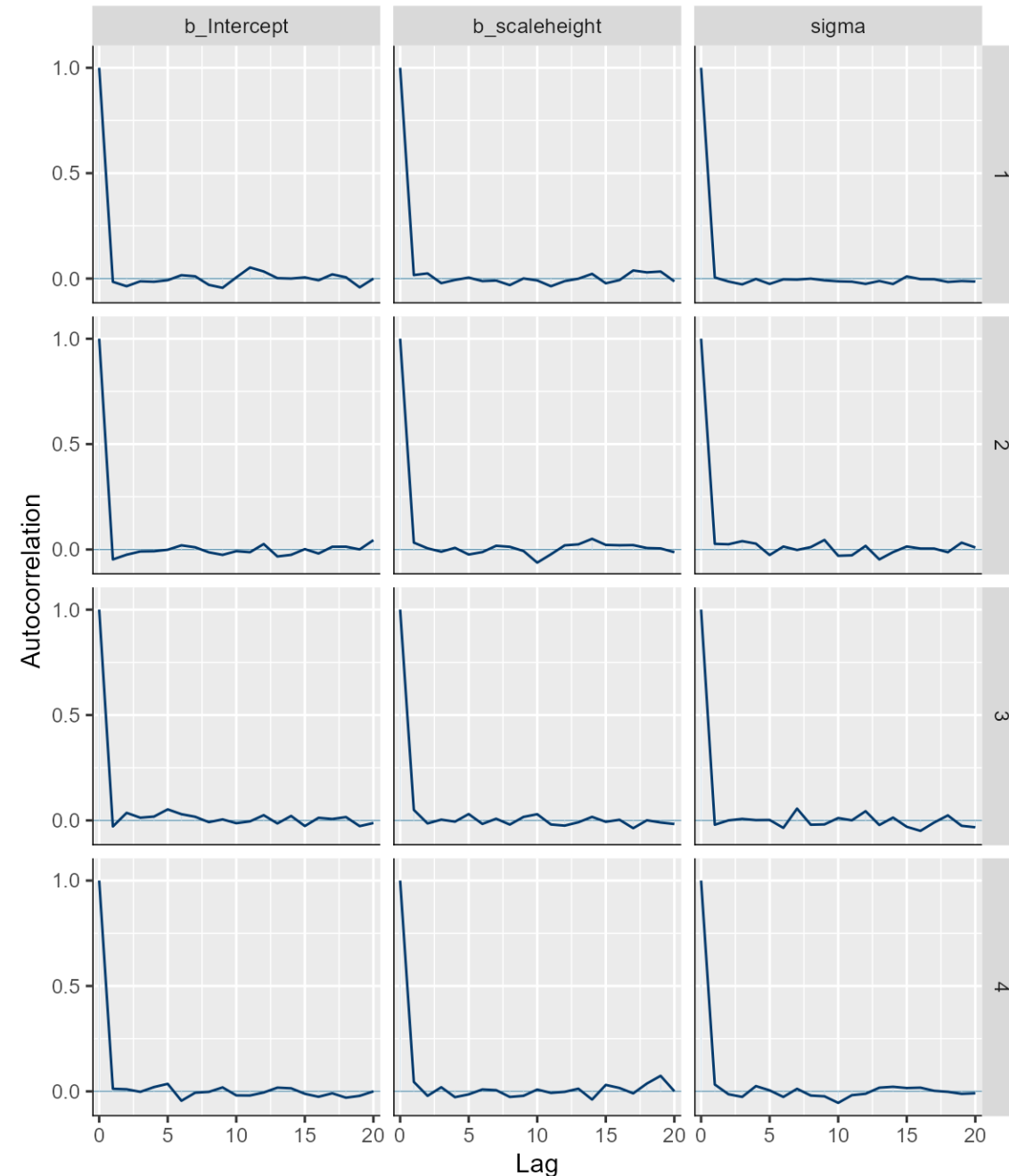
```
brm(weight ~ scale(height),  
     data = dat,  
     family = gaussian(),  
     chains = 4,  
     iter = 2000,  
     warmup = 1000,  
     thin = 1,  
     cores = 4)
```



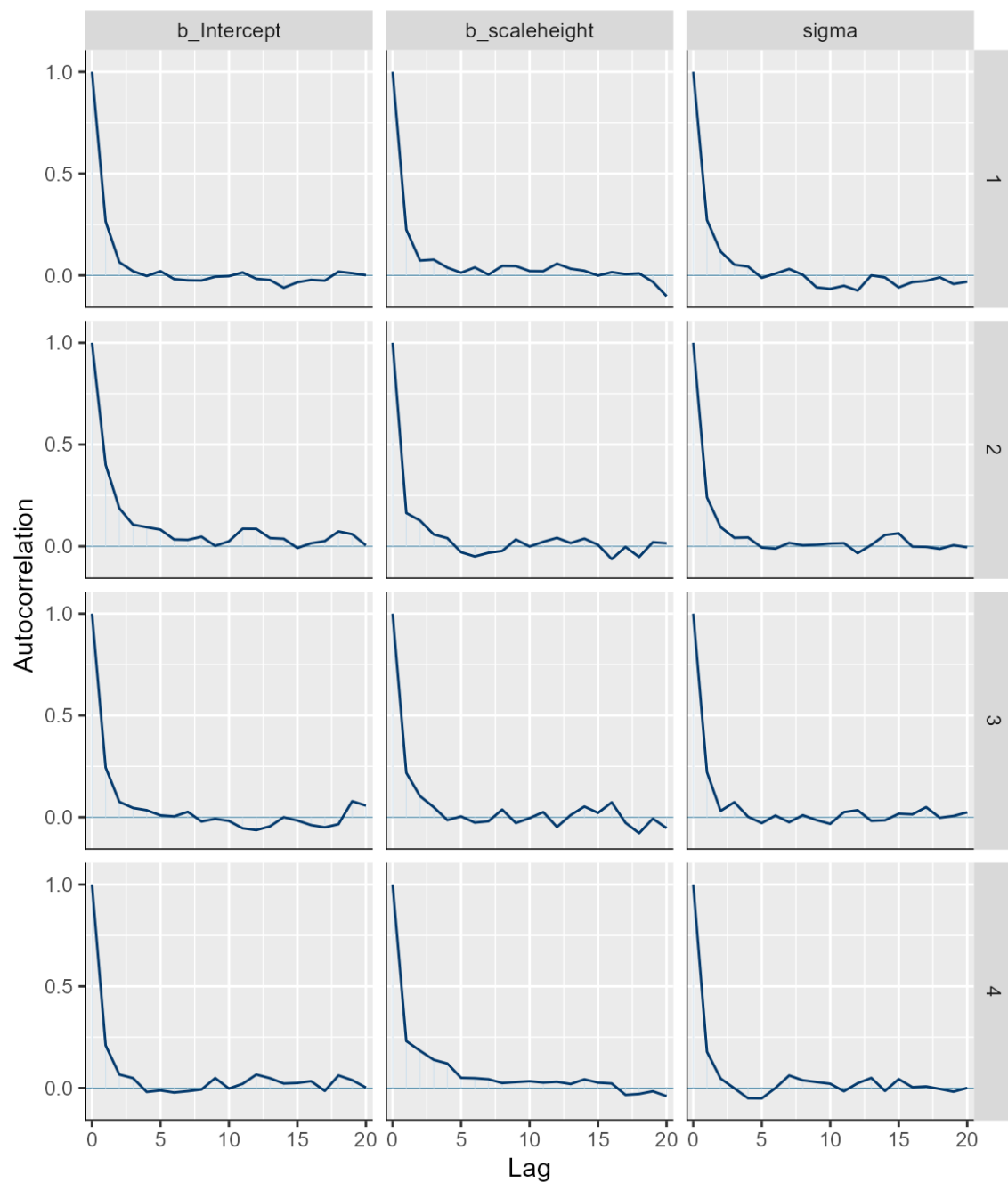
Markov Chains

- Should I thin?
 - Thinning reduces posterior sample size
 - Can reduce / eliminate correlation in MCMC

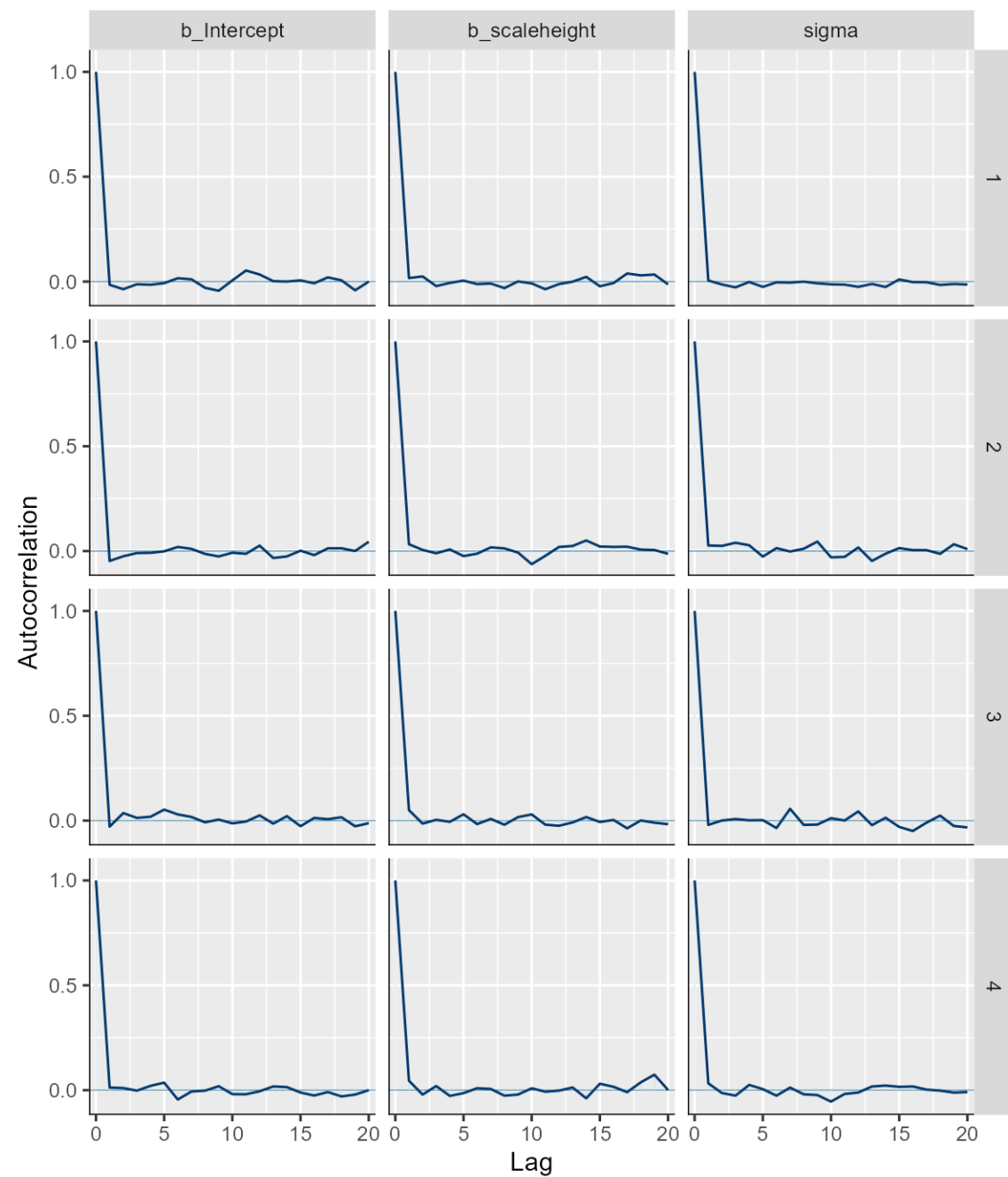
```
brm(weight ~ scale(height),  
     data = dat,  
     family = gaussian(),  
     chains = 4,  
     iter = 2000,  
     warmup = 1000,  
     thin = 5,  
     cores = 4)
```



thin = 1

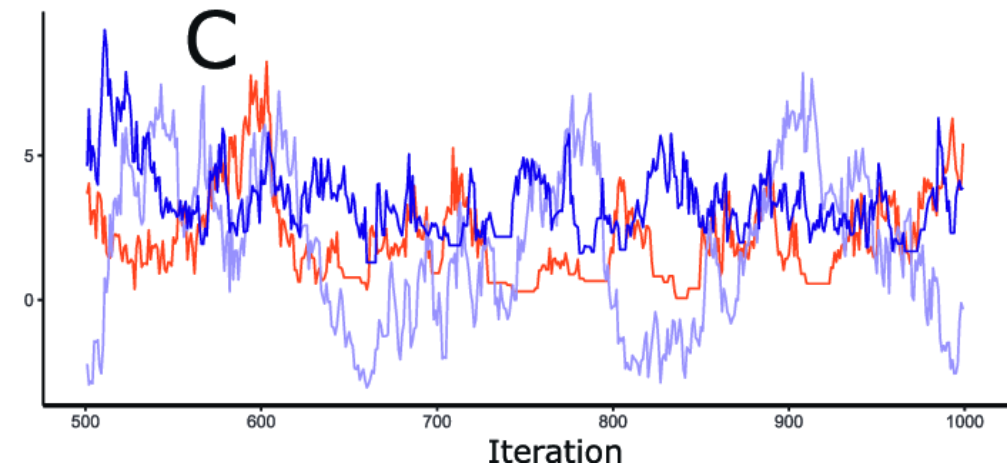
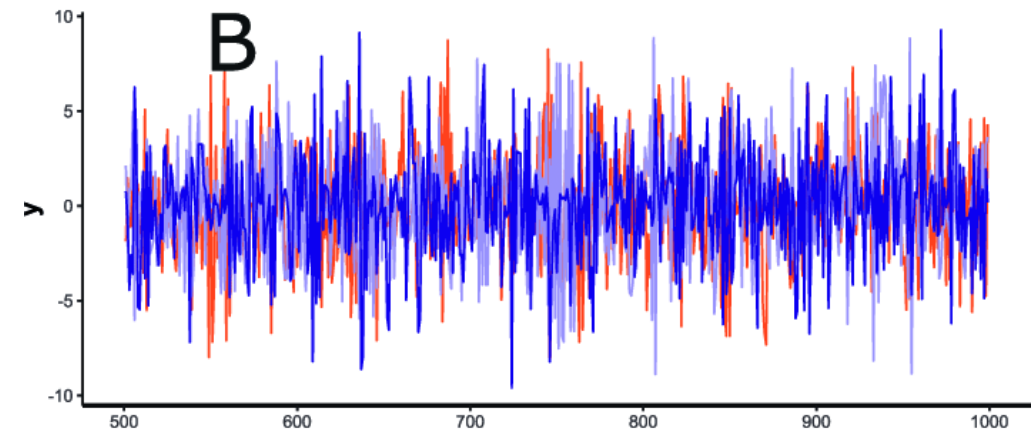
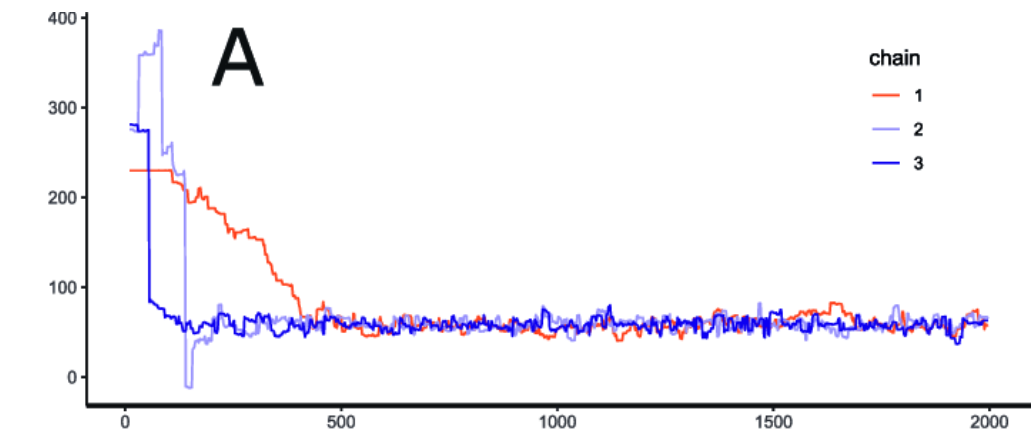


thin = 5



Markov Chains

- Model diagnostics of MCMC
 - Visual inspection
 - Convergence, mixing, stability
 - R-hat
 - Autocorrelation



```
> summary(m3)
```

```
Family: gaussian
```

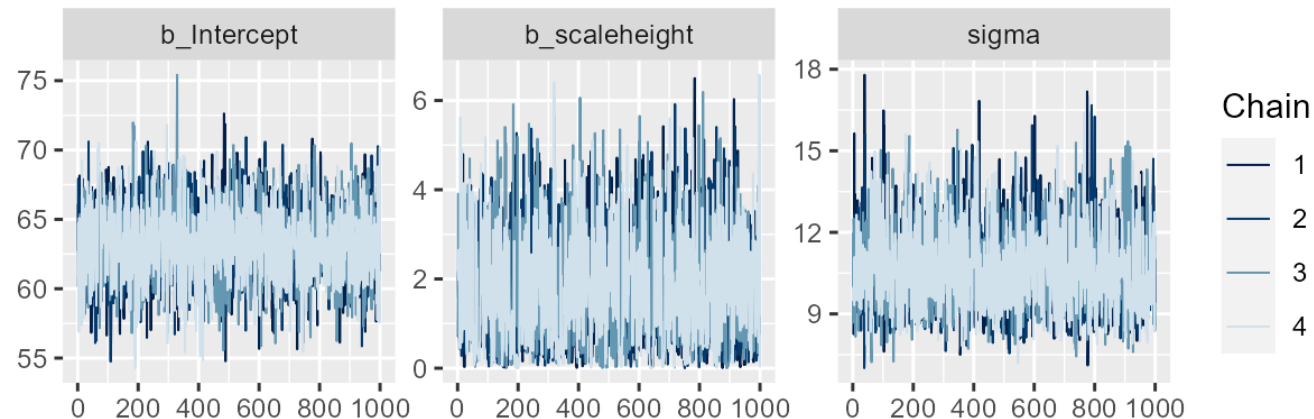
```
Links: mu = identity; sigma = identity
```

```
Formula: weight ~ scale(height)
```

```
Data: d (Number of observations: 15)
```

```
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
```

```
total post-warmup draws = 4000
```



Population-Level Effects:

	Estimate	Est.Error	1-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	63.28	2.69	58.14	68.60	1.00	2005	1766
scaleheight	1.72	1.21	0.06	4.51	1.00	1273	904

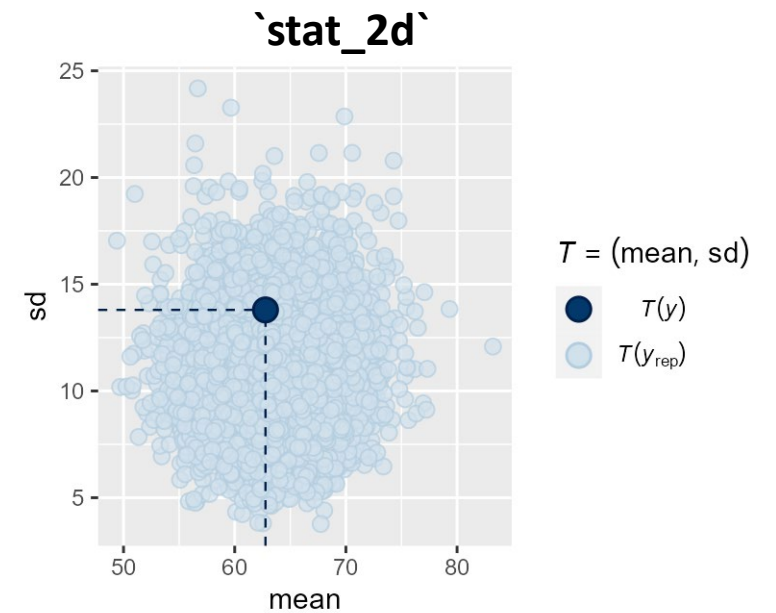
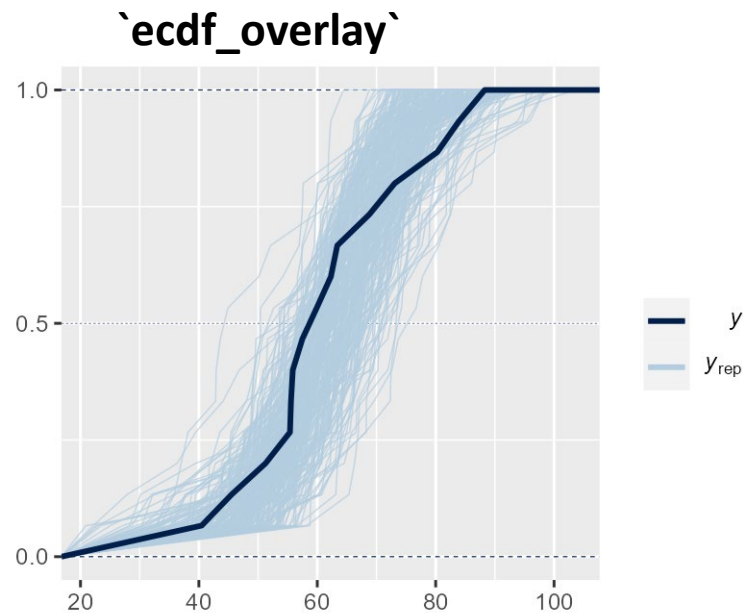
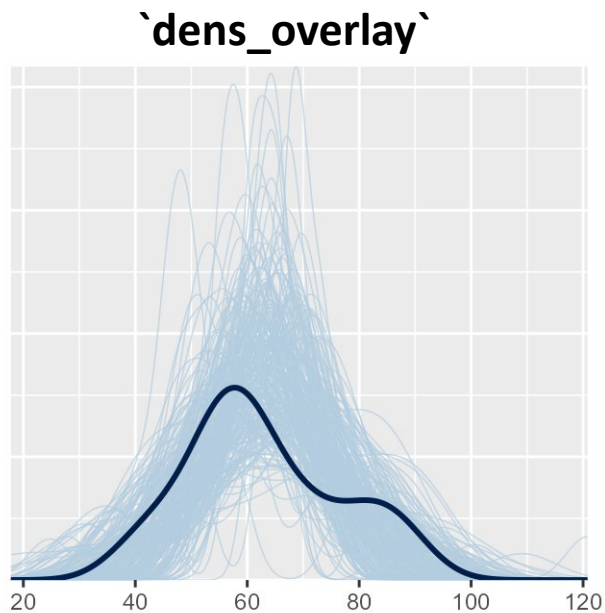
Family Specific Parameters:

	Estimate	Est.Error	1-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	10.68	1.48	8.30	13.93	1.00	2418	2315

Draws were sampled using `sampling(NUTS)`. For each parameter, `Bulk_ESS` and `Tail_ESS` are effective sample size measures, and `Rhat` is the potential scale reduction factor on split chains (at convergence, `Rhat` = 1).

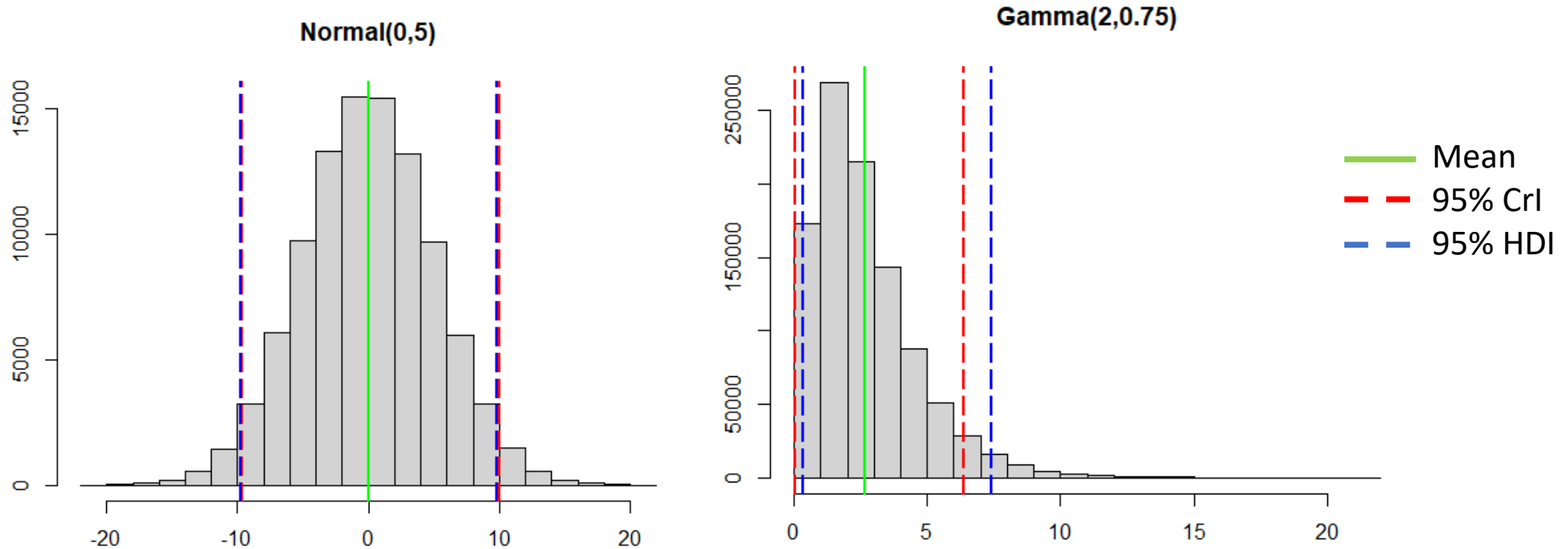
Posterior Predictive Checks

- Does the model generate values that match the data?
- Numerous options available through ``brms`` or ``bayesplot``



Model Summaries

- Summary stats
 - 95% Credible Interval (CrI)
 - Highest Density Interval (HDI)



Model Summaries

- ``bayestestR::describe_posterior``

weight ~ scale(height) [Informed priors]

Parameter	Median	95% CI	pd	ROPE	% in ROPE	Rhat	ESS
(Intercept)	63.17	[58.24, 68.33]	100%	[-1.38, 1.38]	0%	1.000	2462.00
scaleheight	1.63	[0.10, 4.54]	100%	[-1.38, 1.38]	41.16%	1.000	2118.00

weight ~ scale(height) [Uninformed priors]

Parameter	Median	95% CI	pd	ROPE	% in ROPE	Rhat	ESS
(Intercept)	62.64	[55.46, 69.95]	100%	[-1.38, 1.38]	0%	1.001	1290.00
scaleheight	3.21	[-5.18, 10.95]	78.20%	[-1.38, 1.38]	21.37%	1.000	1388.00

pd – Probability of direction = represents the certainty with which an effect goes in a particular direction

Probabilities from Posteriors

- ``bayestestR::describe_posterior``

Parameter	Median	95% CI	pd	ROPE	% in ROPE	Rhat	ESS
(Intercept)	62.66	[57.26, 68.05]	100%	[-1.38, 1.38]	0%	1.002	2390.00
scaleheight	1.63	[0.13, 4.68]	100%	[-1.38, 1.38]	41.95%	1.001	2663.00
sexM	1.79	[0.09, 5.86]	100%	[-1.38, 1.38]	38.97%	1.001	2770.00

Model Warnings

- <https://mc-stan.org/misc/warnings.html#when-can-warnings-be-ignored>
- Adjust control settings

From `brm` function documentation

- In addition to choosing the number of iterations, warmup draws, and chains, users can control the behavior of the NUTS sampler, by using the control argument. The most important reason to use control is to decrease (or eliminate at best) the number of divergent transitions that cause a bias in the obtained posterior draws. Whenever you see the warning "There were x divergent transitions after warmup." you should really think about increasing adapt_delta. To do this, write `control = list(adapt_delta = <x>)`, where <x> should usually be value between 0.8 (current default) and 1. Increasing adapt_delta will slow down the sampler but will decrease the number of divergent transitions threatening the validity of your posterior draws.
- Another problem arises when the depth of the tree being evaluated in each iteration is exceeded. This is less common than having divergent transitions, but may also bias the posterior draws. When it happens, Stan will throw out a warning suggesting to increase max_treedepth, which can be accomplished by writing `control = list(max_treedepth = <x>)` with a positive integer <x> that should usually be larger than the current default of 10. For more details on the control argument see stan.

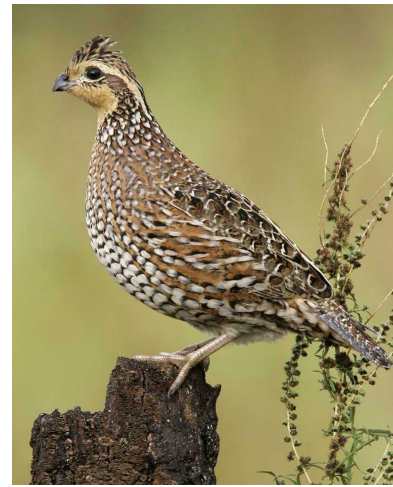
Model Warnings

- <https://mc-stan.org/misc/warnings.html#when-can-warnings-be-ignored>
- Adjust control settings

```
brm(weight ~ scale(height),  
    data = dat,  
    family = gaussian(),  
    chains = 4,  
    iter = 2000,  
    warmup = 1000,  
    thin = 1,  
    cores = 4,  
    control = list(adapt_delta = 0.9,  
                   max_treedepth = 20))
```

Applications

- Many applications for numerous purposes






esa

ECOSPHERE

METHODS, TOOLS, AND TECHNOLOGIES

An integrated population model to project viability of a northern bobwhite population in Ohio

CONNOR J. ROSENBLATT ^{1,†} ROBERT J. GATES,¹ STEPHEN N. MATTHEWS ¹
WILLIAM E. PETERMAN ¹ AND NATHAN J. STRICKER²

¹*School of Environment and Natural Resources, The Ohio State University, 2021 Coffey Road, 210 Kottman Hall, Columbus, Ohio 43210 USA*

²*Ohio Division of Wildlife, 2045 Morse Road, Columbus, Ohio 43229 USA*

Citation: Rosenblatt, C. J., R. J. Gates, S. N. Matthews, W. E. Peterman, and N. J. Stricker. 2021. An integrated population model to project viability of a northern bobwhite population in Ohio. *Ecosphere* 12(9):e03731. 10.1002/ecs2.3731

An integrated population model to project viability of a northern bobwhite population in Ohio

CONNOR J. ROSENBLATT^{1,†}, ROBERT J. GATES¹, STEPHEN N. MATTHEWS¹,
WILLIAM E. PETERMAN¹ AND NATHAN J. STRICKER²

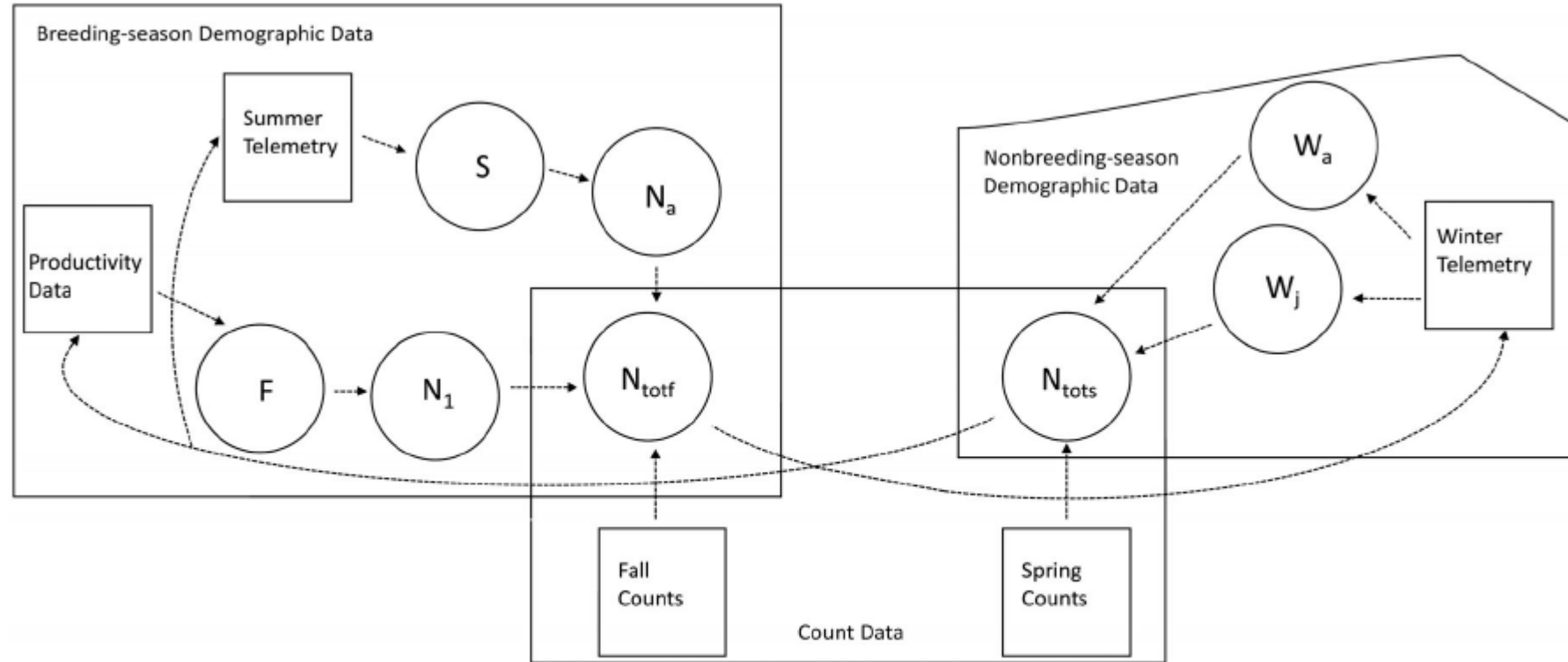


Fig. 2. Graphical representation for the integrated population model used for a northern bobwhite population in southwestern Ohio, USA, during 2007–2015. Squares represent raw data and circles represent parameters estimated from the model. Dashed arrows represent dependence between the nodes. See Table 1.

An integrated population model to project viability of a northern bobwhite population in Ohio

CONNOR J. ROSENBLATT^{1,†}, ROBERT J. GATES¹, STEPHEN N. MATTHEWS¹,
WILLIAM E. PETERMAN¹ AND NATHAN J. STRICKER²

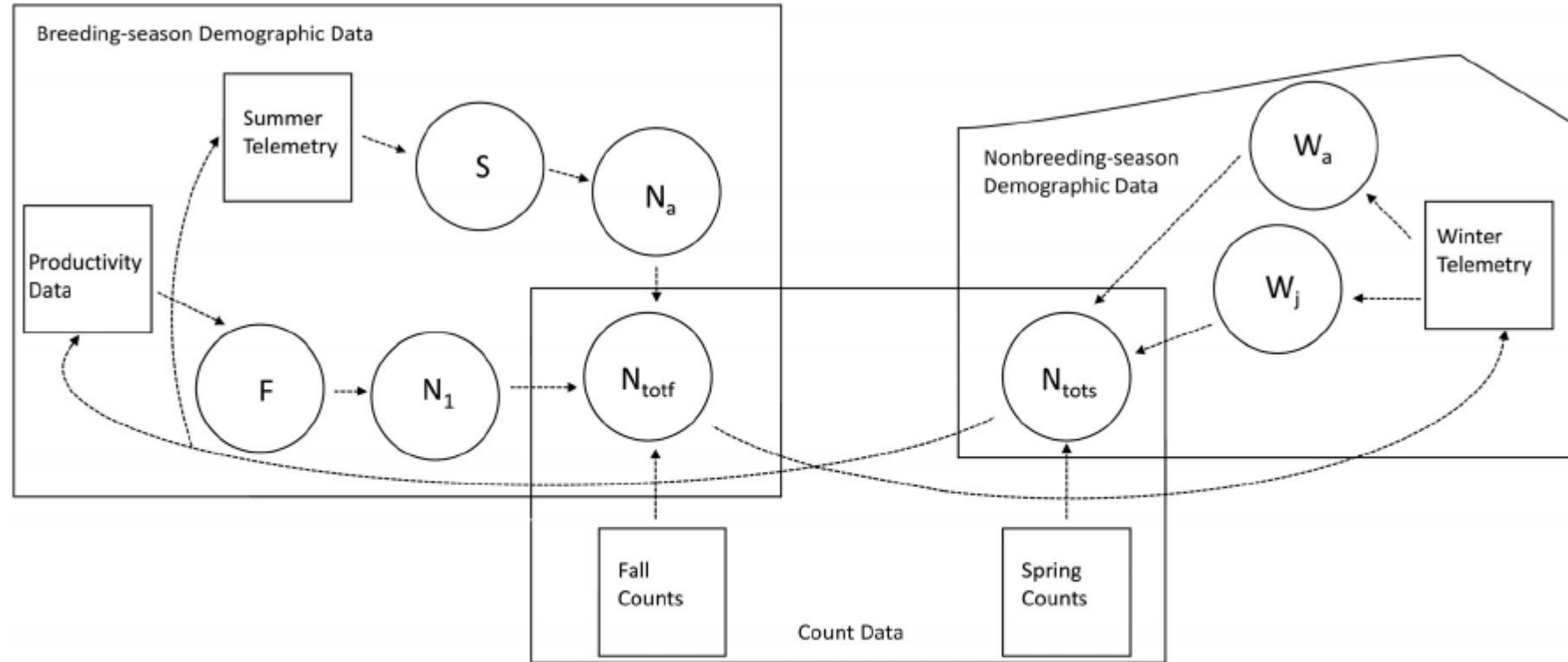


Fig. 2. Graphical representation for the integrated population model used for a northern bobwhite population in southwestern Ohio, USA, during 2007–2015. Squares represent raw data and circles represent parameters estimated from the model. Dashed arrows represent dependence between the nodes. See Table 1.

An integrated population model to project viability of a northern bobwhite population in Ohio

CONNOR J. ROSENBLATT^{1,†}, ROBERT J. GATES¹, STEPHEN N. MATTHEWS¹,
WILLIAM E. PETERMAN¹ AND NATHAN J. STRICKER²

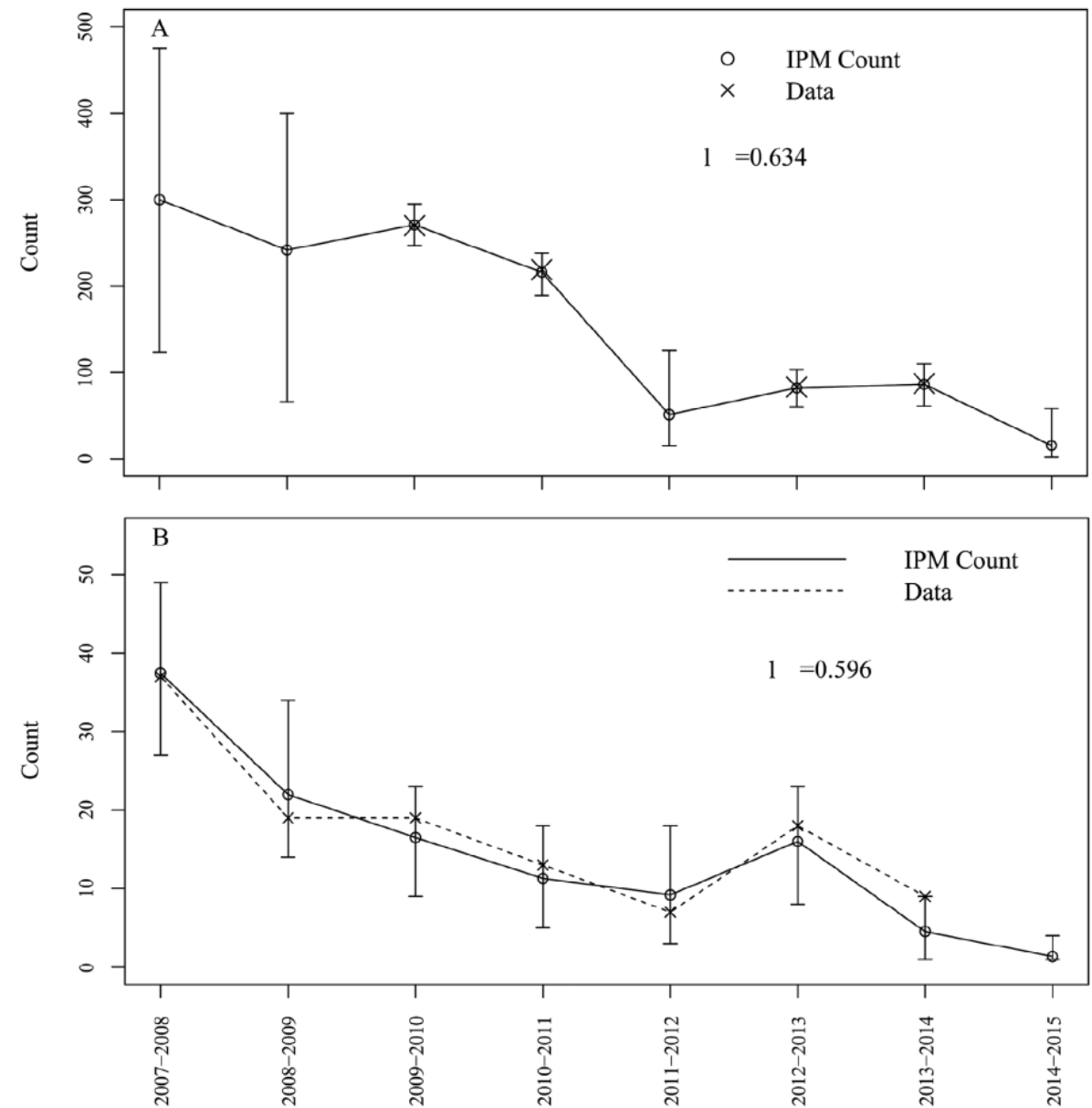


Fig. 4. Population trends during 2007–2015 for a northern bobwhite population in southwestern Ohio, USA, based on counts from the data sources and the predicted counts obtained from an integrated population model. Population trends are shown based on (A) fall counts and (B) spring counts. Vertical bars represent 95% credible intervals.

An integrated population model to project viability of a northern bobwhite population in Ohio

CONNOR J. ROSENBLATT^{1,†}, ROBERT J. GATES¹, STEPHEN N. MATTHEWS¹,
WILLIAM E. PETERMAN¹ AND NATHAN J. STRICKER²

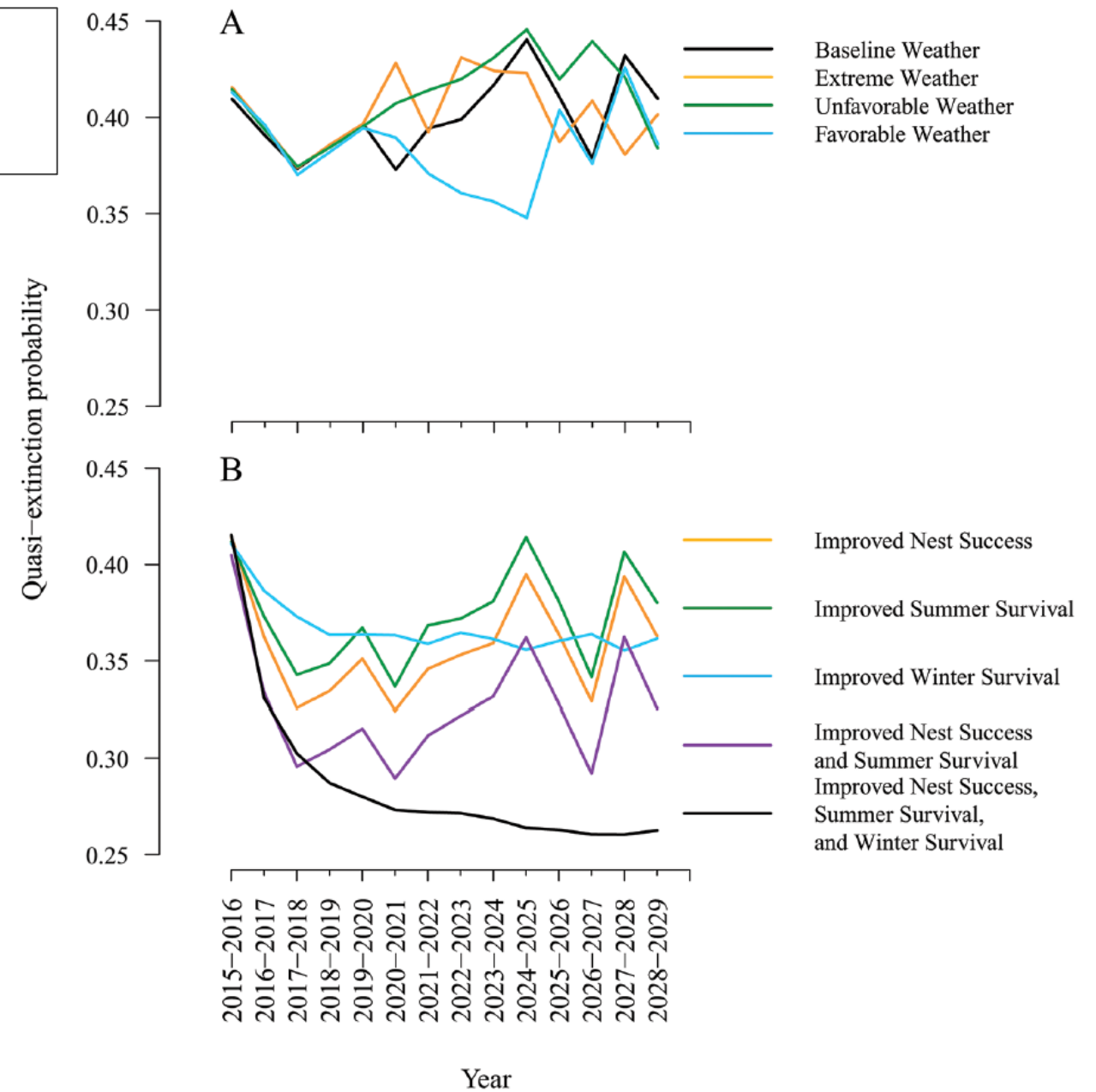


Fig. 6. Quasi-extinction probability over time derived from an integrated population for northern bobwhite in southwestern Ohio, USA, based on fall counts during 2019–2029 under (A) different weather scenarios and (B) different population viability analysis scenarios.

Workshop Materials

- Open RStudio
- Download `usethis` package
- Then execute:

```
usethis::use_course(  
  'https://github.com/wpeterman/workshop_materials/archive/refs/heads/master.zip',  
  destdir = "YOUR DOWNLOAD DIRECTORY")
```