

Introduction to hierarchical (mixed-effects) models

FW 891

Christopher Cahill

25 September 2023



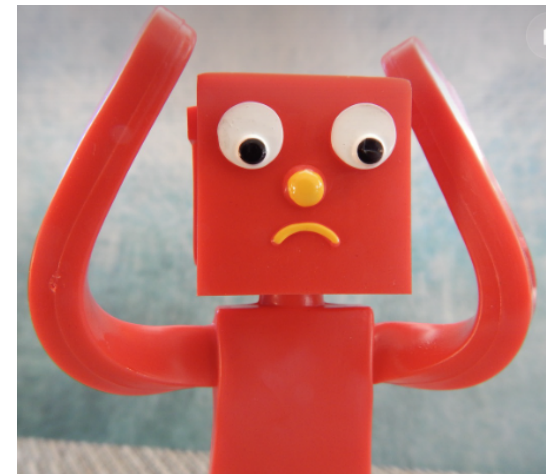
Quantitative Fisheries Center
MICHIGAN STATE UNIVERSITY

Purpose

- Introduce an important extension of all previous models
 - Will have several lectures on mixed effects models
- An example
 - both frequentist and Bayesian
 - Explain in words and algebra what mixed effects models are
- Pros and cons of mixed effects modeling
- Why these models likely deserve to be the default approach to many problems

The polynonomous model (miser)

- These models are known by many names:
 - Mixed effects models (fixed effects + random effects)
 - Multilevel models
 - Hierarchical models
 - Partial-pooling models
- Our example model might also be called:
 - Random-intercepts model



An example to get things started



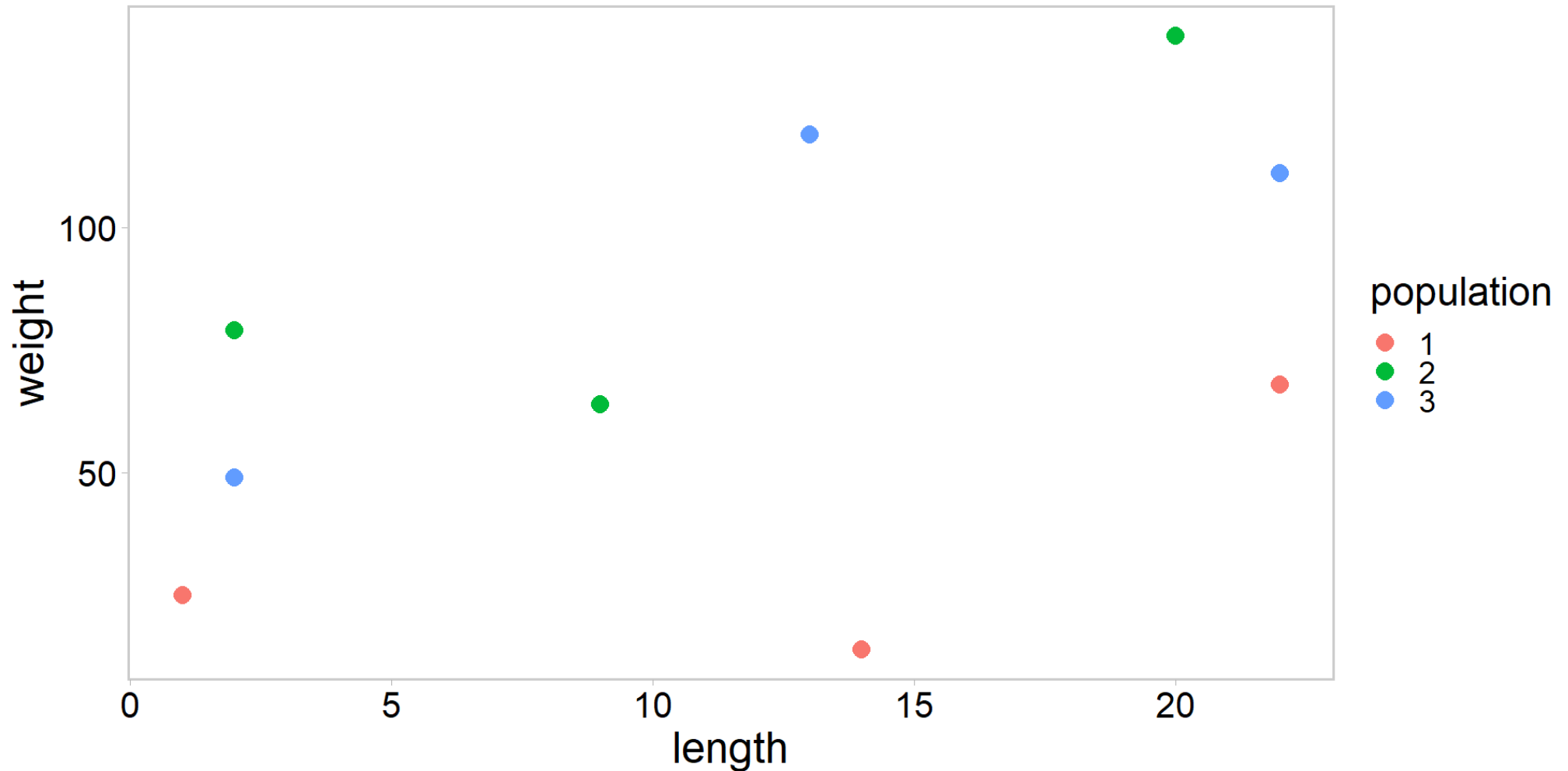
image credit

An example to get things started

- Revisiting the ANCOVA example, but let's change the variable names
 - Plot the weight of emperor scorpions as a function of length

```
1 library(tidyverse)
2 library(ggqfc)
3 weight <- c(25, 14, 68, 79, 64, 139, 49, 119, 111) # obs
4 population <- factor(c(1, 1, 1, 2, 2, 2, 3, 3, 3)) # group
5 length <- c(1, 14, 22, 2, 9, 20, 2, 13, 22) # covariate
6 my_df <- data.frame(weight, population, length)
7
8 my_df %>%
9   ggplot(aes(length, weight, color = population)) +
10   geom_point(pch = 16, size = 3.5) + theme_qfc()
```

An example to get things started



- What does this plot suggest?

An example to get things started

- Linear relationship between weight and length
 - Perhaps with a different intercept for each population?
 - Regression model should account for population differences
- The simplest way to account for population differences:

$$\text{weight}_i = \alpha_{j(i)} + \beta \cdot \text{length}_i + \varepsilon_i$$

$$\varepsilon_i \sim \text{Normal}(0, \sigma^2)$$

- Weight of scorpion i depends on length in a linear way, where each population j has a different intercept, and residuals come from a zero-mean normal distribution

Motivating random effects

- Independent intercepts for each population represents one assumption that we might make about the data
- **Assumption 1: No Pooling**: these are the only three populations we are interested in (no pooling; estimate each α_j separately)

Motivating random effects

- Independent intercepts for each population represents one assumption that we might make about the data
- **Assumption 1: No Pooling**: these are the only three populations we are interested in (no pooling; estimate each α_j separately)
- **Assumption 2: Complete Pooling**: we think these actually represent one population (complete pooling; estimate one α across all three populations)

Motivating random effects

- Independent intercepts for each population represents one assumption that we might make about the data
- **Assumption 1: No Pooling**: these are the only three populations we are interested in (no pooling; estimate each α_j separately)
- **Assumption 2: Complete Pooling**: we think these actually represent one population (complete pooling; estimate one α across all three populations)
- **Assumption 3: Partial Pooling**: these populations merely represent a sample from a larger number of scorpion populations that we could have studied and we want our conclusions to generalize to this larger universe of statistical populations (hierarchical, partial pooling; model parameters are viewed as a sample from a population distribution)

Translating assumption 1 to math

- No pooling:

$$\text{weight}_i = \alpha_{j(i)} + \beta \cdot \text{length}_i + \varepsilon_i$$

$$\varepsilon_i \sim \text{Normal}(0, \sigma^2)$$

Translating assumption 2 to math

- Complete pooling:

$$\text{weight}_i = \alpha + \beta \cdot \text{length}_i + \varepsilon_i$$

$$\varepsilon_i \sim \text{Normal}(0, \sigma^2)$$

- This is literally just a linear regression with one intercept and one slope

Translating assumption 3 to math

- Hierarchical, partial pooling:

$$\text{weight}_i = \alpha_{j(i)} + \beta \cdot \text{length}_i + \varepsilon_i$$

$$\varepsilon_i \sim \text{Normal}(0, \sigma^2)$$

$$\alpha_j \sim \text{Normal}(\mu_\alpha, \sigma_\alpha^2). \quad \text{This line makes } \alpha_j \text{ random!}$$

- Third equation is the *only* difference between a model that treats α_j as a fixed effect (assumption 1) and one where α_j is a random effect (assumption 3)

Translating assumptions 1-3 to R code (Frequentist)

```
1 # ass. 1 - no pooling
2 fit_no <- lm(my_df$weight ~ my_df$population - 1 + my_df$length)
3
4 # ass. 2 - complete pooling:
5 fit_complete <- lm(my_df$weight ~ 1 + my_df$length)
6
7 # ass. 3 - partial pooling:
8 library(lme4)
9 fit_partial <- lmer(my_df$weight ~ 1 + my_df$length + (1|my_df$population))
```

- Don't worry about the computational details underlying how `lmer()` is estimating the random effects

Compare the models (Frequentist)

```
1 AIC(fit_no)
```

```
[1] 88.22643
```

```
1 AIC(fit_complete)
```

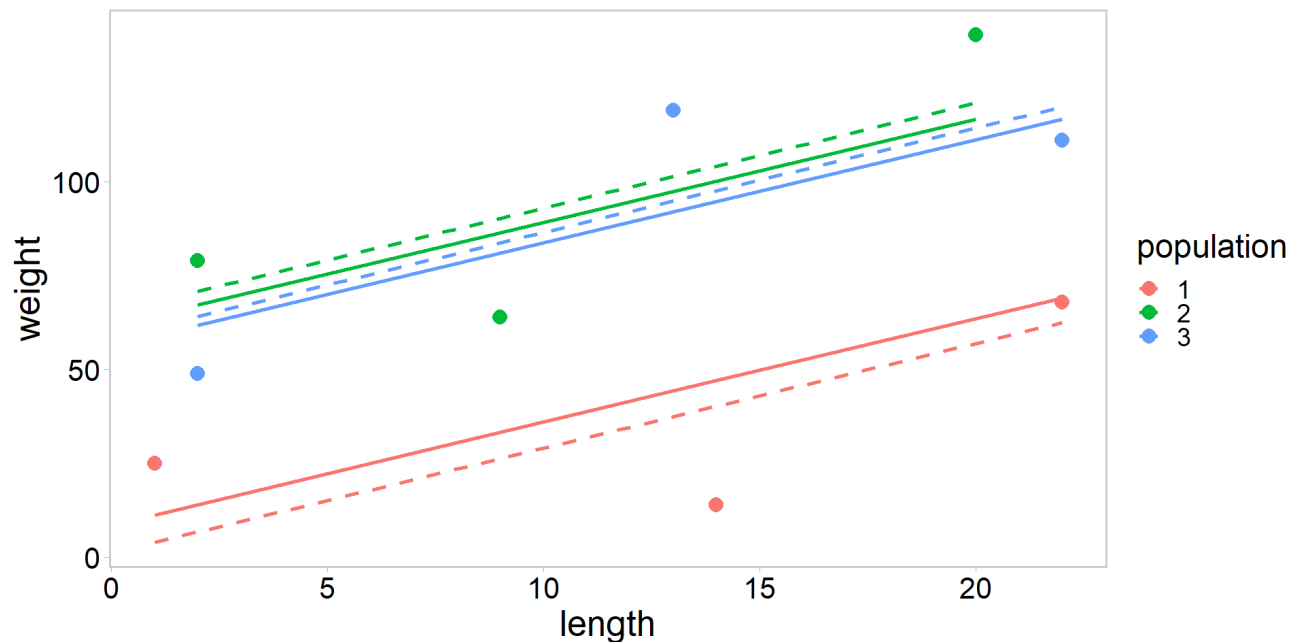
```
[1] 95.11368
```

```
1 AIC(fit_partial)
```

```
[1] 85.08788
```

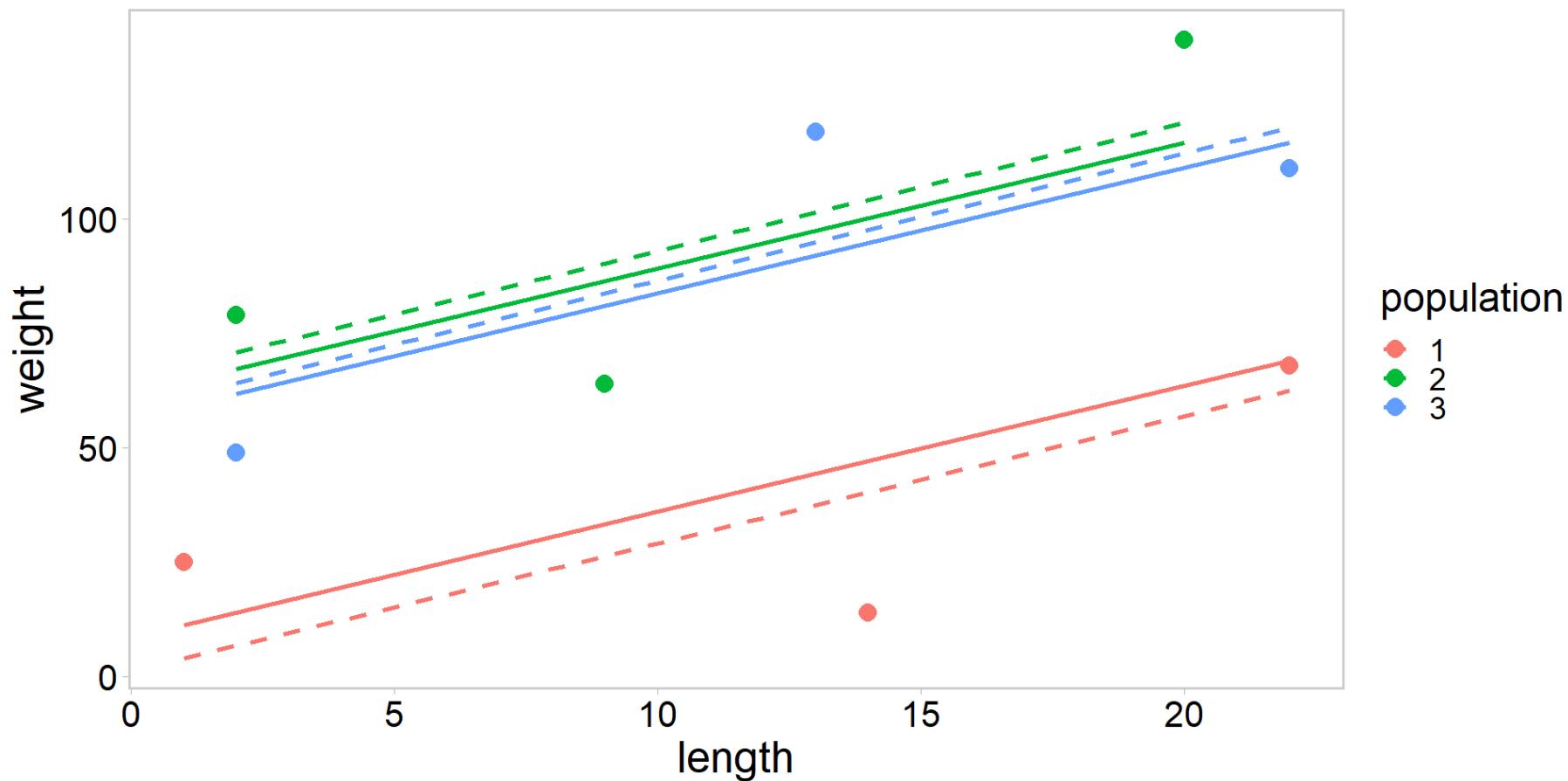
Plot no pooling vs. partial pooling

```
1 my_df$pred_no <- predict(fit_no, newdata = my_df)
2 my_df$pred_partial <- predict(fit_partial, newdata = my_df, level = 0)
3 my_df %>%
4   ggplot(aes(length, weight, color = population)) +
5   geom_point(pch = 16, size = 3.5) +
6   geom_line(data = my_df, aes(y = pred_no), size = 1, lty = 2) +
7   geom_line(data = my_df, aes(y = pred_partial), size = 1) +
8   theme_qfc()
```



What is going on here?

- Solid lines are from mixed-effects model with partial pooling, dashed lines are from ANCOVA with no pooling



Another example of shrinkage

- county-level radon levels

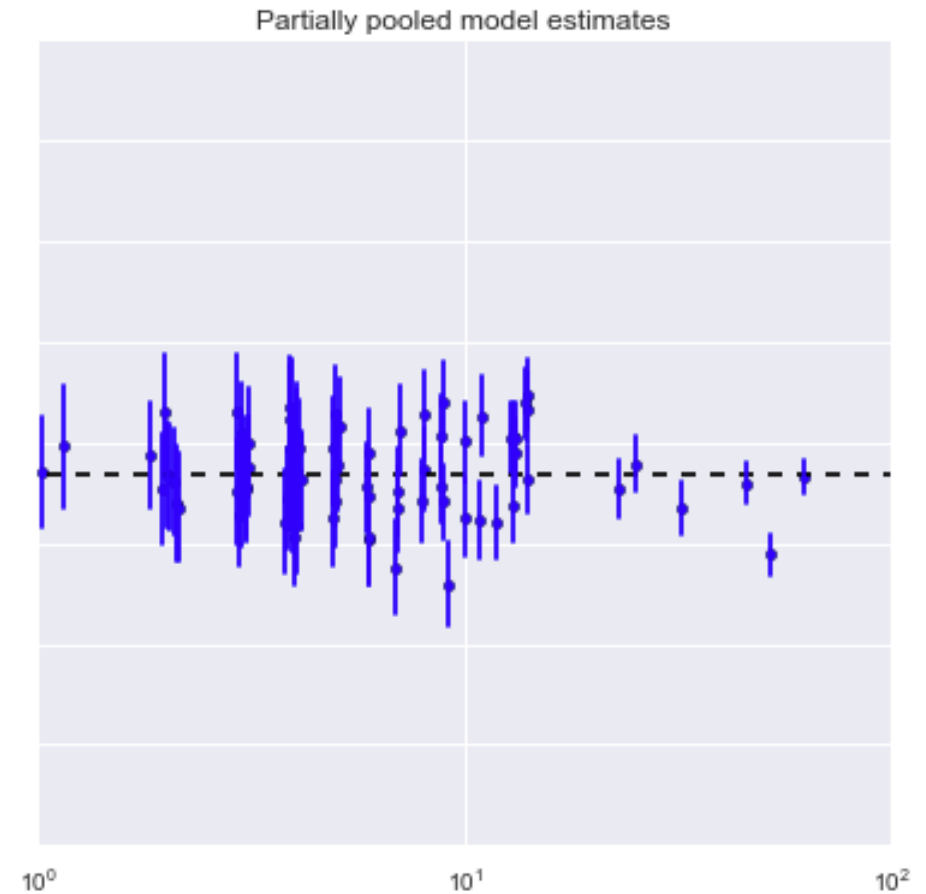
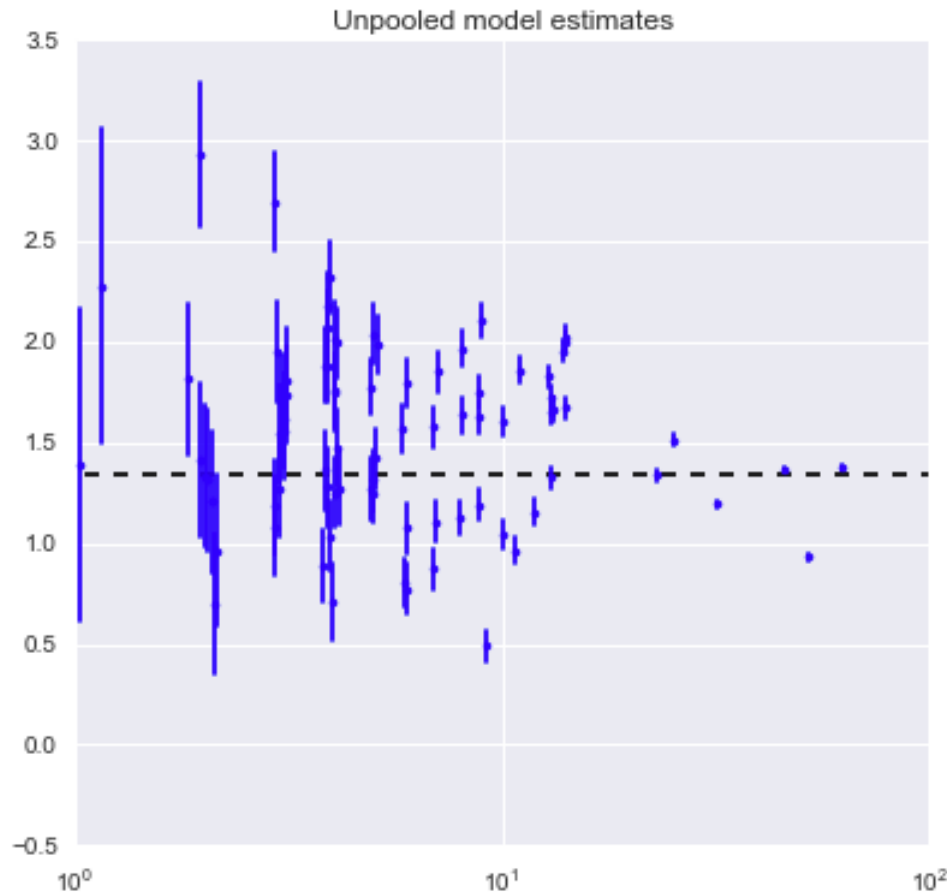


image credit

[The page contains dense, illegible handwritten text.]

Math for the random intercepts model

- Hierarchical, partial pooling:

$$\text{weight}_i = \alpha_{j(i)} + \beta \cdot \text{length}_i + \varepsilon_i$$

$$\varepsilon_i \sim \text{Normal}(0, \sigma^2)$$

$$\alpha_j \sim \text{Normal}(\mu_\alpha, \sigma_\alpha^2). \quad \text{This line makes } \alpha_j \text{ random!}$$

- Third equation is the *only* difference between a model that treats α_j as a fixed effect (**assumption 1**) and one where α_j is a random effect (**assumption 3**)

Stan code for the random intercepts model

```
1 data {  
2   int<lower=0> n;  
3   vector[n] weights;  
4   array[n] int population;  
5   vector[n] lengths;  
6 }  
7 parameters {  
8   real mu_alpha;  
9   vector[3] alpha_j;  
10  real<lower=0> sd_alpha;  
11  real b1;  
12  real<lower=0> sd_obs;  
13 }  
14 // more code below
```


Stan code for the random intercepts model

```
1 //... see code on previous slide
2 transformed parameters {
3   vector[3] alpha_pop;
4   alpha_pop = mu_alpha + alpha_j; // group int is global int + random effect
5 }
6 model {
7   vector[n] w_preds;
8   mu_alpha ~ normal(50, 15);           // hyper prior
9   sd_alpha ~ normal(0, 100);           // hyper prior
10  alpha_j ~ normal(mu_alpha, sd_alpha); // hyper distribution
11  b1 ~ normal(0, 10);                   // slope
12  sd_obs ~ normal(0, 25);               // likelihood error term
13  for(i in 1:n){
14    w_preds[i] = alpha_pop[population[i]] + b1*lengths[i];
15  }
16  weights ~ normal(w_preds, sd_obs);    // likelihood
17 }
```

Compiling the Stan model

```
1 library(cmdstanr)
2 mod <- cmdstan_model("src/random_ints.stan")
3 stan_data <- list(
4   n = length(weight),
5   weights = weight,
6   population = population,
7   lengths = length
8 )
9
10 inits <- function() {
11   list(
12     mu_alpha = 50,
13     sd_alpha = 30,
14     alpha_j = rep(0, 3),
15     b1 = 10,
16     sd_obs = 10
17   )
18 }
```

Running that hawg 🐷 🐷 🐷

```
1 fit <- mod$sample(  
2   data = stan_data,  
3   init = inits,  
4   seed = 1,  
5   chains = 4,  
6   iter_warmup = 1000,  
7   iter_sampling = 1000,  
8   parallel_chains = 4,  
9   refresh = 0,  
10  adapt_delta = 0.9999, # look here  
11  step_size = 1e-3      # look here  
12 )
```

Running MCMC with 4 parallel chains...

Chain 4 finished in 0.5 seconds.

Chain 2 finished in 0.6 seconds.

Chain 1 finished in 0.7 seconds.

Chain 3 finished in 0.8 seconds.

All 4 chains finished successfully.

Mean chain execution time: 0.6 seconds.

Total execution time: 0.9 seconds.

Check the diagnostics (quick and dirty)

```
1 fit$cmdstan_diagnose()
```

```
Processing csv files: C:/Users/Chris/AppData/Local/Temp/RtmpOi11G0/random_ints-202309040845-1-282d3e.csv, C:/Users/Chris/AppData/Local/Temp/RtmpOi11G0/random_ints-202309040845-2-282d3e.csv, C:/Users/Chris/AppData/Local/Temp/RtmpOi11G0/random_ints-202309040845-3-282d3e.csv, C:/Users/Chris/AppData/Local/Temp/RtmpOi11G0/random_ints-202309040845-4-282d3e.csv
```

```
Checking sampler transitions treedepth.  
Treedepth satisfactory for all transitions.
```

```
Checking sampler transitions for divergences.  
No divergent transitions found.
```

```
Checking E-BFMI - sampler transitions HMC potential energy.  
E-BFMI satisfactory.
```

```
Effective sample size satisfactory.
```

```
Split R-hat values satisfactory all parameters.
```

```
Processing complete, no problems detected.
```

Pluck out the posterior summaries and compare to `lmer()`

```
1 fit$summary("alpha_pop") # Bayesian
```

```
# A tibble: 3 × 10
```

	variable	mean	median	sd	mad	q5	q95	rhat	ess_bulk	ess_tail
	<chr>	<num>	<num>	<num>	<num>	<num>	<num>	<num>	<num>	<num>
1	alpha_pop[1]	9.75	8.79	22.9	21.3	-25.3	49.3	1.00	1284.	1097.
2	alpha_pop[2]	61.5	62.3	20.3	18.7	28.3	92.4	1.00	1627.	1153.
3	alpha_pop[3]	56.5	56.8	21.5	19.8	22.2	90.4	1.00	1930.	1829.

```
1 coef(fit_partial)$`my_df$population`[,1] # frequentist
```

```
[1] 8.650009 61.667157 56.276072
```

- Pretty similar estimates (small sample size)
- You have now fitted a Bayesian hierarchical model

Pluck out the posterior summaries and compare to `lmer()`

```
1 fit$summary("b1") # Bayesian
```

```
# A tibble: 1 × 10
```

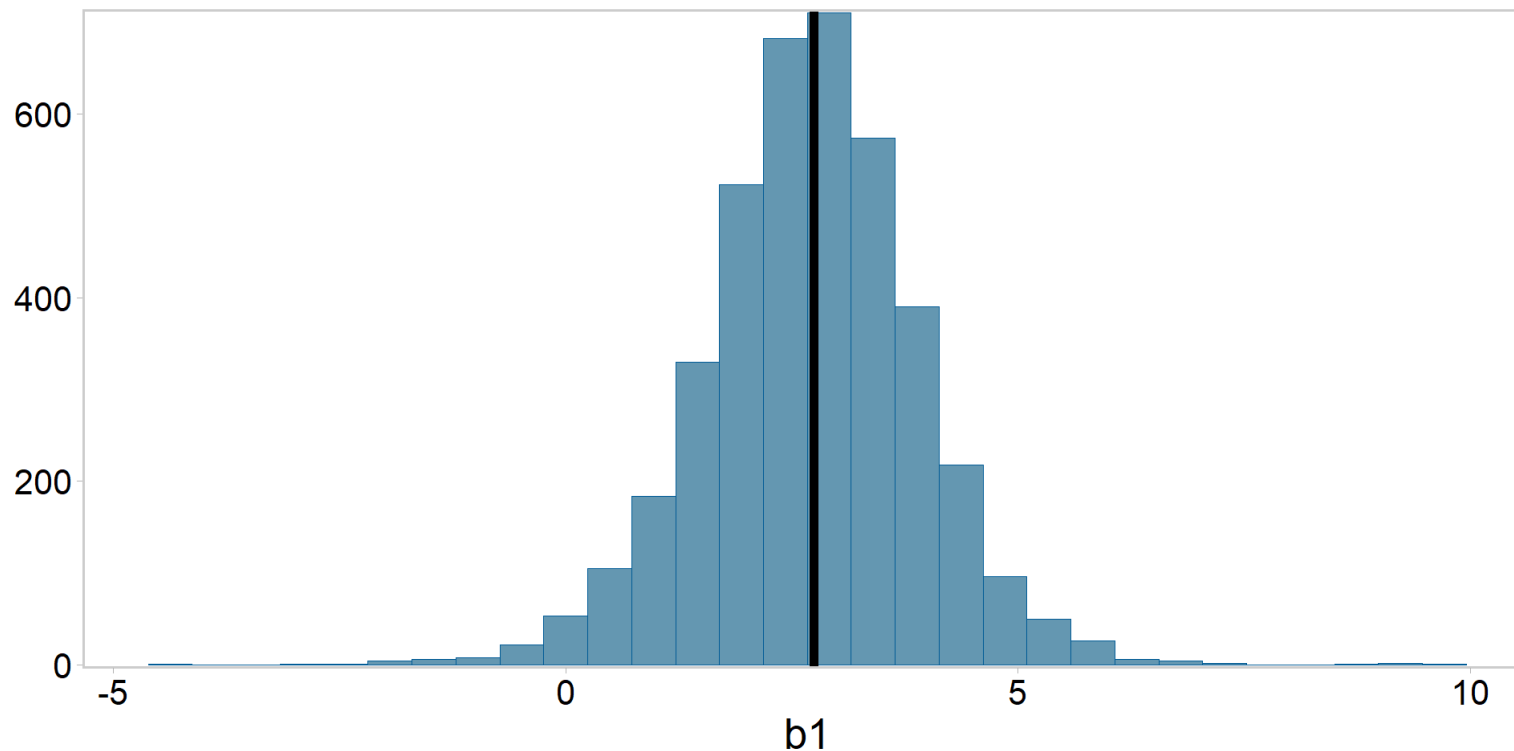
	variable	mean	median	sd	mad	q5	q95	rhat	ess_bulk	ess_tail
	<chr>	<num>	<num>	<num>	<num>	<num>	<num>	<num>	<num>	<num>
1	b1	2.71	2.73	1.19	1.07	0.722	4.58	1.00	1400.	1321.

```
1 fixef(fit_partial)[2] # frequentist
```

```
my_df$length  
2.744955
```

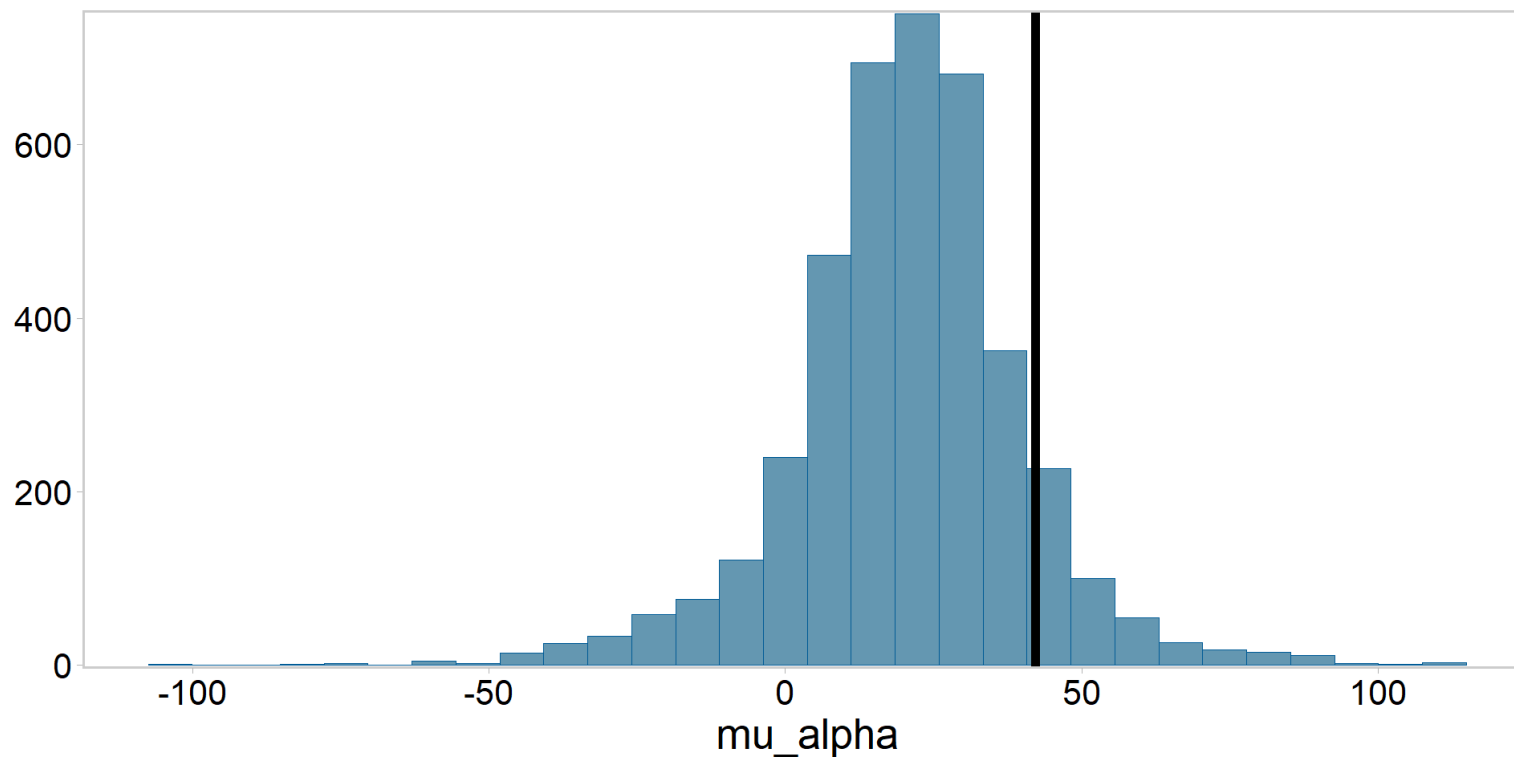
Posterior for β_1 vs. frequentist estimate

```
1 library(bayesplot)
2 posterior <- fit$draws(format = "df")
3 color_scheme_set("blue")
4 p <- mcmc_hist(posterior, par = "b1")
5 p + geom_vline(xintercept = fixef(fit_partial)[2], lwd = 2) + theme_qfc()
```



Posterior for μ_α vs. frequentist estimate

```
1 library(bayesplot)
2 posterior <- fit$draws(format = "df")
3 color_scheme_set("blue")
4 p <- mcmc_hist(posterior, par = "mu_alpha")
5 p + geom_vline(xintercept = fixef(fit_partial)[1], lwd = 2) + theme_qfc()
```



This was meant to be an 'easy' example for a reason

- Probably too few groups (usually want 5-10)
 - Obviously we can do it, but likely means our priors are more influential
- I had to fiddle with this a bit to get it to behave
 - Also low sample size within groups
- This likely explains at least some of the discrepancy among methods

What is a random effect (words)

- Two or more effects or parameters that “belong together” in some way
 - e.g., originating from some common distribution
 - sometimes called “latent variables,” may not be observable
- Bayesians put prior probabilities on all unknown quantities, MCMC
- Frequentists remove random effects from the model via numerical integration
- Exchangeability as a key assumption
 - Ordering of the random effects doesn't matter

The distribution of random effects

- Called the hyperdistribution
- The parameters of the random effects distribution are called hyperparameters
 - The priors of the hyperparameters are called hyperpriors
- No rule against more than one level (i.e., we could have hyper-hyperparameters etc.)
- Key point is to note the hierarchy of effects
- Typically random effects are continuous in ecology
 - however see occupancy models or binomial mixture models

Benefits of hierarchical modeling

- Sounds dope, also scares a lot of people
- *de facto* standard for modern ecological analyses
- Helps you draw (statistical) universe-level inferences if you have a good experimental design
- Borrowing information via a hyperprior
 - Robin-Hooding (see Punt et al. 2011)
- Helps safeguard against overfitting

Some reasons we might use mixed effects models

- To adjust for repeat sampling
- To adjust estimates for imbalanced sampling
- To study variation among groups
- To partition variation among groups
- Borrow information

Some reasons we might use mixed effects models

- Avoiding pseudoreplication
- Improving scope of inference
- To avoid averaging
- Improve understanding
 - Many common classes of models can be reformulated as hierarchical models
- McElreath (2023) argues mixed effects methods should be the default for many modeling problems

Drawbacks of hierarchical modeling?



Why treat any parameter as fixed?

- Treating a factor with very few levels as random results in very imprecise estimates of the hyperparameter
 - Rarely treat factors with fewer than 5-10 levels as random (however see Gelman 2005)
- Assumption of **exchangeability** may not hold, particularly if groups differ in some systematic way

Hierarchical modeling as a philosophical middle ground

- Berliner (1966) offered a definition of hierarchical models based on sub-models and inference from a joint probability distribution:

$$[data|process, parameters][process|parameters][parameters]$$

- observation model component
- process model component
- assumptions about parameters
- A key point:

Hierarchical modeling as a philosophical middle ground

- Berliner (1966) offered a definition of hierarchical models based on sub-models and inference from a joint probability distribution:

$$[data|process, parameters][process|parameters][parameters]$$

- observation model component
- process model component
- assumptions about parameters
- A key point:
 - Hierarchical models admit observation error, conditioned on some (underlying) ecological process of interest and your prior assumptions

Hierarchical modeling as a philosophical middle ground

- Berliner (1966) offered a definition of hierarchical models based on sub-models and inference from a joint probability distribution:

$$[data|process, parameters][process|parameters][parameters]$$

- observation model component
- process model component
- assumptions about parameters
- A key point:
 - Hierarchical models admit observation error, conditioned on some (underlying) ecological process of interest and your prior assumptions
- Whenever possible, prefer a hierarchical model based on an explicit ecological process

Summary and outlook

- Many names for the same general type of model
- Shrinkage as a key concept, which is related to the phrase “partial pooling”
- The difference between fixed effects and random effects, at least for this simple example
- A powerful framework for decomposing complex problems into manageable subcomponents/models
- Many important extensions to this simple model

References

- Berliner, L. M. 1996, Hierarchical Bayesian time series models”, Maximum Entropy and Bayesian Methods, 15-22.
- Gelman and Hill 2007. Data analysis using regression and multilevel models
- Royle and Dorazio 2008. Hierarchical modeling and inference in ecology.
- Kery and Schaub. 2012. Bayesian Population Analysis using WinBUGS. Chapter 3
- McElreath 2023. Statistical Rethinking. Second Edition, Chapters 2 and 9.
- Punt et al. 2011. Among-stock comparisons for improving stock assessments of data-poor stocks: the “Robin Hood” approach. ICES Journal of Marine Sci.