

Hierarchical Models

Objectives

- General Use and Language
- Examples
- Case studies
 - random intercept
 - random intercept & slope
- Considerations

Names

- multilevel model
- mixed model
- random effects model
- nested data model
- split-plot model

Sort of definitions

- models where parameters vary at more than one level
- models in which lower levels are sorted under a hierarchy of successive higher-level units
- Generalization of linear and generalized linear modeling in which regression coefficients are themselves given a model whose parameters are also estimated from data (Gelman)
- Model where unobserved parameters are estimated as a function of other unobserved parameters

Hobbs and Hooten 2015 (pg. 109)

- Representing variation among individuals arising, for example, from genetics, location, or experience.
- Studying phenomena operating at more than one spatial scale or level of ecological organization.
- Modeling a process as well as uncertainty that results from imperfect observations of the process.
- Understanding changes in states of ecological systems that cannot be observed directly. These states arise from “hidden” processes.

General Types

- Random Effect / Multi-level Model
- latent/unobserved state or process

Random Effect

[Andrew Gelman Blog Post](#)

[Paper Link](#)

“People are always asking me if I want to use a fixed or random effects model for this or that.”

. . .

“I always reply that these terms have no agreed-upon definition.”

. . .

“People with their own favorite definitions don’t always realize that other definitions are out there. Worse, people conflate different definitions”.

Example 1

- student grades within classroom, within school



Example 2

- forest cover effect on occurrence of bobcats at different protected areas



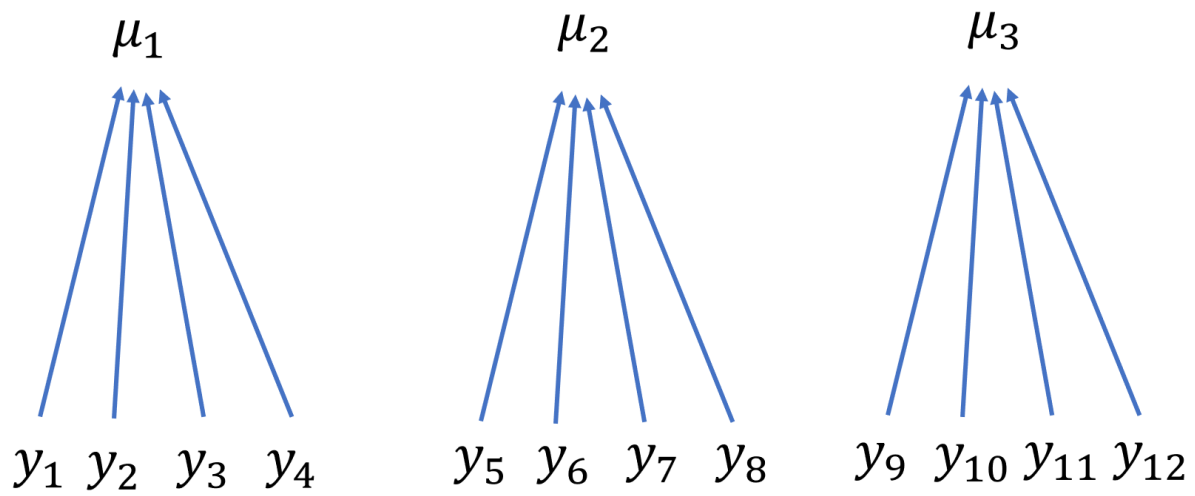
Random Effect / Multi-level Model

- repeated *measurements* of individual within population
- repeated *measurements* at a small spatial scale that is part of a larger one

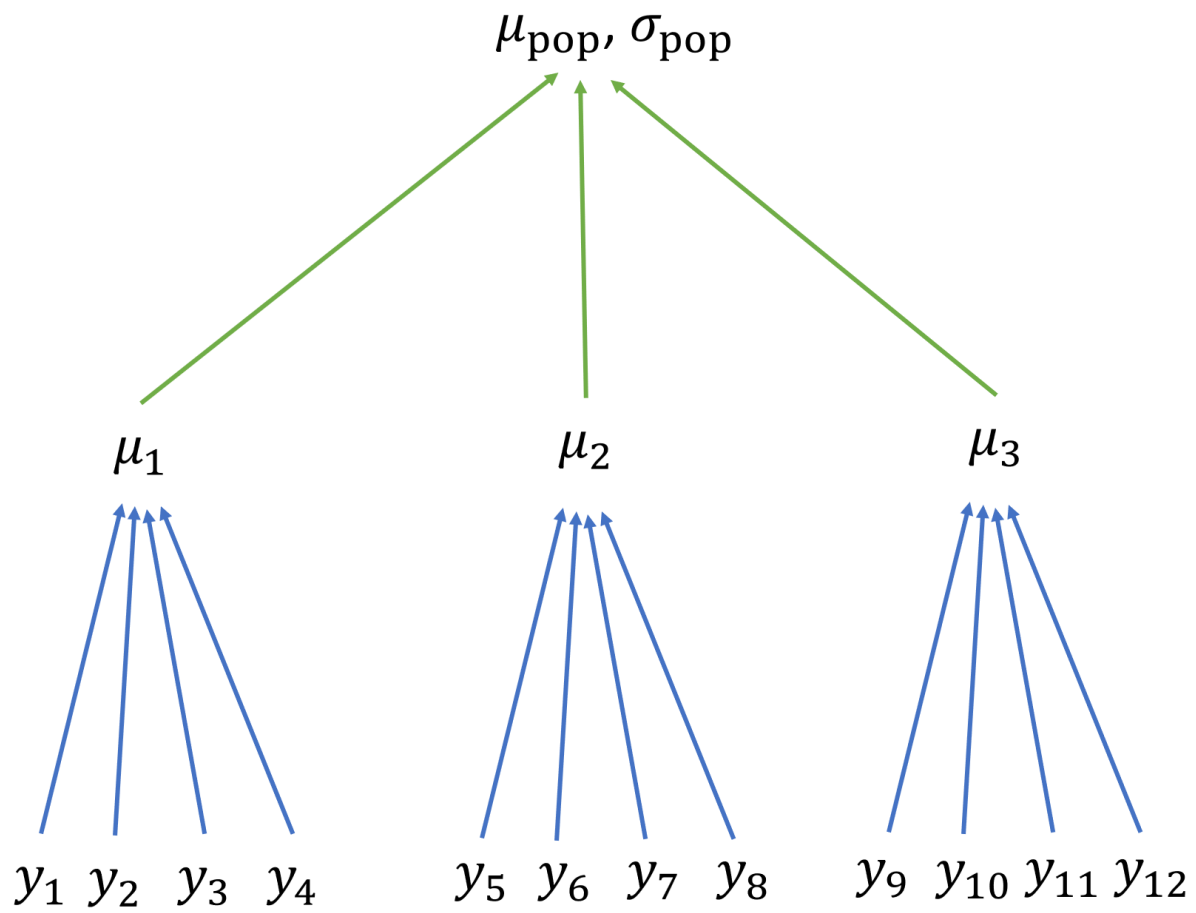
Hierarchical Structure to Data

y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8 y_9 y_{10} y_{11} y_{12}

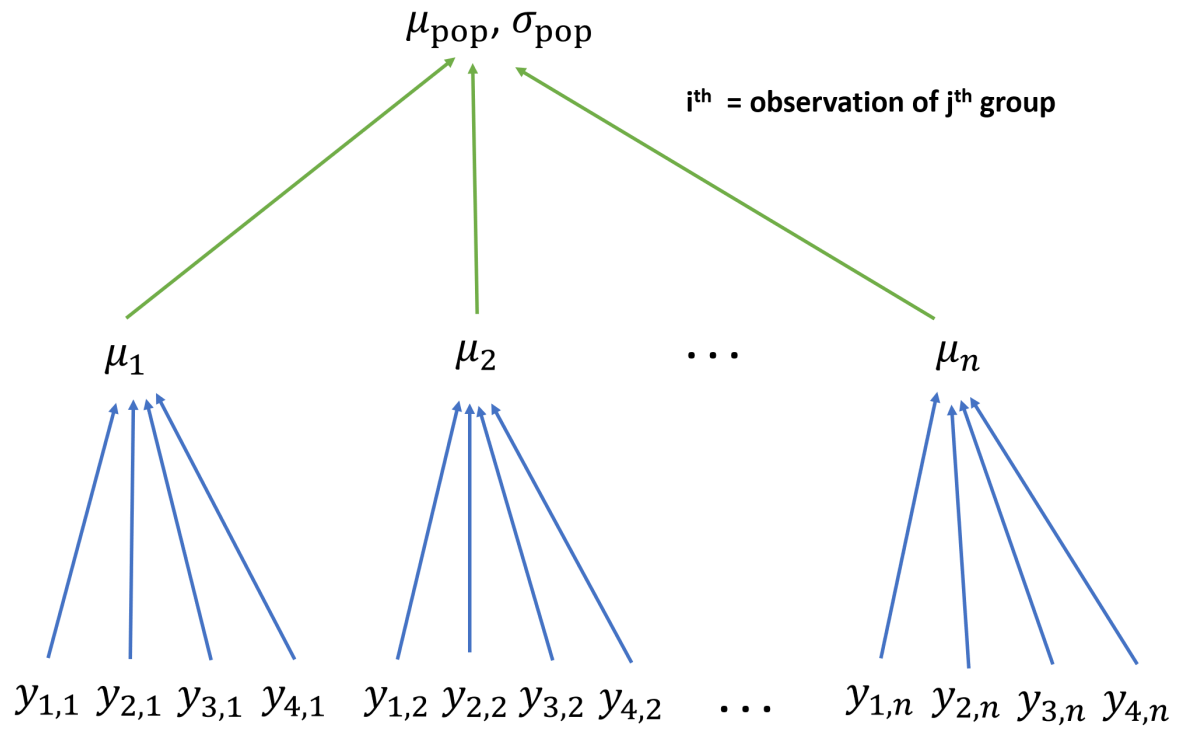
Hierarchical Structure to Data



Hierarchical Structure to Data



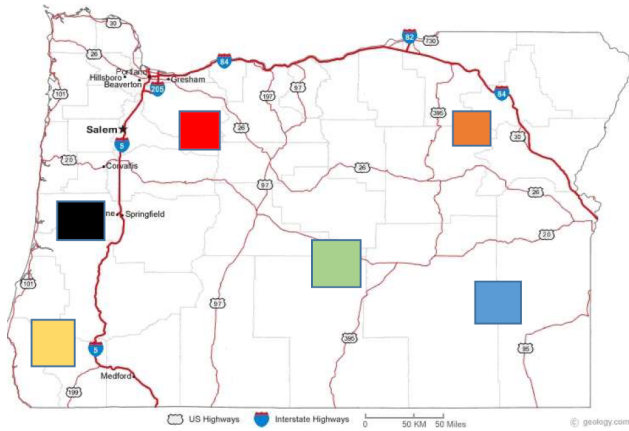
Hierarchical Structure to Data



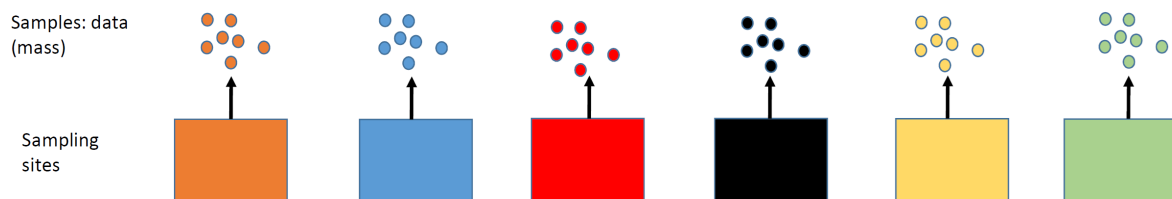
Hierarchical Structure Example

Study goals

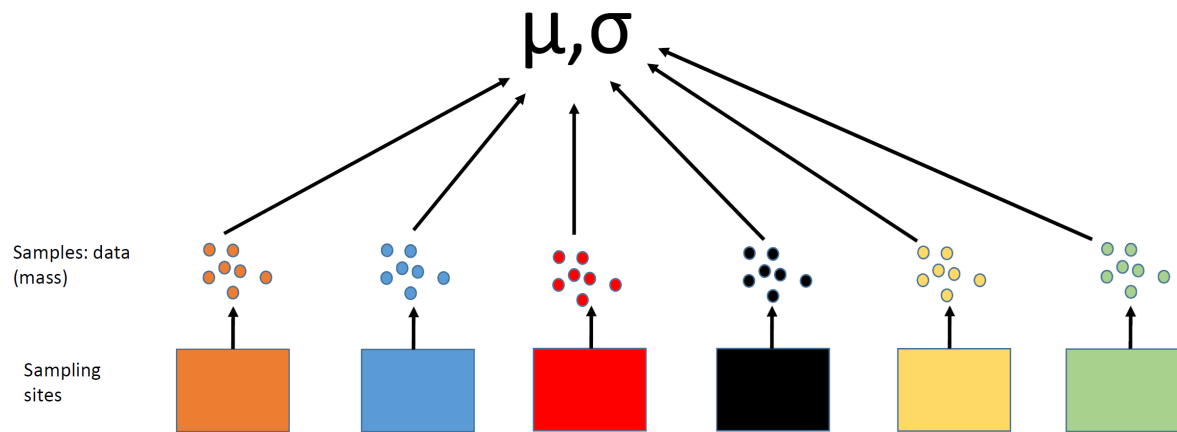
- Estimate state-wide average mass
- Understand site-level variation



Hierarchical Structure Example



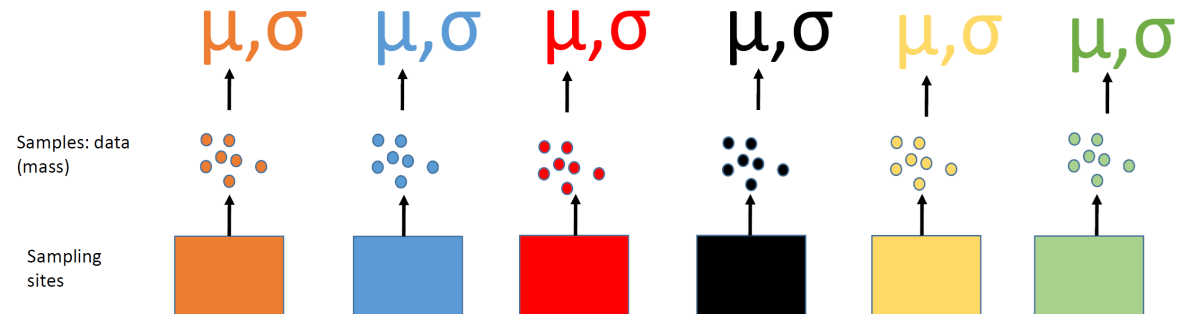
Complete Pooling



...

- Ignores variance across sites, which is a goal of study
- Data within sampling site might not be independent, leading to pseudoreplication (depressed variance)

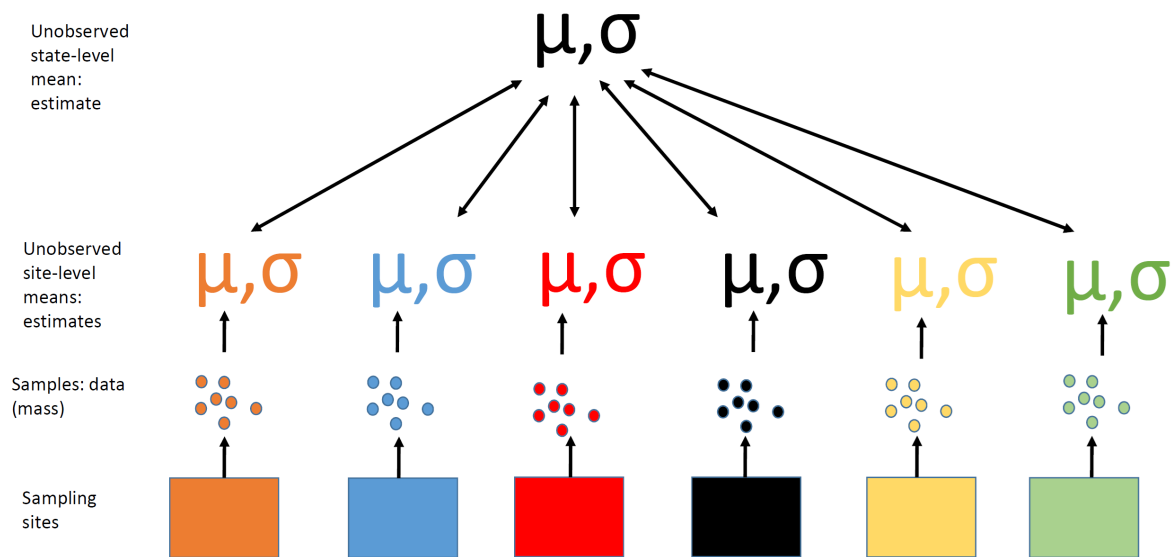
No Pooling



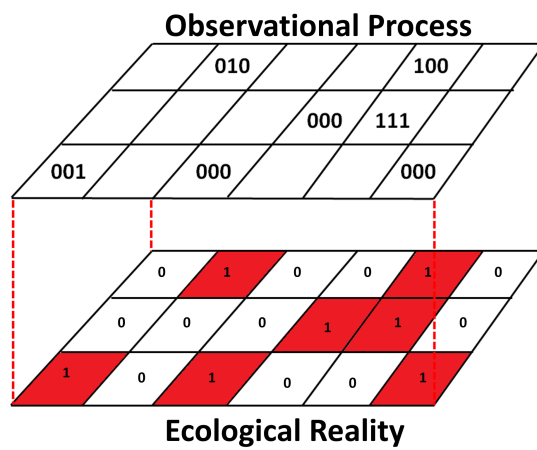
...

- No state-level estimate
- Estimator for each site may be biased, particularly at low sample sizes

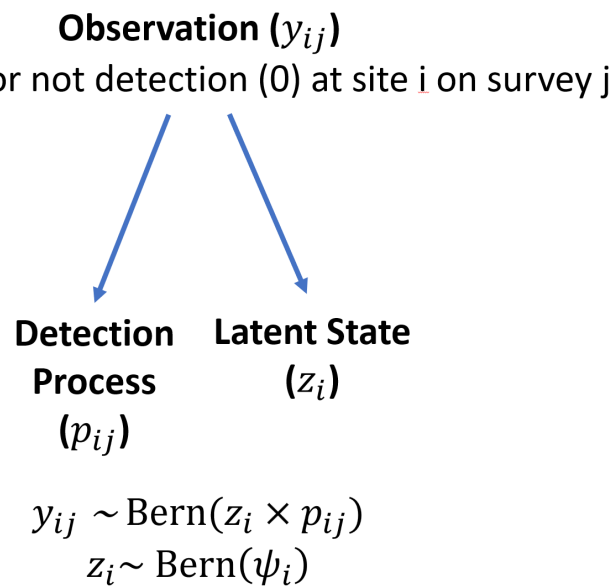
Partial Pooling (Hierarchical Model)



Latent State



Latent State



Data and Problem

We sample Amur Leopards in five different vegetation types of Land of the Leopard National Park.



Interested in detection rate by vegetation type and overall.

Data and Problem

We sample Amur Leopards in five different vegetation types of Land of the Leopard National Park.

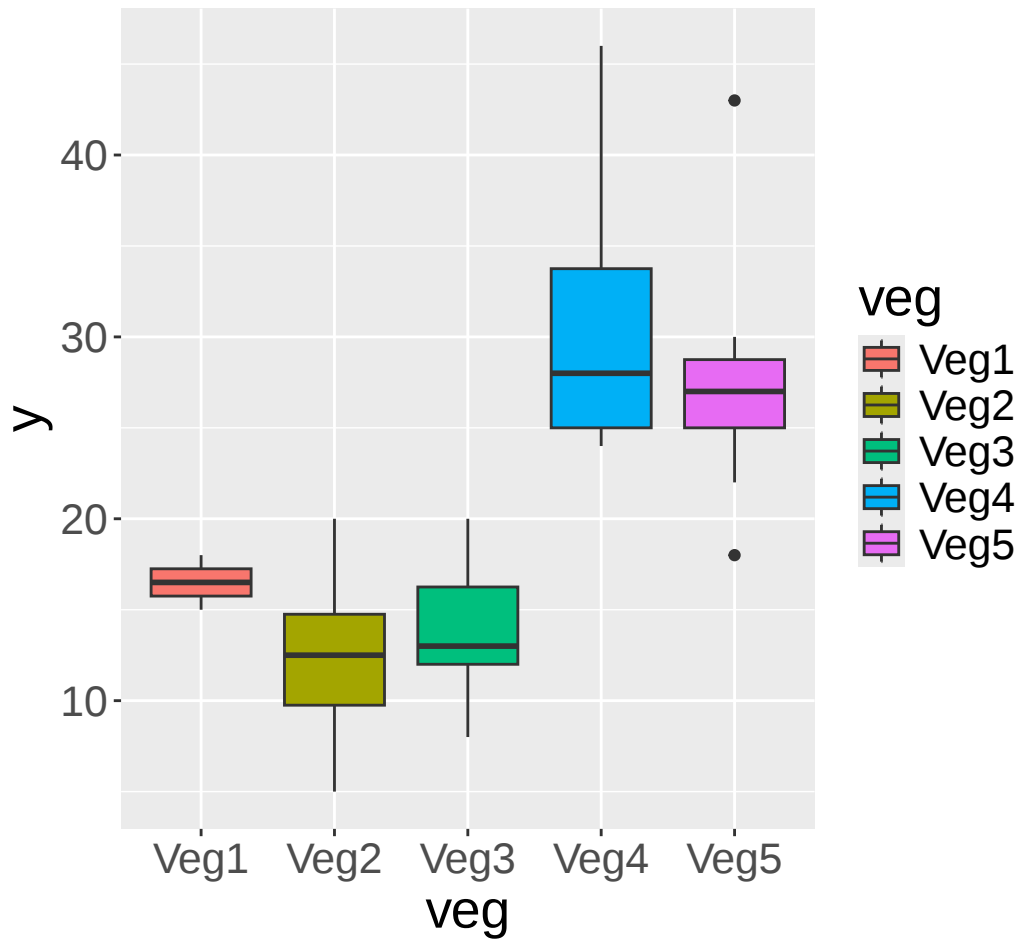


Detection Rate = Independent Counts / Effort

Data

	y	veg	effort
1	15	Veg1	5
2	18	Veg1	5
3	12	Veg2	5
4	14	Veg2	5
5	12	Veg2	5
6	14	Veg2	5

Veg1	Veg2	Veg3	Veg4	Veg5
2	20	20	10	10



Model 1 (Complete Pooling)

```
library(glmTMB)
library(broom.mixed)

model1 = glmTMB(y ~ 1,
                family = poisson(link="log"),
                data = dat
                )
```

...

$$y_i \sim \text{Poisson}(\lambda_i)$$

$$\log(\lambda_i) = \beta_0$$

Model 1 (Complete Pooling)

```
summary(model1)
```

```
Family: poisson ( log )
Formula:          y ~ 1
Data: dat
```

AIC	BIC	logLik	-2*log(L)	df.resid
532.2	534.4	-265.1	530.2	61

Conditional model:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	2.91165	0.02962	98.31	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Model 1 (Complete Pooling)

```
marginaleffects::predictions(model1, type = "response")
```

Warning: For this model type, `marginaleffects` only takes into account the uncertainty in fixed-effect parameters. This is often appropriate when `re.form=NA`, but may be surprising to users who set `re.form=NULL` (default) or to some other value. Call `options(marginaleffects\_safe = FALSE)` to silence this warning.

Estimate	Std. Error	z	Pr(> z)	S	2.5 %	97.5 %
18.4	0.545	33.8	<0.001	827.7	17.3	19.5
18.4	0.545	33.8	<0.001	827.7	17.3	19.5
18.4	0.545	33.8	<0.001	827.7	17.3	19.5

```

      18.4      0.545 33.8    <0.001 827.7  17.3   19.5
      18.4      0.545 33.8    <0.001 827.7  17.3   19.5
--- 52 rows omitted. See ?print.marginalEffects ---
      18.4      0.545 33.8    <0.001 827.7  17.3   19.5
      18.4      0.545 33.8    <0.001 827.7  17.3   19.5
      18.4      0.545 33.8    <0.001 827.7  17.3   19.5
      18.4      0.545 33.8    <0.001 827.7  17.3   19.5
      18.4      0.545 33.8    <0.001 827.7  17.3   19.5
Type: response

```

Model 2 (No Pooling)

```

model2 = glmmTMB(y ~ veg,
                 family = poisson(link="log"),
                 data = dat,
                 contrasts = list(veg = "contr.sum")
                 )

```

...

$$y_i \sim \text{Poisson}(\lambda_i)$$

$$\log(\lambda_i) = \beta_0 + \beta_1 \times \text{veg}_1 + \beta_2 \times \text{veg}_2 + \beta_3 \times \text{veg}_3$$

Model 2 (No Pooling)

```
summary(model2)
```

```

Family: poisson ( log )
Formula:          y ~ veg
Data: dat

```

AIC	BIC	logLik	-2*log(L)	df.resid
369.4	380.1	-179.7	359.4	57

Conditional model:

Estimate	Std. Error	z	value	Pr(> z)
----------	------------	---	-------	----------

```

(Intercept)  2.93764    0.04235   69.37 < 2e-16 ***
veg1         -0.13427    0.14133   -0.95    0.342
veg2         -0.42398    0.06498   -6.52 6.82e-11 ***
veg3         -0.29146    0.06262   -4.65 3.25e-06 ***
veg4          0.47681    0.06138    7.77 7.94e-15 ***

```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Model 2 (No Pooling)

```

marginaleffects::predictions(model2,
                               newdata = data.frame(veg = c("Veg1","Veg2","Veg3","Veg4","Veg5"),
                               re.form = NA,
                               type = "response")

```

Estimate	Std. Error	z	Pr(> z)	S	2.5 %	97.5 %	veg
16.5	2.872	5.74	<0.001	26.7	10.9	22.1	Veg1
12.4	0.786	15.72	<0.001	182.5	10.8	13.9	Veg2
14.1	0.840	16.79	<0.001	207.8	12.5	15.7	Veg3
30.4	1.744	17.44	<0.001	223.7	27.0	33.8	Veg4
27.4	1.655	16.55	<0.001	202.0	24.2	30.6	Veg5

Type: response

Model 3 (Partial Pooling)

```

model3 = glmmTMB(y ~ 1 + (1|veg),
                 family = poisson(link="log"),
                 data = dat
                 )

```

i = counts j = vegetation type

$$\begin{aligned}
 y_i &\sim \text{Poisson}(\lambda_i) \\
 \log(\lambda_i) &= \mu + \beta_{j[i]} \\
 \beta_j &\sim \text{Normal}(0, \sigma_\beta^2)
 \end{aligned}$$

Model 3 (Partial Pooling)

```
summary(model3)
```

```
Family: poisson ( log )  
Formula:          y ~ 1 + (1 | veg)  
Data: dat
```

AIC	BIC	logLik	-2*log(L)	df.resid
384.4	388.6	-190.2	380.4	60

Random effects:

Conditional model:

Groups	Name	Variance	Std.Dev.
veg	(Intercept)	0.1299	0.3604

Number of obs: 62, groups: veg, 5

Conditional model:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	2.9391	0.1662	17.69	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Model 3 (Partial Pooling)

```
confint(model3, component = c("all"))
```

	2.5 %	97.5 %	Estimate
(Intercept)	2.6134002	3.2647754	2.9390878
Std.Dev.(Intercept) veg	0.1896794	0.6846043	0.3603545

```
ranef(model3)
```

```
$veg  
(Intercept)  
Veg1 -0.1103110  
Veg2 -0.4126484
```

```
Veg3 -0.2851561
Veg4  0.4635429
Veg5  0.3612502
```

```
broom.mixed::tidy(model3, effects = "ran_vals", conf.int = TRUE)
```

```
# A tibble: 5 x 9
```

	effect	component	group	level	term	estimate	std.error	conf.low	conf.high
	<chr>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	ran_vals	cond	veg	Veg1	(Interce~	-0.110	0.206	-0.514	0.294
2	ran_vals	cond	veg	Veg2	(Interce~	-0.413	0.173	-0.751	-
						0.0738			
3	ran_vals	cond	veg	Veg3	(Interce~	-0.285	0.172	-0.622	0.0520
4	ran_vals	cond	veg	Veg4	(Interce~	0.464	0.172	0.126	0.801
5	ran_vals	cond	veg	Veg5	(Interce~	0.361	0.173	0.0230	0.699

```
fixef(model3)
```

Conditional model:

```
(Intercept)
      2.939
```

```
#Predictions - does not include RE uncertainty
```

```
preds = predict(model3,
  newdata = data.frame(veg = c("Veg1", "Veg2", "Veg3", "Veg4", "Veg5")),
  type = "link",
  re.form = NULL,
  se.fit = TRUE
)
```

```
preds$LCL = exp(preds$fit - 1.96 * preds$se.fit)
preds$UCL = exp(preds$fit + 1.96 * preds$se.fit)
preds$fit = exp(preds$fit)
```

```
data.frame(preds)
```

	fit	se.fit	LCL	UCL
1	16.92475	0.15877885	12.39843	23.10348

```
2 12.50889 0.06300056 11.05583 14.15291
3 14.20980 0.05890828 12.66030 15.94893
4 30.04303 0.05761047 26.83519 33.63433
5 27.12181 0.06039730 24.09392 30.53021
```

```
#A typical site (non an average site)
# Done by setting the random effect mean to 0
predict(model3,type="response",re.form = NA)[1]
```

```
[1] 18.8986
```

Model 3b (lme4)

```
library(lme4)
```

Loading required package: Matrix

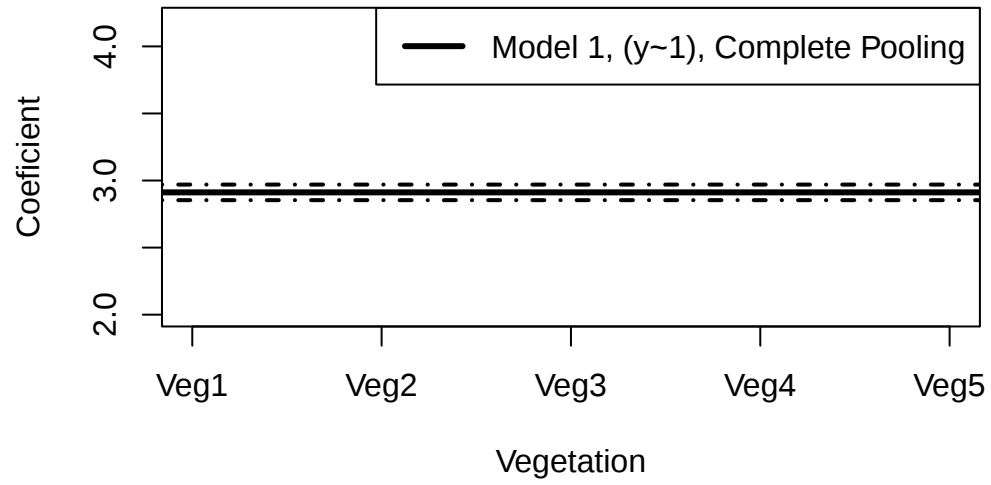
```
model3b = glmer(y ~ 1 + (1|veg),
                family = poisson(link = "log"),
                data = dat
                )
```

```
equationomatic::extract_eq(model3b)
```

$$\begin{aligned}
 y_i &\sim \text{Poisson}(\lambda_i) \\
 \log(\lambda_i) &= \alpha_{j[i]} \\
 \alpha_j &\sim N(\mu_{\alpha_j}, \sigma_{\alpha_j}^2), \text{ for veg } j = 1, \dots, J
 \end{aligned}
 \tag{1}$$

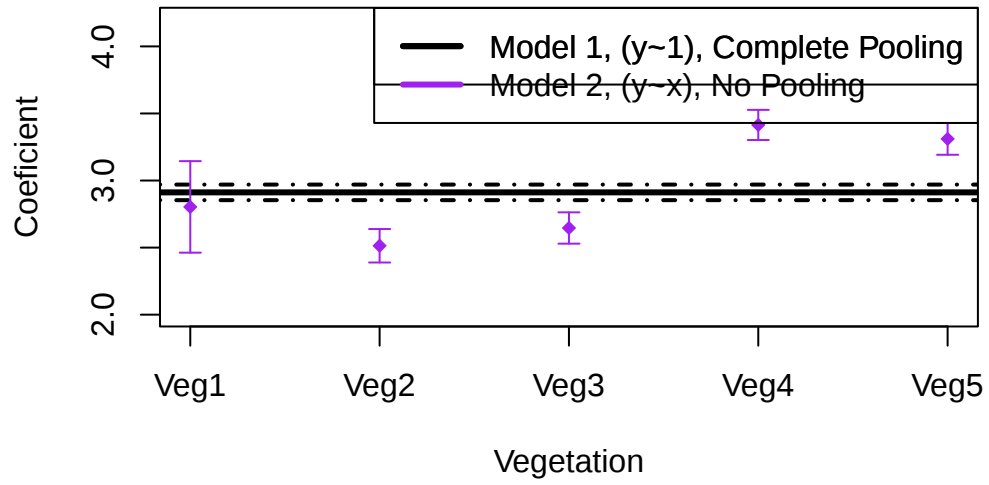
Pooling Information

Estimates



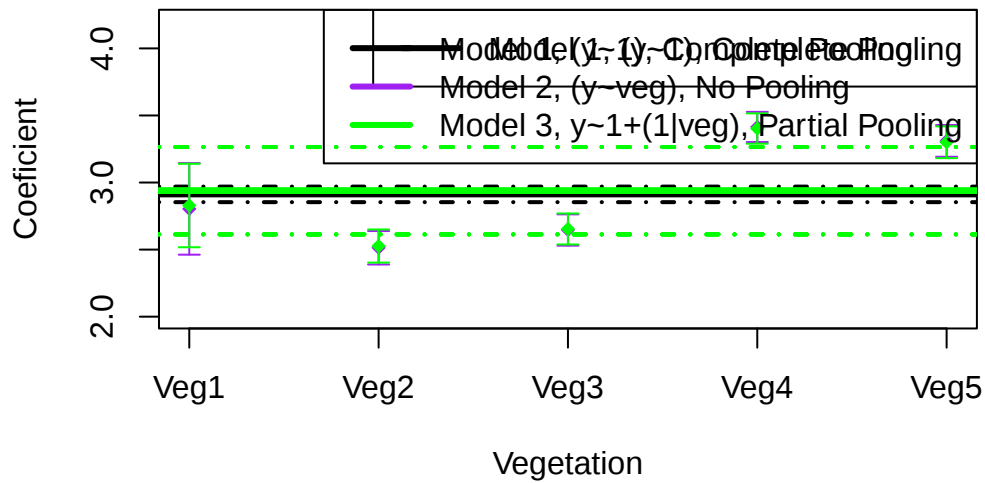
Pooling Information

Estimates



Pooling Information

Estimates



Random Intercept + Slope

New hypothesis: Leopard detection varies by cover and vegetation, where the effect of cover comes from a shared distribution

	y	cov	veg
1	3	0.40161608	Veg1
2	0	-0.12360373	Veg1
3	3	0.62883136	Veg1
4	0	0.07607150	Veg1
5	2	0.04150764	Veg1
6	0	-0.39289231	Veg1

Random Intercept + Slope

```
re.model = glmer(y ~ cov + (cov||veg),  
                  family = poisson(link = "log"),
```

```
data = dat2
)
```

...

i = counts j = vegetation type

$$\begin{aligned}
 y_i &\sim \text{Poisson}(\lambda_i) \\
 \log(\lambda_i) &= \mu_{\text{veg}} + \beta_{\text{veg}_{j[i]}} + \mu_{\text{cov}} + \beta_{\text{cov}_{j[i]}} \\
 \beta_{\text{veg}_j} &\sim \text{Normal}(0, \sigma_{\text{veg}}^2) \\
 \beta_{\text{cov}_j} &\sim \text{Normal}(0, \sigma_{\text{cov}}^2)
 \end{aligned}$$

Random Intercept + Slope

```
summary(re.model)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
Approximation) [glmerMod]
Family: poisson ( log )
Formula: y ~ cov + ((1 | veg) + (0 + cov | veg))
Data: dat2
```

AIC	BIC	logLik	-2*log(L)	df.resid
14272.8	14289.7	-7132.4	14264.8	496

Scaled residuals:

Min	1Q	Median	3Q	Max
-4.796	-2.443	-1.631	-0.946	145.194

Random effects:

Groups	Name	Variance	Std.Dev.
veg	(Intercept)	0.4896	0.6997
veg.1	cov	0.1393	0.3732

Number of obs: 500, groups: veg, 5

Fixed effects:

Estimate	Std. Error	z value	Pr(> z)
----------	------------	---------	----------

```
(Intercept)  1.6210      0.3137    5.167 2.38e-07 ***
cov          0.1128      0.1683    0.670   0.503
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Correlation of Fixed Effects:

```
(Intr)
cov 0.000
```

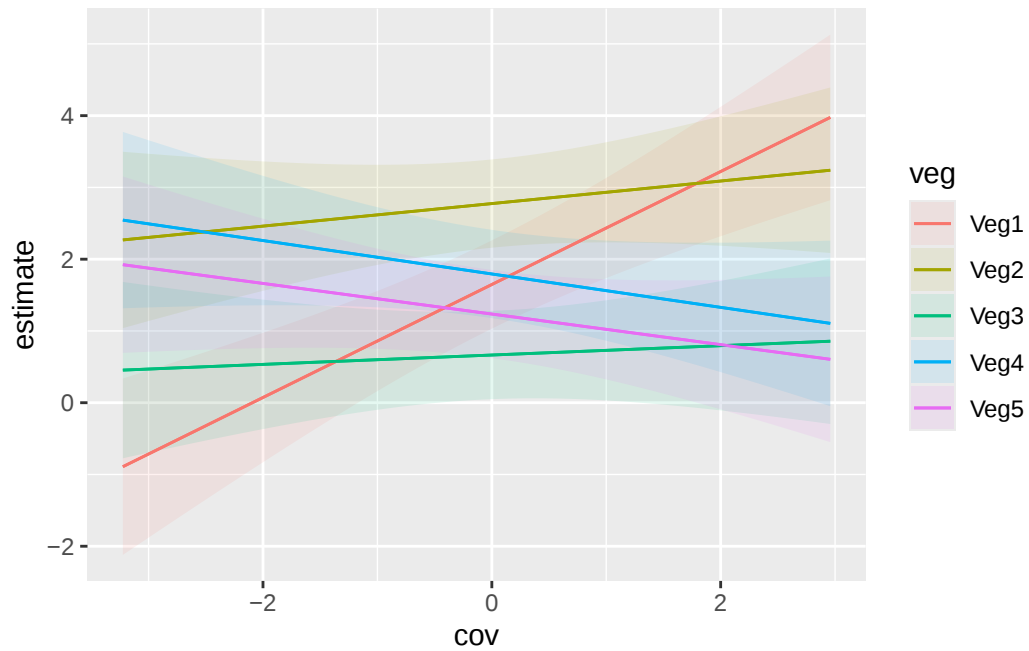
Random Intercept + Slope

- Called conditional mean predictions

```
marginaleffects::plot_predictions(re.model, condition = c("cov","veg"),
                                  type = "link", re.form = NULL)
```

Warning: For this model type, `marginaleffects` only takes into account the uncertainty in fixed-effect parameters. This is often appropriate when `re.form=NA`, but may be surprising to users who set `re.form=NULL` (default) or to some other value. Call `options(marginaleffects\_safe = FALSE)` to silence this warning.

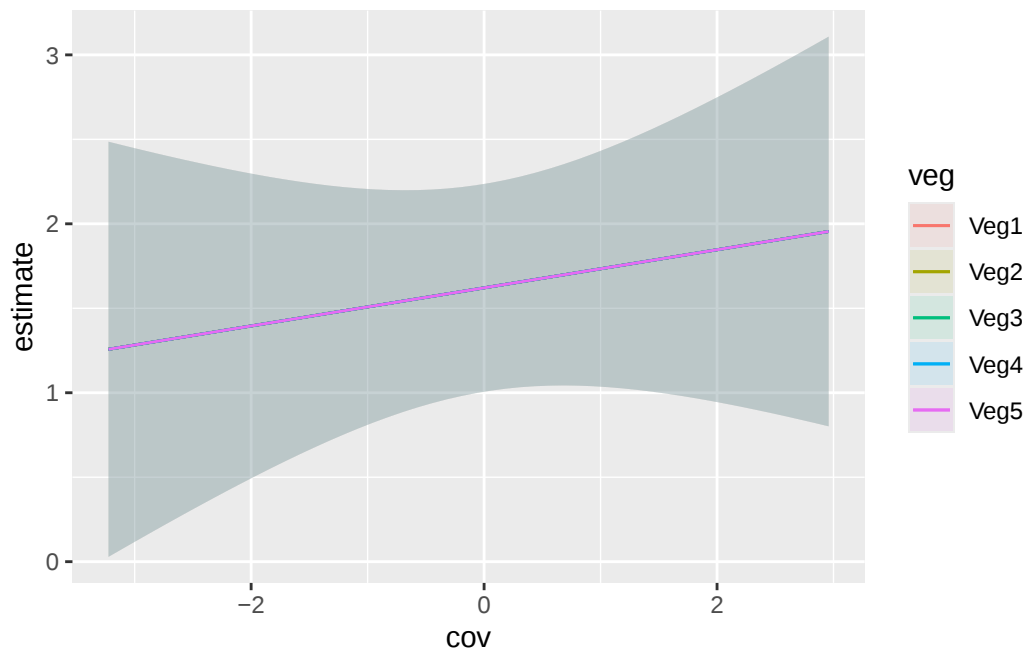
Warning: For this model type, `marginaleffects` only takes into account the uncertainty in fixed-effect parameters. This is often appropriate when `re.form=NA`, but may be surprising to users who set `re.form=NULL` (default) or to some other value. Call `options(marginaleffects\_safe = FALSE)` to silence this warning.



Random Intercept + Slope

- predictions for a ‘typical site’; not an average site [Stats4Ecologists](#)

```
marginalEffects::plot_predictions(re.model, condition=c("cov","veg"),
                                  type = "link", re.form = NA)
```



Random Intercept + Slope

```
equationomatic::extract_eq(re.model)
```

$$\begin{aligned}
 y_i &\sim \text{Poisson}(\lambda_i) \\
 \log(\lambda_i) &= \alpha_{j[i]} + \beta_{1j[i]}(\text{cov}) \\
 \begin{pmatrix} \alpha_j \\ \beta_{1j} \end{pmatrix} &\sim N \left(\begin{pmatrix} \mu_{\alpha_j} \\ \mu_{\beta_{1j}} \end{pmatrix}, \begin{pmatrix} \sigma_{\alpha_j}^2 & 0 \\ 0 & \sigma_{\beta_{1j}}^2 \end{pmatrix} \right), \text{ for veg } j = 1, \dots, J
 \end{aligned} \tag{2}$$

FAQ

GLMM FAQ

- model specifications
- predictions with uncertainty
- Testing RE's

Considerations

- Shrinkage/Regularization
- When to use a random effect?

Lab