

Mixed effects models

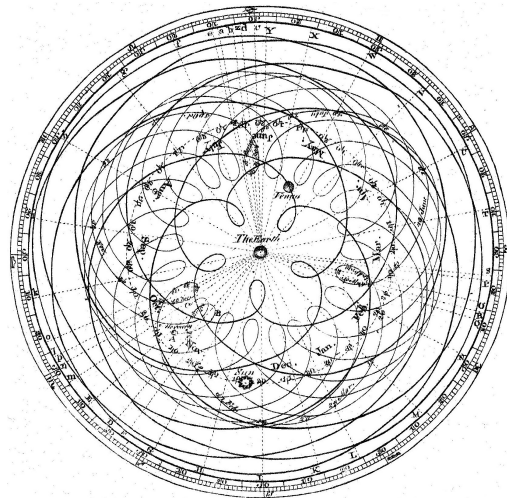
“Wheels within wheels”

Samuel Robinson, Ph.D.

Oct 6, 2023

Outline

- Linear mixed effects models (LMMs)
 - A bit of math
 - Fixed vs. random effects
 - Random intercepts and slopes
- Generalized linear mixed effects models (GLMMs)
 - Residuals checks
 - Some sage advice
- Hypothesis testing and inference
 - Slopes and intercepts
 - Entire terms
 - AIC, R^2 , and Star-Gazing



Part 1: Mixed effects models

Problem: group-level variation

- Sometimes we have to sample within groups: different field sites, individual organisms, etcetera
- However, often we're not really interested in each group *per se*, but in the **average group**
- e.g. "What is the effect of **x** if you remove group-to-group variation?"
- If you have a small number of groups, you can just include it in your model: `lm(y ~ x + group)`
 - If you have few samples for each group, this can create problems
- Another solution is to use **mixed effects models**

What are mixed effects models?

Many different names:

- ① Mixed effects models
- ② Random effects models
- ③ Hierarchical models
- ④ Empirical/Bayesian hierarchical models
- ⑤ Latent variable models
- ⑥ Split-plot models
- ⑦ Variance partitioning

I usually use the term *heirarchical models*, as this is the closest to what I will teach you

Scary math

Unfortunately, we need a review of matrix algebra in order to explain this:

- This is a matrix:

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

- This is a vector:

$$b = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

- Multiplying them looks like this:

$$A \times b = Ab = 1 \times \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + 2 \times \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} + 3 \times \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 30 \\ 36 \\ 42 \end{bmatrix}$$

Why do we call them “linear models”?

- *Linear* mapping of **coefficients** onto a **model matrix** (from your data)

- Coefficients:

$$\beta = \begin{bmatrix} 0.1 & 1.8 & -0.03 \end{bmatrix}$$

- Model matrix:

$$X = \begin{bmatrix} 1 & 1 & 10 \\ 1 & 1 & 12 \\ 1 & 0 & 9 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

- Multiplying them looks like:

$$\hat{y} = X\beta = \begin{bmatrix} 1.60 \\ 1.54 \\ -0.17 \\ \vdots \end{bmatrix}$$

This is exactly what R does to fit models:

```
head(dat)
```

```
##           y           x site
## 1  1.5101095 -4.248450    g
## 2  3.7190900  5.766103    j
## 3 -4.3737644 -1.820462    f
## 4 30.1459331  7.660348    n
## 5  0.2777422  8.809346    o
## 6 -3.6978175 -9.088870    p
```

```
m1 <- lm(y~x,data=dat) #Uses x to predict y
summary(m1)
```

```
##
## Call:
## lm(formula = y ~ x, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.2574  -4.1262   0.0296   3.1854  25.2780
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.7306     0.5772   4.731 4.92e-06 ***
## x              0.7213     0.1020   7.074 4.60e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.3 on 158 degrees of freedom
## Multiple R-squared:  0.2405, Adjusted R-squared:  0.2357
## F-statistic: 50.04 on 1 and 158 DF,  p-value: 4.6e-11
```


This is exactly what R does to fit models (cont.):

```
head(model.matrix(m1))
```

```
##      (Intercept)          x
## 1             1 -4.248450
## 2             1  5.766103
## 3             1 -1.820462
## 4             1  7.660348
## 5             1  8.809346
## 6             1 -9.088870
```

```
coef(m1)
```

```
##      (Intercept)          x
## 2.7305689  0.7212867
```

```
pred2 <- model.matrix(m1) %*% coef(m1) #predicted = matrix * coefs
head(data.frame(pred1=predict(m1),pred2)) #same thing!
```

```
##      pred1      pred2
## 1 -0.3337812 -0.3337812
## 2  6.8895819  6.8895819
## 3  1.4174942  1.4174942
## 4  8.2558758  8.2558758
## 5  9.0846325  9.0846325
## 6 -3.8251119 -3.8251119
```

Groups are coded by “dummy variables” (0s and 1s)

```
m2 <- lm(y~site,data=dat) #Use site to predict y  
head(model.matrix(m2)) #0s and 1s used to identify groups
```

```
## (Intercept) siteb sitec sited sitee sitef siteg siteh sitei sitej sitek sitel  
## 1          1      0      0      0      0      0      1      0      0      0      0  
## 2          1      0      0      0      0      0      0      0      0      1      0  
## 3          1      0      0      0      0      1      0      0      0      0      0  
## 4          1      0      0      0      0      0      0      0      0      0      0  
## 5          1      0      0      0      0      0      0      0      0      0      0  
## 6          1      0      0      0      0      0      0      0      0      0      0  
##      sitem siten siteo sitep  
## 1      0      0      0      0  
## 2      0      0      0      0  
## 3      0      0      0      0  
## 4      0      1      0      0  
## 5      0      0      1      0  
## 6      0      0      0      1
```

```
coef(m2) #This uses the 1st site as the "control" group
```

```
## (Intercept)      siteb      sitec      sited      sitee      sitef  
## 7.192416 -11.998464 -14.632803  1.983649 -7.765354 -4.523079  
##      siteg      siteh      sitei      sitej      sitek      sitel  
## -3.439621 -8.280601 -4.306456 -4.085855 -5.663021 -5.155112  
##      sitem      siten      siteo      sitep  
## -6.226642  8.403599 -8.626661 -10.934182
```

Structure of LMs... now with matrices!

- All linear models take the form:

$$\hat{y} = X\beta = b_0 1 + b_1 x_1 \dots + b_i x_i$$

$$y \sim \text{Normal}(\hat{y}, \sigma)$$

- y is a vector of data you want to predict
- \hat{y} is a vector of *predicted values* for y
- $X = \{1, x_1 \dots\}$ is a matrix of *predictors* for y
- $\beta = \{b_0, b_1, \dots\}$ is a vector of *coefficients*
- $y \sim \text{Normal}(\hat{y}, \sigma)$ means:
 - “ y follows a Normal distribution with mean \hat{y} and SD σ ”

Fixed effects vs. Random effects

Say that X is a model matrix coding for 10 sites¹, and y is something we're interested in predicting

$$\hat{y} = b_0 + X\beta$$
$$y \sim \text{Normal}(\hat{y}, \sigma)$$

- Site coefficients (β) are unrelated to each other
- σ is the SD of *residuals*
- Site is a **fixed effect**

$$\hat{y} = b_0 + X\zeta$$
$$y \sim \text{Normal}(\hat{y}, \sigma)$$
$$\zeta \sim \text{Normal}(0, \sigma_{\text{site}})$$

- Site coefficients (ζ) are related to each other via a *Normal* distribution
- σ is the SD of *residuals*, σ_{site} is the SD of *sites*
- Site is a **random effect**

¹Intercept is a separate variable

Mixed effects = fixed + random effects

A mixed effects model has both **fixed** and **random** effects

$$\hat{y} = X\beta + U\zeta$$

$$y \sim \text{Normal}(\hat{y}, \sigma)$$

$$\zeta \sim \text{Normal}(0, \sigma_{\text{site}})$$

- X = fixed effects matrix (e.g. intercept, temperature)
- β = fixed effects coefficients
- U = random effects matrix (e.g. sites)
- ζ = random effects coefficients
- $\sigma, \sigma_{\text{site}}$ = variance terms

Mixed effect model example

Let's go back to our earlier example:

- We're interested in predicting y using x (fixed effects)
- Data was collected at a number of *sites*, which may affect y "somehow"
- Effect of each site is normally distributed

```
head(dat)
```

```
##           y           x site
## 1  1.5101095 -4.248450    g
## 2  3.7190900  5.766103    j
## 3 -4.3737644 -1.820462    f
## 4 30.1459331  7.660348    n
## 5  0.2777422  8.809346    o
## 6 -3.6978175 -9.088870    p
```

```
library(lme4) #Mixed effects library
mm1 <- lmer(y ~ x + (1|site),data=dat) #site is fit as "random intercepts"
```

Mixed effect model example

```
summary(mm1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ x + (1 | site)
## Data: dat
##
## REML criterion at convergence: 1040.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.50816 -0.71380 -0.02682  0.69401  3.01951
##
## Random effects:
## Groups Name Variance Std.Dev.
## site (Intercept) 26.78 5.175
## Residual 31.68 5.628
## Number of obs: 160, groups: site, 16
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 2.00331 1.38297 1.449
## x 0.76192 0.08083 9.426
##
## Correlation of Fixed Effects:
## (Intr)
## x -0.002
```

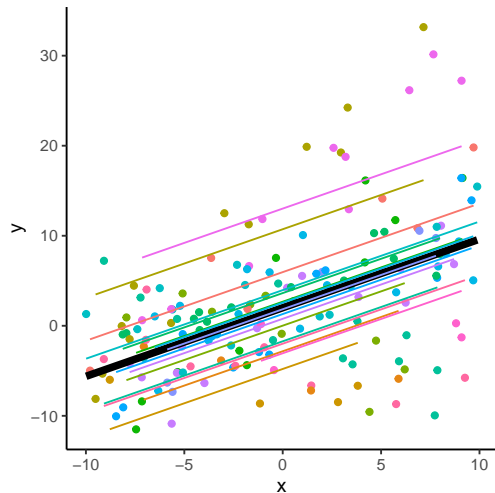
Results from lmer model:

- Random effects:
 - *residual* and *site* variance (σ , σ_{site})
- Fixed effects:
 - Intercept and slope estimates (β)
 - No d.f. and p-value²
 - If you need p-values for parameters, you can use the *lmerTest* package (or just calculate them yourself using means/SEs)

²'lme4' author doesn't think they can be calculated. I somewhat agree

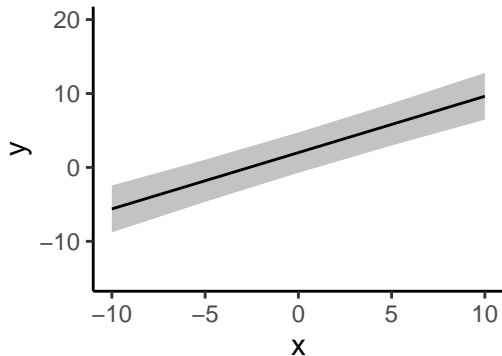
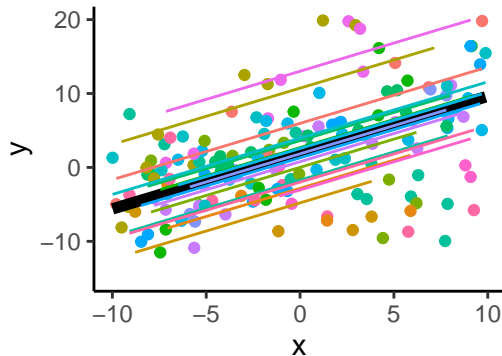
Mixed effect model results

- In a *random intercepts* model, the regression line of x on y is allowed to move up or down around the main regression line for each site
- These changes in intercepts are *normally distributed*



Mixed effect model results (cont.)

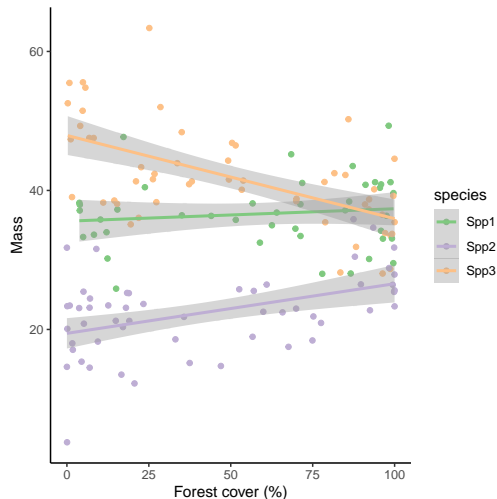
- For plotting, we want a partial effects plot that marginalizes across sites (i.e. “What does the trend look like at the average site?”)
- `ggpredict` works well for this. If you want partial residuals, you’ll have to add them in yourself using `predict` and `residual`



First challenge

How does forest cover influence fish size? Maybe some of the species do better in forested streams?

- You've weighed fish in streams with different forest covers (`fishMass.csv`). However, perhaps some of the variation is caused by "other things" about the site?
- Fit a mixed effects model with the fixed effects you're interested in (**forest cover**, **species**), and include **site** as a random *intercept*
- How does this compare to a simple linear model where you *ignore* site?



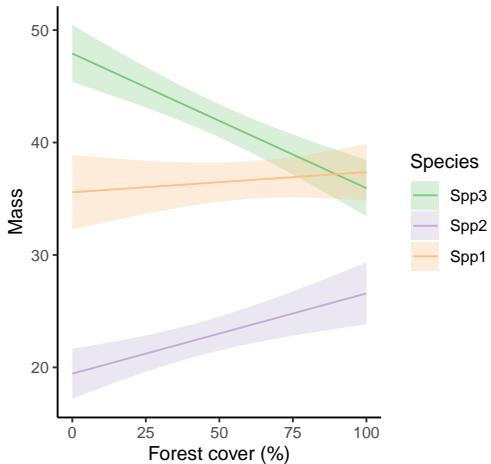
First challenge results

```
##
## Call:
## lm(formula = mass ~ species * forest, data = fishDat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.6767  -3.1422   0.0415   3.3364  18.4631
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    35.57610     1.68256   21.144 < 2e-16 ***
## speciesSpp2    -16.13571     2.02625   -7.963 4.60e-13 ***
## speciesSpp3     12.34080     2.11876    5.825 3.59e-08 ***
## forest          0.01792     0.02413    0.743  0.4590
## speciesSpp2:forest 0.05348     0.03152    1.697  0.0919 .
## speciesSpp3:forest -0.13769     0.03187   -4.321 2.88e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.468 on 144 degrees of freedom
## Multiple R-squared:  0.7391, Adjusted R-squared:  0.73
## F-statistic: 81.57 on 5 and 144 DF,  p-value: < 2.2e-16
```

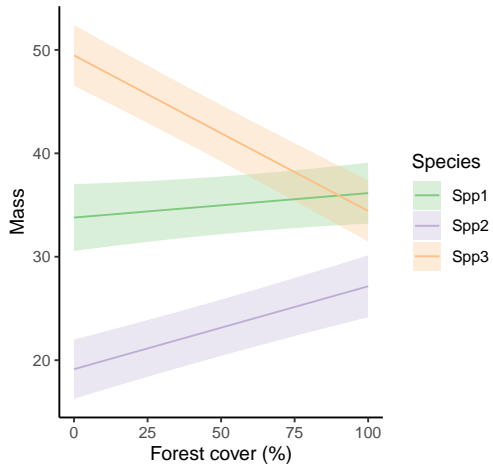
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: mass ~ species * forest + (1 | sites)
## Data: fishDat
##
## REML criterion at convergence: 807.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.3538 -0.5868  0.0548  0.6296  2.1122
##
## Random effects:
## Groups Name Variance Std.Dev.
## sites (Intercept) 25.931  5.092
## Residual          8.381  2.895
## Number of obs: 150, groups: sites, 15
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    33.78928     1.63228  20.701
## speciesSpp2    -14.65711     1.13614 -12.901
## speciesSpp3     15.68085     1.19467  13.126
## forest          0.02365     0.01353   1.747
## speciesSpp2:forest 0.05650     0.01720   3.285
## speciesSpp3:forest -0.17411     0.01785  -9.754
##
## Correlation of Fixed Effects:
##              (Intr) spcsS2 spcsS3 forest spcS2:
## speciesSpp2 -0.501
## speciesSpp3 -0.480  0.681
## forest      -0.513  0.734  0.702
## spcsSpp2:fr  0.390 -0.815 -0.532 -0.771
## spcsSpp3:fr  0.387 -0.546 -0.835 -0.764  0.584
```

First challenge results (cont.)

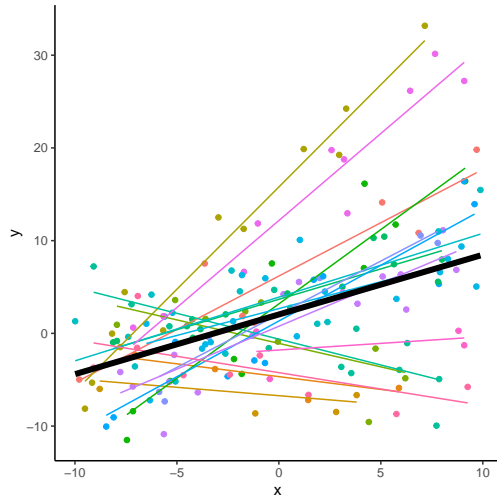
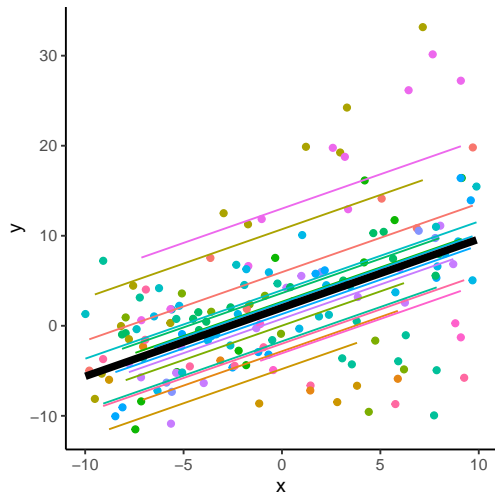
Fixed effects model: $\text{lm}(\text{mass} \sim \text{forest} * \text{species})$



Mixed effects model: $\text{lmer}(\text{mass} \sim \text{forest} * \text{spec})$



More random effects: slopes!



Random slopes + intercepts

Suppose that y wasn't just higher or lower at each site, but that the effect of x on y was higher or lower at each site

$$\hat{y} = X\beta + U\zeta_{int} + U_x\zeta_{slope}$$

$$y \sim \text{Normal}(\hat{y}, \sigma)$$

$$\zeta_{int} \sim \text{Normal}(0, \sigma_{int})$$

$$\zeta_{slope} \sim \text{Normal}(0, \sigma_{slope})$$

- X = fixed effects matrix (e.g. intercept, temperature)
- β = fixed effects coefficients
- U = random intercept matrix (e.g. sites)
- U_x = random slopes matrix (e.g. temperature)
- $\zeta_{int}, \zeta_{slope}$ = random intercept and slope coefficients
- $\sigma, \sigma_{int}, \sigma_{slope}$ = variance terms

Random slope and intercept example:

```
#Intercept varies with site, and slope of x can  
# also vary with site (both hierarchical)  
mm2 <- lmer(y ~ x + (x|site),data=dat)  
summary(mm2)
```

```
## Linear mixed model fit by REML ['lmerMod']  
## Formula: y ~ x + (x | site)  
## Data: dat  
##  
## REML criterion at convergence: 900.6  
##  
## Scaled residuals:  
##      Min       1Q   Median       3Q      Max   
## -2.10500 -0.64857  0.02414  0.61137  2.22996   
##  
## Random effects:  
## Groups   Name      Variance Std.Dev. Corr  
## site     (Intercept) 35.2210  5.9347  
##          x           0.7889  0.8882   0.82  
## Residual                9.3162  3.0522  
## Number of obs: 160, groups: site, 16  
##  
## Fixed effects:  
##              Estimate Std. Error t value  
## (Intercept)   2.0383    1.5091    1.351  
## x             0.6438    0.2275    2.830  
##  
## Correlation of Fixed Effects:  
##      (Intr)  
## x 0.790
```

Results from lmer model:

- Random effects:
 - *residual, slope, and site* variance (σ , σ_{int} , σ_{slope})
 - Correlation b/w intercept and slope = 0.82
 - Sites with higher intercept *also* have a higher slope
- Fixed effects:
 - Intercept and slope estimates

Model matrices

X: Fixed effects model matrix

```
## (Intercept)      x
## 1          1 -4.248450
## 2          1  5.766103
## 3          1 -1.820462
## 4          1  7.660348
## 5          1  8.809346
## 6          1 -9.088870
```

U: Random intercept model matrix

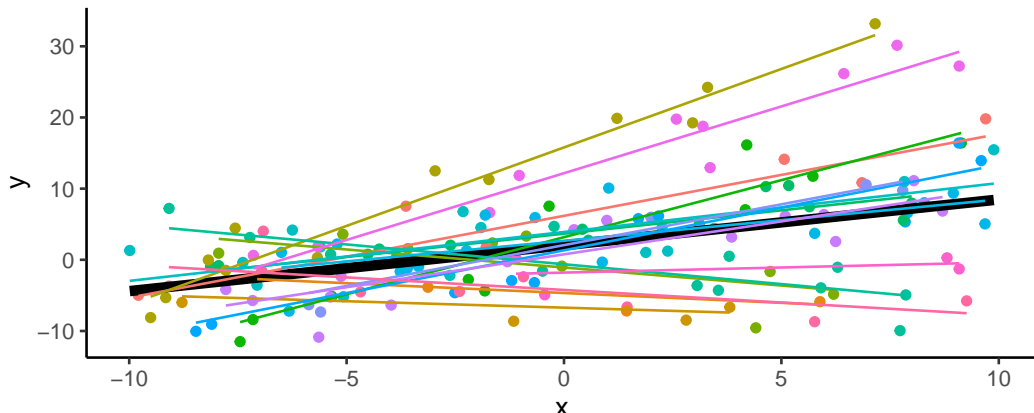
```
## sitea siteb sitec sited sitee sitef siteg siteh sitei
## 1      0      0      0      0      0      0      1      0      0
## 2      0      0      0      0      0      0      0      0      0
## 3      0      0      0      0      0      1      0      0      0
## 4      0      0      0      0      0      0      0      0      0
## 5      0      0      0      0      0      0      0      0      0
## 6      0      0      0      0      0      0      0      0      0
## sitej sitek sitel sitem siten siteo sitep
## 1      0      0      0      0      0      0      0
## 2      1      0      0      0      0      0      0
## 3      0      0      0      0      0      0      0
## 4      0      0      0      0      1      0      0
## 5      0      0      0      0      0      1      0
## 6      0      0      0      0      0      0      1
```

U_x: Random slope model matrix

```
## sitea siteb sitec sited sitee sitef siteg siteh sitei
## 1      0      0      0      0      0 0.00 -4.25      0      0
## 2      0      0      0      0      0 0.00  0.00      0      0
## 3      0      0      0      0      0 -1.82  0.00      0      0
## 4      0      0      0      0      0 0.00  0.00      0      0
## 5      0      0      0      0      0 0.00  0.00      0      0
## 6      0      0      0      0      0 0.00  0.00      0      0
## sitej sitek sitel sitem siten siteo sitep
## 1 0.00      0      0      0 0.00 0.00 0.00
## 2 5.77      0      0      0 0.00 0.00 0.00
## 3 0.00      0      0      0 0.00 0.00 0.00
## 4 0.00      0      0      0 7.66 0.00 0.00
## 5 0.00      0      0      0 0.00 8.81 0.00
## 6 0.00      0      0      0 0.00 0.00 -9.09
```


Mixed effect model results

- Regression line of x on y is allowed to move up or down for each site (random intercepts)
- Slope of regression line can be more or less steep for each site (random slopes)
- Changes in intercepts and slopes are *normally distributed*, and in this example are *correlated* with each other



Why do we need to do any of this?

“My supervisor told me to just use site as a fixed effect. Why can’t I do that?”

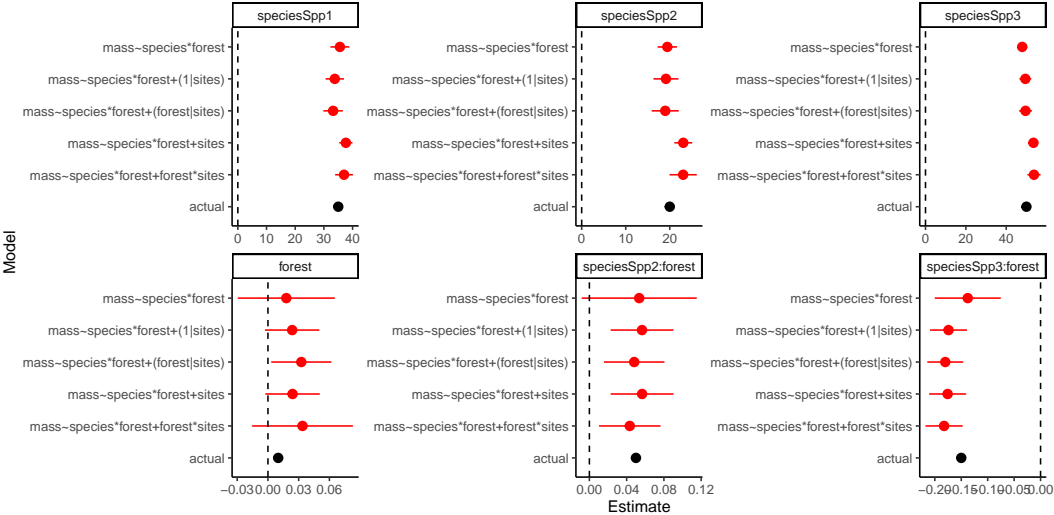
- You can do it this way, but you may encounter the following problems:
 - You lose the *partial pooling* that occurs in mixed effects models = Worse estimates of site effects!
 - You lose 1 d.f. for each site = Type II error \uparrow = You may not find the fixed effect of interest, even if it’s there!
 - Sites with low sample sizes may cause your models to break
 - People³ who have read statistics books published after 1980 may ask questions
- However, if you have a low number of sites (1-10), fixed effects may work better
 - Hard to estimate σ_{site} if number of sites is low
 - If stakes are high, it may be better to be more conservative about site intercepts
 - Easier to interpret (p-values, ANOVA, etc.)

³e.g. me

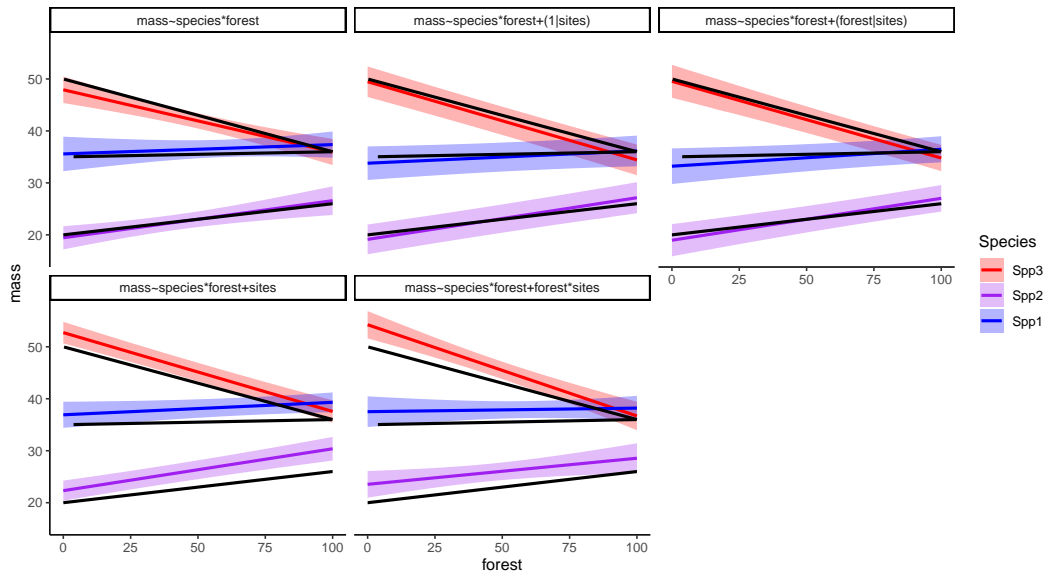
Second challenge

- Let's go back to the fish weight model. . .
- Fit a mixed effects model with the fixed effects you're interested in (**forest cover**, **species**), and include **site** as a random effect (*intercept or slope*)
- Your supervisor doesn't like hierarchical models, and tells you to just use site as another fixed term in an `lm` model. Do you get different results if you use their approach?

Second challenge results



Second challenge results (cont.):



Part 2: GLMMs

What if my response variable is non-normal?

- Linear model (LM)

$$\hat{y} = X\beta$$

$$y \sim \text{Normal}(\hat{y}, \sigma)$$

- Linear mixed effects model (LMM)

$$\hat{y} = X\beta + U\zeta$$

$$y \sim \text{Normal}(\hat{y}, \sigma)$$

$$\zeta \sim \text{Normal}(0, \sigma_{\text{site}})$$

- Generalized linear model (GLM)

$$\text{logit}(\hat{\phi}) = X\beta$$

$$y \sim \text{Binomial}(\hat{\phi})$$

- Generalized linear mixed effects model (GLMM)

$$\text{logit}(\hat{\phi}) = X\beta + U\zeta$$

$$y \sim \text{Binomial}(\hat{\phi})$$

$$\zeta \sim \text{Normal}(0, \sigma_{\text{site}})$$

How do I fit GLMMs?

- `glmer()` and `glmer.nb()` from `lme4` work for Binomial, Poisson, and Negative Binomial data

```
library(lme4)
glmm1 <- glmer.nb(y~x+(x|site),data=dat2) #Negative binomial GLMM
summary(glmm1) #glmer.nb takes a long time to run
```

- `glmmTMB()` from `glmmTMB` works for those above, *plus* a bunch of others (Zero-inflation, Beta-binomial), and it's generally faster

```
library(glmmTMB)
glmm2 <- glmmTMB(y~x+(x|site),data=dat2,family=nbinom2())
summary(glmm2) #Similar results, but quicker
```


Fitting GLMMs

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: Negative Binomial(5.1294) ( log )
## Formula: y ~ x + (x | site)
## Data: dat2
##
##      AIC      BIC    logLik deviance df.resid
##    627.8    646.3   -307.9    615.8      154
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.3745 -0.7098 -0.3946  0.5108  2.5367
##
## Random effects:
##   Groups Name      Variance Std.Dev. Corr
##   site  (Intercept) 1.43500  1.1979
##         x           0.02878  0.1697  0.92
## Number of obs: 160, groups: site, 16
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.32746    0.32166   1.018  0.3087
## x            0.10830    0.04681   2.314  0.0207 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##   (Intr)
## x 0.799
```

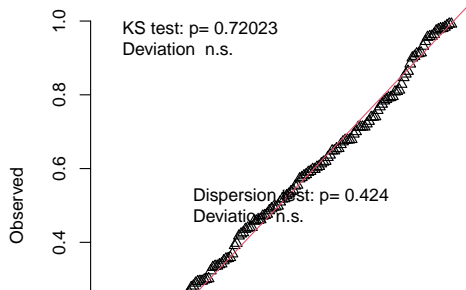
```
## Family: nbinom2 ( log )
## Formula: y ~ x + (x | site)
## Data: dat2
##
##      AIC      BIC    logLik deviance df.resid
##    627.8    646.2   -307.9    615.8      154
##
## Random effects:
##
## Conditional model:
##   Groups Name      Variance Std.Dev. Corr
##   site  (Intercept) 1.43543  1.1981
##         x           0.02892  0.1701  0.92
## Number of obs: 160, groups: site, 16
##
## Dispersion parameter for nbinom2 family (): 5.12
##
## Conditional model:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.34132    0.32172   1.061  0.2887
## x            0.11026    0.04697   2.348  0.0189 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual checks on glmmTMB models

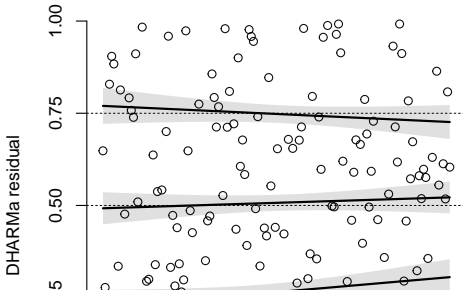
- You can extract deviance residuals and make your own plots, but I usually use `simulateResiduals` from DHARMA (see [here](#))
- DHARMA also has useful functions for checking overdispersion and zero-inflation (found [here](#))
- This works fairly well across a whole range of models, so I would try to learn it!

DHARMA residual

QQ plot residuals

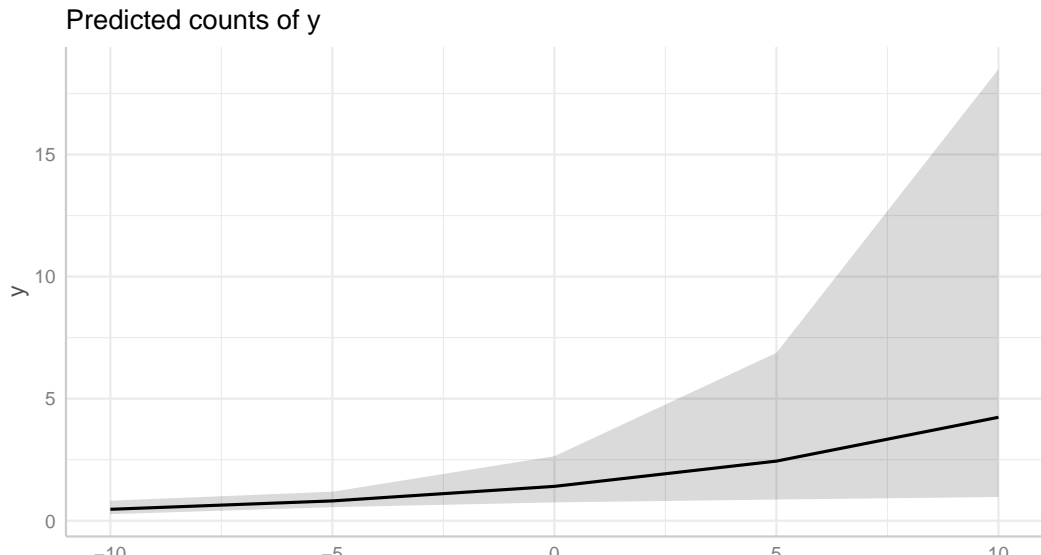


Residual vs. predicted
No significant problems detected



Partial residual plots for glmmTMB models

- `ggpredict()` from `library(ggeffects)` works with `glmmTMB` models



Part 3: Hypothesis testing and inference

I fit a model... now what?

- Congratulations, your model ran and it met the assumptions of regression...
- Time to see if your predictions are supported by your data!
- For each of the terms in your model:
 - Was the term “important” in your model?
 - If so, what direction was the effect in?
- How well did your model fit your data (overall)?
- Some other bits of advice

Step 1: was the term “important”?

In linear models, this is done using an ANOVA F-test (also shown at the bottom of a `summary()` statement):

```
## Analysis of Variance Table
##
## Response: mpg
##           Df Sum Sq Mean Sq F value    Pr(>F)
## disp       1  808.89   808.89  73.9959 1.788e-09 ***
## gear       1   0.14    0.14  0.0132   0.9093
## Residuals 29 317.01   10.93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

But wait, there's more!

Unfortunately, if we have more than 1 term in the model, the order of terms can change your answer:

- ```
Analysis of Variance Table
##
Response: mpg
Df Sum Sq Mean Sq F value Pr(>F)
disp 1 808.89 808.89 73.9959 1.788e-09 ***
gear 1 0.14 0.14 0.0132 0.9093
Residuals 29 317.01 10.93

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```
- ```
## Analysis of Variance Table
##
## Response: mpg
##           Df Sum Sq Mean Sq F value    Pr(>F)
## gear       1  259.75   259.75  23.762 3.595e-05 ***
## disp       1  549.28   549.28  50.248 8.465e-08 ***
## Residuals 29  317.01    10.93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Solution: use a Type II ANOVA

- We're usually interested in the importance of a term on its own, *not just after other terms are accounted for* (Type I ANOVA).
- For this, we use a Type II ANOVA. This can be done using `drop1()` (or `Anova()` in the `car` package)

- ```
Single term deletions
##
Model:
mpg ~ disp + gear
Df Sum of Sq RSS AIC F value Pr(>F)
<none> 317.01 79.383
disp 1 549.28 866.30 109.552 50.2476 8.465e-08 ***
gear 1 0.14 317.16 77.397 0.0132 0.9093

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

- ```
## Single term deletions
##
## Model:
## mpg ~ gear + disp
##      Df Sum of Sq    RSS      AIC F value    Pr(>F)
## <none>                 317.01   79.383
## gear    1      0.14 317.16  77.397  0.0132  0.9093
## disp    1    549.28 866.30 109.552 50.2476 8.465e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Interactions make ANOVA testing a bit strange

- If interactions are present, it doesn't really make sense to test the main terms **because they depend on the interactions**
- For example:

```
## Single term deletions
##
## Model:
## mpg ~ gear * disp
##      Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                 278.60  77.250
## gear:disp  1     38.412 317.01  79.383   3.8604 0.05943 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Anova from the car package tests other terms *after* interactions, but the meaning isn't the same. I prefer to keep things simple and just use drop1()

```
## Anova Table (Type II tests)
##
## Response: mpg
##      Sum Sq Df F value    Pr(>F)
## gear      0.14  1  0.0145   0.90502
## disp    549.28  1 55.2038 4.312e-08 ***
## gear:disp  38.41  1   3.8604  0.05943 .
## Residuals 278.60 28
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

GLMs, LMMs, and GLMMs

- `drop1` also works with GLMs, LMMs, and GLMMs, but we use a χ^2 *likelihood ratio test* rather than an F-test
- Unfortunately, different numbers of data points change your likelihood, so this can wreck LRTs. *This happens a lot if you have NAs in your predictor columns*, so clean up your data before using it in models.

ML vs REML

- Maximum likelihood (ML) estimates of variance (e.g. SD) are always smaller than the actual variance (biased)
- Restricted maximum likelihood (REML) uses a mathematical trick to get around this, but...
- This means that models with different numbers of terms don't have the same REML estimates
- Likelihood between these models technically can't be compared!

Solution:

- ① Use ML if comparing between models with different fixed effects, then...
- ② Re-fit with REML once you've decided on a model

Hypothesis testing (cont.)

How do I know this effect is different from x ? - Use Wald Z-test (2-sided p-value from Z-test)

```
## x  
## 0
```

- `glht` from `library(multcomp)` works with `lmer` models if you are comparing between coefficients (e.g. "Is treatment A different from B and C?")

Model selection

How many terms should be in my model?

- Same principle as in regular linear models: **what do you think the process is?**
 - Just because a term is “not significant” doesn’t mean it should be dropped out!
 - Just because a term is “significant” doesn’t mean it should be left in!
 - Avoid selecting models based on R^2 or AIC alone. Avoid stargazing⁴(hunting around for “better” p-values or AIC scores)
- To test whether terms are important in predicting your data (similar to), use *likelihood-ratio tests*
 - `drop1(model, test='Chisq')`
 - AIC tests usually say the same thing as LR tests

⁴"My God, it's full of stars!" -2001, A Space Odyssey