

Exploring Hierarchical Linear Mixed Models

1/49

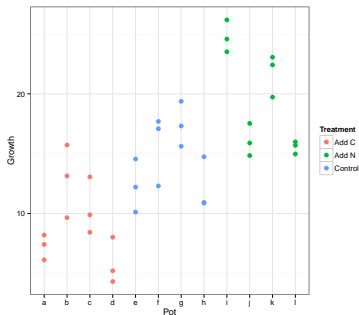
Last time... A Greenhouse Experiment testing C:N Ratios

Sam was testing how changing the C:N Ratio of soil affected plant leaf growth. He had 3 treatments. A control, a C addition, and a N addition. To ensure that any one measurement of one leaf wasn't a fluke, Sam measured 3 leaves per plant. The design is as follows:

- 3 Treatments (Control, C, N)
- 4 Pots of Plants per Treatment
- 3 Leaves Measured Per Pot

2/49

Our Greenhouse Experiment testing C:N Ratios



Leaf Growth = Treatment Effect + Pot Variation + Error

3/49

Fitting a Varying Intercept Model for the Greenhouse Experiment

```
library(nlme)
plantLME <- lme(Growth ~ Treatment, random = ~ 1|Pot, data=plants)
```

4/49

Today

1. Diagnostics in Mixed Models
2. Assessing hypotheses
3. Hierarchical Models
4. Variable Slope-Intercept models
5. A Procedure for Assessing Mixed Model Structure
6. A Taste of Generalized Linear Mixed Models

5/49

Diagnostics in a Mixed Model Framework

6/49

Diagnostics:

1. Is there a relationship between fitted and residual values?
2. Are the residuals normally distributed?
3. Is there a relationship between fitted and residual values at the group level?
4. Are the random effects normally distributed?

7/49

Fitted Values at the Group or Individual Level

```
fitted(plantLME, level=0)
```

```
#      a      a      a      b      b      b
# 9.102532 9.102532 9.102532 9.102532 9.102532 9.102532
#      c      c      c      d      d      d
# 9.102532 9.102532 9.102532 9.102532 9.102532 9.102532
#      e      e      e      f      f      f
# 14.403830 14.403830 14.403830 14.403830 14.403830 14.403830
#      g      g      g      h      h      h
# 14.403830 14.403830 14.403830 14.403830 14.403830 14.403830
#      i      i      i      j      j      j
# 19.534654 19.534654 19.534654 19.534654 19.534654 19.534654
#      k      k      k      l      l      l
# 19.534654 19.534654 19.534654 19.534654 19.534654 19.534654
# attr(,"label")
# [1] "Fitted values"
```

```
fitted(plantLME, level=1)
```

```
#      a      a      a      b      b      b
# 7.457340 7.457340 7.457340 12.425751 12.425751 12.425751
#      c      c      c      d      d      d
# 10.310714 10.310714 10.310714 6.216325 6.216325 6.216325
#      e      e      e      f      f      f
# 12.534204 12.534204 12.534204 15.552437 15.552437 15.552437
#      g      g      g      h      h      h
# 17.099945 17.099945 17.099945 12.428732 12.428732 12.428732
#      i      i      i      j      j      j
# 24.176660 24.176660 24.176660 16.474719 16.474719 16.474719
```

8/49

Residual Values at the Group or Individual Level

```
residuals(plantLME, level=0)

#           a           a           a           b           b
# -2.9707142 -1.6822088 -0.8998211  6.6253576  0.5521013
#           b           c           c           c           d
#  4.0388463  0.7860161  3.9622825 -0.6705262 -1.0759376
#           d           d           e           e           e
# -4.7815614 -3.8838344 -2.1943799  0.1592191 -4.2750746
#           f           f           f           g           g
#  2.6840961  3.2900798 -2.0974722  1.2180073  4.9705525
#           g           h           h           h           i
#  2.9111881 -3.5337961 -3.4706145  0.3381943  5.0448169
#           i           i           j           j           j
#  3.9780774  6.6445239 -4.6891834 -3.6370164 -2.0014874
#           k           k           k           l           l
#  3.5273807  0.1951570  2.8855014 -4.5607985 -3.5464814
#           l
# -3.8404901
# attr(,"label")
# [1] "Residuals"

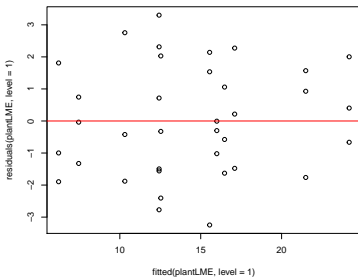
residuals(plantLME, level=1)

#           a           a           a           b
# -1.325521457 -0.037016093  0.745371614  3.302138793
#           b           b           c           c
# -2.771117508  0.715627450 -0.422165179  2.754101256
#           c           c           d           d
#  1.097027517  1.014006070  1.095351014  0.007627061
```

9/49

Residuals v. Fitted at Subsample Level

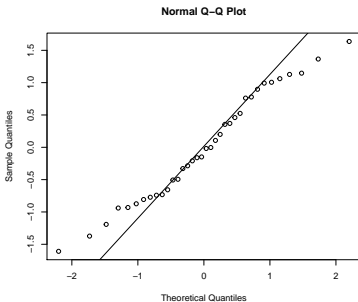
```
plot(fitted(plantLME, level=1), residuals(plantLME, level=1))
abline(a=0, b=0, col="red")
```



10/49

Normality of Residuals at Subsample Level

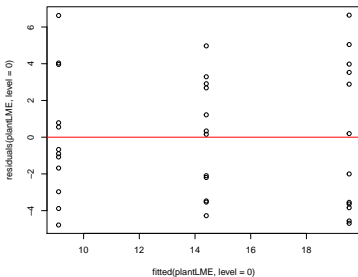
```
qqnorm(residuals(plantLME, type="pearson", level=1))  
qqline(residuals(plantLME, type="pearson", level=1))
```



11/49

Residuals v. Fitted at Group Level

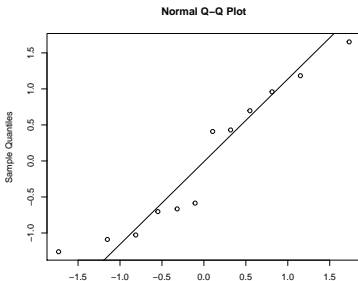
```
plot(fitted(plantLME, level=0), residuals(plantLME, level=0))  
abline(a=0, b=0,col="red")
```



12/49

Normality of Random Effects

```
r<-ranef(plantLME)[,1]
qqnorm(r/sd(r))
qqline(r/sd(r))
```



13/49

Evaluating Model Results

14/49

Evaluating Random Effects

- ▶ Do this first, as random effect structure alters fixed effects outcomes
- ▶ Use χ^2 tests for random effects - for a REML fit without any random effects, use `gls` OR
- ▶ If variance components small, need simulation approaches - see `RLRsim`

15/49

Evaluating the Greenhouse Experiment Random Effects

```
plantGLS <- gls(Growth ~ Treatment, data=plants)
anova(plantLME, plantGLS)

#           Model df      AIC      BIC    logLik  Test
# plantLME      1  5 177.2003 184.6829 -83.60017
# plantGLS      2  4 193.6161 199.6021 -92.80804 1 vs 2
#           L.Ratio p-value
# plantLME
# plantGLS 18.41574 <.0001
```

16/49

Evaluating the Greenhouse Experiment Random Effects

```
library(PLRsim)
exactRLRT(plantLME)

#
# simulated finite sample distribution of RLRT.
#
# (p-value based on 10000 simulated values)
#
# data:
# RLRT = 18.4157, p-value < 2.2e-16
```

More accurate, better if random effects small

17/49

Evaluating the Greenhouse Experiment Fixed Effects

```
anova(plantLME, type="marginal")

#           numDF denDF   F-value p-value
# (Intercept)      1    24 27.149545 <.0001
# Treatment        2     9  8.915836 0.0073
```

DF Denominator = # Groups - DF Treatment - 1

Note type="marginal" - type II

18/49

OR LRT using ML for Fixed Effects

```
plantLME.ml <- lme(Growth ~ Treatment, random = ~ 1|Pot,  
                  data=plants, method="ML")  
Anova(plantLME)  
  
# Analysis of Deviance Table (Type II tests)  
#  
# Response: Growth  
#           Chisq Df Pr(>Chisq)  
# Treatment 17.832  2  0.0001342
```

19/49

Why F-Tests or ML for Fixed Effects?

- ▶ F values calculated using differences in Residual Sums of Squares
- ▶ F tests with $DF = \# \text{ Groups} - DF \text{ Treatment} - 1$ are conservative
- ▶ χ^2 tests with REML for fixed effects are anti-conservative (type I prone)
- ▶ Use χ^2 tests for random effects - for a REML fit without any random effects, use `gl`s
- ▶ If variance components small, need simulation approaches - see `RLRsim`

20/49

Contrasts with Fixed Effects

```
library(contrast)
contrast(plantLME,
  list(Treatment = c("Add C", "Add N")),
  list(Treatment = "Control"))

# lme model parameter contrast
#
# Contrast      S.E.      Lower      Upper      t df
# -5.301297 2.470563 -10.34004405 -0.2625508 -2.15 31
# 5.130825 2.470563 0.09207789 10.1695711 2.08 31
# Pr(>|t|)
# 0.0398
# 0.0462
```

21/49

Exercise: Random Effects on Richness

- ▶ Fit data from RIKZ survey
- ▶ Random Effect of Beach ONLY
- ▶ Compare to No Beach Effect Model (gls)
- ▶ Visualize Random Effects

```
se.ranef <- function(obj)
  ranef(obj, standardized=T)/sapply(ranef(obj), sd)
```

22/49

Fitting Comparison Models

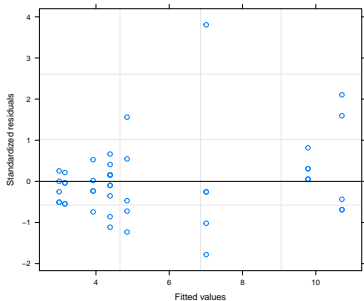
```
rikzInt <- lme(Richness ~ 1, random = ~1|Beach, data=rikz)
#
rikzNoBeach <- gls(Richness ~ 1, data=rikz)
```

```
anova(rikzInt, rikzNoBeach)
```

```
#           Model df      AIC      BIC    logLik  Test
# rikzInt       1  3 267.1142 272.4668 -130.5571
# rikzNoBeach   2  2 274.3695 277.9379 -135.1848 1 vs 2
#           L.Ratio p-value
# rikzInt
# rikzNoBeach 9.255284 0.0023
```

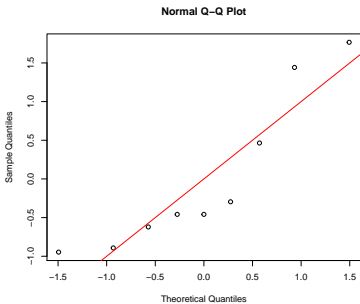
23/49

Fit Is Ok...



24/49

Fit Is Ok...

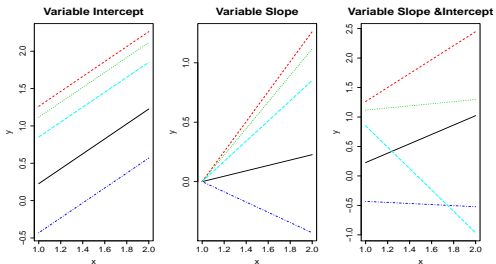


25/49

Full Multilevel/Hierarchical Models

26/49

Types of Multilevel Models



27/49

Variable Intercept Models Useful with Group Level Predictors

$$y_i = \alpha_{j[i]} + \beta_i x_i + \epsilon_{ij}$$

$$\alpha_{j[i]} \sim N(\mu_\alpha + x_j, \sigma_\alpha^2)$$

$$\epsilon_{ij} \sim N(0, \sigma^2)$$

where i = individual sample, j = group

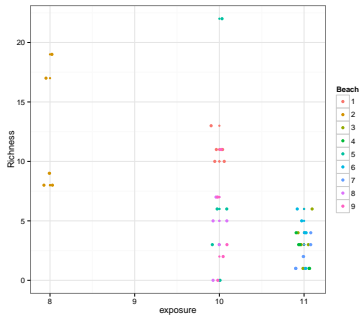
28/49

Each Site has a Unique Exposure - How does it Affect Species Richness?



29/49

Each Site has a Unique Exposure - How does it Affect Species Richness?



Data from Zuur et al. 2009

30/49

A Variable Intercept Model for Wave Exposure

```
varIntExposure<- lme(Richness ~ exposure,  
                    random =~ 1 |  
                    Beach, data=rikz)  
anova(varIntExposure, type="m")  
  
#           numDF denDF  F-value p-value  
# (Intercept)      1   36 21.28013 <.0001  
# exposure         1    7 15.48873 0.0056
```

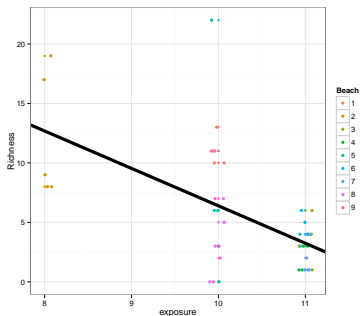
31/49

Plot Fit Using Extracted Components

```
adf <- as.data.frame(t(fixef(varIntExposure)))  
#  
names(adf)[1] <- "intercept"  
#  
p+ geom_abline(data=adf,  
              mapping=aes(intercept=intercept, slope=exposure),  
              lwd=2)
```

32/49

Plot Fit Using Extracted Components



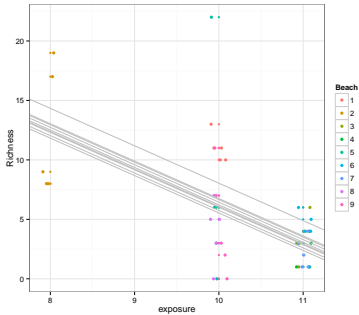
33/49

Plot Fit Using Extracted Components

```
adf2 <- as.data.frame(coef(varIntExposure))
#
names(adf2)[1] <- "intercept"
#
p+ geom_abline(data=adf2,
               mapping=aes(intercept=intercept, slope=exposure),
               color="grey")
```

34/49

Plotting is a Wee Bit Tricky...



35/49

Variable Slope-Intercept Model with No Group Level Predictors

$$y_i = \alpha_{j[i]} + \beta_{[j]i}X + \epsilon_{ij}$$

$$\begin{pmatrix} \alpha_{[i]j} \\ \beta_{[i]j} \end{pmatrix} \sim N \left(\begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix}, \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix} \right)$$

36/49

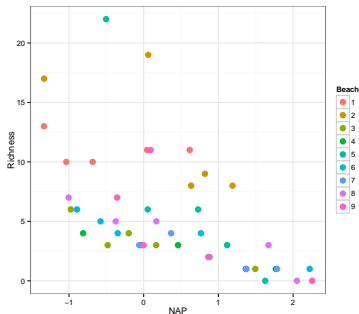
General Protocol for Model Fitting

Variable slope? Intercept? Slope-Intercept? Why do I evaluate Fixed Effects?

1. Start with model with all fixed and random effects that may be important. Evaluate with diagnostics.
2. Evaluate random effects with full model of fixed effects (AIC, χ^2)
3. Evaluate fixed effects with reduced random effects (F Tests)
4. Model diagnostics again...
5. Draw inference from model

37/49

How Important is Tide Height?



38/49

Three Models with Different Random Effects

```
varSlope <- lme(Richness ~ NAP,  
               random = ~ 0 + NAP | Beach, data=rikz)  
#  
varInt <- lme(Richness ~ NAP,  
             random = ~ 1 | Beach, data=rikz)  
#  
varSlopeInt <- lme(Richness ~ NAP,  
                  random = ~ 1 + NAP | Beach, data=rikz)
```

39/49

Does Slope Vary Randomly?

```
ranef(varSlope)  
  
#           NAP  
# 1 -4.920828e-10  
# 2  2.658914e-09  
# 3  2.901381e-09  
# 4 -3.135875e-10  
# 5 -7.360857e-09  
# 6  5.445407e-09  
# 7 -4.259714e-09  
# 8  2.708332e-09  
# 9 -1.287793e-09
```

40/49

SD in Variable Slope Model is Small

```
summary(varSlope)

....

# Random effects:
# Formula: ~0 + NAP | Beach
#               NAP Residual
# StdDev: 0.0001139427 4.159929
#
....
```

41/49

Evaluation of Different Random Effects Models

```
anova(varSlope, varSlopeInt)

#           Model df      AIC      BIC    logLik  Test
# varSlope      1  4 260.2010 267.2458 -126.1005
# varSlopeInt   2  6 244.3839 254.9511 -116.1919 1 vs 2
#           L.Ratio p-value
# varSlope
# varSlopeInt 19.81713 <.0001

anova(varSlopeInt, varInt)

#           Model df      AIC      BIC    logLik  Test
# varSlopeInt   1  6 244.3839 254.9511 -116.1919
# varInt        2  4 247.4802 254.5250 -119.7401 1 vs 2
#           L.Ratio p-value
# varSlopeInt
# varInt        7.096378 0.0288
```

42/49

A Simulation-Based Approach to Random Slope Assessment

```
#Assumes no covarnace between slope-intercept random effect
varSlopeInt2 <- lme(Richness ~ NAP,
                   random = list(~1|Beach, ~0+NAP|Beach), data=rikz)

#To test slope random effect
# m = slope only, mA = all random effect, m0 = model without slope
exactRLRT(m=varSlope, mA=varSlopeInt2, m0=varInt)

#
# simulated finite sample distribution of RLRT.
#
# (p-value based on 10000 simulated values)
#
# data:
# RLRT = 1.4804, p-value = 0.0991
```

43/49

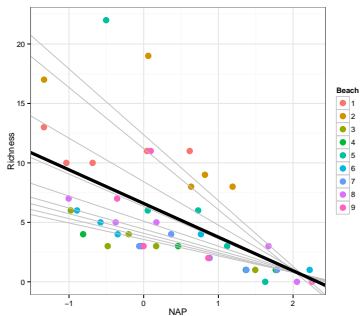
Evaluation of Fixed Effects

```
anova(varSlopeInt, type="m")

#           numDF denDF  F-value p-value
# (Intercept)      1   35 27.13837 <.0001
# NAP              1   35 15.32416 4e-04
```

44/49

Final Model



45/49

Exercise: RIKZ Tide Height and Shoreline Angle

- ▶ Evaluate the effect of angle1 (sample angle) & NAP on Richness
- ▶ Note: You already know the slope of the NAP relationship doesn't vary randomly
- ▶ Check for a NAP*angle1 interaction

46/49

nlme versus lme4

- ▶ nlme - can work like nls for flexible nonlinear specification
- ▶ nlme - can accomodate specified correlation structures
- ▶ lmer - can fit more complex models
- ▶ glmer - can fit Generalized Linear Mixed Models (GLMM)

47/49

lmer for a GLMM

```
library(lme4)
angI_lmer <- glmer(Richness ~ angle1*NAP + (1 | Beach),
                  data=rikz, family=poisson(link="log"))
anova(angI_lmer, type="m")

# Analysis of Variance Table
#           Df Sum Sq Mean Sq F value
# angle1     1  0.321    0.321  0.3207
# NAP        1 41.359   41.359 41.3593
# angle1:NAP 1  4.948    4.948  4.9484
```

Testing GLMMs is trickier, but RLRsim and LRT fixed effects can (usually) get you there - see <http://glimm.wikidot.com/faq> for more!

48/49

Or we can always go Bayesian!

```
library(MCMCglmm)
angI_MCMCglmm <- MCMCglmm(Richness ~ angle1*NAP, random =~ Beach,
                          data=rikz, family="poisson", verbose=F)

summary(angI_MCMCglmm)$Gcovariances

#           post.mean  1-95% CI  u-95% CI  eff.samp
# Beach 0.3892111 0.0588447 0.8634476 117.9257

summary(angI_MCMCglmm)$solutions

#           post.mean      1-95% CI      u-95% CI
# (Intercept) 1.832249814 1.3732623849 2.271485074
# angle1      -0.004186804 -0.0082018560 -0.001161515
# NAP         -0.782624141 -1.0237336673 -0.558918778
# angle1:NAP  0.002903180 0.0009772816 0.006211612
#           eff.samp pMCMC
# (Intercept) 220.58814 0.001
# angle1      54.22571 0.020
# NAP        27.83021 0.001
# angle1:NAP  27.14478 0.024
```

49/49

Ideal because we can get credible intervals on all fixed random effects!