R-Adelaide 2016

Day 3:
Mixed effects models in R

Dr Steven Delean - Benham Rm 103
steven.delean@adelaide.edu.au

School of Biological Sciences - The University of Adelaide

Linear mixed effects models

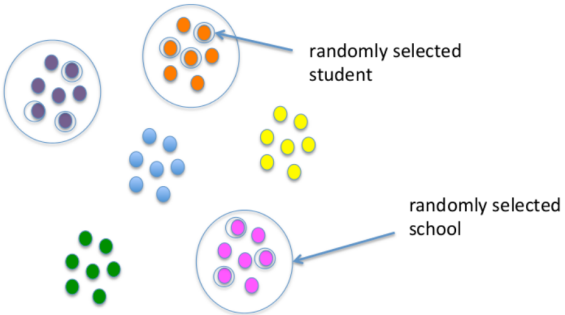
- Linear mixed-effects models are used when you have random effects
  - Occurs when your subjects or units are grouped
  - Groups are assumed to be randomly sampled from a "population" of groups
- Example situations include:
  - when you divide up plots and apply separate treatments to the parts (plot is the random group)
  - when your sampling design is nested, such as quadrats within transects; transects within woodlots; woodlots within districts (transects, woodlots, and districts are the random groups)
  - when you have measurements on related individuals (family is the random group)
  - when you measure subjects or units repeatedly (subject or unit is the random group)

What are we actually talking about?

Alternative names for mixed effcts models

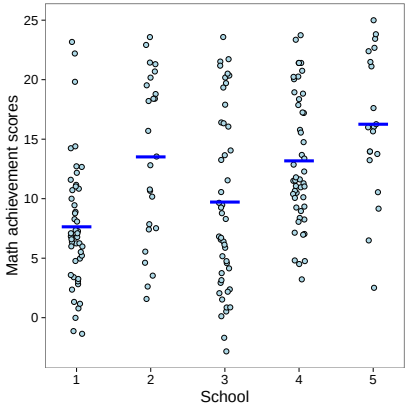
- Mixed model, LMM/GLMM
- Random effects model, mixed eects model
- Longitudinal regression, repeated measures model
- Hierarchical model, multilevel model
- Covariance pattern model

Example of clustered data



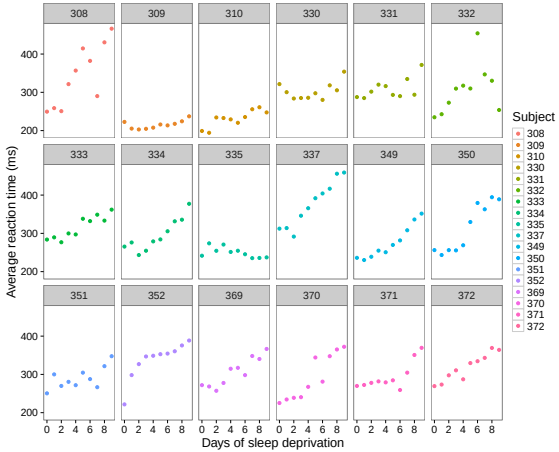
Example of clustered data

- Record students' math achievement scores where students were randomly selected from schools in a state



Example of longitudinal data

- 18 study participants are sleep deprived for several days
- Each day, measure that subject's reaction time



Definition of LME Models

In matrix notation a linear model can be represented as

$$\hat{y}_i = \beta_0 + \beta_1 x_1 + \cdots + \epsilon_i$$

$$y = \boldsymbol{X}\boldsymbol{\beta} + \epsilon$$

Definition of LME Models

In matrix notation an LME model can be represented as

$$y = X\beta + Zb + \epsilon$$

where

- y is a vector of observations, with mean $E(y) = X\beta$
- β is a vector of fixed effects
- $\boldsymbol{b}$ is a vector of random effects with mean $E(b) = 0$ and variance-covariance matrix $\mathrm{var}(b) = G$
- ϵ is a vector of IID random error terms with mean $E(\epsilon) = 0$ and variance $\mathrm{var}(\epsilon) = R$
- X and Z are matrices of regressors relating the observations y to β and b, respectively.

- Fixed effects have coefficients
- Random effects have estimates of variation

Mixed models in R

- Maximum likelihood or restricted maximum likelihood (REML) estimates of the parameters in linear mixed-effects
- Historically, `nlme` was main package for mixed models in R
  - Includes linear and nonlinear fixed effects
  - Variety of variance functions and correlation structures
- Development then shifted to package `lme4`
- The term "mixed effects" denotes model with both fixed- and random-effects terms in a linear predictor

Mixed model formulas - `lmer`

- As for linear models, first two arguments to `lmer`
  - Formula specifying the model
  - Data
- Formula for `lm`:

```
response ~ expr
```

- Formula for `lmer`:

```
response ~ FEexpr + (REexpr1 | factor1) + (REexpr2 | factor2) + ...
```

where,

- FEexpr is fixed-effects model matrix, X
- Random-effects terms (REexpr1 | factor1) and (REexpr2 | factor2) determine random-effects model matrix, Z

Mixed model formulas - `lmer`

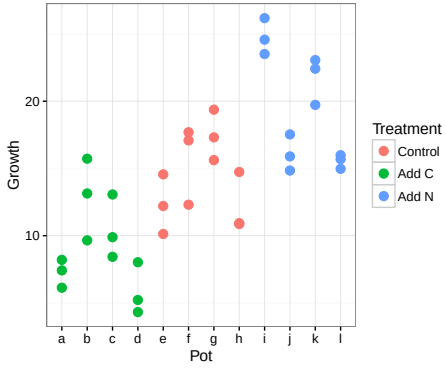
Formula	Alternative	Meaning
(1 | g)	1 + (1 | g)	Random intercept with fixed mean
0 + offset(o) + (1 | g)	-1 + offset(o) + (1 | g)	Random intercept with a priori means
(1 | g1/g2)	(1 | g1)+(1 | g1:g2)	Intercept varying among g1 and g2 within g1
(1 | g1)+(1 | g2)	1 + (1 | g1) + (1 | g2)	Intercept varying among g1 and g2
x + (x | g)	1 + x + (1 + x | g)	Correlated random intercept and slope
x + (x || g)	1 + x + (1 | g) + (0 + x | g)	Uncorrelated random intercept and slope

A Greenhouse Experiment testing C:N Ratios

- Does changing the C:N Ratio of soil affects plant leaf growth?
  - 3 treatments: control, C addition, and N addition
  - 4 pots per treatment
  - 3 leaves per plant

```
plants <- read.csv("data-raw/nestedGrowth.csv")
head(plants)
# Set Control group as reference level
plants$Treatment <- with(plants, relevel(Treatment, ref = "Control"))
# Plot data
plantPlot <- ggplot(data=plants, aes(x=Pot, y=Growth,color=Treatment,
       ↪ group=Pot)) + theme_bw()
plantPlot + geom_point(size=3)
```

```
  X Treatment Pot Growth
1 1     Add C   a   6.13
2 2     Add C   a   7.42
3 3     Add C   a   8.20
4 4     Add C   b  15.73
5 5     Add C   b   9.65
6 6     Add C   b  13.14
```

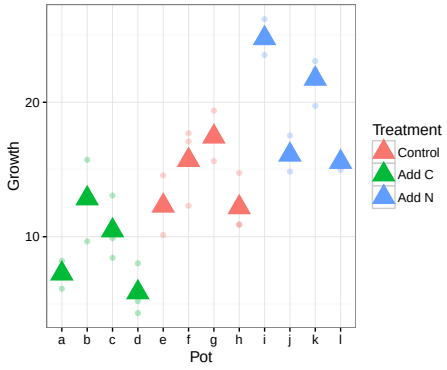


Simplifying analysis - averaging

- If design is balanced (and no interest in within pot variance) average within each pot

```
stat_sum_single <- function(fun, geom="point", ...) {
  stat_summary(fun.y=fun, shape=17, geom=geom, size=6, ...)
}
plantPlot+ geom_point(alpha=0.3) + stat_sum_single(mean)
# ANOVA
library(car)
library(dplyr)
meanPlants <- plants %>% group_by(Pot, Treatment) %>% summarise(Growth =
    ↪ mean(Growth))
plantGrowth <- lm(Growth ~ Treatment, data = meanPlants)
Anova(plantGrowth)
```

```
Anova Table (Type II tests)

Response: Growth
             Sum Sq Df F value    Pr(>F)
Treatment 217.67771  2 8.91584 0.0073312 **
Residuals 109.86628  9
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



Classical ANOVA Error Decomposition with Expected Mean Squares

- $SS_{Total} = SS_{Treatment} + SS_{PotError} + SS_{WithinPotError}$

```r
plantAOV <- aov(Growth ~ Treatment + Error(Pot), data = plants)
summary(plantAOV)
```

```

Error: Pot
          Df   Sum Sq  Mean Sq F value    Pr(>F)
Treatment  2 653.0331 326.5166 8.91584 0.0073312 **
Residuals  9 329.5988  36.6221
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: Within
          Df   Sum Sq  Mean Sq F value Pr(>F)
Residuals 24 97.68967 4.070403
```

Multilevel/Clustered/Hierarchical/Mixed Model

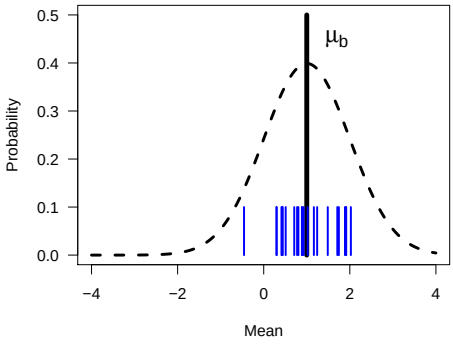
- $i =$ treatment, $j =$ pot, $k =$ subsample

$$\hat{y}_{ijk} = b_{j[i]} + \beta_i X + \epsilon_{ijk}$$

$$b_{j[i]} \sim \mathcal{N}(\mu_b, \sigma_b^2)$$

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

- What is a random effect?
  - Each pot has different parameter value
  - Unlike in linear model, intercept values are constrained around Normal distribution



Some Points about Multilevel Models

- Flexible. Can accomodate varying slope, intercept, intercept-slope models
- Linear mixed models are solved using Restricted Maximum Likelihood (REML)
  - ML estimation produces downward biased estimates of random effect variances
- As group level effects are drawn from the same distribution, Best Linear Unbiased Predictors (BLUPs) are shrunk towards grand mean - basically, we use information from all groups to inform within group means - useful for unbalanced designs

Many R Packages for Multilevel Models

- nlme - from Pinhero and Bates 2009
- lmer - bleeding edge by Doug Bates
- MCMCglmm - uses Bayesian techniques & MCMC (similar syntax to nlme)
- glmmADMB - interface for AD Model Builder
- etc

Fitting a Varying Intercept Model for the Greenhouse Experiment - lmer

```
plantLMM <- lmer(Growth ~ Treatment + (1 | Pot), data=plants)
summary(plantLMM)
```

```
Linear mixed model fit by REML
t-tests use  Satterthwaite approximations to degrees of freedom ['lmerMod
    ↪ ']
Formula: Growth ~ Treatment + (1 | Pot)
   Data: plants

REML criterion at convergence: 167.2

Scaled residuals:
       Min         1Q     Median         3Q        Max
-1.60894241 -0.73479195 -0.08365836  0.76529269  1.63672837

Random effects:
 Groups   Name        Variance  Std.Dev.
 Pot      (Intercept) 10.850564 3.294019
 Residual              4.070403 2.017524
Number of obs: 36, groups:  Pot, 12

Fixed effects:
              Estimate Std. Error       df  t value   Pr(>|t|)
(Intercept)    14.403830   1.746952  8.999998  8.24512 1.7379e-05 ***
TreatmentAdd C -5.301297   2.470563  8.999998 -2.14579   0.060450 .
TreatmentAdd N  5.130825   2.470563  8.999998  2.07678   0.067617 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) TrtmAC
TretmntAddC -0.707
TretmntAddN -0.707  0.500
```

$$\hat{y}_{ijk} = b_{j[i]} + \beta_i X + \epsilon_{ijk}$$

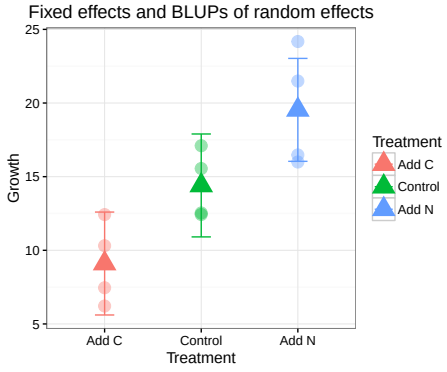
```
plantLMM.Ests <- as.data.frame(cbind(round(coef(plantLMM)$Pot, 3), round(
    ↪ ranef(plantLMM)$Pot, 3)))
names(plantLMM.Ests)[4] <- "BLUP.Random.Intercepts"
print(plantLMM.Ests)
```

	(Intercept)	TreatmentAdd C	TreatmentAdd N	BLUP.Random.Intercepts
a	12.759	-5.301	5.131	-1.645
b	17.727	-5.301	5.131	3.323
c	15.612	-5.301	5.131	1.208
d	11.518	-5.301	5.131	-2.886
e	12.534	-5.301	5.131	-1.870
f	15.552	-5.301	5.131	1.149
g	17.100	-5.301	5.131	2.696
h	12.429	-5.301	5.131	-1.975
i	19.046	-5.301	5.131	4.642
j	11.344	-5.301	5.131	-3.060
k	16.362	-5.301	5.131	1.958
l	10.864	-5.301	5.131	-3.540

Visualizing Fixed Effects and BLUPS

Note use of altered model for ease of plotting

```
plantLMM2 <- lmer(Growth ~ 0 + Treatment + (1 | Pot), data=plants)
summary(plantLMM2)
# Fixed effect estimates
fixDF <- data.frame(summary(plantLMM2)$coef)
fixDF$Treatment <- factor(gsub("Treatment", "", rownames(fixDF)), levels
    ↪ = c("Add C", "Control", "Add N"))
# Random effect BLUPs
ranDF <- data.frame(Estimate = unlist(ranef(plantLMM, condVar = TRUE)$Pot
    ↪ ) + rep(fixDF$Estimate[c(2, 1, 3)], rep(4, 3)))
ranDF$Treatment <- factor(as.character(meanPlants$Treatment), levels = c(
    ↪ "Add C", "Control", "Add N"))
# Plot
p <- ggplot(fixDF, aes(x=Treatment, y=Estimate, color=Treatment)) +
  theme_bw() +
  ylab("Growth") +
  geom_errorbar(aes(ymin=Estimate-Std..Error*2, ymax=Estimate+Std..Error*
      ↪ 2), width = 0.2) +
  geom_point(shape = 17, size = 6) +
  ggtitle("Fixed effects and BLUPs of random effects")
p + geom_point(mapping = aes(x=Treatment, y=Estimate), data = ranDF, size
    ↪ =4, alpha = 0.4)
```

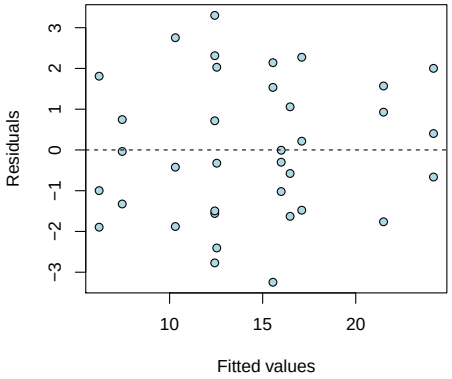


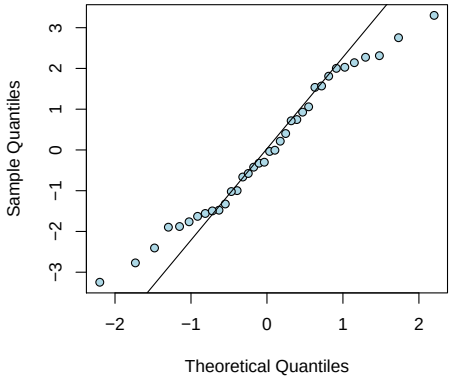
What next?

- Diagnostics in Mixed Models
- Assessing hypotheses
- Hierarchical Models
- Variable Slope-Intercept models
- A Procedure for Assessing Mixed Model Structure
- A Taste of Generalized Linear Mixed Models

Residuals diagnostics for observational units

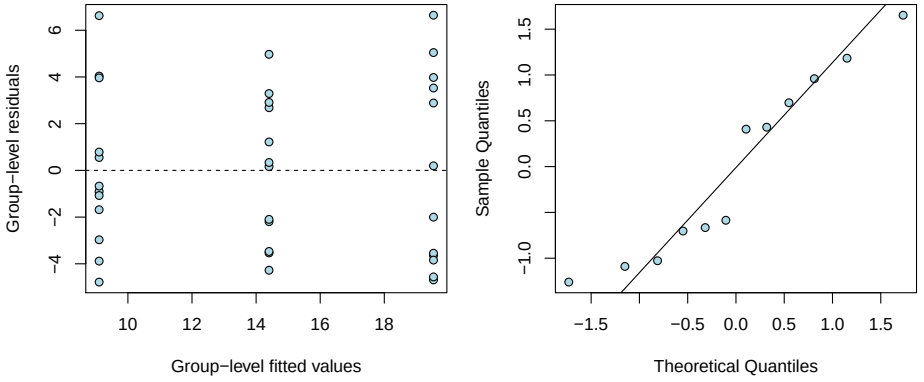
```r
# Plot residuals vs fitted values
plot(fitted(plantLMM), residuals(plantLMM), xlab="Fitted values", ylab="Residuals", pch=21, bg="lightblue")
abline(a=0, b=0, col="black", lty=2)
# Q-Q plot of residuals
qqnorm(residuals(plantLMM, type="pearson"), main="", pch=21, bg="lightblue")
qqline(residuals(plantLMM, type="pearson"))
```





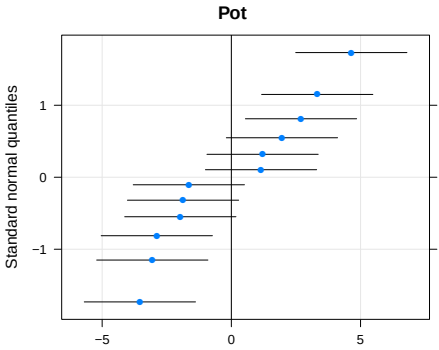
Residuals diagnostics at group-level

```r
# Plot residuals vs fitted values at group level
plot(predict(plantLMM2, re.form=~0), plants$Growth - predict(plantLMM2, re.form=~0), xlab="Group-level fitted values", ylab="Group-level residuals"
    ↪ , pch=21, bg="lightblue")
abline(a=0, b=0, col="black", lty=2)
# Q-Q plot of random effects
r <- unlist(ranef(plantLMM, condVar = TRUE)$Pot)
qqnorm(r/sd(r), main="", pch=21, bg="lightblue")
qqline(r/sd(r))
```



Random effects BLUPS

```
1  library(lattice)
2  qqmath(ranef(plantLMM, condVar = TRUE), strip = FALSE)
3  # Test for heterogeneity
4  library(RLRsim)
5  exactRLRT(plantLMM)
```



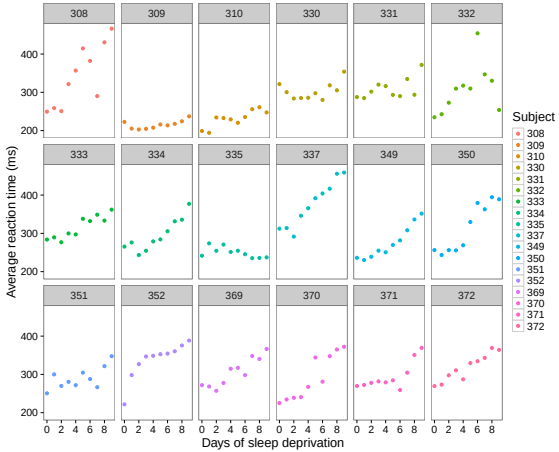
Model inference

```r
# Global testing for fixed effects
drop1(plantLMM, test = "Chisq")
# Using contrasts for planned comparisons
library(lsmeans)
trt.lsm <- lsmeans(plantLMM, "Treatment")
contrast(trt.lsm, "trt.vs.ctrl", adjust = "none")
# Calculating bootstrap confidence intervals - parameter contrasts
confint(plantLMM, method = "boot", boot.type = "basic", nsim = 100,
        ↪ parallel = "multicore", ncpus = 8)
# Calculating bootstrap confidence intervals - Treatment means
confint(plantLMM2, method = "boot", boot.type = "basic", nsim = 100,
        ↪ parallel = "multicore", ncpus = 8)
```

```
Single term deletions

Model:
Growth ~ Treatment + (1 | Pot)
          Df       AIC     LRT   Pr(Chi)
<none>       185.60902
Treatment  2 194.71732 13.1083 0.0014242 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1



                          2.5 %       97.5 %
.sig01             1.06733677269  4.920323813
.sigma             1.34597084395  2.635024473
(Intercept)       11.74738290407 17.594991016
TreatmentAdd C   -10.70452432258 -1.196672772
TreatmentAdd N     0.01244979266  9.969124979

                         2.5 %       97.5 %
.sig01             1.359105917  4.849531572
.sigma             1.411892016  2.549262368
TreatmentControl  10.245785828 18.000970897
TreatmentAdd C     6.097065684 12.975063104
TreatmentAdd N    15.193178109 24.386497203
```

Example dataset: sleepstudy

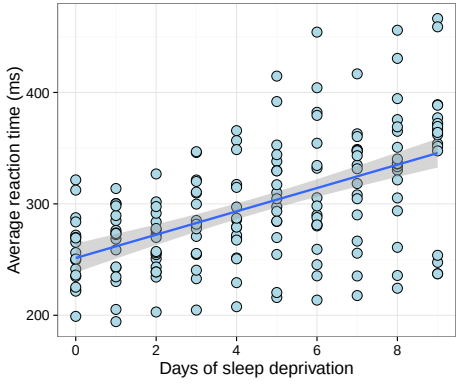
- 18 study participants are sleep deprived for several days
- Each day, measure that subject's reaction time



A simple linear model

```
#+BEGIN_SRC R :results output raw :exports both
lm.simple <- lm(Reaction ~ Days, data=sleepstudy)
summary(lm.simple)
```

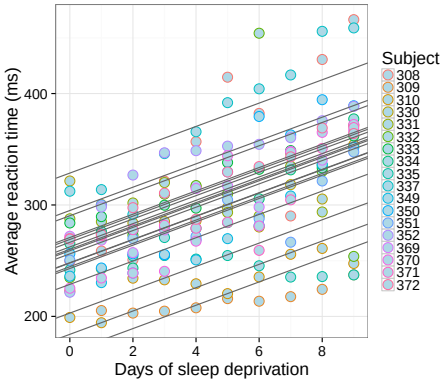
```

Call:
lm(formula = Reaction ~ Days, data = sleepstudy)

Residuals:
    Min      1Q  Median      3Q     Max
-110.85  -27.48    1.55   26.14  139.95

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   251.41       6.61   38.03  < 2e-16 ***
Days           10.47       1.24    8.45  9.9e-15 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 47.7 on 178 degrees of freedom
Multiple R-squared:  0.286,     Adjusted R-squared:  0.282
F-statistic: 71.5 on 1 and 178 DF,  p-value: 9.89e-15
```



Fitting a varying intercepts model with `lm`

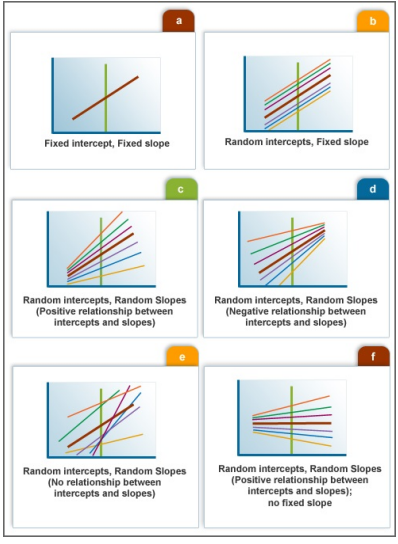
```
#+BEGIN_SRC R :results output raw :exports both
lm.complete <- lm(Reaction ~ 0 + factor(Subject) + Days, data=sleepstudy)
summary(lm.complete)
```

```

Call:
lm(formula = Reaction ~ 0 + factor(Subject) + Days, data = sleepstudy)

Residuals:
    Min      1Q  Median      3Q     Max
-100.54  -16.39   -0.34   15.22  131.16

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
factor(Subject)308  295.031     10.447    28.2   <2e-16 ***
factor(Subject)309  168.130     10.447    16.1   <2e-16 ***
factor(Subject)310  183.898     10.447    17.6   <2e-16 ***
factor(Subject)330  256.119     10.447    24.5   <2e-16 ***
factor(Subject)331  262.333     10.447    25.1   <2e-16 ***
factor(Subject)332  260.199     10.447    24.9   <2e-16 ***
factor(Subject)333  269.056     10.447    25.8   <2e-16 ***
factor(Subject)334  248.199     10.447    23.8   <2e-16 ***
factor(Subject)335  202.967     10.447    19.4   <2e-16 ***
factor(Subject)337  328.618     10.447    31.5   <2e-16 ***
factor(Subject)349  228.732     10.447    21.9   <2e-16 ***
factor(Subject)350  266.500     10.447    25.5   <2e-16 ***
factor(Subject)351  242.995     10.447    23.3   <2e-16 ***
factor(Subject)352  290.319     10.447    27.8   <2e-16 ***
factor(Subject)369  258.932     10.447    24.8   <2e-16 ***
factor(Subject)370  244.599     10.447    23.4   <2e-16 ***
factor(Subject)371  247.881     10.447    23.7   <2e-16 ***
factor(Subject)372  270.783     10.447    25.9   <2e-16 ***
Days                 10.467      0.804    13.0   <2e-16 ***
---
```



Mixed models enable us to account for variability

- Varying intercepts
- Varying slopes



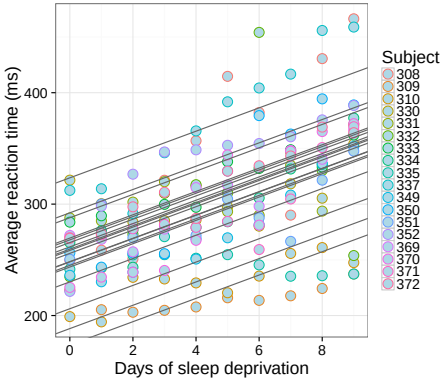
Hence there's gradient between

- Mixed models estimate varying parameters (intercepts and/or slopes) with pooling among levels (rather than considering them fully independent)
- **complete pooling**: Single overall intercept
  - `lm (Reaction ~ Days)`
- **no pooling**: One **independent** intercept for each plot
  - `lm (Reaction ~ Days + factor(Subject)`
- **partial pooling**: Inter-related intercepts
  - `lm (Reaction ~ Days + (1 | Subject))`

Fitting a varying intercepts model with `lmer`

```
#+BEGIN_SRC R :results output raw :exports both
fm.rInt <- lmer(Reaction ~ Days + (1 | Subject), sleepstudy)
summary(fm.rInt)
```

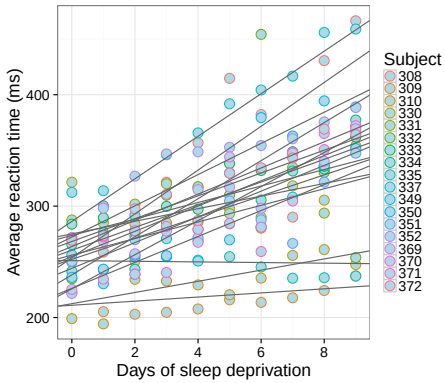
```
Linear mixed model fit by REML
t-tests use  Satterthwaite approximations to degrees of freedom ['lmerMod
        ↪ ']
Formula: Reaction ~ Days + (1 | Subject)
   Data: sleepstudy

REML criterion at convergence: 1786.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.226 -0.553  0.011  0.519  4.251

Random effects:
 Groups   Name        Variance Std.Dev.
 Subject  (Intercept) 1378     37.1
 Residual             960      31.0
Number of obs: 180, groups:  Subject, 18

Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)  251.405      9.747  22.800    25.8   <2e-16 ***
Days          10.467      0.804 161.000    13.0   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
Days -0.371
```



Fitting a varying intercepts and slopes model with `lmer`

```
fm.rIntSlp <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)
summary(fm.rIntSlp)
```

```
Linear mixed model fit by REML
t-tests use  Satterthwaite approximations to degrees of freedom ['lmerMod
    ↪ ']
Formula: Reaction ~ Days + (Days | Subject)
   Data: sleepstudy

REML criterion at convergence: 1743.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.954 -0.463  0.023  0.463  5.179

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Subject  (Intercept) 612.1    24.74
          Days         35.1     5.92    0.07
 Residual             654.9    25.59
Number of obs: 180, groups:  Subject, 18

Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)   251.41       6.82  17.00   36.84  < 2e-16 ***
Days           10.47       1.55  17.00    6.77  3.3e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
Days -0.138
```



Why no p-values in the `lme4` summaries?

- Recall that p-values in the coefficient summary of fitted linear models are the probability of getting a test statistic as large (or larger) if the coefficient was indeed 0.
- Also recall that p-values are determined using null reference distributions
  - Under the null hypothesis, the test statistic has a known distribution
- In LME models, the null reference distribution is technically not known, at least not for unbalanced data
  - Thus the `lme4` authors elected to not output p-values based on a distribution that is not actually the distribution of the test statistic

How do I know if a coefficient is "significant"?

- Quick, common-sense way: look at the t value
- If the t value is greater than 2, then the coefficient is likely significant
  - In other words, the coefficient estimate is more than 2 standard errors away from 0
- Note:
  - t value = Estimate / Std. Error
  - Of course ask yourself if the result is practically significant as well

How do I know if a coefficient is "significant"?

- Another way is to compute 95% confidence intervals using the confint() function
  - If the interval contains 0, it is not significant
  - Further, confidence intervals give an indication of coefficient size and variability
- Say your fitted model object is `lme1` Two ways to compute confidence intervals are as follows:
  - `confint(lme1)` computes a likelihood profile and finds the appropriate cutoffs based on the likelihood ratio test
  - `confint(lme1, method="boot")` parametric bootstrapping (B = 500) with confidence intervals computed from the bootstrap distribution

Confidence intervals for the sleepstudy data

```
fm1 <- lmer(Reaction ~ Days + (Days|Subject), sleepstudy)
fm1.ci <- confint(fm1, method="boot")
fm1.ci
```

```
                2.5 %  97.5 %
.sig01       11.779  34.835
.sig02       -0.508   0.998
.sig03        3.320   8.213
.sigma       22.349  28.566
(Intercept) 238.046 265.029
Days          7.416  13.407
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: Reaction ~ Days + (Days | Subject)
   Data: sleepstudy

REML criterion at convergence: 1743.6

Scaled residuals:
   Min     1Q Median     3Q    Max
-3.954 -0.463  0.023  0.463  5.179

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Subject  (Intercept) 612.1    24.74
          Days         35.1     5.92    0.07
 Residual             654.9    25.59
Number of obs: 180, groups:  Subject, 18

Fixed effects:
            Estimate Std. Error t value
(Intercept)   251.41       6.82    36.8
Days           10.47       1.55     6.8

Correlation of Fixed Effects:
     (Intr)
Days -0.138
```

But I want p-values!

- The `lmerTest` package provides the kind of p-values SAS provides (based on Satterthwaite's approximations)
- Just load it `library(lmerTest)`, run `lmer` as usual and call summary on the object

```
library(lmerTest)
m <- lmer(dv ~ x1 + (x1 | g), data=df)
summary(m)
```

A column of p-values is included in the summary output. Again, they're approximate.

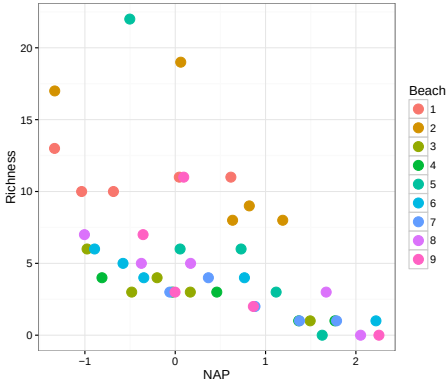
Exercise: Random Effects on Richness using `glmer`

- Poisson distribution is widely used for the description of count data
- Invertebrates species richness in 45 sites on Dutch coast beaches versus height above average sea level
- We will:
  - Fit data from RIKZ survey
  - Random Effect of Beach ONLY
  - Visualize Random Effects

Fitting and summarising a mixed model

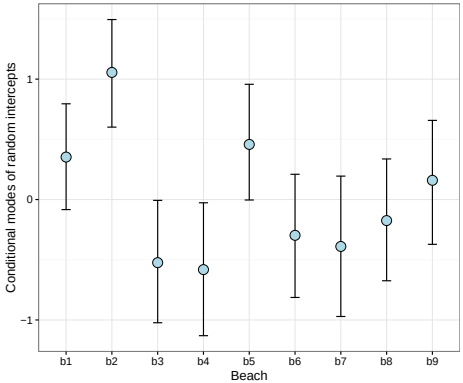
```
rikz <- read.csv("./data-raw/rikz.csv")
rikz$Beach <- as.factor(rikz$Beach)
rikz$angleRad <- rikz$angle1*pi/180
rikz$angleRad
mod1_glmer <- glmer(Richness ~ NAP*angleRad + (1|Beach), data=rikz,
      ↪ family="poisson")
summary(mod1_glmer)
anova(mod1_glmer, type = 'm')
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
      ↪ Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: Richness ~ NAP * angleRad + (1 | Beach)
   Data: rikz

     AIC      BIC   logLik deviance df.resid
   217.2    226.2   -103.6    207.2       40

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.747 -0.629 -0.212  0.273  2.624

Random effects:
 Groups Name        Variance Std.Dev.
 Beach  (Intercept) 0.307    0.554
Number of obs: 45, groups:  Beach, 9

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.850      0.216    8.57  < 2e-16 ***
NAP           -0.767      0.135   -5.70  1.2e-08 ***
angleRad      -0.228      0.105   -2.18    0.030 *
NAP:angleRad   0.184      0.083    2.22    0.026 *
---
```



Parametric bootstrap to estimate variation in random effects

```r
# Random intercepts using sim from the arm package
library(arm)
n.sim <- 1000
simu <- sim(mod1_glmer, n.sims=n.sim)
# Intercept
ranInt <- as.data.frame(t(apply(simu@ranef$Beach[, , 1], 2, quantile,
        ↪ prob=c(0.025, 0.5, 0.975))))
names(ranInt) <- c("lwr", "fit", "upr")
ranInt$Beach <- paste("b", 1:9, sep = "")
```



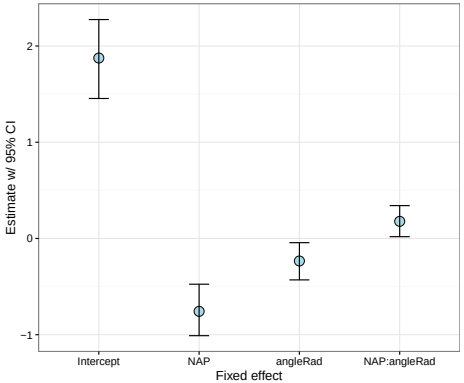
Parametric bootstrap confidence intervals

```
# mod1_glmer_ci <- confint(mod1_glmer, method="boot")
mod1_glmer_ci
```

```
                 2.5 %   97.5 %
.sig01        0.1635  0.80056
(Intercept)   1.4111  2.27656
NAP          -1.0454 -0.50724
angleRad     -0.5559 -0.00781
NAP:angleRad -0.0415  0.40827
```

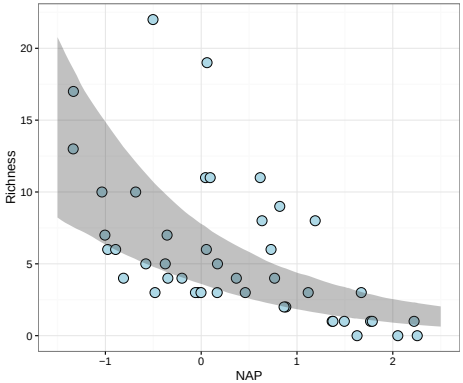
Parametric bootstrap fixed effects estimates

```r
# Predict fixed effects
coeff <- as.data.frame(t(apply(simu@fixef, 2, quantile, prob=c(0.025,
        ↪ 0.5, 0.975))))
names(coeff)[1:3] <- c('lwr', 'fit', 'upr')
coeff$term <- row.names(coeff)
coeff$term[1] <- "Intercept"
coeff$term <- factor(coeff$term, levels=c("Intercept", "NAP", "angleRad",
        ↪  "NAP:angleRad"))
summary(coeff)
```



Parametric bootstrap model predictions

```
# plotting the effect of NAP on the richness
nsim <- 1000
bsim <- sim(mod1_glmer, n.sim=nsim)
newdat <- data.frame(NAP=seq(-1.5, 2.5, length=100), angleRad=mean(rikz$
    ↪ angleRad))
Xmat <- model.matrix(~NAP*angleRad, data=newdat)
predmat <- matrix(ncol=nsim, nrow=nrow(newdat))
predmat <- apply(bsim@fixef, 1, function(x) exp(Xmat%*%x))
newdat$lwr <- apply(predmat, 1, quantile, prob=0.025)
newdat$upr <- apply(predmat, 1, quantile, prob=0.975)
newdat$Richness <- apply(predmat, 1, quantile, prob=0.5)
```



Differences between `lmer` and `lme`

- Package `nlme` contains function `lme`
- Package `lme4` contains function `lmer`
- `lme` has more flexibility in specifying the covariance structure
- `lmer` allows you to fit GLMMs (using `glmer`)
  - Includes `glmer.nb` for negative binomial variance models
- Different specification of random effects in call

```
# lmer
fm.rIntSlp <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)
summary(fm.rIntSlp)
# lme
fm.rIntSlp <- lme(Reaction ~ Days, random = ~ Days | Subject, sleepstudy)
summary(fm.rIntSlp)
```

Challenge

- Growth and maternal effects in mice (`mouse.csv`)
  - In a laboratory experiment, 20 female mice each gave birth to a litter of pups
  - Each litter was reared in a single cage, and each pup was weighed at several timepoints
- Some biological questions of interest are:
  - Do males and females differ with respect to their average growth rates?
  - Is there a maternal effect? (i.e., do individuals from the same litter tend to be similar?)
- See R script for data and code!