# Lecture 5:
# Inference with mixed effects models

# Choosing best models and trusting the statistics??

- How to choose best models with mixed model systems?

  …OR…

- How to evaluate the model parameter coefficients??

# Making inferences with lmer

- Open your practical and run the code to:

    '?pvalues'     Read it.


- No p-values provided by default.

- Largely because it is currently unclear how to calculate denominator dfs

- "Simple" methods provided here

- Simulation methods introduced at the end

# P values and all that

- P values are traditionally worked out by comparison to a theoretical distribution

- Significant parameter estimates are inferred by comparing $\beta/se > t_{den.df}$

- Significant variances are inferred by comparing $MS_{[variable]}/MS_{[resid]} > F_{num.df,\ den.df}$

- <u>Problem</u>: difficult to estimate d.f. properly with mixed effects models.

  => df dependent tests cannot be used for inference.

Do you need a P value??

# Some simpler approaches

- Likelihood ratio tests of nested models with comparison to $\chi^2$ distribution. AIC methods (recall AIC depends on likelihood)
  - Easiest, but possibly wrong for random effects

- Take F or t-values and estimate denominator dfs
  - Approximate, Not possible for crossed designs.
  - When high number of replicates, possibly OK (because of small changes in t-distribution for large n)

- Use the Kenward-Roger or Satterthwaite approximation of dfs
  - Pros: Inbuilt, theoretically justified.
  - Cons: In development – may give odd results.

# Likelihood ratio test

- Very simple – standard approach using nested models

    mod1 <- lmer(y~x + (1|group), …)

    mod0 <- update(mod1, ~.- x)

➢ anova(mod1, mod0)


➢ df here based on the <u>difference</u> in number of parameters in each model

➢ start from the largest model and move to smaller models..

➢ Previously: Start with random effects first, then deal with fixed effects later (Baayen et al. 2008)

➢ Now: only use for fixed effects?

# Likelihood ratio test

- So why not just use LRTs/AIC and be done with it?

- Problem comes in when comparing models with different random effects.

> mod1 <- lmer(y~x + (1+x|group), data)

> mod2 <- lmer(y~x + (1|group), data)

These are different variance components ($\sigma_a^2$, $\sigma_b^2$ etc)

- LRT based on maximum likelihood , but ML estimates not reliable when $\sigma^2 \to 0$ (=> LRT no longer reliable either)

- So need to do something else here to get reliable estimates

# Estimate error DF
# Calculate DFs by hand

- The "real" dfs are realistically going to be between
  - Number of reps (n) – 1 - number of parameters (p)
  - Number of reps (n) –  p - number of random groups (q)
  - Sometimes hard to work out what the correct number of groups are (e.g. cross-nested designs?)
- How do we count the random-effects?
  - Number of variances estimated?
  - Number of levels?

# Estimate error DF
# Satterthwaite Approximation

1. Satterthwaite's method

-> run package "`lmertest`", then `lmer()` give probs

```
> library(lmerTest)
> mod1<- lmer(radon~floor + (1|county), data=radon)
> summary(mod1)
…
Random effects:
 Groups   Name       Variance Std.Dev.
 county   (Intercept) 0.257    0.507
 Residual             1.200    1.095
Number of obs: 2369, groups:  county, 68

Fixed effects:
             Estimate Std. Error       df      t value    Pr(>|t|)
(Intercept)   1.3369    0.0698  69.6652      19.1   <2e-16 ***
floor        -0.7930    0.0751 2335.2309   -10.6   <2e-16 ***
```

# Estimate error DF
# Kenward-Rogers Approximation

- Based on approximating the denominator dfs and scaling the estimate of F.

- Can use instead of a likelihood ratio test

- Implemented using the `pbkrtest` package –

- `> library(pbkrtest)`

- `> KRmodcomp(mod0, mod1)`

- there is also an `anova` like wrapper in the `car` package, which performs the same test.

`> library(car)`

`> Anova(model, test='F')`

# Code 5.1

*Inference with lmer*

# Exercise 5.1: Plant damage dataset

Data: plants grown in different greenhouses with two levels of light and 3 levels of defoliation (damage). Light levels are applied at the level of each greenhouse. In each greenhouse plants are defoliated at 3 levels of defoliation (damage).

The scientist wants to know what effect light and damage have on plant growth.

1.  Open the dataset and examine it structure
2.  Fit a mixed effects model using `lmer` to test if light and damage affect growth.
3.  Does damage significantly affect growth?
4.  Is there an interaction?

[Ignore species, slides and blocks]

# Bootstrap simulation tests for parameters

- An analysis of parameter estimates
- Simulate data from the model (many times)
- Refit the model to these new response data
- Look at the distribution of the new coefficients.
- Quantiles of these parameters are the confidence intervals (95% quantiles = 95% confidence intervals)
- Can use the `confint()` function to do this

# Code 5.2

*Bootstrapped confidence intervals
for parameters*

# Order for evaluating mixed model

1.  Write out full model with all fixed and random effects

2.  Evaluate the random effects first, and reduce the random effects as appropriate (use bootstrapped estimates)

3.  Evaluate the fixed effects and reduce the fixed effects as appropriate (LRT, info theoretic approaches, estimated DF methods, bootstraps)

4.  Present data and parameter estimates on final selected model (use REML estimates)

# Exercise 5.2

Let's revisit the plantdamage data set.

1.  Evaluate whether damage effects in different greenhouses are explaining significant amounts of variation in the data using bootstrapping.

2.  Then simplify the fixed effects model as appropriate