Random Effects Models

Jesse Wheeler

Contents

1 Overview 1

2 Models for Random Effects 2

3 Hypothesis Testing 3

4 Estimation: REML and LRT 4

5 Factorial Random Effects Models 9

6 Mixed Effects Models 12

1 Overview

Objectives and overview

- This material is based Chapter 11–12 on the textbook "A First Course in Design and Analysis of Experiments" (Oehlert, 2000).
- Chapter outcomes include:
  - Students will be able to distinguish between a random effect and a fixed effect.
  - Students will be able to identify when random effects are appropriate.
  - Students will know how to implement random effects and mixed effects models.
  - Students will gain some familiarity with the `lme4` package in R.

Introduction

- So far, the treatment effects that we have been working with can be classified as *fixed-effects*: we wish to estimate the effect of specific (fixed) treatments.
- What if the treatments themselves are a random selection from a larger population?

Example: Farming Equipment
A farmer has 100 pieces of equipment, and would like to know how the equipment used may impact crop-yield. Rather than testing all 100 pieces of equipment, they *randomly select* g pieces of equipment, and performs a CRD with these equipment.

Random Effects
A random effect is appropriate to model treatments that are a *random sample* from a population of potential treatments.
Random effects are useful if the focus of inference is on the *population*, not the individual treatment effects.

- When *designing* for a random effect, the treatments should be a random draw from a larger population.

2 Models for Random Effects

Example: Carton factory

- We will build intuition of Random Effects models and designs using an example.

Carboard company
Suppose a company makes cardboard cartons for storing goods. They have 50 machines to make cartons, and want to understand the variation in quality for finished products.
They choose 10 machines at random from the 50, and make 40 cartons on each machine. The resulting cartons are tested for strength.

- This is a completely randomized design (CRD), with $g = 10$ treatments (the different machines), and $n_i = 40$, for a total of $N = 400$ EUs.
- The model we have been using for this type of data is:

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij}, \quad \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2).$$

- We are still going to use a very similar model. However, we are not as interested in the exact α_i for the 10 randomly chosen machines. Instead, we want to make inference on the larger population.
- We could proceed as usual, but that only tells us about the 10 specific machines that were randomly chosen.

  We would get an entirely different set of machines if we repeated the experiment!
- Thus, we want to introduce a *random effect*: let the effect α_i represent a random draw from a population of possible effects:

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij}, \qquad \alpha_i \overset{iid}{\sim} N(0, \sigma^2_\alpha), \quad \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2).$$

- We no longer assume $\sum_i \alpha_i = 0$, rather $E[\alpha_i] = 0$.
- We have introduced a new parameter, σ^2_α, which describes the spread of the effects.
- *We assume that the two random components are independent.* That is, α_i and ϵ_{ij} are independent.
- If they are independent, the variance of the observed data is increased:

$$\text{Var}(y_{ij}) = \text{Var}(\mu + \alpha_i + \epsilon_{ij}) = \sigma^2_\alpha + \sigma^2.$$

- The parameters σ^2_α and σ^2 are called *variance components*.
- If observations belong to the same treatment group, then they now have non-zero correlation:

$$\text{Cor}(y_{ij}, y_{ik}) = \begin{cases} 1 & j = k \\ \sigma^2_\alpha/(\sigma^2_\alpha + \sigma^2) & j \neq k \end{cases}$$

- We call this within treatment group correlation the *intraclass* correlation.

Key summary

- In a random effects model, the parameters are μ, σ^2, and σ_α^2.
- We are not interested in making inferences about the specific α_i and ϵ_{ij}, which are treated as random variables, not parameters.
- Inference wise, we are now interested in confidence intervals and hypothesis tests regarding μ and σ^2.

Practical Advice
In fixed effects, we generally prefer larger n and smaller g; for random effects, it's the opposite!

- We really would like a minimum of $g = 10$ groups for reliable estimates; otherwise, confidence intervals will be very large / meaningless.
- $g = 15$ or more is preferred.

3 Hypothesis Testing

Hypothesis Testing

- If we are testing for population effects based on our random sample, there are multiple ways to do this formally.
- The hypothesis we are generally interested in testing in this scenario is:

$$H_0 : \sigma_\alpha^2 = 0, \quad \text{vs} \quad H_0 : \sigma^2 > 0.$$

- Surprisingly, this leads to the same F-tests that we have been using before!
- Specifically, we have said that $E[\alpha_i] = 0$; thus, if H_0 is true, then $\text{Var}(\alpha_i) = 0$, and by necessity, $\alpha_i = 0$. Thus,

$$\sigma_\alpha^2 = 0 \iff \alpha_i = 0 \iff \mu_i = \mu_j \, \forall (i, j).$$

Example data in R

- The following example uses data synthetically generated with $\alpha_i \sim N(0, 1.37^2)$, $\epsilon_{ij} \sim N(0, 2^2)$, and $\mu = 10$.

```
synthData <- read.csv("synthData.csv")
mod <- aov(response ~ factor(trt), data = synthData)
summary(mod)

            Df Sum Sq Mean Sq F value  Pr(>F)
factor(trt) 11  115.5  10.501   3.101 0.00321 **
Residuals   48  162.6   3.387
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Above, we see a significant (random) treatment effect, rejecting the hypothesis that $\sigma_\alpha^2 = 0$.

- Note in this problem, the F statistic that is used matches the R output:

$$F_{DF_A, DF_E} = \frac{MS_A}{MS_E}.$$

  This will *not* be the case for factorial models or if the data are unbalanced.

- For factorials models, the denominator changes. We *can* do it by hand, but the output of `aov` does *not* match!

Estimation variance components

Estimation of Variance Components
In perfectly balanced data, the `aov()` output can be used to obtain estimates for σ^2_α; this follows a similar approach to estimating σ^2, like we did back in module 3.

- Recall: $\hat{\sigma}^2 = \text{MS}_\text{E}$.
- If the data are balanced, then we can subtract off the error variance, and divide by the number of observations (per-group):
$$\hat{\sigma}^2_\alpha = \frac{MS_\text{A} - MS_\text{E}}{n}$$
- In the previous R example, this is $(10.501 - 3.387)/5 = 1.4228$
- **NOTE:** This can be problematic if $MS_A < MS_E$, because we will get negative estimates! It will also not work if the data are not balanced, because then n is not the same for all groups. The approach we will discuss later does not have this same problem.

4 Estimation: REML and LRT

Beyond ANOVA: Likelihood Based Inference

- For factorial random effects models, or unbalanced data, the default `aov()` output will not work!
- Classic solutions rely on Expected Mean Squares (EMS) (See Oehlert (2000, Sections 11.3–11.6)). These approaches can be computed by hand.
- Unfortunately, these approaches have their weaknesses.
- We will shift from the textbook and rely on *likelihood-based methods*, which is in-line with modern practices.
- For the type of models considered so far (fixed-effects, or balanced random effect), the two approaches are the same.
- **Maximum Likelihood (ML)** finds the parameter estimates (mean and variances) that make the observed data most probable; confidence intervals are obtained using likelihood ratio tests.
- Roughly speaking, the likelihood is "what is the probability of observing what we observed, IF the parameter value is X".

  For an example, consider flipping a coin 10 times. There is a parameter of interest: p, the probability of heads. If we don't know if the coin is fair or not, we will want to perform an experiment (flip the coin 10 times), and count the number of heads.

  Now suppose we saw $n = 8$ heads. IF $p = 0$, then what is the probability that we saw 8 heads? This probability is zero! That means we can conclude that p is certainly not zero.

  What about if $p = 0.1$? What's the probability that we would get 8 heads? The probability of this? The probability is smaller than 3.7×10^{-7}. This is such a small outcome, it's highly unlikely that $p = 0.1$.

  Now what about $p = 0.8$? If this is the case, the probability of getting $n = 8$ heads is about 0.30. This is much more likely, and in fact we might be inclined to think that p *could* be 0.8.

  If we do this for all possible numbers $0 \leq p \leq 1$, and we treat this as a function, then we get the **likelihood funtion**: $L(p)$. More generally, we denote this as $L(\theta)$, and it can be computed

for any probability-based model. The core idea / takeaway: **Parameter values with higher-likelihood are better descriptions of reality; low-likelihood values do not correspond to observed data, so all else being equal, we should favor values with higher likelihood values.**

One of the main concerns of the likelihood function is "how large of a likelihood is large enough?" In our coin-flipping example, $L(0.8)$ happens to be the *Maximum of the Likelihood Function*, so we call $p = 0.8$ the *maximum likelihood estimate*. However, the likelihood of $p = 0.8$ is not all that different from the likelihood of $p = 0.81$ or $p = 0.79$, or even $p = 0.6$ (or at least the difference isn't large enough to convince us that $p = 0.6$ is not a real possibility).

It's often possible to derive an exact cutoff of the likelihood value that satisfies a Type-I controlled level-α confidence interval. In fact, our F-tests are exactly this type! In many cases, however, the exact cutoff to get a level-α confidence interval is not available. In these cases, we can often rely on Wilks' Theorem to give us an approximate answer! **If the log-likelihood of a single parameter model is within 1.92 units of the maximum, then it is "high-likelihood", and belongs in a 95% confidence interval.**

Restricted likelihood

- However, standard ML has a flaw when estimating variance components: it does not account for the degrees of freedom used to estimate the fixed effects (like the grand mean).
- This makes standard ML variance estimates *biased* (they underestimate the true variance).

  Example: For a normal distribution model, $y_{ij} \sim N(\mu_i, \sigma^2)$, the MLE of (μ, σ^2) is:

$$\left(\hat{\mu}_{MLE}, \hat{\sigma}^2_{MLE}\right) = \left(\bar{Y}_i, \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right).$$

  However, you may recall from you intro stats class that we usually don't estimate variance that way, instead we divide by n_1:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

  That's because the MLE for the variance component is biased low: on average, our estimate will be slightly smaller than the truth.

Restricted Maximum Likelihood (REML)

- To fix the bias in standard ML, modern software uses *Restricted Maximum Likelihood (REML).*
- REML accounts for the degrees of freedom lost in estimating fixed effects, REML provides *unbiased* estimates of σ^2 and σ^2_α.

  Example: building on the normally distributed model from before, the MLE for σ^2 was biased low:

$$\hat{\sigma}^2_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2.$$

  The REML estimate for this model is unbiased:

$$\hat{\sigma}^2_{REML} = s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

- Additionally, REML handles unbalanced data effortlessly and prevents the impossible "negative variance" estimates that can occur when using classical EMS methods.

Estimating variance components in R

- For variance components (random effects) We will shift from our `aov()` workflow and use the `lmer` function from the `lme4` package to do this in R:

```
# install.packages("lme4")
library(lme4)
lmer(response ~ 1 + (1 | randomEffect), data = data)
```

On the next slide, we will use this for the same `synthData` dataset we used the `aov` function for.

Example: Synthetic data

```
# Old Code:
# aov(response ~ factor(trt), data = synthData)
synthData$trt <- factor(synthData$trt)
lmer(response ~ (1 | trt), data = synthData)

Linear mixed model fit by REML ['lmerMod']
Formula: response ~ (1 | trt)
   Data: synthData
REML criterion at convergence: 255.9462
Random effects:
 Groups   Name        Std.Dev.
 trt      (Intercept) 1.193
 Residual             1.840
Number of obs: 60, groups:  trt, 12
Fixed Effects:
(Intercept)
      10.59
```

The important thing to note in the output above is that the estimated values match what we did by hand with the `aov()` output. Specifically, our estimate for treatment (`trt`) had $\hat{\sigma}_\alpha = 1.193$; if we square it, we get the same value as the by-hand calculation of $\hat{\sigma}^2_\alpha = 1.423$.

Example: Penicillin

- We will first consider the built-in `Penicillin` data in R, which contains the reduced diameter of bacterial growth, after penicillin application. Each measurement is taken on a specific `plate`, which is our random effect.

```
data("Penicillin")

output <- lmer(
  diameter ~ 1 + (1 | plate),
  data = Penicillin, REML = TRUE  # default
)
```

- What is the notation here doing? `1 + (1 | plate)`.
  - The model we are fitting is $y_{ij} = \mu + \alpha_i + \epsilon_{ij}$.
  - The `1` in the beginning is not strictly needed, but represents the grand mean μ.

  - The `(1 | plate)` represents the random effect. In `lme4`, the syntax `(A | B)` implies that B is the random effect grouping variable, and A is the effect that is random. For a simple random effects model, the mean (`1`) is always the thing that is random.

- Now let's look at the output.

```
summary(output)

Linear mixed model fit by REML ['lmerMod']
Formula: diameter ~ 1 + (1 | plate)
   Data: Penicillin

REML criterion at convergence: 613.3

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.41583 -0.55523  0.01212  0.59232  1.95241

Random effects:
 Groups   Name        Variance Std.Dev.
 plate    (Intercept) 0.09509  0.3084
 Residual             4.03333  2.0083
Number of obs: 144, groups:  plate, 24

Fixed effects:
            Estimate Std. Error t value
(Intercept)  22.9722     0.1788   128.5
```

Comparing to `aov`

- Since the data are balanced, the estimated parameters of this new `lmer` function will match what we can compute with `aov`:

Here, we can tell $n = 6$ because there are $g = 24$ plates, so $120 + 24 = 144$ total observations. Since it is balanced, there are $n = 144/24 = 6$ observations per group.

```
# Compute (MS_plate - MS_E) / n, n = 6
summary(aov(diameter ~ plate, data = Penicillin))

             Df Sum Sq Mean Sq F value Pr(>F)
plate        23  105.9   4.604   1.141  0.313
Residuals   120  484.0   4.033
```

Hypothesis testing

- The `lmer` function uses REML to get parameter estimates, i.e., $\hat{\sigma}^2_\alpha$.
- If we want to perform a hypothesis test for $\sigma^2_\alpha > 0$, we have two main options:
  1. Refit the model with `aov()`, and use the built-in F-test.
     - **Advantage:** Exact F-tests. These are more *powerful*.
     - **Disadvantage:** Only works if data are balanced, and the output P-value only matches if there is a single random factor in the model (otherwise, can compute by hand).

2. Use likelihood based inference procedures (shown below).
   - **Advantage:** Works whether or not data is balanced, and output P-values always "correct". More robust to violations of assumptions.
   - **Disadvantage:** Based on large-sample sizes, so less powerful if samples are small.

- We have already seen analysis of 1-factor random effects with `aov`, now let's do the second approach.

Likelihood Ratio Tests (LRT)

- The profile likelihood confidence intervals are based on the Likelihood Ratio Test.
- Instead of using exact F-tests derived from ratios of Mean Squares, modern mixed modeling often uses the *Likelihood Ratio Test (LRT)* for hypothesis testing.
- The LRT explicitly compares two competing models:
  1. A **Full Model** that includes the random effect.
  2. A **Reduced Model** (Null Model) that removes the random effect.
- The test measures how much the likelihood (fit) of the data drops when the random effect is removed.

$$\Lambda = \frac{\text{Likelihood}_{\text{reduced}}}{\text{Likelihood}_{\text{Full}}}, \quad \Longrightarrow \quad -2\log\Lambda \sim \chi^2_d$$

- Using properties of logarithms, we often consider differences in log-likelihood:

$$\ell_{\text{full}} - \ell_{\text{reduced}} \sim \frac{1}{2}\chi^2_d,$$

  where d is the difference in number parameters between the models.
- The test can be extended for restricted likelihood ratio tests (RLRT); the idea is similar: fit two models (full vs reduced) using REML, compare differences in the restricted log-likelihoods.
- Rather than fitting two models manually and comparing, we will use the `lmerTest` package to do this automatically for us.

```
# install.packages("lmerTest")
lmerTest::ranova(output)  # Will do RLRT or LRT, depending on object.

ANOVA-like table for random-effects: Single term deletions

Model:
diameter ~ 1 + (1 | plate)
            npar  logLik    AIC     LRT Df Pr(>Chisq)
<none>         3 -306.63 619.26
(1 | plate)    2 -306.71 617.43 0.17402  1     0.6766
```

Confidence intervals

- We can get confidence intervals for our estimates using the `confint` function.
- The two primary methods will be `profile` (default) and `boot`.
- This computes *profile likelihood* confidence intervals, which are effectively the same as a **likelihood ratio test** (LRT).
- Mathematically, we can obtain a confidence interval using profiled restricted likelihoods, but unfortunately there's not an immediate way to do this in R; we can use `method = "boot"` for a bootstrap interval for REML estimates, or `method = "profile"` for a profile likelihood. Both should give very similar output.

```
set.seed(123)
confint(output, signame = FALSE, level = 0.95, method = "boot")

                          2.5 %     97.5 %
sd_(Intercept)|plate  0.000000  0.8138926
sigma                 1.771368  2.2218709
(Intercept)          22.656619 23.3378052

confint(output, signame = FALSE)

                          2.5 %     97.5 %
sd_(Intercept)|plate  0.000000  0.8458197
sigma                 1.778727  2.2792231
(Intercept)          22.614951 23.3294931
```

5 Factorial Random Effects Models

Factorial models

- Just like with fixed effects, we may want to consider random effects with multiple random treatment factors.
- Again, we use random effects when we aren't interested in the specific treatment effects chosen, but the larger population of treatment effects.

Example: Carton Factory (continued)

- Let's expand our motivating example of the Carton Factory.

Two-way random effects model

Suppose the carton factory has multiple machines and would like to know the variance in the effect of these machines, but there are also different machine operators, and their effect may be random.

- We randomly select machines and operators, and have each operator work on each each machine and produce $n = 4$ cartons each.
- We have a two-way, completely randomized factorial design.

- In the design proposed above, a two-way random effects model is natural:

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk},$$

  with

$$\alpha_i \overset{iid}{\sim} N(0, \sigma^2_\alpha) \quad \beta_j \overset{iid}{\sim} N(0, \sigma^2_\beta) \quad (\alpha\beta)_{ij} \overset{iid}{\sim} N(0, \sigma^2_{\alpha\beta}) \quad \epsilon_{ijk} \overset{iid}{\sim} N(0, \sigma^2)$$

- The variance of each observation is

$$\text{Var}(y_{ijk}) = \sigma^2_\alpha + \sigma^2_\beta + \sigma^2_{\alpha\beta} + \sigma^2.$$

- This idea once again can generalize to higher-order factorial models

Parameter estimation

- The parameter estimation will still happen in a similar way: we can use ML or REML.
- If we are primarily interested in the error components, REML is unbiased, and generally preferred.

```
carton <- read.csv('carton.csv')
output <- lmer(
  # thickness ~ 1 + (1 | Machine) + (1 | Operator) + (1 | Machine:Operator),
  thickness ~ (1 | Machine * Operator),
  data = carton,
  REML = TRUE
)
```

```
summary(output)

Linear mixed model fit by REML ['lmerMod']
Formula: thickness ~ (1 | Machine * Operator)
   Data: carton

REML criterion at convergence: 1486.2

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.27713 -0.58130  0.00118  0.54914  2.77623

Random effects:
 Groups           Name        Variance Std.Dev.
 Machine:Operator (Intercept)  3.389   1.841
 Machine          (Intercept) 12.985   3.604
 Operator         (Intercept)  7.867   2.805
 Residual                      1.307   1.143
Number of obs: 384, groups:
Machine:Operator, 96; Machine, 12; Operator, 8

Fixed effects:
            Estimate Std. Error t value
(Intercept)   49.838      1.451   34.36
```

Hypothesis Testing (F-test)

- Recall that the F-test is exact, and valid for testing the hypothesis $\sigma^2_A, \sigma^2_B, \sigma^2_{AB} > 0$.
- To do this, we would re-fit the model with `aov()`.
- *HOWEVER*: we must be extremely careful. The output P-values are INCORRECT, because of the factorial structure.
- Exact P-values can be computed by hand (visit office hours if you're interested). However, we will instead use the likelihood based inference methods.

  For the notes only, here are a few details regarding the "by-hand" P-value calculations.
- In the output ANOVA table, the data are balanced, meaning that the Mean Sq values that are output are correct. However, if we have factorial random effects structure, then the correct F-test does NOT match the output of `aov`. In the output ANOVA, the F-value for fixed effects is computed as MS_{effect}/MS_E. For random effects, the denominator must change. For the Factor A test, the F value is MS_A/MS_{AB}; this also impacts the degrees of freedom. Our textbook (**?**) goes into a lot of these details if you're interested. However, pure factorial random effects models are not that common in practice. We'll later go into mixed-effects models, at which point we will strictly need to use `lme4::lmer`. Because of this, it makes sense to just highlight this now, rather than going over tedious by-hand computations that are not often seen in practice.

Confidence intervals

- We can obtain confidence intervals the same way as before. (can be slow).

```
set.seed(444)
# Or method = 'boot'
confint(output, method = 'profile', signame = FALSE)
```

```
                                    2.5 %    97.5 %
sd_(Intercept)|Machine:Operator  1.557155  2.205129
sd_(Intercept)|Machine           2.411323  5.659272
sd_(Intercept)|Operator          1.711321  4.973523
sigma                            1.055678  1.243132
(Intercept)                     46.919476 52.756306
```

Hypothesis testing (likelihood based)

- We can also do hypothesis testing, but it can get tricky due to the interaction effect.
- The `lmerTest::ranova` function just drops the term from the model, and compares REML (or ML). This may not be meaningful for all variables in the presence of an interaction effect.
- Instead, we often will rely on the confidence intervals.

```
# May not be meaningful for main (random) effects if interaction is present.
lmerTest::ranova(output)
```

```
ANOVA-like table for random-effects: Single term deletions

Model:
thickness ~ 1 + (1 | Machine) + (1 | Operator) + (1 | Machine:Operator)
                      npar  logLik    AIC     LRT Df
<none>                   5 -743.10 1496.2
(1 | Machine)            4 -790.71 1589.4  95.226  1
(1 | Operator)           4 -779.39 1566.8  72.583  1
(1 | Machine:Operator)   4 -861.14 1730.3 236.073  1
                      Pr(>Chisq)
<none>
(1 | Machine)          < 2.2e-16 ***
(1 | Operator)         < 2.2e-16 ***
(1 | Machine:Operator) < 2.2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Summary

- We fit a random effects model if we are interested in the variance components specifically, rather than specific treatment effects.

6 Mixed Effects Models

Mixed Effects: an introduction

- One of the most common ways we see a random effect is in the context of a *mixed effects model*.

  Mixed Effects Modes: Motivation

  '**Idea:** We wish to estimate the fixed effects of specific treatments, while accounting for nuisance or contextual factors that are randomly drawn from a larger population.

- Mixed effects models are common for *block designs*:
  - We have a treatment factor A of direct interest (**fixed effect**).
  - We have a blocking factor B (**random effect**).
- Appropriate if blocks are randomly drawn from a population of possible blocks.

Model Formulation

- Consider a Randomized Complete Block Design (RCBD) with a fixed treatments and b randomly selected blocks:
$$y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij}$$
- Here, α_i are **fixed parameters** (we assume $\sum_i \alpha_i = 0$), while β_j and ϵ_{ij} are **independent random variables**:
$$\beta_j \overset{iid}{\sim} N(0, \sigma_\beta^2), \qquad \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2).$$
- Because β_j is random, observations within the same block share a common random shift, inducing correlation:
$$\mathrm{Var}(y_{ij}) = \sigma_b^2 + \sigma^2, \qquad \mathrm{Cor}(y_{ij}, y_{ij'}) = \frac{\sigma_b^2}{\sigma_b^2 + \sigma^2} \quad (j \neq j').$$

- **Key advantage:** Treating blocks as random allows our conclusions about the fixed treatments (α_i) to generalize to the broader population of blocks, rather than being restricted to the specific blocks in our experiment.

Inference in Mixed Effects Models

- When our model contains both fixed and random effects, we separate our inference workflow into two distinct steps using `lme4` and `lmerTest`:

- **1. Fixed Effects (Treatments):** Use F-tests via `anova(lmer_model)`.
  - `lmerTest` uses REML to estimate variances and applies the *Satterthwaite approximation* to calculate fractional denominator degrees of freedom.

- **2. Random Effects (Blocks):** Use Restricted Likelihood Ratio Tests via `lmerTest::ranova(model)`.
  - If your model includes a random interaction (e.g., Block × Treatment), test the interaction first.
  - If the interaction is significant, retain it and stop. Do not test the main random effect.
  - If non-significant, you can test the main random block effects.

Example: Ergonomic Stools (RCBD)

- Consider the `ergoStool` dataset (from the `nlme` package).
- The data come from a Randomized Complete Block study, with the treatment being stool-type ($g = 4$ levels), and the block the test subject ($r = 9$).
- We measure the physical effort required to arise from 4 stool types (**fixed treatments**) across 9 randomly selected subjects (**random blocks**).

```r
library(lme4)
library(lmerTest)
data(ergoStool, package = "nlme")

# Fit mixed model: Subject is random, Type is fixed
mod_ergo <- lmer(effort ~ Type + (1 | Subject), data = ergoStool)
```

Testing Fixed vs. Random Effects

- First, we test the fixed treatment effect (`Type`) using the Satterthwaite approximated F-test:

```r
anova(mod_ergo, type = 3)

Type III Analysis of Variance Table with Satterthwaite's method
     Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
Type 81.194  27.065     3    24  22.356 3.935e-07 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Next, we may be interested in the estimated variance component for the subject-to-subject variation:

```
# Either summary, or just inspect the object
mod_ergo

Linear mixed model fit by REML ['lmerModLmerTest']
Formula: effort ~ Type + (1 | Subject)
   Data: ergoStool
REML criterion at convergence: 121.1308
Random effects:
 Groups   Name        Std.Dev.
 Subject  (Intercept) 1.332
 Residual             1.100
Number of obs: 36, groups:  Subject, 9
Fixed Effects:
(Intercept)       TypeT2       TypeT3       TypeT4
     8.5556       3.8889       2.2222       0.6667
```

- Next, we test whether the baseline random variance between subjects (σ_b^2) is significant using a restricted likelihood ratio test:

```
ranova(mod_ergo)

ANOVA-like table for random-effects: Single term deletions

Model:
effort ~ Type + (1 | Subject)
              npar  logLik    AIC    LRT Df Pr(>Chisq)
<none>           6 -60.565 133.13
(1 | Subject)    5 -67.304 144.61 13.477  1  0.0002415

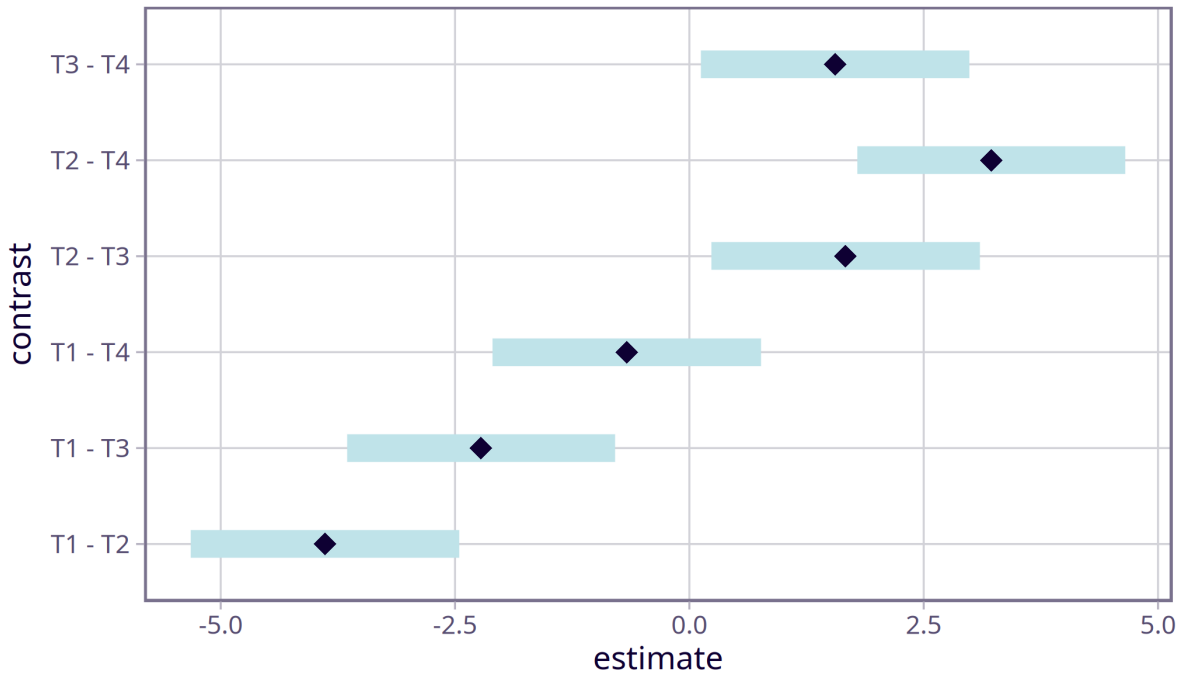
<none>
(1 | Subject) ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Because this is a perfectly balanced RCBD with no interactions, notice that the Satterthwaite denominator degrees of freedom for the fixed effect is exactly 24: $(a-1)(b-1) = (3)(8)$. This yields the exact same F-statistic as classical `aov()`, but using `lmer()` safely allows for missing data, robust post-hoc comparisons, and straightforward variance component extraction.

Post-Hoc Comparisons with Mixed Models

- Just like with fixed-effects models, we can use the `emmeans` package on an `lmer` object to estimate marginal means and perform pairwise comparisons for our fixed treatments.
- `emmeans` automatically incorporates the random effect variance into the standard errors and uses the Satterthwaite degrees of freedom.

```r
library(emmeans)
em_mod <- emmeans(mod_ergo,  ~ Type)
plot(pairs(em_mod))
```



Factorial Mixed Models: The Interaction Rule

- If we had replicated data (multiple trials per subject per stool), we would include a Subject × Type interaction.
- Because Subject is random, the interaction is also a *random effect*:

$$y_{ijk} = \mu + \alpha_i + b_j + (\alpha b)_{ij} + \epsilon_{ijk}.$$

- **In R:** `lmer(y  Type + (1 | Subject) + (1 | Type:Subject), data = df)`
- Using `anova(lmer_mod)` automatically tests the fixed main effect against the random interaction variance (using fractional degrees of freedom)—avoiding complex manual Expected Mean Squares algebra entirely!

Acknowledgments

- Compiled on November 17, 2026 using R version 4.6.2.

- Licensed under the Creative Commons Attribution-NonCommercial license.  Please share and remix non-commercially, mentioning its origin.

- We acknowledge students and instructors for previous versions of this course / slides.

References

Oehlert GW (2000). *A First Course in Design and Analysis of Experiments*. W. H. Freeman. ISBN 0-7167-3510-5. 1, 7