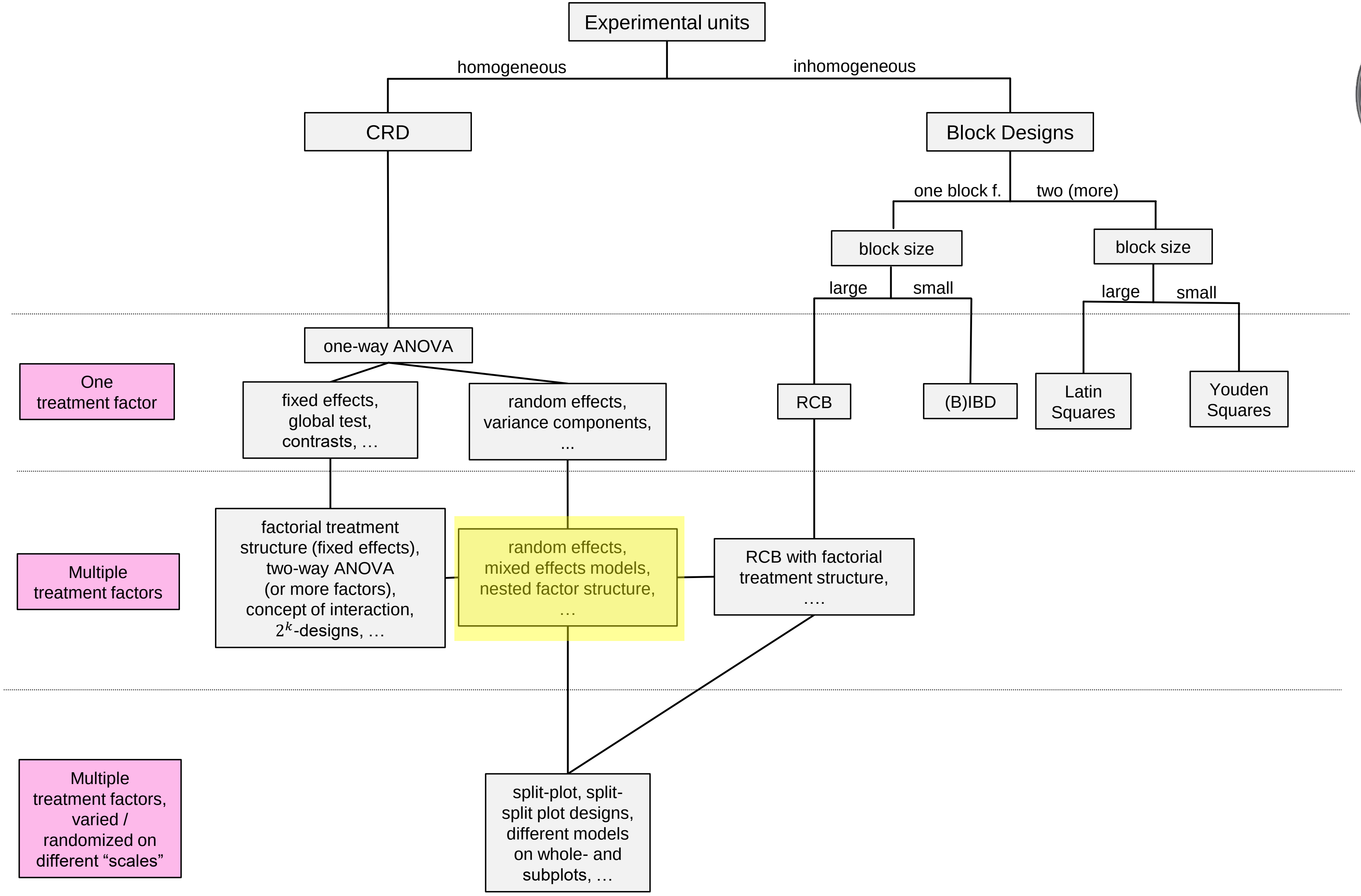
Nesting and Mixed Effects: Part I



Similar to Lawson (2015)

Where do we stand?

- So far:
  - Fixed effects
  - Random effects
  - Both in the factorial context

- Now:
  - **Nested** factor structure
  - **Mixed models**: A combination of fixed and random effects.

Remember: Crossed Factors

- With **crossed factors** A and B we **see** (by definition) **all possible combinations** of the factor levels, i.e. we can set up a data-table of the following form:

Factor A / Factor B	***1***	***2***	***3***
1	$\times$	$\times$	$\times$
2	$\times$	$\times$	$\times$
3	$\times$	$\times$	$\times$
4	$\times$	$\times$	$\times$

Think of n observations here

- This means: We see **every level** of factor A at **every level** of factor B (and vice versa).
- Factor level 1 of factor A has the same meaning **across all levels** of factor B.

Example: Student Performance (Roth, 2013)

- Want to analyze student performance.
- Data from different **classes** from different **schools** (on **student** level).
- How large is (grade) variability
  - between different schools?
  - between classes **within** the **same** school?
  - between students **within** the **same** class?
- This looks like a **new design**, as classes are clearly **not** crossed with schools, similarly for students.
- This leads us to a new definition…

New: Nested Factor Structure

- We call factor B **nested** in factor A if we have **different levels** of B **within each level** of A.

Factor A / Factor B	1	2	3	4	5	6	7	8	9	10	11	12
1	$\times$	$\times$										
2			$\times$	$\times$								
3					$\times$	$\times$						
4							$\times$	$\times$				
5									$\times$	$\times$		
6											$\times$	$\times$

- E.g., think of A = school, B = class.
- We also write: $B(A)$.
- Data is **not** necessarily presented in this form…

Nested Factors: Example (Roth, 2013)

- Presented data:

	Class 1	*Class 2*
***School* 1**	×	×
***School* 2**	×	×
***School* 3**	×	×

- Underlying data structure:

	Class 1	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Class 6*
***School* 1**	×	×				
***School* 2**			×	×		
***School* 3**					×	×

Hence, **class is nested in school** because class 1 in school 1 has nothing do with class 1 in school 2 etc.

Nested Factors

- Note: Just because classes are labelled 1 and 2 doesn't mean that it is a crossed design!

- Hence: Always ask yourself whether factor level "1" really corresponds to the **same "object" across all levels of the other factor**.

- Typically we use **parentheses** in the index to indicate nesting, i.e. the model is written as

$$Y_{ijk} = \mu + \underbrace{\alpha_i}_{\text{school}} + \underbrace{\beta_{j(i)}}_{\text{class \textbf{within} school}} + \underbrace{\epsilon_{k(ij)}}_{\text{error}}$$

- Here, we also wrote the **errors** in "nested notation". Errors are always nested, we have just ignored this so far.

Why Use Nesting?

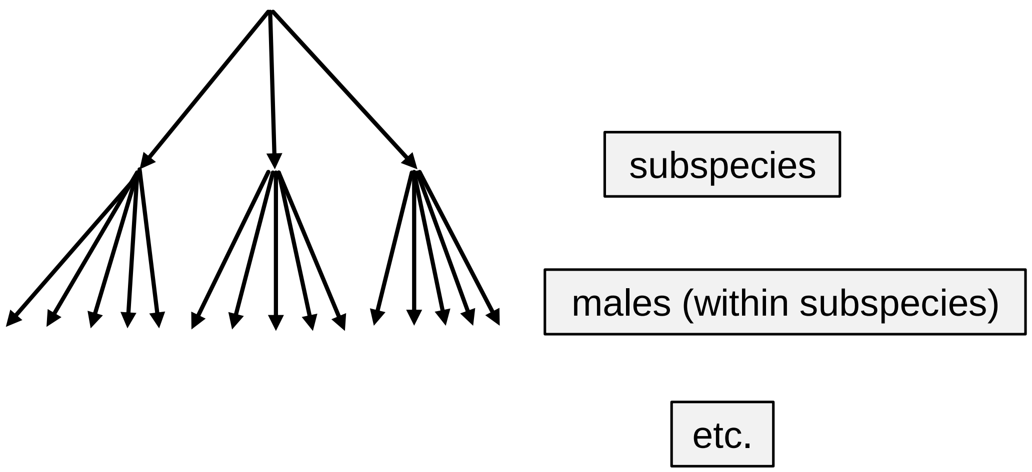
- Typically we use a nested structure due to practical / logistical constraints.
- For example:
  - Patients are nested in hospitals as we don’t want to send patients to all clinics across the country.
  - Samples are nested in batches (in quality control).
  - …

Example: Fully Nested Design

- We call a design **fully nested** if **every** factor is **nested** in its **predecessor**.

- Genomics example in Oehlert (2010):
  - Consider three subspecies.
  - Randomly choose five males from each subspecies (= 15 males).
  - Each male is mated with four different females of the same subspecies (= 60 females).
  - Observe three offsprings per mating (= 180 offsprings).
  - Make two measurements per offspring (= 360 measurements).

- Picture:



Example: Fully Nested Design

- We use the model

$$Y_{ijklm} = \mu + \alpha_i + \beta_{j(i)} + \gamma_{k(ij)} + \delta_{l(ijk)} + \varepsilon_{m(ijkl)}$$

weight | subspecies | male **within** subspecies | female **within** male | offspring **within** female | error

- To calculate the corresponding sums of squares, we use the following decomposition

deviation of the subspecies mean

deviation of the male mean **within** subspecies

$$(y_{ijklm} - \bar{y}_{.....}) = (\bar{y}_{i....} - \bar{y}_{.....}) + (\bar{y}_{ij...} - \bar{y}_{i....}) +$$
$$(\bar{y}_{ijk..} - \bar{y}_{ij...}) + (\bar{y}_{ijkl.} - \bar{y}_{ijk..}) + (y_{ijklm} - \bar{y}_{ijkl.})$$

deviation of the female mean **within** male animal

deviation of the offspring mean **within** female animal

residual

take the square and the sum over all indices.

ANOVA Table for Fully Nested Design (Balanced Design)

OLD

- This leads us to the decomposition

$$SS_{Total} = SS_A + SS_{B(A)} + SS_{C(AB)} + SS_{D(ABC)} + SS_E$$

- Assuming we have only **random effects** and a **balanced** design, we have the following ANOVA table:

Source	df	E[MS] NEW
A	$a-1$	$\sigma^2 + n\sigma_\delta^2 + nd\sigma_\gamma^2 + ncd\sigma_\beta^2 + nbcd\sigma_\alpha^2$
$B(A)$	$a(b-1)$ NEW	$\sigma^2 + n\sigma_\delta^2 + nd\sigma_\gamma^2 + ncd\sigma_\beta^2$
$C(AB)$	$ab(c-1)$ NEW	$\sigma^2 + n\sigma_\delta^2 + nd\sigma_\gamma^2$
$D(ABC)$	$abc(d-1)$ NEW	$\sigma^2 + n\sigma_\delta^2$
Error	$abcd(n-1)$ NEW	σ^2

- With this information we can again construct **tests** and **estimators** for the different variance components.

ANOVA for Fully Nested Designs

$\mathcal{OLD}$

- F-tests are constructed by taking the ratio of “neighboring” mean squares as they just differ by the variance component of interest.

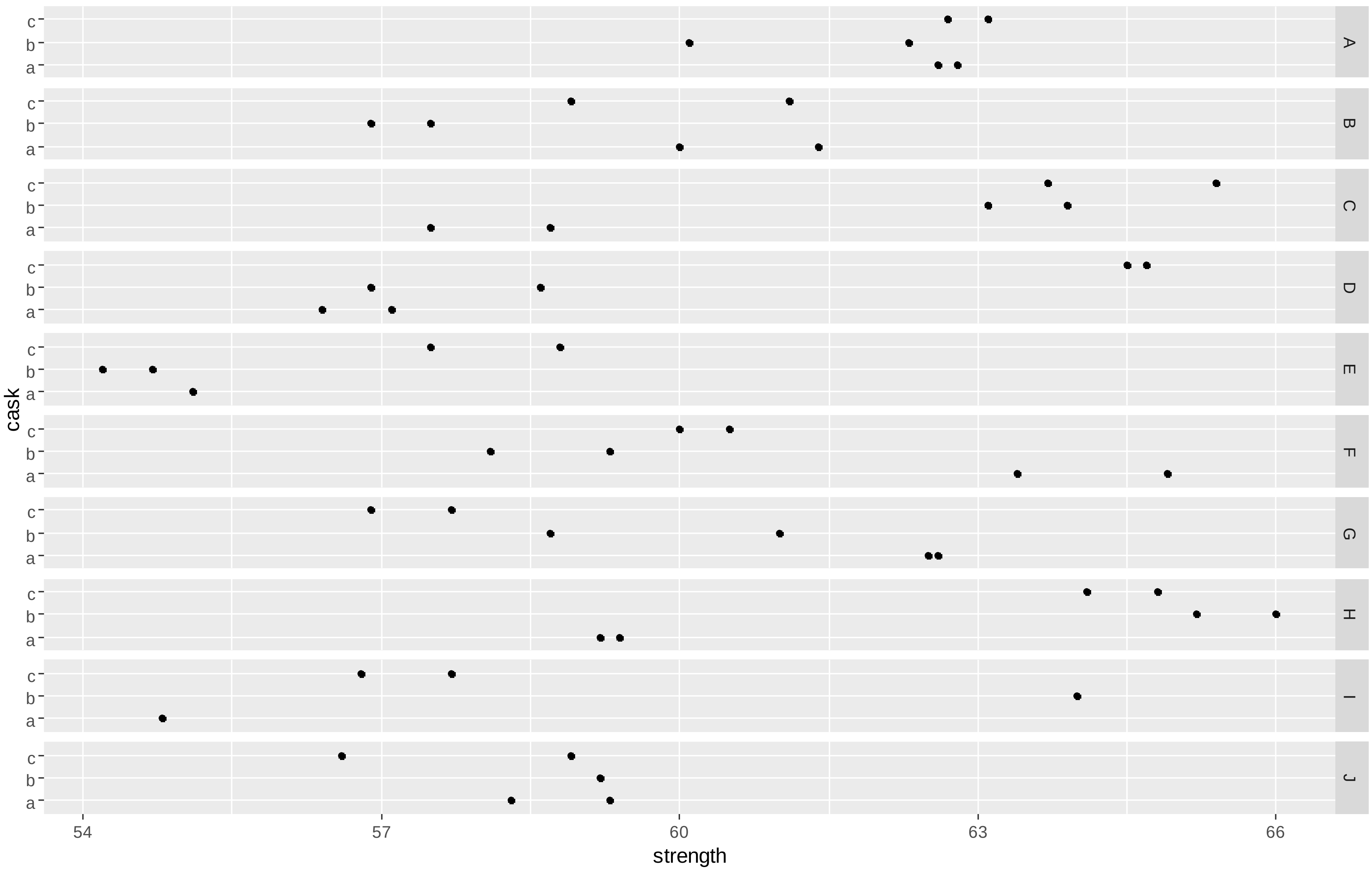
- This means that we always use the mean square of the successor in the hierarchy tree as denominator.

- E.g., use $F = \frac{\overbrace{MS_A}^{\text{“between } A\text{”}}}{\underbrace{MS_{B(A)}}_{\text{“within } A\text{”}}}$ to test $H_0: \sigma_\alpha^2 = 0$ vs. $H_A: \sigma_\alpha^2 > 0$.

Example: Pastes Strength

- Dataset from the `lme4` package.
- Chemical paste product contained in casks.
- 10 **deliveries** (**batches**) were **randomly** selected.
- From each delivery, 3 **casks** were **randomly** selected.
- Per cask: Make two **measurements**.

Example: Pastes Strength - Visualization



Example: Pastes Strength

- Model:

$$\underbrace{Y_{ijk}}_{\text{strength}} = \mu + \overbrace{\underbrace{\alpha_i}_{\text{effect of batch}}}^{N(0,\sigma_\alpha^2)} + \overbrace{\underbrace{\beta_{j(i)}}_{\text{effect of cask }\textbf{within}\text{ batch}}}^{N(0,\sigma_\beta^2)} + \overbrace{\underbrace{\epsilon_{k(ij)}}_{\text{error term}}}^{N(0,\sigma^2)}$$

- Dataset in R:

```
> str(Pastes)
'data.frame':   60 obs. of  4 variables:
 $ strength: num  62.8 62.6 60.1 62.3 62.7 63.1 60 61.4 57.5
 $ batch   : Factor w/ 10 levels "A","B","C","D",..: 1 1 1 1
 $ cask    : Factor w/ 3 levels "a","b","c": 1 1 2 2 3 3 1 1
 $ sample  : Factor w/ 30 levels "A:a","A:b","A:c",..: 1 1 2
```

- Be careful! Why?

Analysis Using lmer

notation for **nesting** structure

See alternative formulations in the corresponding R-file

```
> fm1 <- lmer(strength ~ (1 | batch/cask), data = Pastes)
> summary(fm1)
Linear mixed model fit by REML
t-tests use  Satterthwaite approximations to degrees of freedom ['merModLmerTest']
Formula: strength ~ (1 | batch/cask)
   Data: Pastes

REML criterion at convergence: 247

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.4798 -0.5156  0.0095  0.4720  1.3897

Random effects:
 Groups     Name        Variance Std.Dev.
 cask:batch (Intercept) 8.434    2.9041
 batch      (Intercept) 1.657    1.2874
 Residual               0.678    0.8234
Number of obs: 60, groups:  cask:batch, 30; batch, 10

Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)  60.0533     0.6769  9.0000   88.72 1.49e-14 ***
```

$\hat{\sigma}_\beta$

$\hat{\sigma}_\alpha$

$\hat{\sigma}$

Check if model was interpreted correctly

$\hat{\mu}$

Analysis Using lmer

- Get **confidence intervals** using

```
> confint(fm1, oldNames = FALSE)
Computing profile confidence intervals ...
                                2.5 %    97.5 %
sd_(Intercept)|cask:batch  2.1579337  4.053589
sd_(Intercept)|batch       0.0000000  2.946591
sigma                      0.6520234  1.085448
(Intercept)               58.6636504 61.443016
```

- (Conservative) tests of the variance components can be obtained with:

```
> fm2 <- lmer(strength ~ (1 | batch) + (1 | cask:batch), data = Pastes)
> rand(fm2)
Analysis of Random effects Table:
           Chi.sq Chi.DF p.value
batch       0.658      1     0.4
cask:batch 54.605      1   1e-13 ***
```

- Alternative: Package `RLRsim` or bootstrap based methods (will not discuss this here).

General Situation

- The fully nested design is only a (very) special case.
- A design can of course have **both crossed and nested** factors.
- In addition, a model can contain both **random** and **fixed** effects.
- If this is the case, we call it a **mixed effects model**.
- Let’s have a look at an example.

Cheese Tasting (Oehlert, 2010, Example 12.2)

- How do **urban** and **rural** consumers rate cheddar cheese for bitterness?

- **Four** 50-pound blocks of **different cheese types** are available.

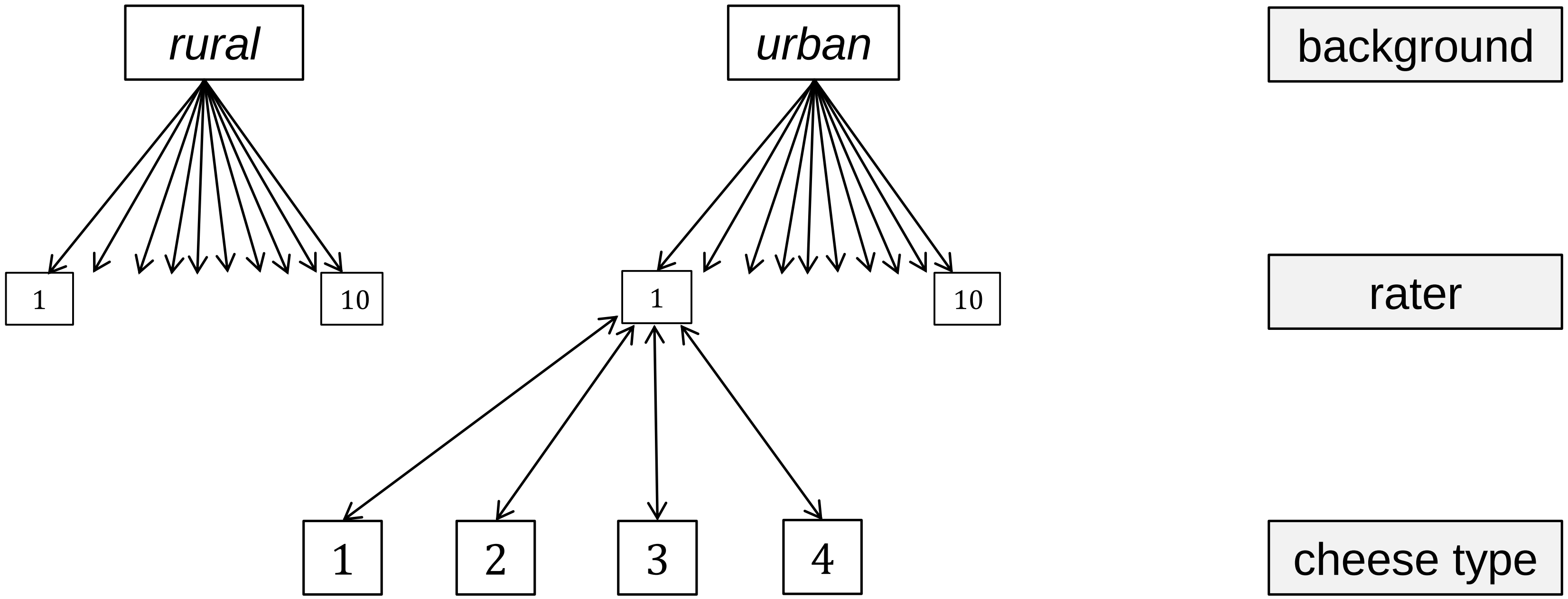
- We use food science students as our raters:
  - Choose 10 students at random with **rural background**.
  - Choose 10 students at random with **urban background**.

- Each rater will taste 8 bites of cheese (presented in random order).

- The eight bites consist of two from each cheese type. Hence, every rater gets every cheese type **twice**.

Cheese Tasting (Oehlert, 2010, Example 12.2)

- What factors do we have here?
  - A: **background**, levels = {“rural”, “urban”}
  - B: **rater**, levels = {1, …,10} (or 20); **nested** in background
  - C: **cheese type**, levels = {1, 2, 3, 4}
- Relationship: $A \times B(A) \times \ C$ (both A and B are crossed with C)



Cheese Tasting (Oehlert, 2010, Example 12.2)

- A model to analyze this data could be

$$Y_{ijkl} = \mu + \alpha_i + \beta_{j(i)} + \gamma_k + (\alpha\gamma)_{ik} + (\beta\gamma)_{jk(i)} + \epsilon_{l(ijk)}$$

  where
  - α_i are the **fixed** effects of background
  - $\beta_{j(i)}$ are the **random** effects of rater (within background)
  - γ_k are the **fixed** effects of cheese type
  - $(\alpha\gamma)_{ik}$ is the (**fixed**) interaction effect between background and cheese type
  - $(\beta\gamma)_{jk(i)}$ is the (**random**) interaction between rater and cheese type.

- The interaction between a fixed effect and a random effect is **random** (as it includes a random component).

Cheese Tasting (Oehlert, 2010, Example 12.2)

- Interpretation of parameters:

Term	*Interpretation*
α_i	Main effect of background.
$\beta_{j(i)}$	**Random** effect of rater: Allows for an individual “general cheese liking” level of a rater.
γ_k	Main effect of cheese type.
$(\alpha\gamma)_{ik}$	Fixed interaction effect between background and cheese type: Allows for a background specific cheese type preference.
$(\beta\gamma)_{jk(i)}$	**Random** interaction between rater and cheese type: Allows for an individual deviation from the population average “cheese type” effect.

- See R-File for results (later).

Appendix

Analysis Using aov

OLD

- Using the `aov` command

```
> fit <- aov(strength ~ batch + cask %in% batch, data = Pastes)
> summary(fit)
            Df Sum Sq Mean Sq F value   Pr(>F)    
batch        9  247.4  27.489   40.55 2.28e-14 ***
batch:cask  20  350.9  17.545   25.88 9.79e-14 ***
Residuals   30   20.3   0.678                     
```

- Output can be used to manually calculate the different variance components by solving the equations with the corresponding expected mean squares:

$$\hat{\sigma}^2 = 0.678$$

$$\hat{\sigma}_\beta^2 = \frac{17.545 - 0.678}{2} = 8.43$$

$$\hat{\sigma}_\alpha^2 = \frac{27.489 - 17.545}{2 \cdot 3} = 1.66$$

	E[MS]
batch	$\sigma^2 + 2 \cdot \sigma_\beta^2 + 2 \cdot 3 \cdot \sigma_\alpha^2$
cask(batch)	$\sigma^2 + 2 \cdot \sigma_\beta^2$
Error	σ^2

Analysis Using aov

OLD

- Similarly, tests have to be calculated **manually**.
- E.g., for the batch variance component:

$$F = \frac{27.489}{17.545} = 1.567$$

- Use $F_{9,\,20}$-distribution to calculate p-value:

```
> pf(27.489 / 17.545, df1 = 9, df2 = 20, lower.tail = FALSE)
[1] 0.1925487
```

- Hence, **cannot** reject $H_0\!: \sigma_\alpha^2 = 0$.
- Note that the default output is **wrong** here as the model was interpreted as a fixed effects model (using the wrong denominator mean square)!

Appendix: Data Generating Mechanisms

- **Unrestricted model**:

  Random effects (including interactions!) have the following assumptions:
  - independent
  - normally distributed with mean zero
  - effects corresponding to the same term have a common variance: $\sigma_\alpha^2, \sigma_\beta^2, \sigma_{\alpha\beta}^2$ etc.

  Fixed effects:
  - have the usual sum-to-zero constraint (across **any** subscript).

- **Restricted model:**
  - As above, with the exception that interactions between random and fixed factors (which are random!) follow the sum-to-zero constraint over **any subscript** corresponding to a **fixed factor**.
  - This induces a **negative correlation** within these random effects, hence they are **not independent** anymore.

Appendix: Data Generating Mechanisms

- Assume only two factors:
  - A **fixed** (with a levels) and
  - B **random** (with b levels)

- Think of hypothetical data-table

A/B	*1*	*2*	*3*	*4*	*5*	*6*	*…*													
1																				
2																				
…																				
a																				

with **lot's** of columns.

- To get the observed data-table, we randomly select b columns.

Appendix: Data Generating Mechanisms

- This means: If we repeat the experiment and select the same column twice, we get the very same column (and of course the same column total).
- This implicitly means: The interaction effects are “attached” to the column. This is called the **restricted model.**
- In the restricted model we assume that the interaction effects add to zero when summed across a fixed effect (they are random but **restricted**!)
- The alternative is the **unrestricted model** which treats interaction effects independently from the main effects.
- `lmer` uses the unrestricted model.

Appendix: Data Generating Mechanisms

- In the tasting experiment: Would prefer restricted model because interaction is “attached” to raters.

- Reason: The rater will not change his particular taste (remember meaning of parameters).