

Linear Mixed Models

Lieven Clement

In this chapter we introduce linear mixed models. We start from simple designs in which the response is measured repeatedly on the same experimental unit and analyse them first with a **two-stage analysis**: we summarise the data of each experimental unit and analyse the summaries. The linear mixed model does both stages in one step. The theory part follows Sections 4.1–4.8 of Ruppert et al. (2003): best linear unbiased prediction (BLUP), ML and REML estimation and inference.

```
library(tidyverse)
library(lme4)
library(Matrix)
library(QFeatures)
library(BiocFileCache)
```

Experimental and observational units

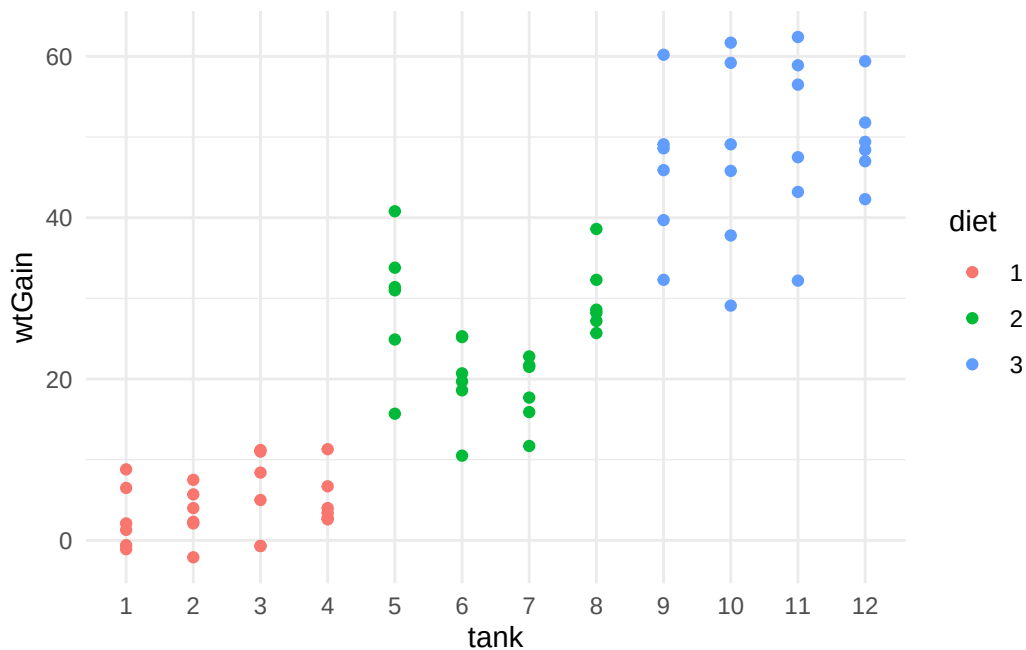
Whether we need random effects depends on the design: on which unit the treatment is randomised (the **experimental unit**) and on which unit the response is measured (the **observational unit**). We first illustrate this with a simple example and then with one protein from two proteomics experiments that are also used in the other lectures.

Pseudo-replication: fish in tanks

Researchers want to assess the effect of three diets on the weight gain of fish. They have 12 tanks with the same number of fish. Each diet is randomly assigned to 4 tanks, and at the end of the experiment the weight gain of 6 fish in each tank is recorded.

```
fish <- data.frame(
  diet = factor(rep(1:3, each = 24)),
  tank = factor(rep(1:12, each = 6)),
  wtGain = c(
    8.8, 6.5, -1.1, 1.3, -0.6, 2.1,
    5.7, 4, -2.1, 2.3, 2.1, 7.5,
    -0.7, 11.2, -0.7, 11, 8.4, 5,
    3.4, 6.7, 11.3, 4, 2.6, 2.8,
    31, 15.7, 40.8, 33.8, 24.9, 31.4,
    10.5, 25.3, 18.6, 19.7, 20.7, 25.2,
    21.7, 15.9, 22.8, 21.5, 11.7, 17.7,
    38.6, 25.7, 32.3, 28.2, 28.6, 27.2,
    49.1, 39.7, 45.9, 48.6, 32.3, 60.2,
    61.7, 29.1, 49.1, 59.2, 45.8, 37.8,
    58.9, 43.2, 56.5, 47.5, 32.2, 62.4,
    59.4, 49.4, 42.3, 47, 48.4, 51.8))

ggplot(fish, aes(x = tank, y = wtGain, colour = diet)) +
  geom_point() +
  theme_minimal()
```



Naive data analysis:

```
naiveFish <- lm(wtGain ~ diet, data = fish)
summary(naiveFish)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.229167	1.528121	2.767560	7.244420e-03
diet2	20.333333	2.161090	9.408834	5.408316e-14
diet3	44.000000	2.161090	20.360099	6.933229e-31

Experimental units and observational units?

- The diet is randomised to the tanks: the 12 tanks are the experimental units.
- The weight gain is measured on the fish: the 72 fish are the observational units.
- Fish in the same tank share the same environment (water quality, temperature, feeding, ...): they are **pseudo-replicates**. Weighing more fish of the same tank does not give more independent tanks.

Two-stage analysis

The diet effect has to be assessed with the variability between the experimental units, the tanks. A simple and correct analysis therefore works in two stages:

- **Stage I:** summarise the data of each experimental unit, here the average weight gain in each tank.
- **Stage II:** analyse the summaries with an ordinary linear model: 4 tanks per diet.

```
stageIFish <- fish |>
  group_by(tank, diet) |>
  summarise(wtGain = mean(wtGain), .groups = "drop")
stageIFish
```

```
# A tibble: 12 x 3
  tank diet wtGain
  <fct> <fct> <dbl>
1 1     1     2.83
2 2     1     3.25
3 3     1     5.7
4 4     1     5.13
5 5     2    29.6
6 6     2    20
7 7     2    18.6
```

8	8	2	30.1
9	9	3	46.0
10	10	3	47.1
11	11	3	50.1
12	12	3	49.7

```
twoStageFish <- lm(wtGain ~ diet, data = stageIFish)
summary(twoStageFish)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.229167	1.907782	2.216798	5.384469e-02
diet2	20.333333	2.698011	7.536416	3.554557e-05
diet3	44.000000	2.698011	16.308310	5.445651e-08

If we act as if the 72 fish are independent, the standard errors are too small: the analysis treats the 24 fish per diet as independent repeats, while there are only 4 independent tanks per diet.

```
summary(naiveFish)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.229167	1.528121	2.767560	7.244420e-03
diet2	20.333333	2.161090	9.408834	5.408316e-14
diet3	44.000000	2.161090	20.360099	6.933229e-31

Note that we cannot correct for the tanks with a fixed tank effect as in a block design: once we know the tank, we also know the diet. The tank effects are aliased with the diet effects.

```
coef(lm(wtGain ~ diet + tank, data = fish))
```

(Intercept)	diet2	diet3	tank2	tank3	tank4
2.8333333	27.2666667	46.8833333	0.4166667	2.8666667	2.3000000
tank5	tank6	tank7	tank8	tank9	tank10
-0.5000000	-10.1000000	-11.5500000	NA	-3.7500000	-2.6000000
tank11	tank12				
0.4000000	NA				

Linear mixed model

The two-stage analysis is easy, but it has drawbacks:

- Information is lost when the data of a tank are summarised by a single number.
- It is only valid when all Stage I summaries are equally precise, i.e. when the design is balanced. With unequal numbers of fish per tank, or missing values, the precision of the tank averages differs, and Stage II does not account for that.

A **linear mixed model** combines both stages into one model:

$$\begin{cases} y_{tk} &= \beta_0 + \beta_{d(t)} + b_t + \epsilon_{tk} \\ b_t &\sim N(0, \sigma_b^2), \\ \epsilon_{tk} &\sim N(0, \sigma_\epsilon^2) \end{cases}$$

with y_{tk} the weight gain of fish k in tank t , $\beta_{d(t)}$ the effect of the diet of tank t and b_t a **random effect** for tank t . (Note, that the random effects and errors are assumed to be independent.)

- The diet effects are **fixed effects**: they are the parameters of interest.
- The tank effects are **random effects**: the 12 tanks are a random sample of tanks, and we are not interested in the effect of these particular tanks. They only cost one extra parameter, the **variance component** σ_b^2 , the between-tank variance. σ_ϵ^2 is the within-tank variance.

```
lmmFish <- lmer(wtGain ~ diet + (1|tank), data = fish)
summary(lmmFish)
```

Linear mixed model fit by REML ['lmerMod']

Formula: wtGain ~ diet + (1 | tank)

Data: fish

REML criterion at convergence: 481.9

Scaled residuals:

	Min	1Q	Median	3Q	Max
	-2.60555	-0.45452	-0.05768	0.50128	1.97624

Random effects:

Groups	Name	Variance	Std.Dev.
tank	(Intercept)	6.001	2.450
Residual		51.348	7.166

Number of obs: 72, groups: tank, 12

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	4.229	1.908	2.217
diet2	20.333	2.698	7.536
diet3	44.000	2.698	16.308

Correlation of Fixed Effects:

	(Intr)	diet2
diet2	-0.707	
diet3	-0.707	0.500

Comparisons between analyses

```
rbind(  
  wrong = summary(naiveFish)$coef["diet2", 1:3],  
  twoStage = summary(twoStageFish)$coef["diet2", 1:3],  
  mixed = summary(lmmFish)$coef["diet2", 1:3])
```

	Estimate	Std. Error	t value
wrong	20.33333	2.161090	9.408834
twoStage	20.33333	2.698011	7.536416
mixed	20.33333	2.698011	7.536416

The two-stage analysis and the mixed model give identical estimates, standard errors and t -statistics. For a balanced design with m tanks per diet and n fish per tank, the variance of a diet mean is

$$\text{var}[\bar{y}_d] = \frac{\sigma_b^2}{m} + \frac{\sigma_\epsilon^2}{mn}.$$

The between-tank variance is only averaged over the $m = 4$ independent tanks: weighing more fish per tank cannot reduce it. The mixed model accounts for both sources of variability.

Induced correlation

The random tank effect induces a correlation between fish of the same tank ($k \neq k'$):

$$\text{cov}[y_{tk}, y_{tk'}] = \text{var}[b_t] = \sigma_b^2, \quad \text{corr}[y_{tk}, y_{tk'}] = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_\epsilon^2}.$$

Fish of different tanks remain independent. All fish of the same tank have the same variance $\sigma_b^2 + \sigma_\epsilon^2$ and the same correlation: this covariance structure is called **compound symmetry**. It is natural here, because the fish within a tank are exchangeable.

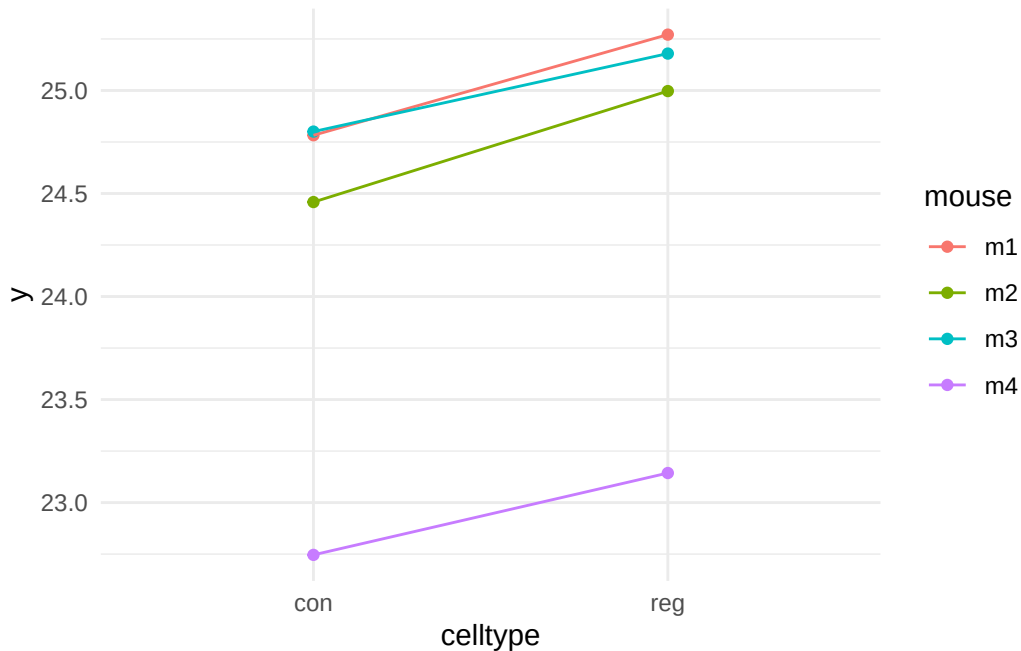
Blocking: mouse T-cells

In the [mouse T-cell experiment](#), regulatory (Treg) and conventional (Tconv) T-cells are purified from each of 4 mice. Here, the cell type varies **within** each mouse: the mouse is a **block**, a randomised complete block (RCB) design.

```
bfc <- BiocFileCache()
mouse <- readRDS(bfc$path(bfc, "https://github.com/statOmics/msqrob2data/raw/refs/heads/main/"))

mouseProt <- data.frame(
  y = assay(mouse[["proteins"]])["B9EKR1", ],
  celltype = mouse$celltype,
  mouse = mouse$mouse)

ggplot(mouseProt, aes(x = celltype, y = y, group = mouse, colour = mouse)) +
  geom_point() +
  geom_line() +
  theme_minimal()
```



We compare

- The two-stage analysis now summarises each mouse by the difference between its Treg and Tconv intensity (Stage I) and tests whether the average difference is zero (Stage II): a paired analysis.
- ignoring the mouse effect
- an RCB analysis with a fixed effect for mouse and
- an RCB analysis with a random intercept for mouse.

```
stageIMouse <- mouseProt |>
  pivot_wider(names_from = celltype, values_from = y) |>
  mutate(diff = reg - con)
twoStageMouse <- lm(diff ~ 1, data = stageIMouse)

wrongMouse <- lm(y ~ celltype, data = mouseProt)
rcbMouse <- lm(y ~ celltype + mouse, data = mouseProt)
lmmMouse <- lmer(y ~ celltype + (1|mouse), data = mouseProt)

rbind(
  twoStage = summary(twoStageMouse)$coef[1, 1:3],
  wrong = summary(wrongMouse)$coef[2, 1:3],
```

```
RCB = summary(rcbMouse)$coef[2, 1:3],
mixed = summary(lmmMouse)$coef[2, 1:3])
```

	Estimate	Std. Error	t value
twoStage	0.4507435	0.03788201	11.8986157
wrong	0.4507435	0.70334325	0.6408585
RCB	0.4507435	0.03788201	11.8986157
mixed	0.4507435	0.03788201	11.8986157

- The two-stage (paired) analysis, the RCB analysis with a fixed block effect and the mixed model with a random block effect give identical results.
- Ignoring the mouse effect inflates the standard error, because the between-mouse variability then ends up in the residual variance.

The cell type effect is estimated **within** each mouse. In the difference $y_{\text{reg},i} - y_{\text{con},i} = \beta_{\text{reg}} + \epsilon_{\text{reg},i} - \epsilon_{\text{con},i}$ the mouse effect b_i cancels, so

$$\text{var} [\bar{y}_{\text{reg}} - \bar{y}_{\text{con}}] = \frac{2\sigma_{\epsilon}^2}{m},$$

which only involves the within-mouse variability. This is the advantage of blocking: the between-mouse variability is removed from the comparison.

Summary

Design	Treatment varies	Unit of the random effect	Variance of treatment effect	Without random effect
Pseudo-replication	between units	experimental unit (tank, bio-repeat)	between + within unit	SE too small: liberal
Blocking (RCB)	within units	block (mouse)	within unit only	SE too large: conservative

- In both designs the random intercept induces the correlation between observations of the same unit
- The mixed model automatically uses the correct variance for the treatment effect.
- In balanced designs it gives the same result as the two-stage analysis, but it also works for unbalanced designs.

Open issues:

- Mixed model is problematic if random effect variance is estimated to be zero. Then data are treated as independent
- Inference relies on asymptotic theory
- Approximate inference in small samples is possible using approximations for the degrees of freedom
- Unclear how to borrow strength across features for both random effect variance and error variance, and how to approximate the degrees of freedom in this context.

Pseudo-replication in proteomics

- Experiments with technical repeats for each biological repeat (see tutorials)
- Inference at protein-level from peptide level data

Peptide level vs protein level analysis

- Inference at protein level desired
- Observational unit: precursors/PSM level
- We perform two-stage analysis:
 1. Summarisation
 2. DA on protein summaries
- Problem: theoretically invalid due to unbalanced data due to missingness
- However, shown to work well in practice

With msqrob we started using peptide level models and explicitly addressed pseudo-replication:

$$y_{ip} = \beta_0 + \mathbf{x}_i \beta^{\text{design}} + \beta_p^{\text{peptide}} + b_i^{\text{run}} + \epsilon_{ip}$$

protein-level:

- $\beta^{\text{design}} \rightarrow$ DA estimates immediately estimated at the protein level from peptide level data.
- $b^{\text{run}} \sim N(0, \sigma_{\text{run}}^2)$: random effect addressing pseudo-replication

peptide-level:

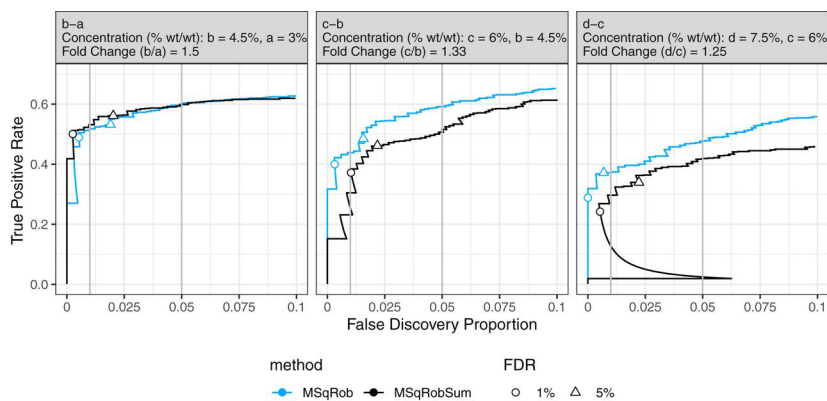
- β_p^{peptide}
- $\epsilon_{ip} \sim N(0, \sigma_\epsilon^2)$

→ Still available in msqrob2 via `msqrobAggregate()`

Problems:

- Dissemination
- No protein level summaries for plotting
- Unclear which degrees of freedom to use
- For some proteins random run effect variance is set at zero: then we act as if precursor/psm/peptide level data are independent
- Computational complexity

See [Sticker et al. 2020, 10.1074/mcp.RA119.001624](#) for comparison between peptide level and protein level models, showing that summarisation based approaches work fine when appropriate summarisation is used.

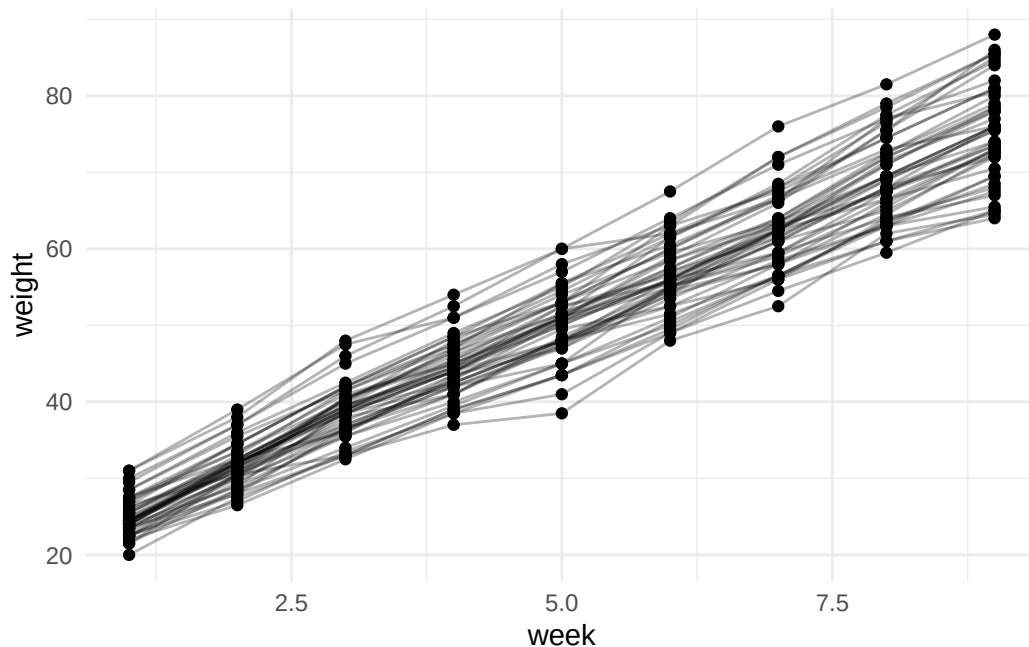


Longitudinal data: pig weights

The weight of 48 pigs was measured in 9 successive weeks (Ruppert et al. 2003). Each pig is measured repeatedly over time: the data are **longitudinal**.

```
pigs <- read.csv("data/pigweights.csv")
pigs$pig <- as.factor(pigs$pig)

ggplot(pigs, aes(x = week, y = weight, group = pig)) +
  geom_point() +
  geom_line(alpha = 0.3) +
  theme_minimal()
```



- The growth curve of an individual pig is much less variable than the cloud of all points.
- Each pig grows approximately linearly, but pigs differ in their weight at the start and in their growth rate.

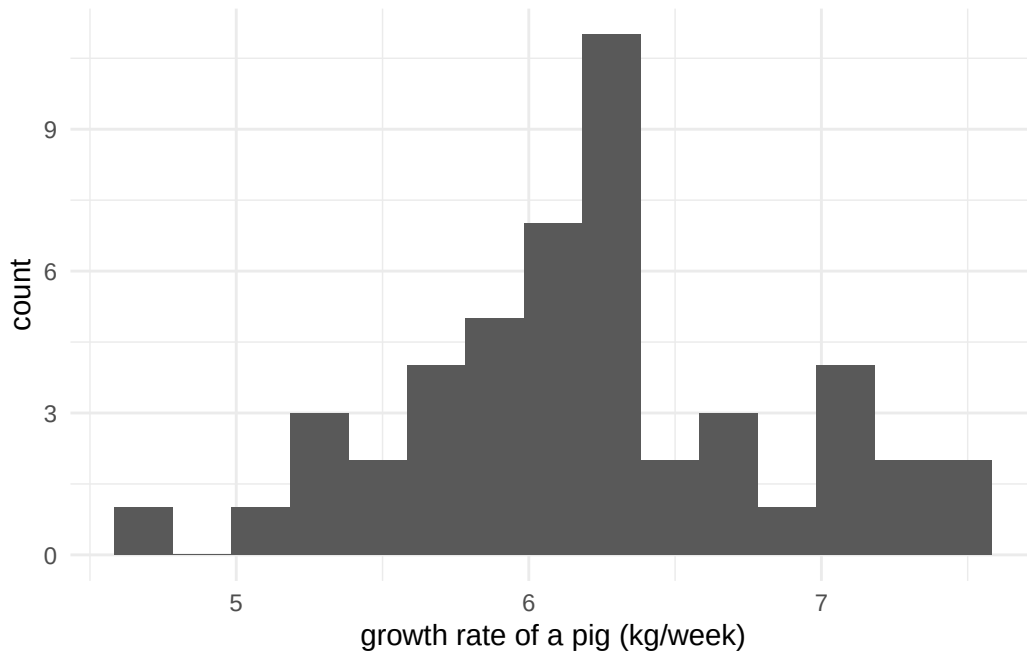
We are interested in the average growth rate of pigs in the population.

Two-stage analysis

- **Stage I:** fit a regression line for each pig i separately: $y_{ij} = \beta_{0i} + \beta_{1i}\text{week}_j + \epsilon_{ij}$.
- **Stage II:** analyse the pig-specific intercepts $\hat{\beta}_{0i}$ and slopes $\hat{\beta}_{1i}$, e.g. estimate the average slope and its standard error.

```
stageIPig <- pigs |>
  group_by(pig) |>
  summarise(
    intercept = coef(lm(weight ~ week))[1],
    slope = coef(lm(weight ~ week))[2])

ggplot(stageIPig, aes(x = slope)) +
  geom_histogram(bins = 15) +
  xlab("growth rate of a pig (kg/week)") +
  theme_minimal()
```



```
twoStagePig <- lm(slope ~ 1, data = stageIPig)
summary(twoStagePig)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.209896	0.09203821	67.47085	1.935218e-48

The standard error of the average growth rate is based on the variability of the growth rates between the 48 pigs, the experimental units.

The two-stage analysis has the same drawbacks as before, and some extra ones that are typical for longitudinal data:

- Information is lost when the data of a pig are summarised by an intercept and a slope.
- The precision of $\hat{\beta}_{0i}$ and $\hat{\beta}_{1i}$ depends on the number of measurements of pig i and on the time points at which they were taken. In practice, longitudinal data are often unbalanced, e.g. due to drop-out, and Stage II does not account for the differences in precision.
- Stage II replaces the true β_{0i} and β_{1i} by estimates, which introduces extra variability that is not modelled.

Linear mixed model with a random intercept and slope

The mixed model combines both stages:

$$y_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})\text{week}_j + \epsilon_{ij}, \quad \begin{bmatrix} b_{0i} \\ b_{1i} \end{bmatrix} \sim N(\mathbf{0}, \mathbf{D}), \quad \epsilon_{ij} \text{ i.i.d. } N(0, \sigma_\epsilon^2),$$

with

- fixed effects β_0 and β_1 : the average intercept and growth rate in the population of pigs,
- random effects b_{0i} and b_{1i} : the deviation of the intercept and the growth rate of pig i from the population averages, independent between pigs,
- variance components σ_ϵ^2 and $\mathbf{D} = \begin{bmatrix} \sigma_{b_0}^2 & \sigma_{b_0 b_1} \\ \sigma_{b_0 b_1} & \sigma_{b_1}^2 \end{bmatrix}$.

```
lmmPigSlope <- lmer(weight ~ week + (week|pig), data = pigs)
summary(lmmPigSlope)
```

Linear mixed model fit by REML ['lmerMod']

Formula: weight ~ week + (week | pig)

Data: pigs

REML criterion at convergence: 1740.9

Scaled residuals:

Min	1Q	Median	3Q	Max
-3.6202	-0.5474	0.0150	0.5486	2.9939

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
pig	(Intercept)	6.986	2.6432	
	week	0.380	0.6164	-0.06
Residual		1.597	1.2637	

Number of obs: 432, groups: pig, 48

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	19.35561	0.40387	47.93
week	6.20990	0.09204	67.47

Correlation of Fixed Effects:

	(Intr)
week	-0.132

```

rbind(
  `two-stage` = summary(twoStagePig)$coef[1, 1:2],
  `mixed model` = summary(lmmPigSlope)$coef["week", 1:2])

```

	Estimate	Std. Error
two-stage	6.209896	0.09203821
mixed model	6.209896	0.09203775

The mixed model gives the same estimate and standard error of the average growth rate as the two-stage analysis, because the design is balanced: every pig is weighed in the same 9 weeks.

The random intercept and slope imply that the variance increases over time and that the correlation between two measurements of the same pig depends on when they were taken:

$$\text{cov}[y_{ij}, y_{ij'}] = \begin{bmatrix} 1 & \text{week}_j \end{bmatrix} \mathbf{D} \begin{bmatrix} 1 \\ \text{week}_{j'} \end{bmatrix} + \sigma_\epsilon^2 \delta_{jj'} = \sigma_{b_0}^2 + \sigma_{b_0 b_1} (\text{week}_j + \text{week}_{j'}) + \sigma_{b_1}^2 \text{week}_j \text{week}_{j'} + \sigma_\epsilon^2 \delta_{jj'},$$

with $\delta_{jj'} = 1$ if $j = j'$ and 0 otherwise.

Why not a random intercept only?

We compare the mixed model with three simpler models:

1. Ignore the pigs: $y_{ij} = \beta_0 + \beta_1 \text{week}_j + \epsilon_{ij}$.
2. A fixed intercept α_i for each pig, as in a block design: $y_{ij} = \alpha_i + \beta_1 \text{week}_j + \epsilon_{ij}$.
3. A random intercept for each pig: $y_{ij} = \beta_0 + b_i + \beta_1 \text{week}_j + \epsilon_{ij}$, b_i i.i.d. $N(0, \sigma_b^2)$.

```

lmOLS <- lm(weight ~ week, data = pigs)
lmFixed <- lm(weight ~ pig + week, data = pigs)
lmmPig <- lmer(weight ~ week + (1|pig), data = pigs)

rbind(
  `model 1: ignore pigs` = summary(lmOLS)$coef["week", 1:2],
  `model 2: fixed pig intercepts` = summary(lmFixed)$coef["week", 1:2],
  `model 3: random intercept` = summary(lmmPig)$coef["week", 1:2],
  `two-stage: slope per pig` = summary(twoStagePig)$coef[1, 1:2],
  `random intercept + slope` = summary(lmmPigSlope)$coef["week", 1:2])

```

	Estimate	Std. Error
model 1: ignore pigs	6.209896	0.08184093
model 2: fixed pig intercepts	6.209896	0.03906329
model 3: random intercept	6.209896	0.03906329
two-stage: slope per pig	6.209896	0.09203821
random intercept + slope	6.209896	0.09203775

- Model 2 (a block analysis with a fixed effect for each pig) and model 3 (random intercept) give a standard error that is less than half as large as the two-stage analysis: they are too liberal.
- They remove the differences in level between pigs, but assume that all pigs grow at the same rate.
- The variation of the growth rates between pigs (the pig $\times$ week interaction) then ends up in the residuals, and the slope is tested against the within-pig variability only.
- This is again a form of pseudo-replication: the 9 measurements of a pig are treated as 9 independent pieces of information on the growth rate in the population.
- Model 2 also uses 49 parameters, and its intercepts only describe *these* 48 pigs.
- Model 1 is closer to the correct standard error, but only because two errors partly cancel each other out.

Whenever an effect is assessed repeatedly within an experimental unit, its variation between the units has to be part of its standard error!

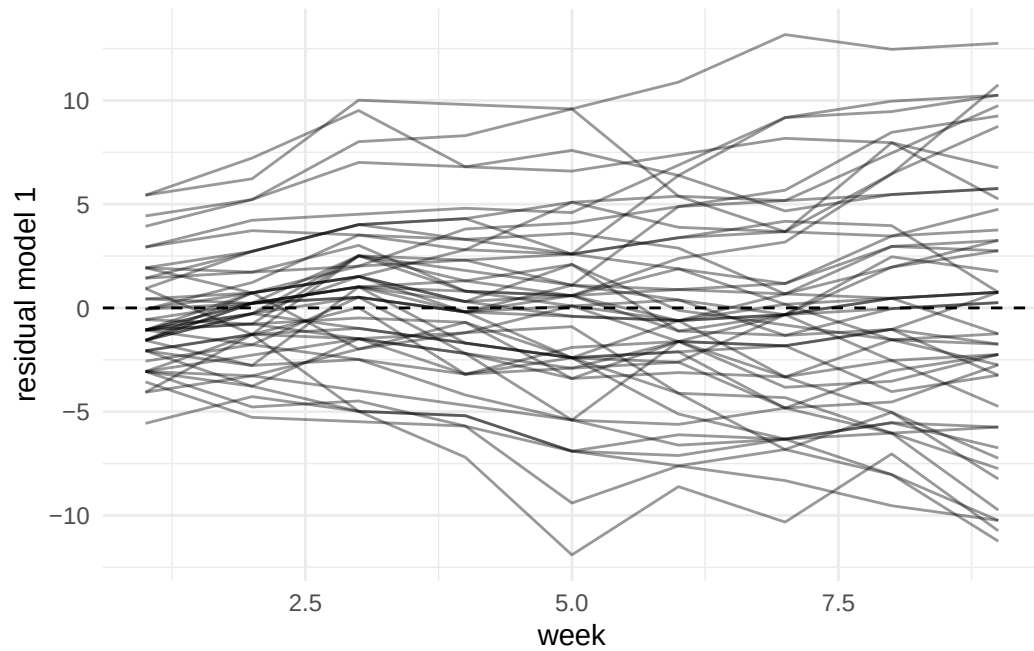
- In the mouse T-cell example there is only one observation per mouse and cell type
- so the mouse $\times$ cell type interaction automatically ends up in the residual and the block analysis is correct.
- If each mouse were measured in several MS-runs per cell type, a random effect for mouse $\times$ cell type, $(1|\text{mouse}) + (1|\text{mouse:celltype})$, would be needed.

Residuals of model 1

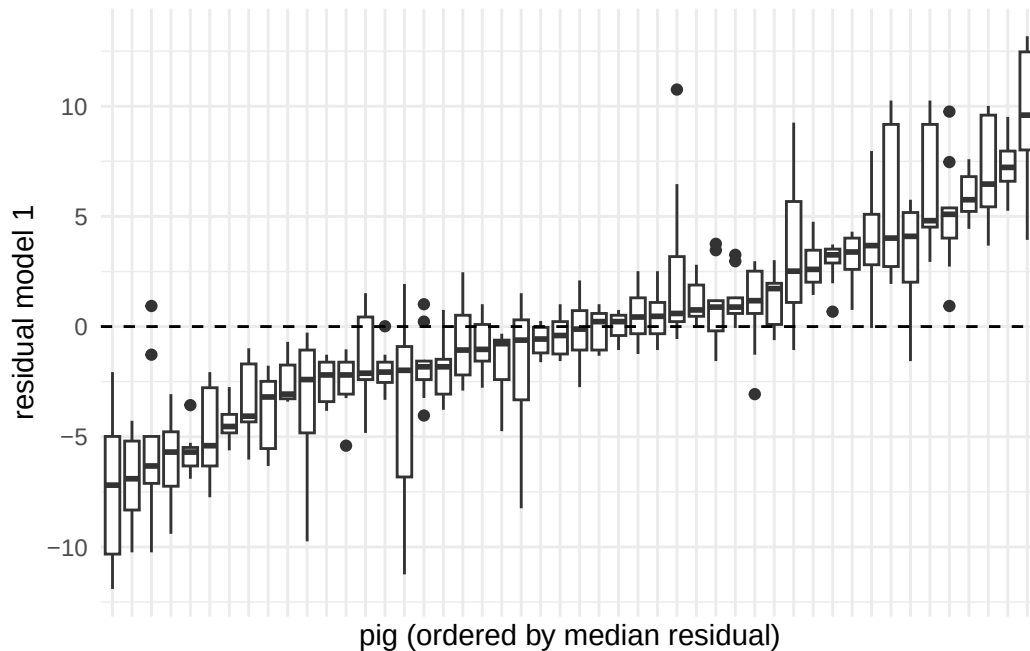
The residuals of model 1, which ignores the pigs, show the correlation structure in the data.

```
pigs$resOLS <- residuals(lmOLS)

ggplot(pigs, aes(x = week, y = resOLS, group = pig)) +
  geom_line(alpha = 0.4) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  ylab("residual model 1") +
  theme_minimal()
```



```
pigs |>
  mutate(pig = fct_reorder(pig, resOLS, .fun = median)) |>
  ggplot(aes(x = pig, y = resOLS)) +
  geom_boxplot() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("pig (ordered by median residual)") +
  ylab("residual model 1") +
  theme_minimal() +
  theme(axis.text.x = element_blank())
```



```
resWide <- pigs |>
  dplyr::select(pig, week, resOLS) |>
  pivot_wider(names_from = week, values_from = resOLS)
```

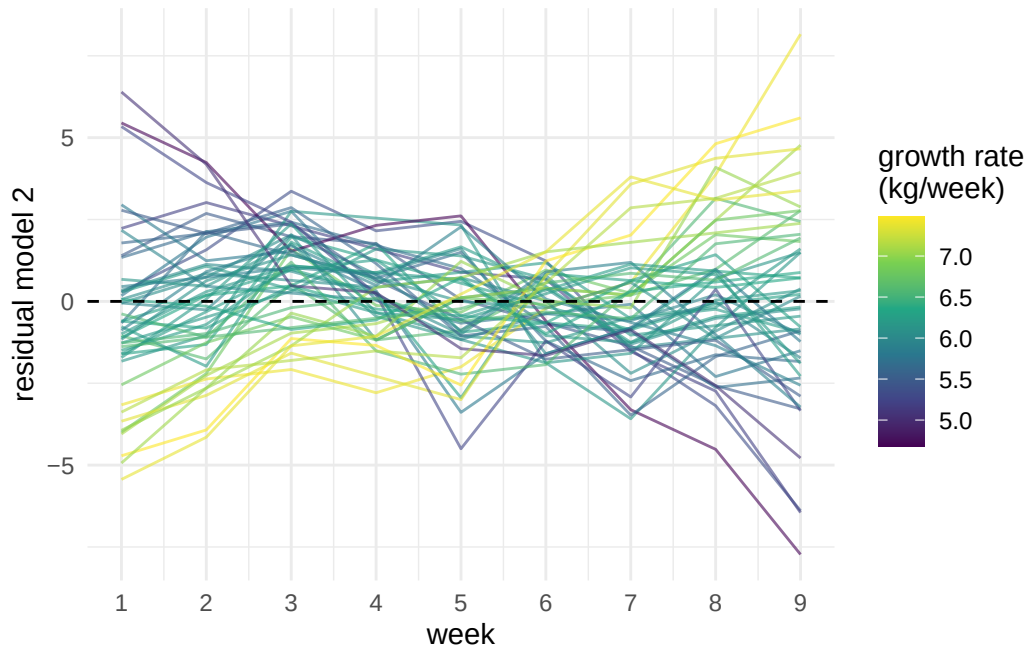
- A pig that is heavier than average in week 1 tends to remain heavier than average in all other weeks: the residuals of the same pig are strongly correlated.
- The residuals of different pigs are centred around very different values: for many pigs the whole box lies above or below zero, and the boxes of the lightest and heaviest pigs do not overlap.
- The residuals fan out: their variance increases over time.

Residuals of model 2

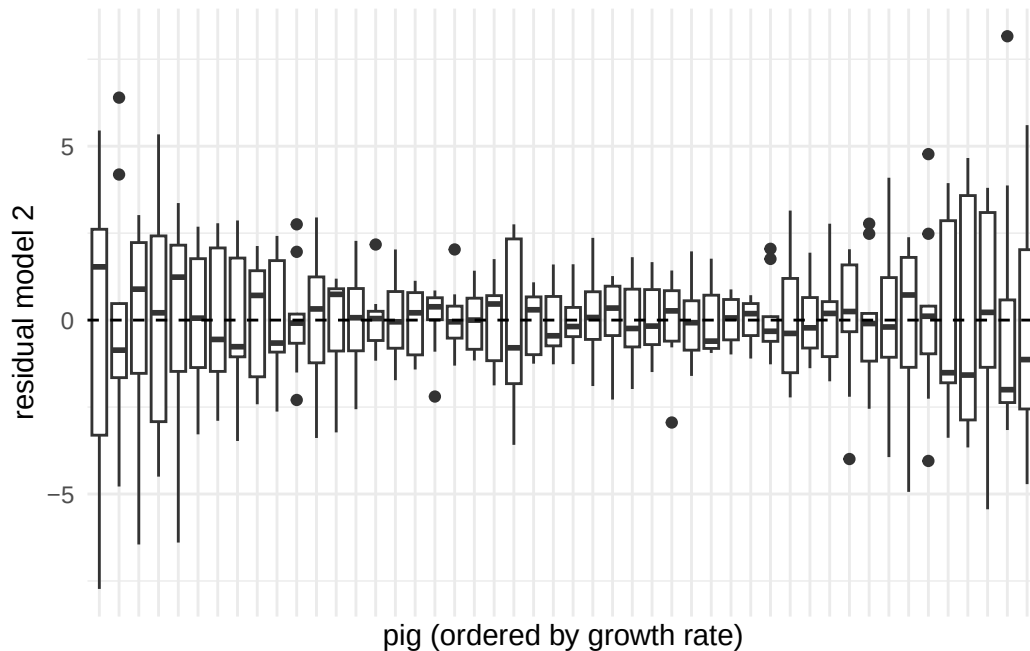
The block analysis of model 2 removes the differences in level between the pigs. Do its residuals still show a correlation structure? We colour the pigs according to their growth rate from Stage I of the two-stage analysis.

```
pigs$resFixed <- residuals(lmFixed)
pigs <- pigs |>
  left_join(stageIPig |> dplyr::select(pig, slope), by = "pig")
ggplot(pigs, aes(x = week, y = resFixed, group = pig, colour = slope)) +
```

```
geom_line(alpha = 0.6) +
geom_hline(yintercept = 0, linetype = "dashed") +
scale_colour_viridis_c(name = "growth rate\n(kg/week)") +
scale_x_continuous(breaks = 1:9) +
ylab("residual model 2") +
theme_minimal()
```



```
pigs |>
mutate(pig = fct_reorder(pig, slope)) |>
ggplot(aes(x = pig, y = resFixed)) +
geom_boxplot() +
geom_hline(yintercept = 0, linetype = "dashed") +
xlab("pig (ordered by growth rate)") +
ylab("residual model 2") +
theme_minimal() +
theme(axis.text.x = element_blank())
```



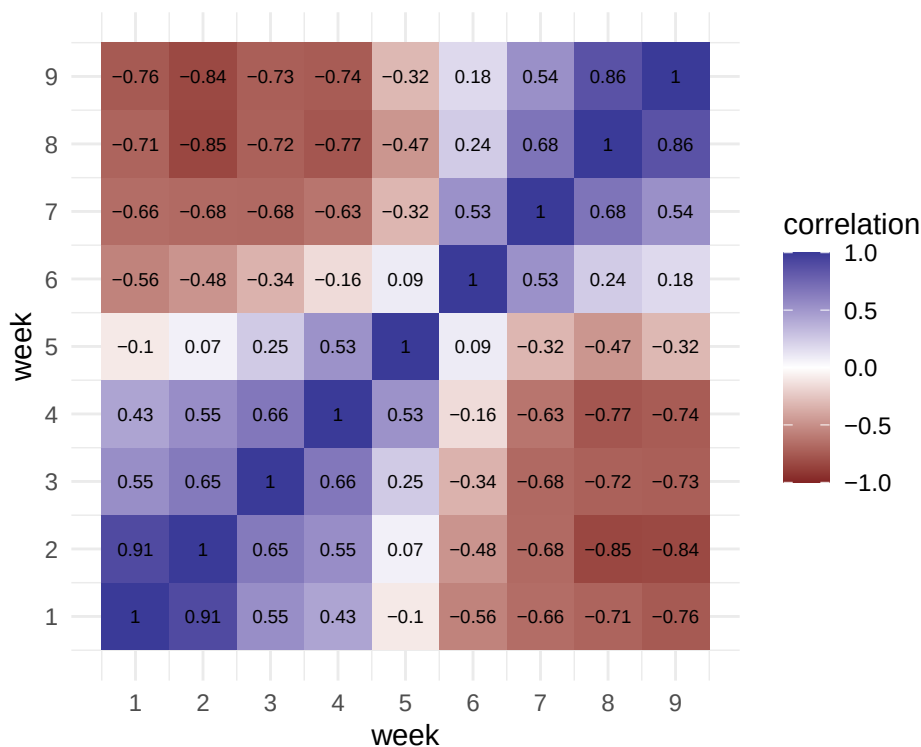
- All boxes are now centred around zero: the fixed pig effects remove the differences in level between the pigs.
- But the residuals of a pig are not random noise around zero: pigs that grow faster than average have negative residuals in the early weeks and positive residuals in the late weeks, and vice versa for pigs that grow slowly.
- The residuals of a pig thus follow a line whose slope is the deviation of its growth rate from the average growth rate: the pig $\times$ week interaction ends up in the residuals.
- The residual variance is smallest in the middle weeks and largest in the first and last weeks.

```
resFixedWide <- pigs |>
  dplyr::select(pig, week, resFixed) |>
  pivot_wider(names_from = week, values_from = resFixed)

corFixed <- cor(resFixedWide[, -1])

tibble(
  week1 = rep(1:9, times = 9),
  week2 = rep(1:9, each = 9),
  correlation = as.vector(corFixed)) |>
  ggplot(aes(x = week1, y = week2, fill = correlation)) +
  geom_tile() +
  geom_text(aes(label = round(correlation, 2)), size = 2.5) +
```

```
scale_fill_gradient2(limits = c(-1, 1)) +
scale_x_continuous(breaks = 1:9) +
scale_y_continuous(breaks = 1:9) +
coord_equal() +
xlab("week") +
ylab("week") +
theme_minimal()
```



- Residuals of the same pig in nearby weeks are positively correlated, residuals in early and late weeks are negatively correlated.
- This is exactly the pattern induced by pig-specific slopes, and it violates the independence assumption of model 2.
- Model 2 treats the 9 residuals of a pig as independent, so the variation of the growth rates between pigs is used as if it were within-pig noise, while the growth rate is assessed within each pig. Its standard error for the average growth rate is therefore too small.

Note that the within-pig error is not underestimated by model 2: its residual standard deviation is even larger than that of the random slope model.

```
c(`model 2` = sigma(lmFixed), `random intercept + slope` = sigma(lmmPigSlope))
```

```
model 2 random intercept + slope
2.096356 1.263658
```

The problem is that model 2 uses the wrong error term to assess the growth rate. We can see this by splitting the residual sum of squares of model 2 into two parts by adding the pig $\times$ week interaction, i.e. a separate slope for each pig:

```
lmPigSlopes <- lm(weight ~ pig + week + pig:week, data = pigs)
ssTab <- anova(lmFixed, lmPigSlopes)

decomp <- tibble(
  source = c(
    "differences in slope between pigs (pig x week)",
    "scatter around the line of each pig",
    "residual model 2 (pooled)" ),
  SS = c(ssTab$`Sum of Sq`[2], ssTab$RSS[2], ssTab$RSS[1]),
  df = c(ssTab$Df[2], ssTab$Res.Df[2], ssTab$Res.Df[1])) |>
  mutate(MS = SS / df)
decomp
```

```
# A tibble: 3 x 4
```

source	SS	df	MS
<chr>	<dbl>	<dbl>	<dbl>
1 differences in slope between pigs (pig x week)	1147.	47	24.4
2 scatter around the line of each pig	537.	336	1.60
3 residual model 2 (pooled)	1683.	383	4.39

- The growth rate is assessed within each pig, so its standard error has to be based on how much the growth rates vary between the pigs: the first row, with 47 degrees of freedom.
- Model 2 does not know about this split: it pools the slope differences with the scatter around the pig-specific lines and divides by 383 instead of 47 degrees of freedom.
- Because there are many more degrees of freedom for the scatter around the lines than for the differences between pigs, the variability of the growth rates is diluted and the error term becomes much too small.
- The mean square of the second row, 1.6, is an estimate of the pure within-pig error variance, close to $\hat{\sigma}_\epsilon^2 = 1.6$ of the random slope model.

With $n = 48$ pigs and $S_{xx} = \sum_{j=1}^9 (\text{week}_j - \overline{\text{week}})^2 = 60$, the standard error of the average growth rate is $\sqrt{\text{MS}/(nS_{xx})}$:

```
Sxx <- sum((1:9 - mean(1:9))^2)
n <- nlevels(pigs$pig)
c(`MS pig x week (correct)` = sqrt(decomp$MS[1] / (n * Sxx)),
  `MS pooled (model 2)` = sqrt(decomp$MS[3] / (n * Sxx)))
```

MS pig x week (correct)	MS pooled (model 2)
0.09203821	0.03906329

- With the mean square of the pig $\times$ week interaction we recover the standard error of the two-stage analysis and of the random slope model.
- With the pooled mean square we recover the much too small standard error of model 2.
- This is the same reasoning as in a split-plot ANOVA: an effect has to be tested against the variability of the units at which it varies.

The random intercept model (model 3) implies **compound symmetry**:

- all measurements of the same pig have the same variance

$$\sigma_b^2 + \sigma_\epsilon^2$$

and

- the same correlation whatever the time between them:

$$\sigma_b^2 / (\sigma_b^2 + \sigma_\epsilon^2).$$

- That is natural for fish in a tank or MS-runs of a bio-repeat, but not for repeated measurements over time when subjects grow at different rates.

With a random intercept and slope, the correlation between two measurements of the same pig depends on the weeks at which they were taken:

$$\text{corr}[y_{ij}, y_{ij'}] = \frac{\sigma_{b_0}^2 + \sigma_{b_0 b_1}(\text{week}_j + \text{week}_{j'}) + \sigma_{b_1}^2 \text{week}_j \text{week}_{j'}}{\sqrt{v_j v_{j'}}}$$

$$v_j = \sigma_{b_0}^2 + 2\sigma_{b_0 b_1} \text{week}_j + \sigma_{b_1}^2 \text{week}_j^2 + \sigma_\epsilon^2.$$

We compare the observed correlations with the correlations implied by both models.

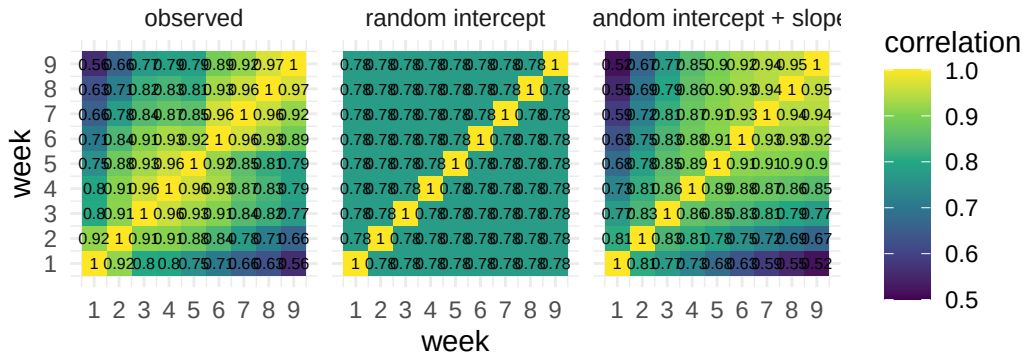
```

# implied correlation matrix of the 9 measurements of a pig
impliedCor <- function(fit, Zi) {
  D <- bdiag(VarCorr(fit)) |> as.matrix()
  Vi <- Zi %*% D %*% t(Zi) + diag(nrow(Zi)) * sigma(fit)^2
  cov2cor(Vi)
}

corMats <- list(
  observed = cor(resWide[, -1]),
  `random intercept` = impliedCor(lmmPig, matrix(1, 9, 1)),
  `random intercept + slope` = impliedCor(lmmPigSlope, cbind(1, 1:9)))

corMats |>
  map(\(corMat) tibble(
    week1 = rep(1:9, times = 9),
    week2 = rep(1:9, each = 9),
    correlation = as.vector(corMat))) |>
  bind_rows(.id = "matrix") |>
  mutate(matrix = factor(matrix, levels = names(corMats))) |>
  ggplot(aes(x = week1, y = week2, fill = correlation)) +
  geom_tile() +
  geom_text(aes(label = round(correlation, 2)), size = 2) +
  scale_fill_viridis_c(limits = c(0.5, 1)) +
  scale_x_continuous(breaks = 1:9) +
  scale_y_continuous(breaks = 1:9) +
  coord_equal() +
  facet_wrap(~ matrix) +
  xlab("week") +
  ylab("week") +
  theme_minimal()

```



- The random intercept model implies the same correlation for all pairs of weeks.
- In the observed data the correlation decreases with the time between the measurements, and the random intercept and slope model captures this pattern. In the early weeks the weight of a pig is mainly determined by its random intercept, in the later weeks by its random slope, whose contribution grows with time. Hence, measurements in the later weeks are more strongly correlated than measurements in the early weeks.
- Measurements in consecutive weeks are still somewhat more correlated than implied by the random slope model, especially in the early weeks. Growth curves are not exactly linear for every pig, which could be modelled with a serial correlation structure or a random quadratic term. We do not pursue this here.

Random treatment effects in block designs

The random slope for the pigs is a random treatment effect: the effect of the “treatment” (time) varies from block (pig) to block. Do we always need such a random block $\times$ treatment effect when we have repeated measures within a block? No: we need it when the residual no longer contains the variability of the treatment effect between blocks. This happens as soon as there is more than one observation per block $\times$ treatment combination.

Design	Example	Model	Random block $\times$ treatment effect
One observation per block $\times$ treatment	mouse T-cells: one Treg and one Tconv sample per mouse	$y \sim \text{treatment} + (1 \mid \text{block})$	not needed: confounded with ϵ
Technical repeats of each block $\times$ treatment	Treg and Tconv sample of each mouse run in technical triplicate	$y \sim \text{treatment} + (1 \mid \text{block}) + (1 \mid \text{block}:\text{treatment})$	always needed: block $\times$ treatment is the experimental unit
Several experimental units per block $\times$ treatment	several mice per litter per treatment	$y \sim \text{treatment} + (1 \mid \text{block}) + (1 \mid \text{block}:\text{treatment})$	needed when the treatment effect varies between blocks
Continuous treatment within block	pig weights over time	$y \sim \text{week} + (\text{week} \mid \text{pig})$	random slope, needed when slopes vary between pigs

- **Technical repeats.** The block $\times$ treatment combination is the experimental unit and the technical repeats are observational units. Omitting b_{ij} for block i and treatment j is pseudo-replication, as in the fish tanks: the treatment effect is then evaluated against the technical variability only, and its standard error is too small. The variance of b_{ij} combines the biological variability and a possible block $\times$ treatment interaction, which cannot be separated. In a balanced design, the mixed model is equivalent to averaging the technical repeats per block $\times$ treatment and an RCB analysis on these averages.
- **Several experimental units per block $\times$ treatment.** Now the block $\times$ treatment interaction $b_{ij} \sim N(0, \sigma_{bt}^2)$ is a genuine interaction that can be separated from the variability between experimental units. With n blocks, the variance of the estimated treatment effect contains $2\sigma_{bt}^2/n$. If $\sigma_{bt}^2 > 0$ and we omit b_{ij} , the standard error is too small and the degrees of freedom are too large: the inference is liberal. It can be omitted if $\sigma_{bt}^2 = 0$, which can be tested with the boundary test of Section . With few blocks, however, σ_{bt}^2 is poorly estimated and keeping it is the safe default. The price is a loss of power: the treatment effect is evaluated against the variability between blocks, with roughly $n - 1$ degrees of freedom.
- **Continuous treatment.** For the pigs, the slopes clearly vary between pigs. Omitting the random slope gives a standard error of 0.039 instead of 0.092 for the average growth rate.
- **Peptide-level models.** In the msqrob peptide-level model, the random run effect b_i^{run} plays the role of the block $\times$ treatment effect: the peptides are repeated measures within each run, which is the experimental unit.

In balanced designs, a two-stage analysis on the block $\times$ treatment averages, or on the block-specific slopes or contrasts, is a quick check of the correct standard error.

Prediction of random effects

The fixed effects and the variance components are estimated, but the random effects are **predicted**: the target is random, not a fixed parameter.

Best prediction

Suppose we observe $y = v + \epsilon$ with $v \sim N(0, 1)$ and $\epsilon \sim N(0, 4)$ independent. What is the best guess for the unobserved v ?

The **best predictor** minimises $E[(\tilde{v} - v)^2]$ and equals $\tilde{v} = E[v|y]$.

Regression view. Because v and y are jointly normal, $E[v|y]$ is the regression line of v on y :

$$E[v|y] = E[v] + \frac{\text{cov}[v, y]}{\text{var}[y]} \{y - E[y]\},$$

with $\text{cov}[v, y] = \text{cov}[v, v + \epsilon] = \text{var}[v] = 1$, $\text{var}[y] = \text{var}[v] + \text{var}[\epsilon] = 5$ and $E[v] = E[y] = 0$. Hence

$$\tilde{v} = E[v|y] = \frac{y}{5}.$$

Only 1/5 of the variability in y comes from v , so we only keep 1/5 of what we observe: the prediction is a *shrunk* version of y .

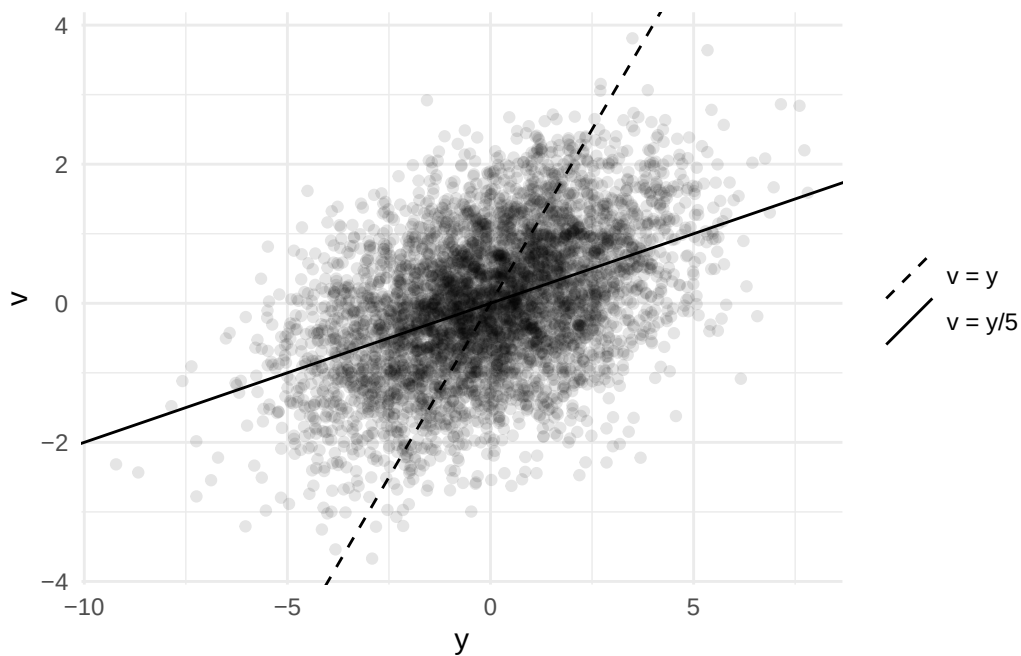
We can see this by simulation: the regression of v on y has slope 1/5, not 1.

```
set.seed(1)
sim <- tibble(
  v = rnorm(5000, mean = 0, sd = 1),
  y = v + rnorm(5000, mean = 0, sd = 2))

coef(lm(v ~ y, data = sim))
```

```
(Intercept)          y
0.001607151 0.208870743
```

```
ggplot(sim, aes(x = y, y = v)) +
  geom_point(alpha = 0.1) +
  geom_abline(
    data = tibble(slope = c(1, 1/5), line = c("v = y", "v = y/5")),
    aes(intercept = 0, slope = slope, linetype = line)) +
  scale_linetype_manual(name = NULL, values = c("dashed", "solid")) +
  theme_minimal()
```



Precision-weighted view. We have two sources of information on v :

- the prior mean $E[v] = 0$, with precision $1/\text{var}[v] = 1$,
- the observation y , with precision $1/\text{var}[\epsilon] = 1/4$.

The best predictor is their precision-weighted average:

$$\tilde{v} = \frac{1 \times 0 + \frac{1}{4} \times y}{1 + \frac{1}{4}} = \frac{y}{5}.$$

The noisier the observation, the lower its precision and the stronger the shrinkage towards the mean of v .

Fish tanks

For the random intercept model of the fish, the best predictor of the effect of tank t is

$$\tilde{b}_t = \frac{n\sigma_b^2}{\sigma_\epsilon^2 + n\sigma_b^2}(\bar{y}_{t\cdot} - \beta_0 - \beta_{d(t)}),$$

i.e. the deviation of the tank average from its diet mean, shrunk towards 0. This is again a precision-weighted average: the tank average has precision n/σ_ϵ^2 and the prior mean 0 of b_t has precision $1/\sigma_b^2$, so

$$\tilde{b}_t = \frac{\frac{1}{\sigma_b^2} \times 0 + \frac{n}{\sigma_\epsilon^2}(\bar{y}_{t\cdot} - \beta_0 - \beta_{d(t)})}{\frac{1}{\sigma_b^2} + \frac{n}{\sigma_\epsilon^2}}.$$

The shrinkage is strong when there are few observations per unit or when σ_ϵ^2 is large compared to σ_b^2 .

```
vcFish <- as.data.frame(VarCorr(lmmFish))
sigma2b <- vcFish$vcov[vcFish$grp == "tank"]
sigma2EpsFish <- vcFish$vcov[vcFish$grp == "Residual"]
nFish <- 6

shrinkFish <- nFish * sigma2b / (sigma2EpsFish + nFish * sigma2b)
shrinkFish
```

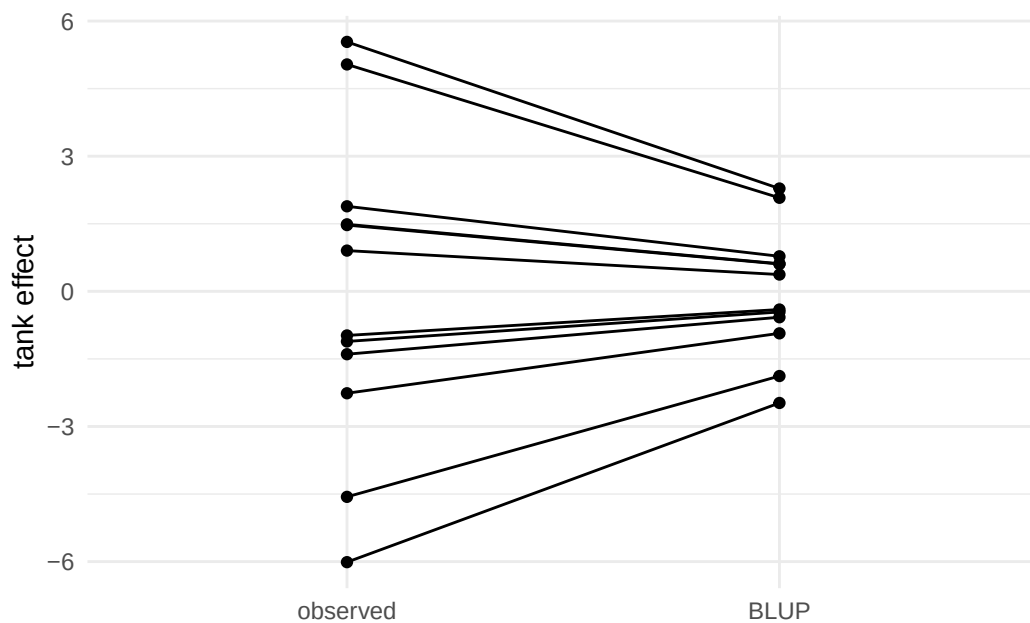
```
[1] 0.4121706
```

```
dietMeans <- predict(lmmFish, newdata = stageIFish, re.form = NA)
tankEffects <- tibble(
  tank = stageIFish$tank,
  `tank average - diet mean` = stageIFish$wtGain - dietMeans,
  byHand = shrinkFish * `tank average - diet mean`,
  lme4 = raneff(lmmFish)$tank[, 1])
tankEffects
```

```
# A tibble: 12 x 4
  tank `tank average - diet mean` byHand    lme4
  <fct>                <dbl>    <dbl>    <dbl>
1 1 -1.40 -0.575 -0.575
2 2 -0.979 -0.404 -0.404
```

3	3	1.47	0.606	0.606
4	4	0.904	0.373	0.373
5	5	5.04	2.08	2.08
6	6	-4.56	-1.88	-1.88
7	7	-6.01	-2.48	-2.48
8	8	5.54	2.28	2.28
9	9	-2.26	-0.933	-0.933
10	10	-1.11	-0.459	-0.459
11	11	1.89	0.778	0.778
12	12	1.49	0.613	0.613

```
tankEffects |>
  dplyr::select(-byHand) |>
  pivot_longer(-tank) |>
  mutate(name = factor(name, levels = c("tank average - diet mean", "lme4"), labels = c("obs", "BLUP"))) |>
  ggplot(aes(x = name, y = value, group = tank)) +
  geom_point() +
  geom_line() +
  xlab("") +
  ylab("tank effect") +
  theme_minimal()
```



With only 6 fish per tank and a within-tank variance that is much larger than the between-tank

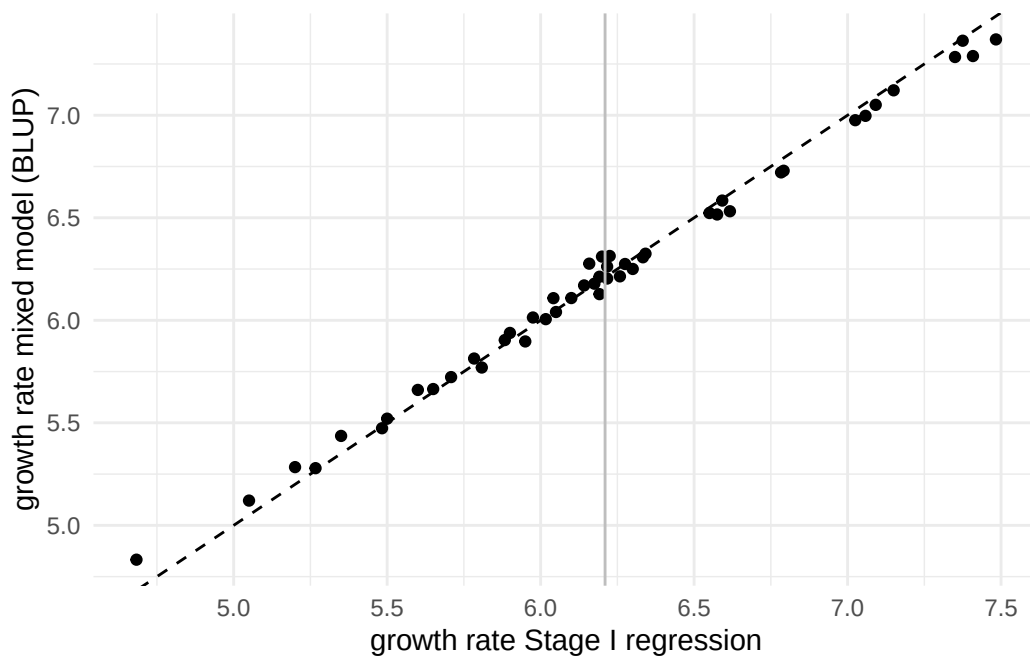
variance, the tank effects are shrunken strongly towards zero.

Pig-specific growth curves

For the pigs, the mixed model predicts a pig-specific intercept $\beta_0 + b_{0i}$ and growth rate $\beta_1 + b_{1i}$. They are shrunken towards the population averages compared to the Stage I regression lines of each pig.

```
blupPig <- coef(lmmPigSlope)$pig |>
  rownames_to_column("pig")

stageIPig |>
  mutate(pig = as.character(pig)) |>
  left_join(blupPig, by = "pig") |>
  ggplot(aes(x = slope, y = week)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  geom_vline(xintercept = fixef(lmmPigSlope)["week"], colour = "grey") +
  xlab("growth rate Stage I regression") +
  ylab("growth rate mixed model (BLUP)") +
  theme_minimal()
```



The growth rates are shrunk towards the average growth rate (grey line), but only mildly: each pig has 9 measurements and the within-pig variance is small compared to the variance of the growth rates between pigs.

The linear mixed model

In general, the linear mixed model (LMM) is

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon}, \quad E \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\epsilon} \end{bmatrix} = \mathbf{0}, \quad \text{cov} \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix},$$

so that

$$\mathbf{V} = \text{var}[\mathbf{y}] = \mathbf{Z}\mathbf{G}\mathbf{Z}^T + \mathbf{R}.$$

For the pig model with a random intercept and slope:

- $\mathbf{X}$ has a column of ones and a column with the week,
- $\mathbf{b} = (b_{0,1}, \dots, b_{0,48}, b_{1,1}, \dots, b_{1,48})^T$ contains the 48 random intercepts and the 48 random slopes,
- $\mathbf{Z}$ is a 432×96 matrix with the pig indicators and the pig indicators multiplied by the week,
- $\mathbf{G}$ contains $\sigma_{b_0}^2$, $\sigma_{b_1}^2$ and $\sigma_{b_0 b_1}$ for the random effects of the same pig and zeros elsewhere,
- $\mathbf{R} = \sigma_{\epsilon}^2 \mathbf{I}_{432}$.

```
X <- model.matrix(~ week, pigs)
Z <- cbind(
  model.matrix(~ -1 + pig, pigs),
  model.matrix(~ -1 + pig:week, pigs))
y <- pigs$weight
n <- nrow(pigs)
m <- nlevels(pigs$pig)

D <- bdiag(VarCorr(lmmPigSlope)) |> as.matrix()
sigma2Eps <- sigma(lmmPigSlope)^2

G <- kronecker(D, diag(m))
R <- sigma2Eps * diag(n)
V <- Z %*% G %*% t(Z) + R

V[1:9, 1:9] |> round(1)
```

	1	2	3	4	5	6	7	8	9
1	8.8	7.4	7.7	8.0	8.3	8.5	8.8	9.1	9.4
2	7.4	9.7	8.7	9.4	10.1	10.7	11.4	12.0	12.7
3	7.7	8.7	11.4	10.8	11.9	12.9	13.9	15.0	16.0
4	8.0	9.4	10.8	13.8	13.7	15.1	16.5	17.9	19.3
5	8.3	10.1	11.9	13.7	17.0	17.2	19.0	20.8	22.6
6	8.5	10.7	12.9	15.1	17.2	21.0	21.6	23.8	26.0
7	8.8	11.4	13.9	16.5	19.0	21.6	25.8	26.7	29.3
8	9.1	12.0	15.0	17.9	20.8	23.8	26.7	31.2	32.6
9	9.4	12.7	16.0	19.3	22.6	26.0	29.3	32.6	37.5

$\mathbf{V}$ is block diagonal with one 9×9 block per pig: measurements of different pigs are independent. Within a pig, the variance increases over time.

The LMM can be viewed in two ways.

- **Hierarchical (conditional) model:** conditionally on the random effects, the data follow an ordinary linear model,

$$\mathbf{y}|\mathbf{b} \sim N(\mathbf{X}\beta + \mathbf{Zb}, \mathbf{R}), \quad \mathbf{b} \sim N(\mathbf{0}, \mathbf{G}).$$

Given its random effects, the weights of pig i vary independently around the pig-specific growth line.

- **Marginal model:** if we do not condition on the random effects, the data are correlated,

$$\mathbf{y} \sim N(\mathbf{X}\beta, \mathbf{V}), \quad \mathbf{V} = \mathbf{ZGZ}^T + \mathbf{R}.$$

Weights of the same pig are correlated because they share the same random effects.

The marginal likelihood is used to estimate the fixed effects and the variance components; the joint likelihood of $(\mathbf{y}, \mathbf{b})$ from the hierarchical model is used to predict the random effects.

Estimation and prediction in the LMM

Fixed effects: generalised least squares

Write $\mathbf{y} = \mathbf{X}\beta + \epsilon^*$ with $\epsilon^* = \mathbf{Zb} + \epsilon$ and $\text{var}[\epsilon^*] = \mathbf{V}$. For known $\mathbf{V}$, the best linear unbiased estimator (BLUE) of β is the generalised least squares estimator

$$\tilde{\beta} = (\mathbf{X}^T \mathbf{V}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{V}^{-1} \mathbf{y}.$$

Random effects: BLUP

For known β , the best predictor of $\mathbf{b}$ is $E[\mathbf{b}|\mathbf{y}] = \mathbf{GZ}^T\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\beta)$. The **best linear unbiased predictor** (BLUP) is this best predictor in which the unknown β is replaced by its GLS estimate $\tilde{\beta}$:

$$\tilde{\mathbf{b}} = \mathbf{GZ}^T\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\tilde{\beta}).$$

```
Vinv <- solve(V)
betaTilde <- solve(t(X) %*% Vinv %*% X, t(X) %*% Vinv %*% y)
uTilde <- G %*% t(Z) %*% Vinv %*% (y - X %*% betaTilde)

cbind(GLS = betaTilde[, 1], lme4 = fixef(lmmPigSlope))
```

	GLS	lme4
(Intercept)	19.355613	19.355613
week	6.209896	6.209896

```
head(cbind(
  intercept = uTilde[1:m, 1],
  slope = uTilde[m + 1:m, 1],
  ranef(lmmPigSlope)$pig))
```

	intercept	slope	(Intercept)	week
1	0.2402199	-0.3964045	0.2402199	-0.3964045
2	-1.5915118	0.5113575	-1.5915118	0.5113575
3	-3.5374496	0.3218426	-3.5374496	0.3218426
4	1.9744827	-0.7738000	1.9744827	-0.7738000
5	1.3087301	-0.9259323	1.3087301	-0.9259323
6	-1.1464377	-0.5451286	-1.1464377	-0.5451286

BLUP as penalised least squares

Henderson's justification assumes $\mathbf{y}|\mathbf{b} \sim N(\mathbf{X}\beta + \mathbf{Zb}, \mathbf{R})$ and $\mathbf{b} \sim N(\mathbf{0}, \mathbf{G})$ and maximises the joint likelihood of $(\mathbf{y}, \mathbf{b})$, which amounts to minimising

$$(\mathbf{y} - \mathbf{X}\beta - \mathbf{Zb})^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{X}\beta - \mathbf{Zb}) + \mathbf{b}^T\mathbf{G}^{-1}\mathbf{b}.$$

This is **generalised least squares with a ridge penalty on the random effects** only. With $\mathbf{C} = [\mathbf{X} \ \mathbf{Z}]$ and $\mathbf{B} = \text{blockdiag}(\mathbf{0}, \mathbf{G}^{-1})$ the solution is

$$\begin{bmatrix} \tilde{\beta} \\ \tilde{\mathbf{b}} \end{bmatrix} = (\mathbf{C}^T \mathbf{R}^{-1} \mathbf{C} + \mathbf{B})^{-1} \mathbf{C}^T \mathbf{R}^{-1} \mathbf{y}.$$

```
C <- cbind(X, Z)
B <- bdiag(matrix(0, 2, 2), solve(G)) |> as.matrix()
Rinv <- diag(1 / sigma2Eps, n)
A <- solve(t(C) %*% Rinv %*% C + B)
ridge <- A %*% t(C) %*% Rinv %*% y

head(cbind(ridge = ridge[, 1], GLS_BLUP = c(betaTilde, uTilde)))
```

	ridge	GLS_BLUP
(Intercept)	19.3556134	19.3556134
week	6.2098958	6.2098958
pig1	0.2402199	0.2402199
pig2	-1.5915118	-1.5915118
pig3	-3.5374496	-3.5374496
pig4	1.9744827	1.9744827

Setting the derivative of Henderson's criterion with respect to $\mathbf{b}$ to zero shows where the shrinkage comes from. With $\mathbf{R} = \sigma_\epsilon^2 \mathbf{I}$:

$$\tilde{\mathbf{b}} = (\mathbf{Z}^T \mathbf{Z} + \sigma_\epsilon^2 \mathbf{G}^{-1})^{-1} \mathbf{Z}^T (\mathbf{y} - \mathbf{X} \tilde{\beta}),$$

while least squares with $\mathbf{b}$ as fixed effects would give $(\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T (\mathbf{y} - \mathbf{X} \tilde{\beta})$: the Stage I regression line of each pig minus the population line. The term $\sigma_\epsilon^2 \mathbf{G}^{-1}$ shrinks the random effects towards zero. For a random intercept model with n observations per unit, $\mathbf{Z}^T \mathbf{Z} = n \mathbf{I}$ and $\mathbf{G} = \sigma_b^2 \mathbf{I}$, which gives the shrinkage factor $n\sigma_b^2 / (n\sigma_b^2 + \sigma_\epsilon^2)$ of the fish tanks.

```
res <- y - X %*% betaTilde
uShrunk <- solve(t(Z) %*% Z + sigma2Eps * solve(G), t(Z) %*% res)
uLS <- solve(t(Z) %*% Z, t(Z) %*% res)

# random slopes
head(cbind(LS = uLS[m + 1:m, 1], BLUP = uShrunk[m + 1:m, 1], lme4 = ranef(lmmPigSlope)$pig$w
```

	LS	BLUP	lme4
pig1:week	-0.4265625	-0.3964045	-0.3964045
pig2:week	0.5734375	0.5113575	0.5113575

```

pig3:week  0.4067708  0.3218426  0.3218426
pig4:week -0.8598958 -0.7738000 -0.7738000
pig5:week -1.0098958 -0.9259323 -0.9259323
pig6:week -0.5598958 -0.5451286 -0.5451286

```

Variance components: ML and REML

The variance components in $\mathbf{V}$ are estimated by maximising a likelihood of the marginal model $\mathbf{y} \sim N(\mathbf{X}\beta, \mathbf{V})$.

- **ML** maximises the profile log-likelihood

$$\ell_P(\mathbf{V}) = -\frac{1}{2} \left\{ \log |\mathbf{V}| + (\mathbf{y} - \mathbf{X}\tilde{\beta})^T \mathbf{V}^{-1} (\mathbf{y} - \mathbf{X}\tilde{\beta}) + n \log(2\pi) \right\}.$$

- **REML** (restricted maximum likelihood) maximises the likelihood of linear combinations of $\mathbf{y}$ that do not depend on β :

$$\ell_R(\mathbf{V}) = \ell_P(\mathbf{V}) - \frac{1}{2} \log |\mathbf{X}^T \mathbf{V}^{-1} \mathbf{X}|.$$

What are linear combinations that do not depend on β ? Take $\mathbf{w} = \mathbf{A}^T \mathbf{y}$ with $\mathbf{A}^T \mathbf{X} = \mathbf{0}$. Then

$$\mathbf{w} = \mathbf{A}^T \mathbf{X} \beta + \mathbf{A}^T (\mathbf{Z} \mathbf{b} + \epsilon) = \mathbf{A}^T (\mathbf{Z} \mathbf{b} + \epsilon) \sim N(\mathbf{0}, \mathbf{A}^T \mathbf{V} \mathbf{A}),$$

so the fixed effects drop out and the distribution of $\mathbf{w}$ only depends on the variance components. There are only $n-p$ linearly independent combinations of this kind, the so-called error contrasts, with p the number of fixed effects. You can think of $\mathbf{w}$ as the **residuals** after fitting the fixed effects with least squares, $\{\mathbf{I} - \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T\} \mathbf{y}$: their distribution does not depend on β . REML maximises the likelihood of $\mathbf{w}$, which results in $\ell_R(\mathbf{V})$ above whatever choice of $\mathbf{A}$ is made.

ML estimates the variance components as if $\hat{\beta}$ were the true β , while REML only uses the information that is left after removing the fixed effects. Hence, REML accounts for the degrees of freedom used to estimate the fixed effects. For a sample $X_1, \dots, X_n$ from $N(\mu, \sigma^2)$: $\hat{\sigma}_{\text{ML}}^2 = \frac{1}{n} \sum_i (X_i - \bar{X})^2$ while $\hat{\sigma}_{\text{REML}}^2 = \frac{1}{n-1} \sum_i (X_i - \bar{X})^2$. REML is based on the contrasts $X_i - \bar{X}$, which do not depend on μ , and only $n-1$ of them are linearly independent because they sum to zero. For small samples REML is preferred; for large samples both give similar results. REML is the default in `lmer`.

```
lmmPigSlopeML <- lmer(weight ~ week + (week|pig), data = pigs, REML = FALSE)
tibble(
  component = c("var intercept", "var slope", "cov intercept slope", "var residual"),
  ML = as.data.frame(VarCorr(lmmPigSlopeML))$vcov,
  REML = as.data.frame(VarCorr(lmmPigSlope))$vcov)
```

```
# A tibble: 4 x 3
  component      ML    REML
  <chr>      <dbl> <dbl>
1 var intercept    6.82    6.99
2 var slope        0.372   0.380
3 cov intercept slope -0.0984 -0.103
4 var residual     1.60    1.60
```

Estimated BLUP

In practice $\mathbf{G}$ and $\mathbf{R}$ are unknown and replaced by their (RE)ML estimates:

$$\hat{\beta} = (\mathbf{X}^T \hat{\mathbf{V}}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \hat{\mathbf{V}}^{-1} \mathbf{y}, \quad \hat{\mathbf{b}} = \hat{\mathbf{G}} \mathbf{Z}^T \hat{\mathbf{V}}^{-1} (\mathbf{y} - \mathbf{X} \hat{\beta}).$$

These are **estimated BLUPs** (EBLUPs). They are what `fixef()` and `ranef()` return (and what we computed above, because we plugged in the REML estimates).

Inference

Standard errors

$$\text{var} [\tilde{\beta}] = (\mathbf{X}^T \mathbf{V}^{-1} \mathbf{X})^{-1}$$

and the standard errors are obtained by plugging in $\hat{\mathbf{V}}$.

```
cbind(byHand = sqrt(diag(solve(t(X) %*% Vinv %*% X))),
      lme4 = summary(lmmPigSlope)$coef[, 2])
```

```
              byHand      lme4
(Intercept) 0.40386734 0.40386734
week        0.09203775 0.09203775
```

This ignores the uncertainty in the estimated variance components. That is negligible in large samples but not in small experiments. Small-sample corrections approximate the degrees of freedom of the resulting t -statistics, e.g. Satterthwaite or Kenward-Roger (packages `lmerTest` and `pbkrtest`). The precision of the BLUPs follows from $\text{var} \begin{bmatrix} \tilde{\beta} \\ \tilde{\mathbf{b}} - \mathbf{b} \end{bmatrix} = (\mathbf{C}^T \mathbf{R}^{-1} \mathbf{C} + \mathbf{B})^{-1}$.

Tests for fixed effects

- **Wald test:** $z = \hat{\beta}_k / \text{se}(\hat{\beta}_k) \approx N(0, 1)$ under $H_0 : \beta_k = 0$.
- **Likelihood ratio test (LRT):** $-2 \log \text{LR} = -2\{\ell(\hat{\theta}_0) - \ell(\hat{\theta})\} \approx \chi_\nu^2$, with ν the difference in the number of parameters.

Use ML, not REML, to compare models with different fixed effects: the restricted likelihood is the likelihood of the residuals after fitting the fixed effects, so it is not comparable across different fixed effects models.

```
lmmPigSlopeML0 <- lmer(weight ~ 1 + (week|pig), data = pigs, REML = FALSE)
anova(lmmPigSlopeML0, lmmPigSlopeML)
```

Data: pigs

Models:

lmmPigSlopeML0: weight ~ 1 + (week | pig)

lmmPigSlopeML: weight ~ week + (week | pig)

	npar	AIC	BIC	logLik	-2*log(L)	Chisq	Df	Pr(>Chisq)
lmmPigSlopeML0	5	1967.9	1988.3	-978.97	1957.9			
lmmPigSlopeML	6	1749.9	1774.3	-868.96	1737.9	220.01	1	< 2.2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Testing a variance component

Is there between-unit variability, e.g. $H_0 : \sigma_b^2 = 0$ versus $H_1 : \sigma_b^2 > 0$ in the random intercept model?

Under H_0 the parameter is on the **boundary** of its parameter space $[0, \infty)$, so the classical χ_1^2 approximation of the LRT does not hold. When the number of independent subjects is large, $-2 \log \text{LR}$ follows a 50:50 mixture of a point mass at zero and a χ_1^2 distribution (Self and Liang 1987): in half of the datasets $\hat{\sigma}_b^2 = 0$. Using χ_1^2 therefore gives p-values that are twice too large.

We can check this by simulating data under H_0 , i.e. from the fitted model 1 that ignores the pigs, and computing the LRT for each simulated dataset.

```

set.seed(1)
lrtNull <- replicate(200, {
  ySim <- simulate(lmOLS)[, 1]
  ml1 <- lmer(ySim ~ week + (1|pig), data = pigs, REML = FALSE)
  ml0 <- lm(ySim ~ week, data = pigs)
  max(0, 2 * (as.numeric(logLik(ml1)) - as.numeric(logLik(ml0))))
})

mean(lrtNull < 1e-6) # proportion of LRT statistics equal to zero

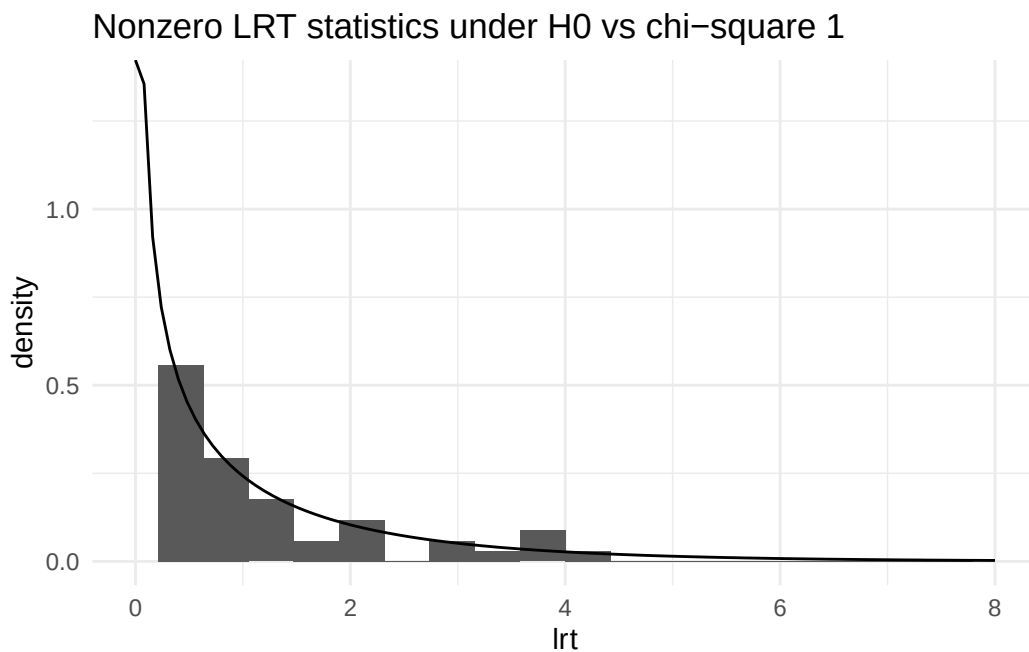
```

```
[1] 0.595
```

```

ggplot(data.frame(lrt = lrtNull[lrtNull > 1e-6]), aes(x = lrt)) +
  geom_histogram(aes(y = after_stat(density)), bins = 20) +
  geom_function(fun = dchisq, args = list(df = 1)) +
  xlim(0, 8) +
  ggtitle("Nonzero LRT statistics under H0 vs chi-square 1") +
  theme_minimal()

```



About half of the simulated LRT statistics are zero; in finite samples this proportion is typically somewhat larger than $1/2$ (Ruppert et al. 2003), so the 50:50 mixture is slightly conservative.

To test whether we need a random slope, we compare the random intercept model with the random intercept and slope model. Under H_0 the variance of the random slope is zero and the covariance between the random intercept and slope then also vanishes. The LRT then follows a 50:50 mixture of a χ_1^2 and a χ_2^2 distribution. Both models have the same fixed effects, so we can also use their restricted likelihoods.

```
anova(lmmPig, lmmPigSlope, refit = FALSE)
```

Data: pigs

Models:

lmmPig: weight ~ week + (1 | pig)

lmmPigSlope: weight ~ week + (week | pig)

	npars	AIC	BIC	logLik	-2*log(L)	Chisq	Df	Pr(>Chisq)
lmmPig	4	2041.8	2058.1	-1016.90	2033.8			
lmmPigSlope	6	1752.9	1777.3	-870.44	1740.9	292.93	2	< 2.2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
lrtSlope <- 2 * (as.numeric(logLik(lmmPigSlope)) - as.numeric(logLik(lmmPig)))
c(LRT = lrtSlope,
  pval = 0.5 * pchisq(lrtSlope, df = 1, lower.tail = FALSE) +
    0.5 * pchisq(lrtSlope, df = 2, lower.tail = FALSE))
```

	LRT	pval
	2.929256e+02	1.290343e-64

The random slope is clearly needed. With few subjects or many observations per subject the mixture approximations are also inaccurate. In those cases the null distribution is best obtained by simulation, as above (parametric bootstrap).

Summary

Two-stage analysis

summarise each experimental unit (Stage I),
analyse the summaries (Stage II); only valid
for balanced designs

Model

$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon}$, $\text{var}[\mathbf{b}] = \mathbf{G}$, $\text{var}[\boldsymbol{\epsilon}] = \mathbf{R}$

Marginal variance

$\mathbf{V} = \mathbf{ZGZ}^T + \mathbf{R}$

Fixed effects (GLS)

$\tilde{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{V}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{V}^{-1} \mathbf{y}$

Random effects (BLUP)	$\tilde{\mathbf{b}} = \mathbf{GZ}^T \mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}})$
Ridge form	$(\tilde{\boldsymbol{\beta}}^T, \tilde{\mathbf{b}}^T)^T = (\mathbf{C}^T \mathbf{R}^{-1} \mathbf{C} + \mathbf{B})^{-1} \mathbf{C}^T \mathbf{R}^{-1} \mathbf{y}$
Variance components	ML or REML: $\ell_R(\mathbf{V}) = \ell_P(\mathbf{V}) - \frac{1}{2} \log \mathbf{X}^T \mathbf{V}^{-1} \mathbf{X} $
Standard errors	$\text{var} [\tilde{\boldsymbol{\beta}}] = (\mathbf{X}^T \mathbf{V}^{-1} \mathbf{X})^{-1}$, plug in $\hat{\mathbf{V}}$
Test a variance component	boundary problem: mixture of χ^2 distributions, or simulate

Ruppert, D., M. P. Wand, and R. J. Carroll. 2003. *Semiparametric Regression*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Self, Steven G., and Kung-Yee Liang. 1987. “Asymptotic Properties of Maximum Likelihood Estimators and Likelihood Ratio Tests Under Nonstandard Conditions.” *Journal of the American Statistical Association* 82 (398): 605–10.