WHERE ARE TREATMENTS ASSIGNED?

CLASS 8: SHARED PENS AND SPLIT PLOTS

Gavin Simpson
Aarhus University

Wednesday, 28 October 2026

TODAY'S ROUTE

Time	Work
12:30–12:40	Opening recall: animals and pens
12:40–13:20	Random effects, wider slope variation and centred time
13:20–13:40	Split plots: two treatment assignments
13:40–13:55	Break
13:55–14:25	Latin squares: block in two directions
14:25–14:55	Unequal group sizes
14:55–15:10	Break
15:10–16:00	Practice, project audit and exit

OPENING RECALL: WHAT RECEIVED THE DIET?

Four rooms contain four pens each. Each pen receives one diet. We measure six animals in every pen.

Draw **room → pen → animal**, and mark where diet was assigned.

If we measure six more animals in each pen, how many new diet assignments have we made?

SET UP YOUR R SESSION

Open the supplied `08-glmm-analysis` Posit Cloud project. Run these lines once before reading the pig records.

```
library("readr")
library("dplyr", warn.conflicts = FALSE)
library("tibble")
library("tidyr")
library("ggplot2")
library("lme4")
library("performance")
library("marginaleffects")
```

- `readr` reads CSV files; `dplyr` modifies the recorded columns.
- `tibble` keeps tables and IDs together; `tidyr` creates combinations.
- `ggplot2` makes plots.
- `lme4` fits the Gaussian mixed models; `lm()` fits models without random effects.
- `performance` checks models; `marginaleffects` estimates predictions and comparisons.

READ THE PIG RECORDS

Each row is one animal's gain in grams per day. These data are invented for teaching.

```
d <- readr::read_csv("pen-animals.csv", show_col_types = FALSE)
d |>
  select(room, pen, diet, gain_g_day) |>
  slice_head(n = 2)
```

```
# A tibble: 2 × 4
  room  pen    diet       gain_g_day
  <chr> <chr>  <chr>           <dbl>
1 R1    R1-P01 High fibre       783.
2 R1    R1-P01 High fibre       787.
```

COUNT PENS BEFORE FITTING

`factor()` tells R which named groups a column contains. The first diet is the reference for comparisons.

```
d <- d |>
  mutate(
    room = factor(room),
    pen = factor(pen),
    diet = factor(diet, levels = c("Control", "High fibre"))
  )
pen_design <- d |>
  distinct(room, pen, diet)
pen_design |>
  count(diet, name = "pens")
```

```
# A tibble: 2 × 2
  diet        pens
  <fct>      <int>
1 Control        8
2 High fibre     8
```

Researchers feed each diet to **eight pens**. The 48 pigs eating that diet share their pen's feed and conditions.

PLOT THE PIGS BEFORE FITTING A MODEL

Each point will show one pig. Give pens separate positions, show each room in its own panel, and use colour for the diet.

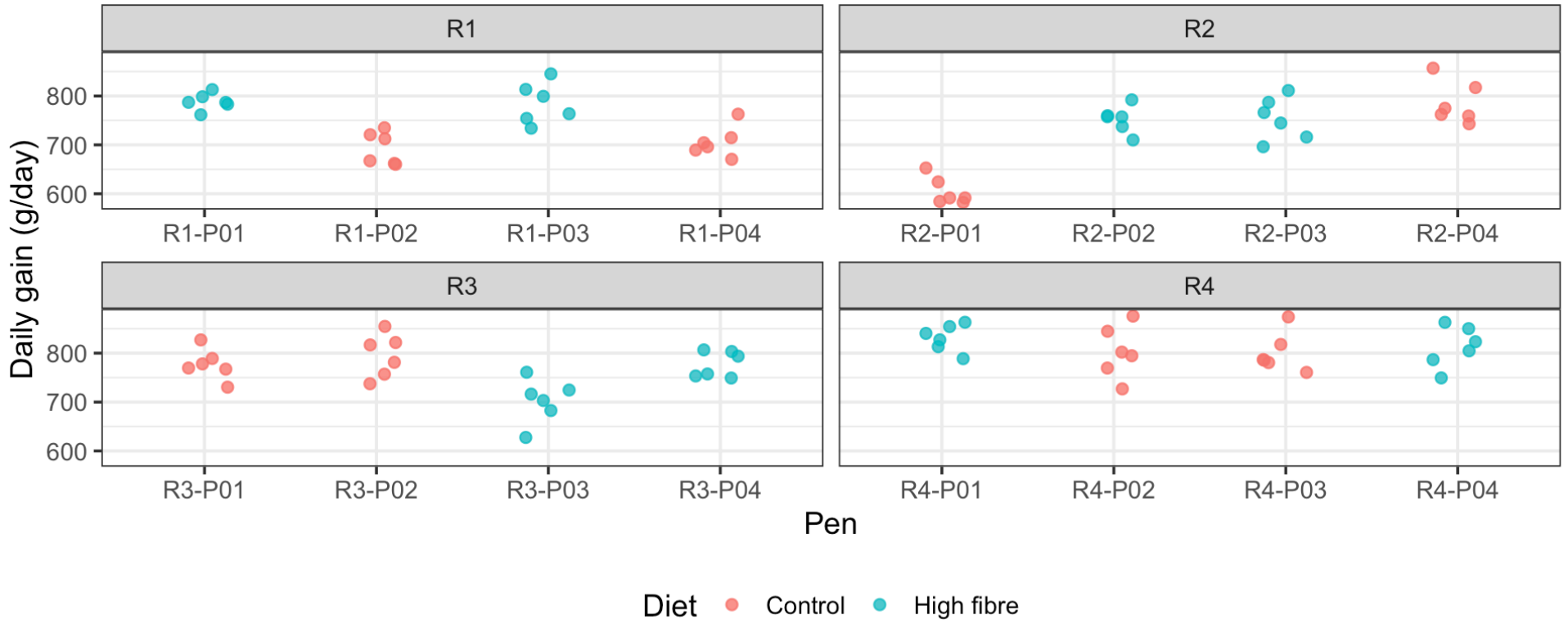
```
pig_plot <- ggplot(d, aes(pen, gain_g_day, colour = diet)) +
  geom_point(
    position = position_jitter(width = 0.15, height = 0, seed = 828),
    size = 2,
    alpha = 0.8
  ) +
  facet_wrap(~ room, scales = "free_x", nrow = 2) +
  labs(x = "Pen", y = "Daily gain (g/day)", colour = "Diet") +
  theme_bw(base_size = 14) +
  theme(legend.position = "bottom")
```

The horizontal jitter separates overlapping points. It does not change a pig's daily gain.

COMPARE PIGS WITHIN AND BETWEEN PENS

Researchers assign one diet to each pen. All six pigs in that pen eat the same diet.

```
pig_plot
```



Within a room, do pigs in some pens tend to gain more than pigs in other pens?

WHY ADD A PEN DIFFERENCE?

Animals in one pen share conditions even after allowing for diet and room.

Think of an animal's gain as three parts:

Diet-and-room mean + shared pen difference + remaining animal difference.

A **random effect** represents a group difference using a common distribution. Here, pen differences follow a bell-shaped distribution centred on zero, with a spread estimated from the data.

READ THE FORMULA IN WORDS

```
gain_g_day ~ diet + room + (1 | pen)
```

Part	Meaning in this study
`gain_g_day ~`	Explain gain in grams per day.
`diet + room`	Estimate diet differences and differences among these four rooms. These are **fixed effects**.
`(1 | pen)`	Give each pen a shared difference, with their spread described by one common distribution.

Using fixed and random effects together gives a **mixed model**.

READ `(1 | pen)` ONE SYMBOL AT A TIME

Symbol	Meaning
`1`	Allow the overall level, called the **intercept**, to differ. It does not mean a difference of one gram.
`|`	Read this as "for each group identified by". It does not mean division.
`pen`	The column identifying the groups. Animals with the same pen ID share its difference.

The shared difference moves the whole pen's expected gain up or down. This is a **random intercept**.

R includes an overall intercept by default. The `1` inside the brackets adds pen differences around the diet-and-room mean.

"RANDOM" DOES NOT MEAN RANDOMLY ASSIGNED

The diet comparison asks whether the two diets differ.

The random pen effect describes how pens still vary after diet and room are allowed for. It also represents the similarity of animals sharing a pen.

Random assignment supports the treatment comparison. A random effect describes variation; it does not create random assignment or extra treatment replication.

FIT A SHARED PEN DIFFERENCE

```r
# REML accounts for estimating
# the diet and room means.
shared_pen <- lme4::lmer(
  gain_g_day ~
    diet + room + (1 | pen),
  data = d,
  REML = TRUE
)
summary(shared_pen)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: gain_g_day ~ diet + room + (1 | pen)
   Data: d

REML criterion at convergence: 959

Scaled residuals:
   Min     1Q Median     3Q    Max
-2.195 -0.668 -0.100  0.676  2.099

Random effects:
 Groups   Name        Variance Std.Dev.
 pen      (Intercept) 2454     49.5
 Residual             1375     37.1
Number of obs: 96, groups:  pen, 16

Fixed effects:
              Estimate Std. Error t value
(Intercept)       728.2       29.0   25.15
dietHigh fibre     29.9       25.9    1.15
roomR2            -19.2       36.6   -0.52
roomR3             19.8       36.6    0.54
roomR4             68.7       36.6    1.88

Correlation of Fixed Effects:
            (Intr) dtHghf roomR2 roomR3
dietHighfbr -0.447
roomR2      -0.632  0.000
roomR3      -0.632  0.000  0.500
roomR4      -0.632  0.000  0.500  0.500
```

SEE THE FITTED PEN DIFFERENCES

`ranef()` extracts the fitted group differences. Here we select the pen part of the model; each row is a pen, in grams/day.

```
pen_differences <- ranef(shared_pen)$pen |>
  rownames_to_column("pen")
pen_differences |>
  slice_head(n = 4)
```

```
     pen (Intercept)
1 R1-P01        27.7
2 R1-P02       -32.2
3 R1-P03        24.6
4 R1-P04       -20.1
```

```
VarCorr(shared_pen)
```

```
 Groups   Name        Std.Dev.
 pen      (Intercept) 49.5
 Residual             37.1
```

A positive pen value means higher expected gain than its diet-and-room mean. These estimates are pulled towards zero when a pen provides limited information. This is called **partial pooling**.

A SEPARATE ILLUSTRATION: FORTY CALVES

Suppose we follow forty calves from week 0 to week 8. To make the shapes clear, we specify their mean lines directly.

- Overall starting weight: **35 kg**; average gain: **1.8 kg/week**.
- Calves differ in starting weight; their differences have SD **4 kg**.
- For the second plot, weekly gain also differs; its differences have SD **0.9 kg/week**.

These are invented mean lines, not observed weights or fitted pen results. SD describes spread.

CREATE THE FORTY CALF DIFFERENCES

`rnorm()` draws bell-shaped differences centred on zero. The fixed seed makes the illustration reproducible.

```
start_sd <- 4
gain_sd <- 0.9
set.seed(82727)
calf_effects <- tibble(
  calf = factor(sprintf("C%02d", 1:40)),
  start_difference = rnorm(40, mean = 0, sd = start_sd),
  gain_difference = rnorm(40, mean = 0, sd = gain_sd)
)
calf_effects |>
  slice_head(n = 3)
```

```
# A tibble: 3 × 3
  calf  start_difference gain_difference
  <fct>            <dbl>           <dbl>
1 C01            -13.8            1.33
2 C02              0.680         -1.79
3 C03              2.32           0.453
```

Starting differences are in kg; gain differences are in kg/week. For every calf we use the same two distributions. We draw each calf's pair separately. We will return to this assumption with a QQ plot after the second break.

BUILD BOTH SETS OF MEAN LINES

Create each calf's record at every week, then join its starting and gain differences by calf ID. Keep the same starting differences in both plots.

```
lines <- expand_grid(calf = calf_effects$calf, week = 0:8) |>
  left_join(calf_effects, by = "calf") |>
  mutate(
    intercept_only = 35 + 1.8 * week + start_difference,
    intercept_and_slope = intercept_only + gain_difference * week
  )
weight_limits <- range(
  pull(lines, intercept_only),
  pull(lines, intercept_and_slope)
)
lines |>
  select(calf, week, intercept_only, intercept_and_slope) |>
  slice_head(n = 2)
```

```
# A tibble: 2 × 4
  calf   week intercept_only intercept_and_slope
  <fct> <int>          <dbl>               <dbl>
1 C01       0           21.2                21.2
2 C01       1           23.0                24.3
```

BUILD THE TWO PLOTS

Both plots use the same calves, colours, axes and dashed overall mean. Only the vertical values change in the second plot.

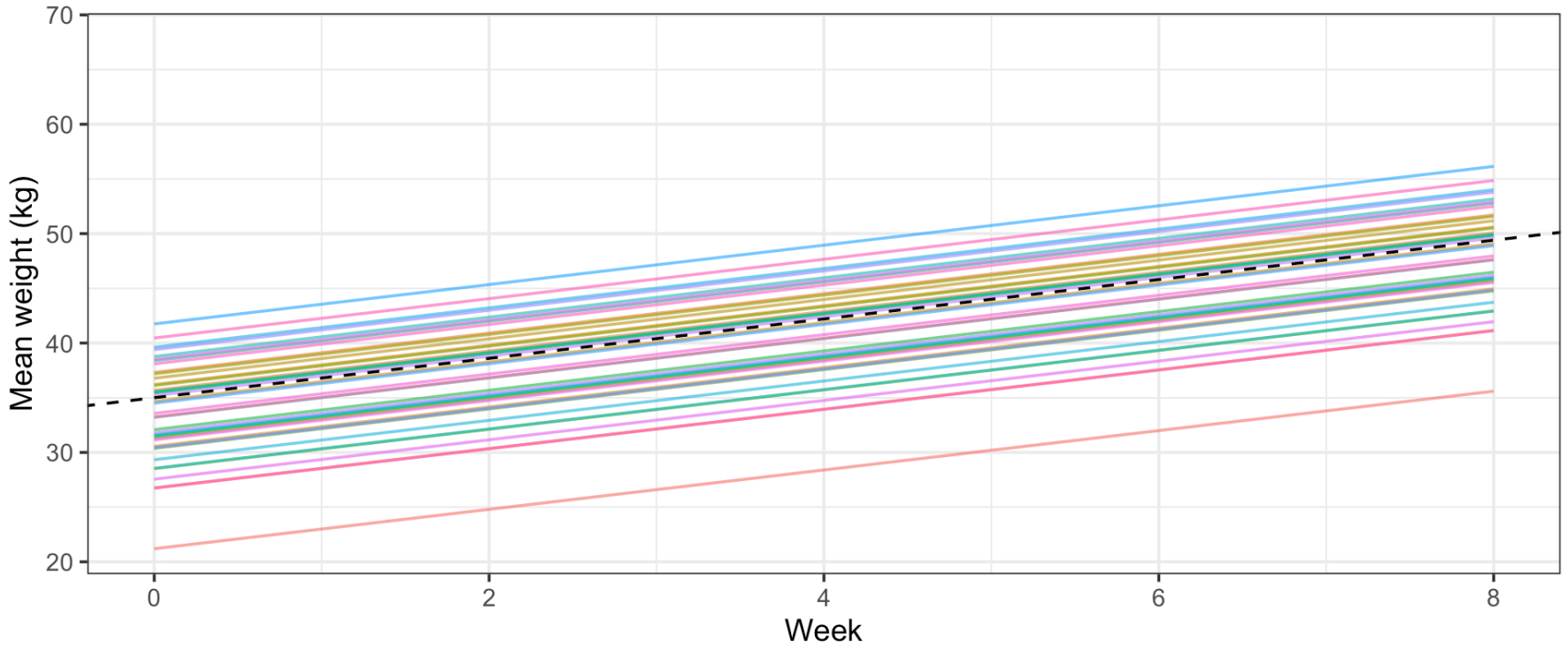
```
p_intercepts <- ggplot(
  lines,
  aes(week, intercept_only, group = calf, colour = calf)
) +
  geom_line(alpha = 0.65) +
  geom_abline(intercept = 35, slope = 1.8, linetype = "dashed") +
  coord_cartesian(ylim = weight_limits) +
  labs(x = "Week", y = "Mean weight (kg)") +
  theme_bw(base_size = 14) +
  theme(legend.position = "none")
p_slopes <- p_intercepts + aes(y = intercept_and_slope)
```

`group = calf` joins records from the same calf. We omit a forty-entry legend so the line shapes remain the focus.

RANDOM INTERCEPTS: PARALLEL LINES

`weight_kg ~ week + (1 | calf)` gives different starting weights but the same weekly gain.

```
p_intercepts
```

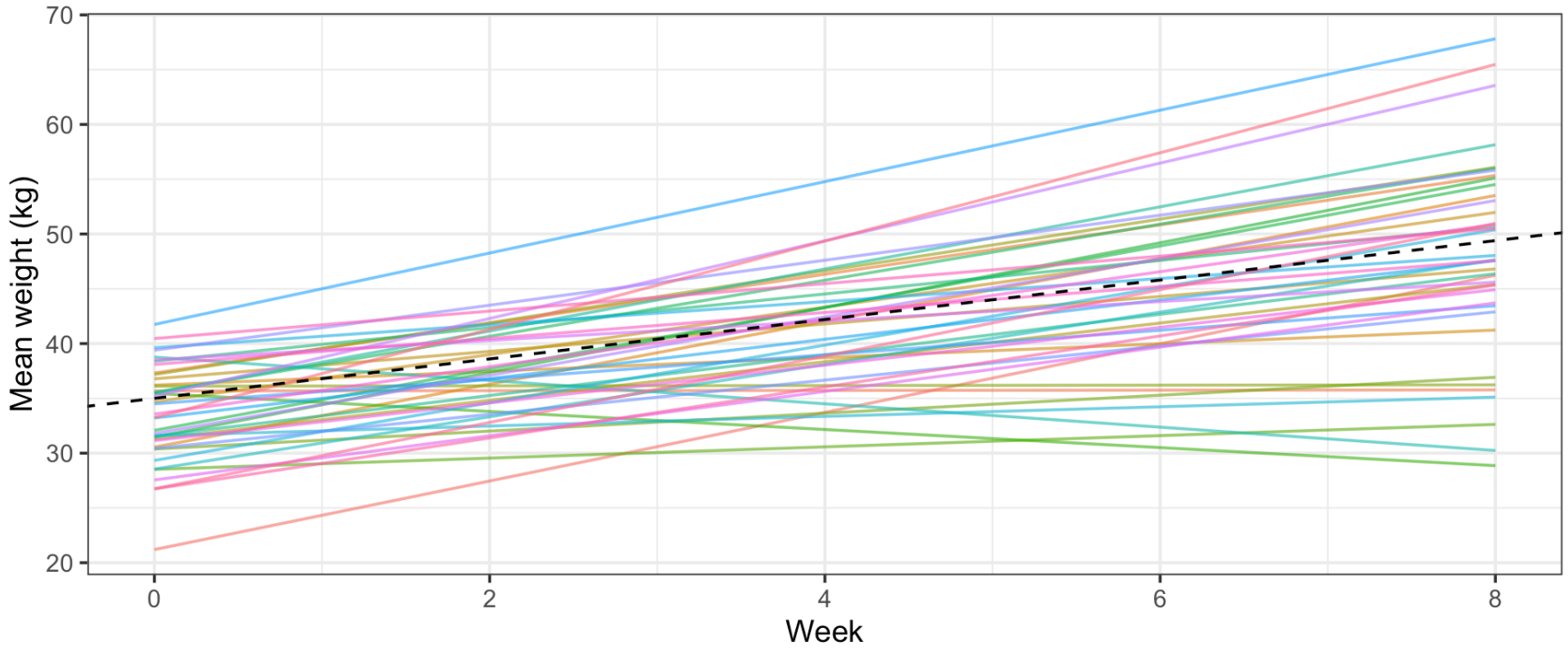


Each coloured line is a calf mean; the dashed line is the assumed overall mean.

RANDOM INTERCEPTS AND SLOPES

`weight_kg ~ week + (1 + week | calf)` also lets weekly gain differ. A **slope** is weight change per week.

```
p_slopes
```



The same starting weights, now with different slopes.

READ `(1 + week | calf)`

R includes an overall intercept automatically: `~week` is shorthand for `~ 1 + week`.

Part	Meaning
`week` outside the brackets	Average change in weight per week.
`1` inside the brackets	A calf's difference in starting weight at week 0.
`week` inside the brackets	A calf's difference from the average weekly gain.
`| calf`	Starting and gain differences belong to the same calf.

This notation also allows starting and gain differences to be related. The two plots show specified mean lines; we are not fitting the calf models today.

MOVE THE ZERO OF THE WEEK VARIABLE

Researchers might want each calf's intercept to describe its weight at **week 4**, halfway through follow-up, rather than at the start of follow-up.

Subtract four from every recorded week. This is called **centring the covariate**: here, the covariate is the recorded week.

```
lines <- lines |>
  mutate(week_c = week - 4)
week_mapping <- lines |>
  distinct(week, week_c)
week_mapping |>
  filter(week %in% c(0, 4, 8))
```

```
# A tibble: 3 × 2
   week week_c
  <int>  <dbl>
1     0     -4
2     4      0
3     8      4
```

No weights or weekly gains have changed. Only the label on the time axis has changed.

EXPRESS THE SAME CALF PATHS AROUND WEEK 4

The overall mean at week 4 is 35 + 1.8 × 4 kg. A calf's difference at that week includes its starting difference and four weeks of its gain difference.

```
mean_at_four <- 35 + 1.8 * 4
lines <- lines |>
  mutate(
    middle_difference = start_difference + 4 * gain_difference,
    centered_mean = mean_at_four + 1.8 * week_c +
      middle_difference + gain_difference * week_c
  )
mean_at_four
```

```
[1] 42.2
```

```
lines |>
  summarise(maximum_difference = max(abs(centered_mean - intercept_and_slope)))
```

```
# A tibble: 1 × 1
  maximum_difference
               <dbl>
1           1.42e-14
```

The tiny maximum difference is computer rounding. We have kept every calf's mean path.

BUILD THE CENTRED VERSION

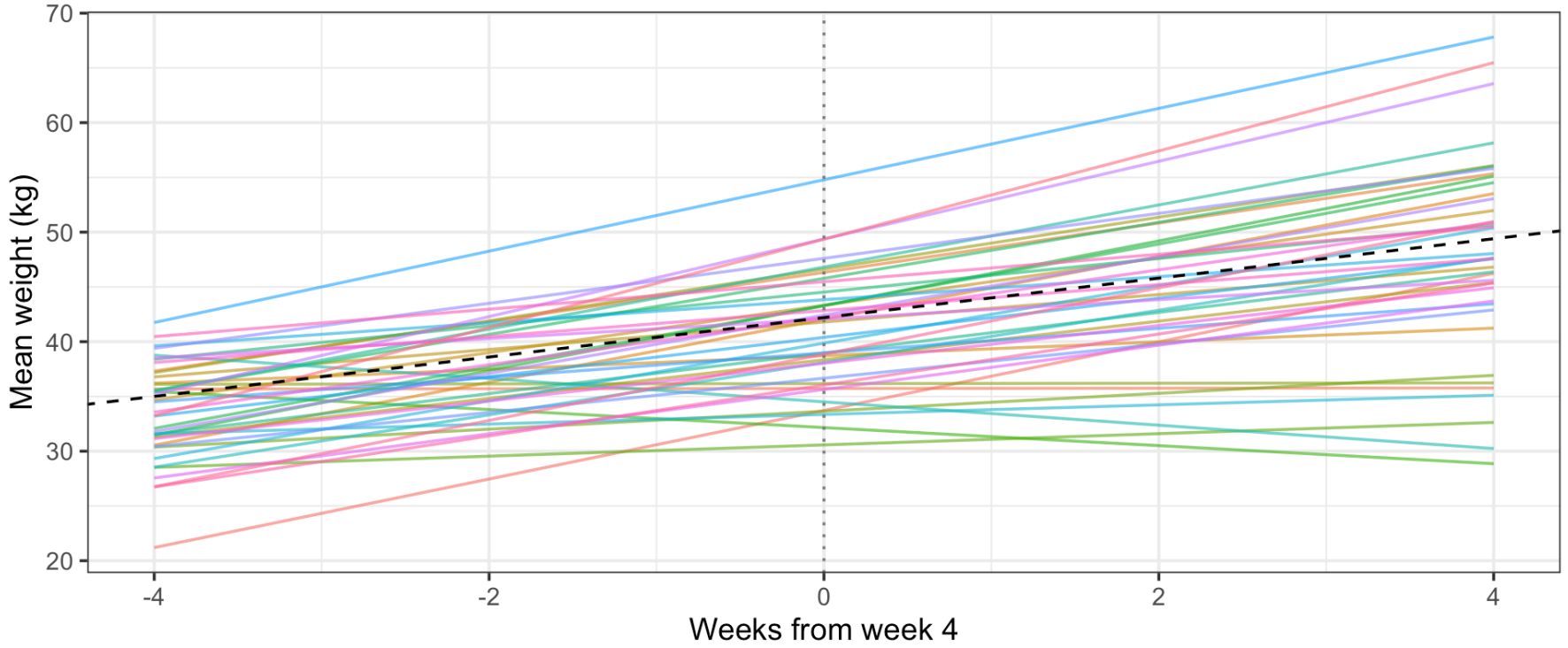
Keep the calf colours and weight scale. The vertical dotted line marks week 4, where the centred week is zero.

```
p_centered <- ggplot(
  lines,
  aes(week_c, centered_mean, group = calf, colour = calf)
) +
  geom_line(alpha = 0.65) +
  geom_abline(intercept = mean_at_four, slope = 1.8, linetype = "dashed") +
  geom_vline(xintercept = 0, colour = "grey50", linetype = "dotted") +
  coord_cartesian(ylim = weight_limits) +
  labs(x = "Weeks from week 4", y = "Mean weight (kg)") +
  theme_bw(base_size = 14) +
  theme(legend.position = "none")
```

INTERCEPTS NOW DESCRIBE WEEK 4

`weight_kg ~ week_c + (1 + week_c | calf)` permits calf differences in weight at week 4 and in weekly gain.

```
p_centered
```



Compare with the previous plot: the paths and slopes are the same.

STARTING POINT CHANGES THE INTERCEPT'S MEANING

At week 0 we drew starting-weight and gain differences separately. By week 4, a faster-growing calf will usually be further above the overall mean.

The week-4 weight differences and weekly-gain differences can therefore be related. Centring does not guarantee that they become unrelated.

The formula allows these two calf differences to be related. We have changed where we read the intercept, while keeping the same mean paths.

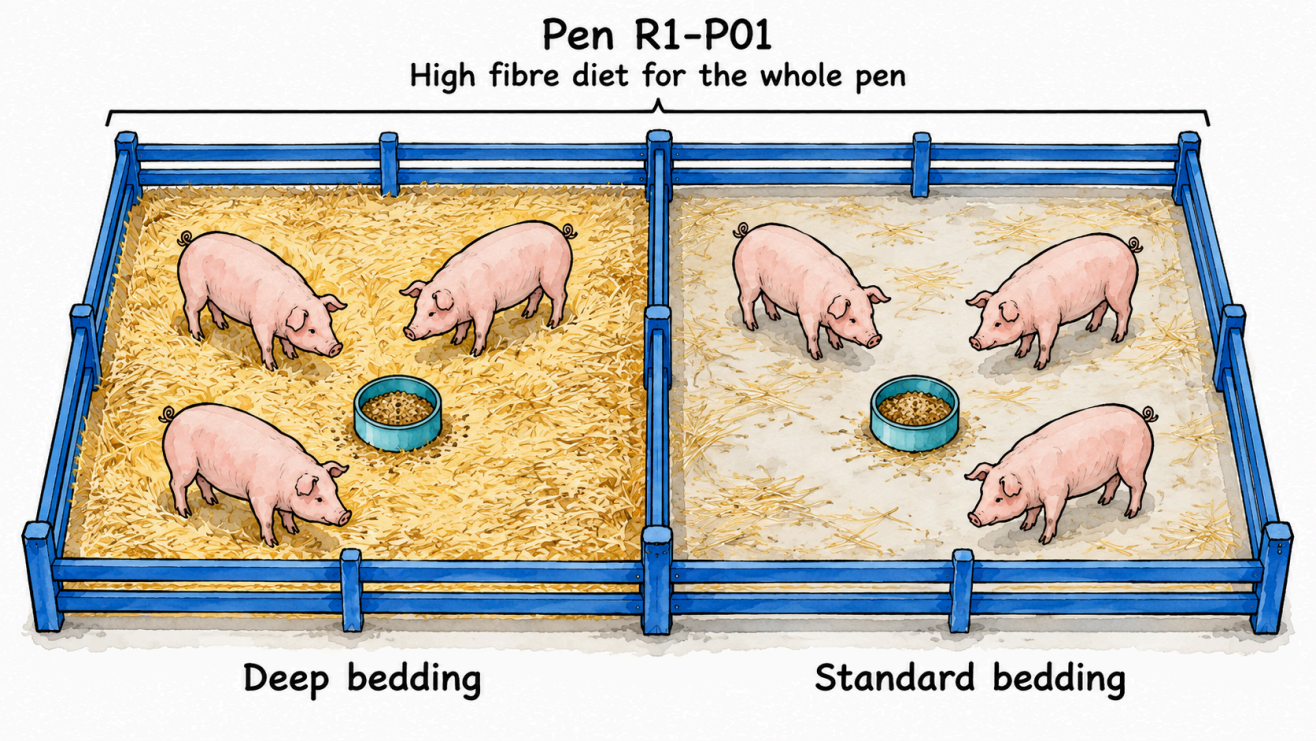
CHOOSE THE TERM FROM THE STUDY

To estimate different weekly gains for calves, we need repeated weights at different weeks within each calf.

Each pen in our diet study receives just one diet. It cannot show a within-pen change from Control to High fibre.

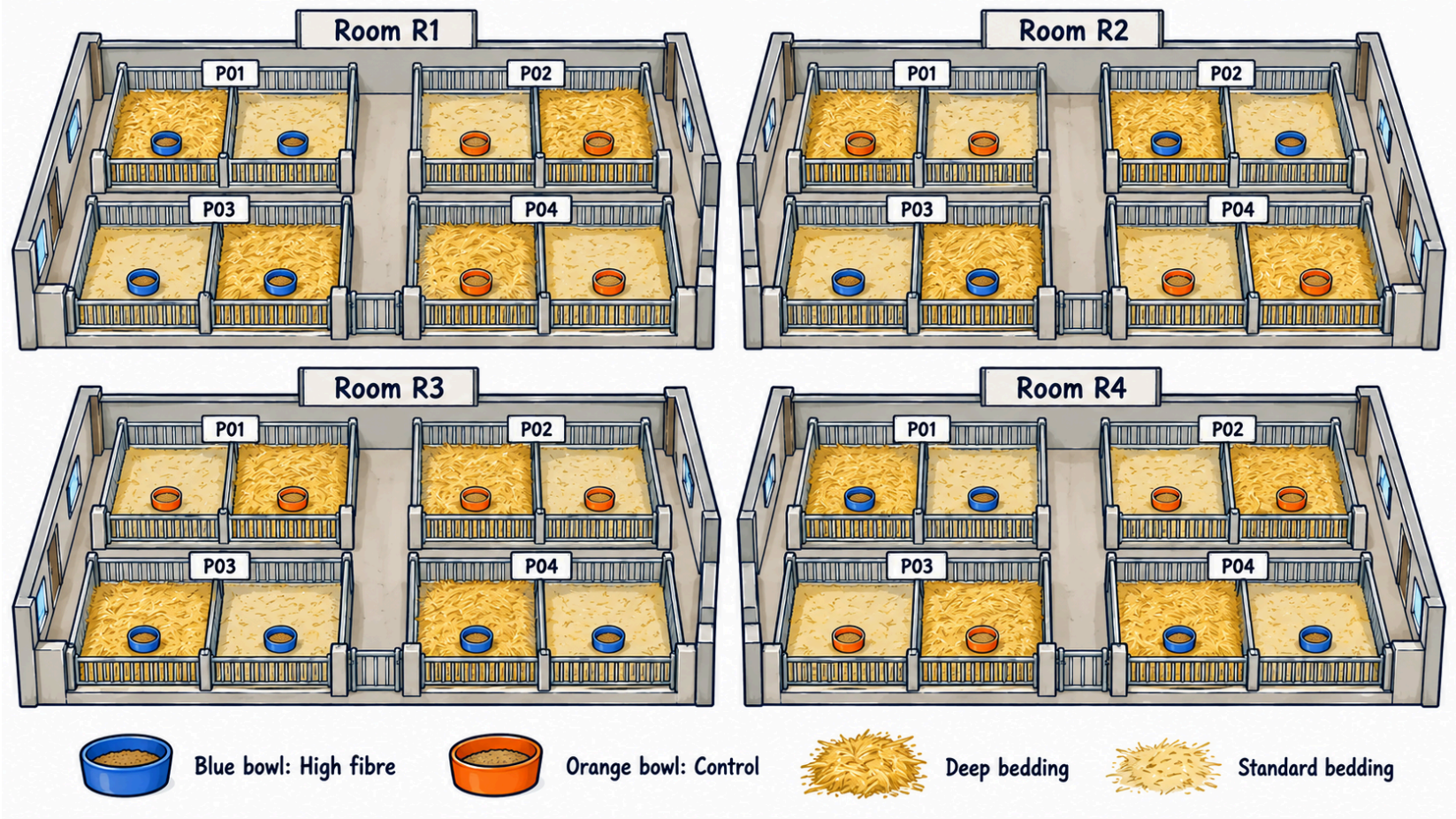
Explain why `(1 | pen)` describes shared pen conditions, while `(1 + diet | pen)` would ask for pen-specific diet changes the design cannot supply.

A NEW EXPERIMENT: TWO ASSIGNMENTS



Researchers assign diet to the whole pen, then randomly choose which section receives deep bedding. The other receives standard bedding. This is a **split-plot design**.

FOUR ROOMS, SIXTEEN PENS



Eight pens receive each diet. Record one average gain from each fenced section: **32 section records** altogether.

SEE THE TWO ASSIGNMENTS

The new file has **one row per section**, rather than one row per animal. `section` means Left or Right.

```
s <- read_csv("split-plot.csv", show_col_types = FALSE) |>
  mutate(
    room = factor(room),
    pen = factor(pen),
    diet = factor(diet, levels = c("Control", "High fibre")),
    bedding = factor(bedding, levels = c("Standard", "Deep"))
  )
s |>
  select(pen, section, diet, bedding) |>
  slice_head(n = 4)
```

```
# A tibble: 4 × 4
  pen    section diet       bedding
  <fct>  <chr>   <fct>      <fct>
1 R1-P01 Left    High fibre Deep
2 R1-P01 Right   High fibre Standard
3 R1-P02 Left    Control    Standard
4 R1-P02 Right   Control    Deep
```

WHICH COMPARISON USES WHICH UNITS?

For this split-plot study, explain each row before opening the answer.

Comparison	What was assigned?
Diet	Eight whole pens per diet
Bedding	One of each bedding within every pen
Does bedding change the diet difference?	Both assignments together

Would treating 32 sections as 32 independent diet assignments describe this experiment?

FIT BOTH TREATMENT QUESTIONS

`diet * bedding` allows the diet difference to change with bedding. This changing difference is called an **interaction**.

```
split_fit <- lme4::lmer(
  gain_g_day ~ room + diet * bedding + (1 | pen),
  data = s,
  REML = TRUE
)
summary(split_fit)
```

The remaining variation is between sections after allowing for their shared pen.

FIT BOTH TREATMENT QUESTIONS

```
Linear mixed model fit by REML ['lmerMod']
Formula: gain_g_day ~ room + diet * bedding + (1 | pen)
   Data: s

REML criterion at convergence: 250

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.3498 -0.4822 -0.0602  0.5286  1.3209

Random effects:
 Groups   Name        Variance Std.Dev.
 pen      (Intercept) 3567     59.7
 Residual              127     11.3
Number of obs: 32, groups:  pen, 16

Fixed effects:
                           Estimate Std. Error t value
(Intercept)                  716.56      33.80   21.20
roomR2                         2.62      42.60    0.06
roomR3                        42.94      42.60    1.01
roomR4                        76.40      42.60    1.79
dietHigh fibre                 9.97      30.39    0.33
beddingDeep                   16.42       5.63    2.92
dietHigh fibre:beddingDeep     8.75       7.96    1.10

Correlation of Fixed Effects:
            (Intr) roomR2 roomR3 roomR4 dtHghf bddngD
roomR2      -0.630
roomR3      -0.630  0.500
roomR4      -0.630  0.500  0.500
dietHighfbr -0.450  0.000  0.000  0.000
beddingDeep -0.083  0.000  0.000  0.000  0.093
dtHfbr:bddD  0.059  0.000  0.000  0.000 -0.131 -0.707
```

DEFINE THE COMPARISON POPULATION

We want each of the four rooms to count equally within each bedding group.

- By default, `avg_comparisons()` averages over the observed records. Rooms with more records receive more weight.
- `newdata = "balanced"` creates every combination of the model's categories and gives those combinations equal weight.
- `by = "bedding"` reports a separate diet difference for Standard and Deep bedding.
- `re.form = NA` leaves out individual fitted pen differences.

ASK FOR USEFUL DIFFERENCES

For each bedding, how much does average gain change from Control to High fibre?

```
avg_comparisons(
  split_fit,
  variables = "diet",
  newdata = "balanced",
  by = "bedding",
  re.form = NA
)
```

```
  bedding Estimate Std. Error     z Pr(>|z|)   S 2.5 % 97.5 %
 Standard     9.97       30.4 0.328    0.743 0.4 -49.6   69.5
 Deep        18.72       30.4 0.616    0.538 0.9 -40.8   78.3

Term: diet
Type: response
Comparison: High fibre - Control
```

All values are grams/day. For this Gaussian model, averaging over the assumed zero-mean pen differences gives the same mean.

FROM PREDICTIONS TO DIET DIFFERENCES

For this study, `avg_comparisons()`:

- Predicts mean gain on Control and High fibre for each room and bedding.
- Sets the fitted difference for each particular pen to zero (`re.form = NA`).
- Subtracts Control from High fibre within the same room and bedding.
- Uses the balanced grid to average the four room differences separately for each bedding, giving every room equal weight.
- Uses the model's uncertainty to calculate a standard error and approximate **95% confidence interval**.

AVERAGE OVER THE OBSERVED SECTIONS

For every recorded section, predict gain twice: once with diet set to Control, then with diet set to High fibre. Keep its room and bedding unchanged.

```
avg_comparisons(
  split_fit,
  variables = "diet",
  by = "bedding",
  re.form = NA
)
```

```
  bedding Estimate Std. Error     z Pr(>|z|)   S 2.5 % 97.5 %
 Standard     9.97       30.4 0.328    0.743 0.4 -49.6   69.5
 Deep        18.72       30.4 0.616    0.538 0.9 -40.8   78.3

Term: diet
Type: response
Comparison: High fibre - Control
```

Subtract Control from High fibre for each row, then average within each bedding. Every section record receives equal weight.

No sections are missing: every room contributes four records per bedding. Both methods therefore give each room equal weight and produce the same results.