ALLOCATION WORKSHOP

BUILD, AUDIT, RANDOMISE WITHIN ROOMS

Gavin Simpson

gavin@anivet.au.dk

Aarhus University

Wednesday, 21 October 2026

BUILD AN ALLOCATION

13:55–14:35

THE CASE

A feeding experiment has:

- 24 pig pens in North, Central and South rooms
- eight pen positions within each room
- four diet treatments
- six independently treated pens per diet

> Create a balanced, reproducible, completely randomised allocation of **pens** to diets.

REQUIRED OUTPUTS

By 14:35, retain:

1. the seed and exact R code
2. an allocation table with one row per pen
3. a table confirming six pens per diet
4. one sentence explaining why the method is random and balanced

Compare results using group seeds **101, 202, 303 and 404** after your seed-123 allocation works.

START HERE

```r
library("tibble")
library("dplyr")
library("ggplot2")

pens <- tibble(
  pen_id = sprintf("P%02d", 1:24),
  room = rep(c("North", "Central", "South"), each = 8),
  position = rep(1:8, times = 3)
)

treatments <- rep(
  c("Control", "Diet A", "Diet B", "Diet C"),
  each = 6
)
```

WORK TIME

25 minutes

- Set seed 123
- Shuffle the treatment labels
- Attach the shuffled labels to the pen table
- Count assignments by diet
- Re-run from the seed and compare
- Change only the seed; compare within your group

BUILD CHECK

```
set.seed(123)

allocation <- pens |>
  mutate(diet = sample(treatments))

allocation |>
  count(diet)
```

```
# A tibble: 4 × 2
  diet        n
  <chr>   <int>
1 Control     6
2 Diet A      6
3 Diet B      6
4 Diet C      6
```

> Each diet has six pens. Re-running from seed 123 recreates the same ordered allocation.

AUDIT THE ALLOCATION

14:35–14:55

AUDIT QUESTIONS

For seed 123, check:

- Are all 24 pen IDs present exactly once?
- Does every diet have six pens?
- How are diets distributed among rooms?
- Does the layout contain a predictable repeating treatment rule?
- Can another person reproduce the table?

USEFUL CHECKS

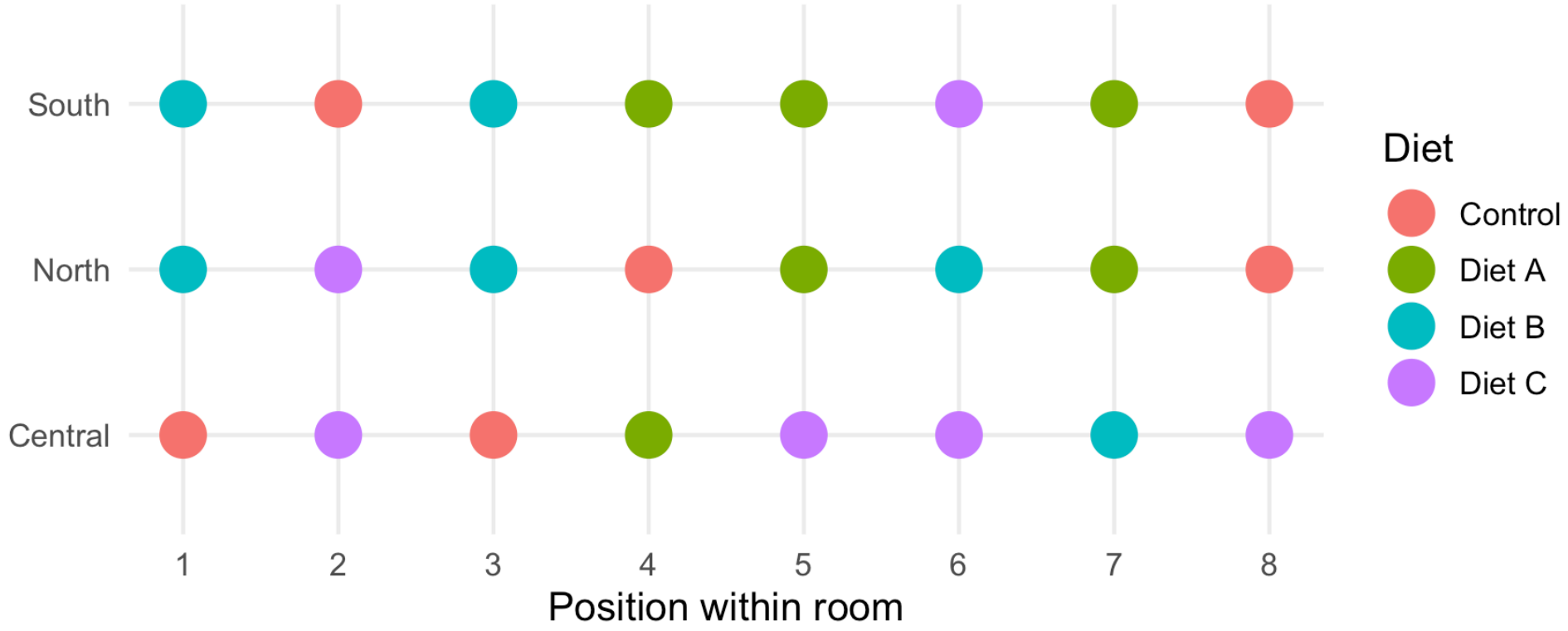
```
allocation |> summarise(
  rows = n(),
  distinct_pens = n_distinct(pen_id)
)
```

```
# A tibble: 1 × 2
   rows distinct_pens
  <int>         <int>
1    24            24
```

```
allocation |> count(room, diet)
```

```
# A tibble: 12 × 3
   room    diet        n
   <chr>   <chr>   <int>
 1 Central Control     2
 2 Central Diet A      1
 3 Central Diet B      1
 4 Central Diet C      4
 5 North   Control     2
 6 North   Diet A      2
 7 North   Diet B      3
 8 North   Diet C      1
 9 South   Control     2
10 South   Diet A      3
11 South   Diet B      2
12 South   Diet C      1
```

PLOT THE ROOMS



STOP AND COMPARE

With a partner using a different assigned seed:

1. Which features are identical by design?
2. Which assignments differ because of chance?
3. Does either result show room imbalance?
4. Would you reject an allocation merely because it looks uneven?

SEED 123 AUDIT

Room	Control	Diet A	Diet B	Diet C
North	2	2	3	1
Central	2	1	1	4
South	2	3	2	1

> The allocation is random, reproducible and balanced overall. It is not balanced within rooms.

RANDOMISE WITHIN ROOMS

15:10–16:00

A JUSTIFIED RESTRICTION

Suppose room conditions may affect the response.

Require each room to contain:

- two Control pens
- two Diet A pens
- two Diet B pens
- two Diet C pens

Within each room, chance still decides which pen positions receive those labels.

RANDOMISE WITHIN ROOMS

```
set.seed(123)

allocation_blocked <- pens |>
  group_by(room) |>
  mutate(
    diet = sample(
      rep(c("Control", "Diet A", "Diet B", "Diet C"), each = 2)
    )
  ) |>
  ungroup()
```

CHECK THE RESTRICTION

```
allocation_blocked |>
  count(room, diet)
```

```
# A tibble: 12 × 3
   room    diet        n
   <chr>   <chr>   <int>
 1 Central Control     2
 2 Central Diet A      2
 3 Central Diet B      2
 4 Central Diet C      2
 5 North   Control     2
 6 North   Diet A      2
 7 North   Diet B      2
 8 North   Diet C      2
 9 South   Control     2
10 South   Diet A      2
11 South   Diet B      2
12 South   Diet C      2
```

Every room must have two pens per diet. Then visualise the layout and inspect pen uniqueness.

> A repeating Control–A–B–C pattern would satisfy the counts, but would not be random.

GROUP CHALLENGE

30 minutes

For the assigned case in your group's four-card set:

1. identify the experimental unit and block
2. construct the complete unit table
3. randomise treatments within blocks
4. check every stated constraint
5. make a layout plot
6. prepare a 60-second explanation and justification

EXPLAIN AND JUSTIFY

Use your 60 seconds to explain and justify the allocation:

- Why is this the correct experimental unit?
- Why is the grouping variable a plausible source of variation?
- Which feature is fixed by the restriction?
- Which part remains random?
- What records would allow an audit?

DEBRIEF

Compare the cases completed by the four groups.

- A **block** is a group of similar experimental units
- Treatments are randomised among units within each block
- Each block in today's examples contains all treatment levels
- This is a **randomised complete block design**

Week 4 will construct blocked and factorial designs with Lego.

EXIT QUESTION

Answer individually in two or three sentences:

> A researcher assigns equal numbers of cages to treatments using the fixed repeating pattern A, B, A, B. Is the allocation random, balanced and defensible? What single change would improve it?

SOURCES

Ruxton, Graeme D., and Nick Colegrave. 2016. *Experimental Design for the Life Sciences*. 4th ed. Oxford: Oxford University Press. Chapter 7, sections 7.1–7.4, pp. 101–108. ISBN 978-0-19-871735-5. Publisher page.

R Core Team. 2026. *R: A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing. R project.

Wickham, Hadley, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. doi:10.21105/joss.01686.