SIMULATE BEFORE ANALYSING

WEEK 5: THE ESSENTIALS FOR TODAY'S ACTIVITY

Gavin Simpson

gavin@anivet.au.dk

Aarhus University

Wednesday, 21 October 2026

TODAY'S CLASS

Time	Work
12:30–12:40	Opening recall of responses and experimental units
12:40–13:20	Essential GLMs and R randomisation recap
13:20–13:40	Workshop: design and assumptions
13:40–13:55	Break
13:55–14:55	Generate, inspect and repeat responses
14:55–15:10	Break
15:10–16:00	Fit, audit, discuss and answer the exit question

OPENING RECALL

With your group of three, discuss one response:

Group	One observation from each pen
1	Mean daily weight gain, g/day
2	Any pig required treatment during the study, yes/no
3	Number of veterinary treatment events during the study
4	Feed conversion ratio, kg feed/kg gain

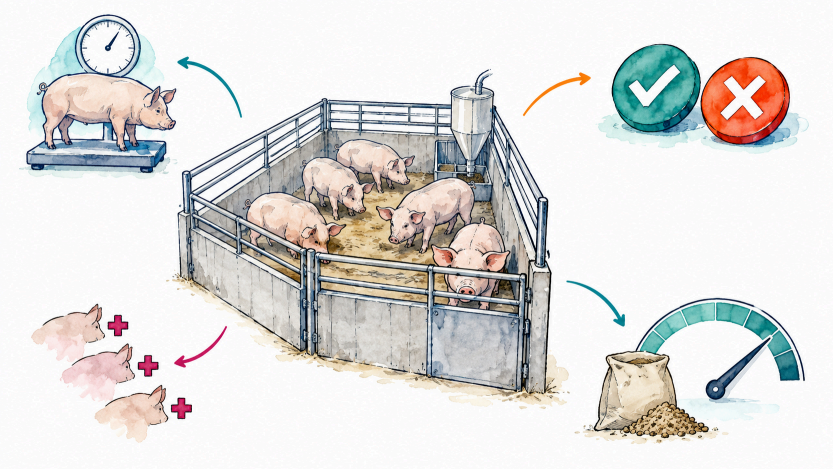
Discuss what one observation could equal. Could it be zero, negative, a fraction, or a count? Name the experimental unit.

POSSIBLE OBSERVATIONS

A model starts with the response definition.

A binary observation is 0 or 1. A count is a non-negative integer. A feed conversion ratio is positive in this scenario, which assumes positive gain.

Mean daily gain is continuous and can be negative if animals lose mass.



OUR PROPOSED EXPERIMENT

Sixteen pens occupy four rooms, with four pens in each room. Each pen contains six pigs and has a shared feeder. We compare control and high-fibre diets over the same fixed observation period.

Each room receives two pens on each diet. The **pen is the experimental unit**, and room is the block.

We will generate one response per pen. The numerical assumptions are teaching scenarios, not evidence about diet effects.

WHAT THE GLM SUPPLIES

A GLM gives us an expected response for each pen and a distribution from which to draw its observation.

1. Combine diet and room effects to calculate a score, η.
2. Convert the score into a valid expected response, μ.
3. Draw a random observation from the chosen family.

The conversion in step 2 is the **inverse link**. The supplied group recipe tells you which conversion and random generator to use.

EXPECTED RESPONSE AND OBSERVATION

For a binary pen response, an expected value of **0.30** means a 30% probability that at least one pig needs treatment.

The observation is still **0 or 1**. Two otherwise similar pens can have different outcomes.

In R, `plogis(eta)` produces the probability. `rbinom(..., size = 1, prob = mu)` draws the observation.

> Which quantity belongs between 0 and 1? Which can equal only 0 or 1?

THE GROUP RECIPES

Group	Family	Expected response	Random observation
1	Gaussian	`mu = eta`	`rnorm(n(), mu, sd = 60)`
2	Binomial	`mu = plogis(eta)`	`rbinom(n(), 1, mu)`
3	Poisson	`mu = exp(eta)`	`rpois(n(), mu)`
4	Gamma	`mu = exp(eta)`	`rgamma(n(), 25, scale = mu/25)`

The family supplies the response values. The inverse link gives an appropriate range for the mean. Each group gets its complete recipe in the handout.

A WORKED PEN

For the count group in Room 1:

$$\eta = \log(3) + \log(0.65) \times \text{high-fibre indicator}$$

The control mean is 3 events. The high-fibre mean is $3 \times 0.65 = 1.95$ events.

A pen cannot have 1.95 events. `rpois(1, lambda = 1.95)` draws an integer. Room effects modify the score before we calculate the mean.

PACKAGE CHOICE

We use **tidyverse** because students already know its table verbs from randomisation. It lets us inspect design columns and model calculations with the same syntax. Its `ggplot2` component supplies the workshop plot.

The `stats` package supplied with R contains the random generators and `glm()`.

```
library("tidyverse")
```

ONE ROW PER PEN

```
pens <- tibble(
  pen_id = sprintf("P%02d", 1:16),
  room = rep(paste0("Room ", 1:4), each = 4),
  position = rep(1:4, times = 4)
)
pens
```

`rep(..., each = 4)` gives four rows per room. `position` cycles through the four pens in each room.

The response column will come later. Treatment allocation exists before any observations.

BALANCED TREATMENT LABELS

```
labels <- rep(c("Control", "High fibre"), each = 2)
labels
```

There are two labels of each kind. We shuffle all four **without replacement within each room**.

`rep()` creates the required labels. `sample()` randomises their order. Sampling each pen independently would not guarantee this balance.

BLOCKED RANDOMISATION

```
set.seed(505)
design <- pens |>
  group_by(room) |>
  mutate(diet = sample(labels)) |>
  ungroup() |>
  mutate(diet = factor(diet,
                       levels = c("Control", "High fibre")))
```

Grouping repeats the shuffle separately for each room. The seed reproduces the allocation. The factor order makes control the reference diet.

ALLOCATION CHECKS

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

```
# A tibble: 8 × 3
  room   diet           n
  <chr>  <fct>      <int>
1 Room 1 Control        2
2 Room 1 High fibre     2
3 Room 2 Control        2
4 Room 2 High fibre     2
5 Room 3 Control        2
6 Room 3 High fibre     2
7 Room 4 Control        2
8 Room 4 High fibre     2
```

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

```
# A tibble: 1 × 2
   rows distinct_pens
  <int>         <int>
1    16            16
```

We need two pens per diet in every room, 16 rows, and 16 unique pen IDs.

THE MODEL SCORE IN R

```
simulated <- design |>
  mutate(
    room_effect = c(0, 0.20, 0.40, 0.30)[
      match(room, paste0("Room ", 1:4))],
    eta = log(3) + log(0.65) * (diet == "High fibre") +
      room_effect,
    mu = exp(eta)
  )
```

The comparison `(diet == "High fibre")` acts as 0 or 1. The room lookup supplies one fixed effect per room. The expected response is now visible in `mu`.

A RANDOM RESPONSE

```
set.seed(1503)
simulated <- simulated |>
  mutate(response = rpois(n(), lambda = mu))
simulated |> select(pen_id, diet, eta, mu, response)
```

`n()` supplies the number of rows. Every pen receives one random response.

A new response seed leaves the design, score and mean unchanged. It produces another possible set of observations.

READY FOR THE ACTIVITY

Explain to your neighbour:

- why this study has 16 experimental units
- how the code guarantees balance within rooms
- why `mu` and `response` can have different values and different ranges

You will use one supplied model recipe, inspect the resulting data, repeat the response draw, and fit a supplied GLM.

SOURCES

- R Core Team. R statistical documentation: families, random generators and GLMs.
- Wickham et al. (2019). Welcome to the Tidyverse.
- McCullagh and Nelder (1989). *Generalized Linear Models*, 2nd ed. Chapters 1–2.