POWER WORKSHOP

WEEK 6: CURVES, ASSUMPTIONS AND DESIGN RECOMMENDATIONS

Gavin Simpson

Aarhus University

Wednesday, 7 October 2026

WORKSHOP ROUTE

Time	Task
13:55–14:15	Repetition and one sample size
14:15–14:40	Build or open the baseline curve
14:40–14:55	Interpret the curve and null reference
14:55–15:10	Break
15:10–15:25	Reports and assumptions
15:25–15:40	Recommendation and sensitivity prediction
15:40–15:55	Group project briefing
15:55–16:00	Individual exit

ROTATE ROLES

Driver: run the code.

Model translator: explain what is being simulated and tested.

Design auditor: check units, assumptions and unresolved results.

Rotate now, and again at 15:10.



LOADING THE WORKSHOP TOOLS

Run from your `06-power-analysis` project folder. The demonstration uses Group 3; your group script supplies your own scenario and seeds.

```
source("power-helpers.R")
s <- scenario(3)
s$label
```

```
[1] "Poisson events"
```

`source()` defines our helpers and loads dplyr, tibble, ggplot2 and readr. `s` holds the assumed data generating process.

TWENTY REPEATED EXPERIMENTS

`run_simulations()` repeats generation, fitting and the test. `small` has one row per attempted experiment.

```
small <- run_simulations(s, total_pens = 16,
                         B = 20, seed = 6203)
```

```
small |> slice(1:10) |>
  select(iteration, detected) |>
  print(n = 10)
```

```
# A tibble: 10 × 2
   iteration detected
       <int> <lgl>
 1         1 TRUE
 2         2 TRUE
 3         3 FALSE
 4         4 FALSE
 5         5 FALSE
 6         6 FALSE
 7         7 TRUE
 8         8 FALSE
 9         9 TRUE
10        10 FALSE
```

```
small |> slice(11:20) |>
  select(iteration, detected) |>
  print(n = 10)
```

```
# A tibble: 10 × 2
   iteration detected
       <int> <lgl>
 1        11 FALSE
 2        12 TRUE
 3        13 FALSE
 4        14 FALSE
 5        15 FALSE
 6        16 TRUE
 7        17 FALSE
 8        18 FALSE
 9        19 TRUE
10        20 FALSE
```

Count TRUE across **all 20 attempts** before advancing.

FROM DECISIONS TO A PROPORTION

First count detections directly. Then use `summarise_runs()` to count unresolved analyses and calculate a Monte Carlo interval.

```
sum(small$detected) / nrow(small)
```

```
[1] 0.35
```

```
small_summary <- summarise_runs(small)
small_summary |>
  select(attempted, detections, unresolved, detection, lower, upper)
```

```
# A tibble: 1 × 6
  attempted detections unresolved detection lower upper
      <int>      <int>      <int>     <dbl> <dbl> <dbl>
1        20          7          0      0.35 0.181 0.567
```

Twenty repetitions give a rough illustration, not a stable recommendation.

ESTIMATE AT 16 PENS

Create `one_size` using the same design and assumptions, with **200** repetitions.

```
one_size <- run_simulations(s, total_pens = 16,
                            B = 200, seed = 6303)
one_summary <- summarise_runs(one_size)
one_summary |>
  select(attempted, detections, unresolved, detection, lower, upper)
```

```
# A tibble: 1 × 6
  attempted detections unresolved detection lower upper
      <int>      <int>      <int>     <dbl> <dbl> <dbl>
1       200         61          2     0.305 0.245 0.372
```

Compare with 20 repetitions. What became more precise? What stayed the same?

READING THE SUMMARY

Output	Meaning
`attempted`	All simulated experiments
`detections`	Results passing numerical screens with $p < 0.05$
`unresolved`	Attempts not suitable for the declared test decision
`detection`	Detections / all attempts
`lower`, `upper`	95% Monte Carlo interval

An unresolved result stays in the denominator.

FOUR CANDIDATE DESIGNS

Total pens	Pens per room	Pens per diet
16	4	8
32	8	16
48	12	24
64	16	32

Four rooms and six pigs per pen remain fixed. More room capacity must be justified before adopting a larger design.

BUILD YOUR CURVE

`run_curve()` repeats the calculation at each requested size. Combine `other_sizes` with the existing 16-pen `one_size` results.

```
other_sizes <- run_curve(s, sizes = c(32, 48, 64),
                         B = 200, seed = 6403)
baseline_runs <- bind_rows(one_size, other_sizes)
baseline_summary <- summarise_runs(baseline_runs)
baseline_summary |>
  select(total_pens, detection, unresolved_rate, lower, upper)
```

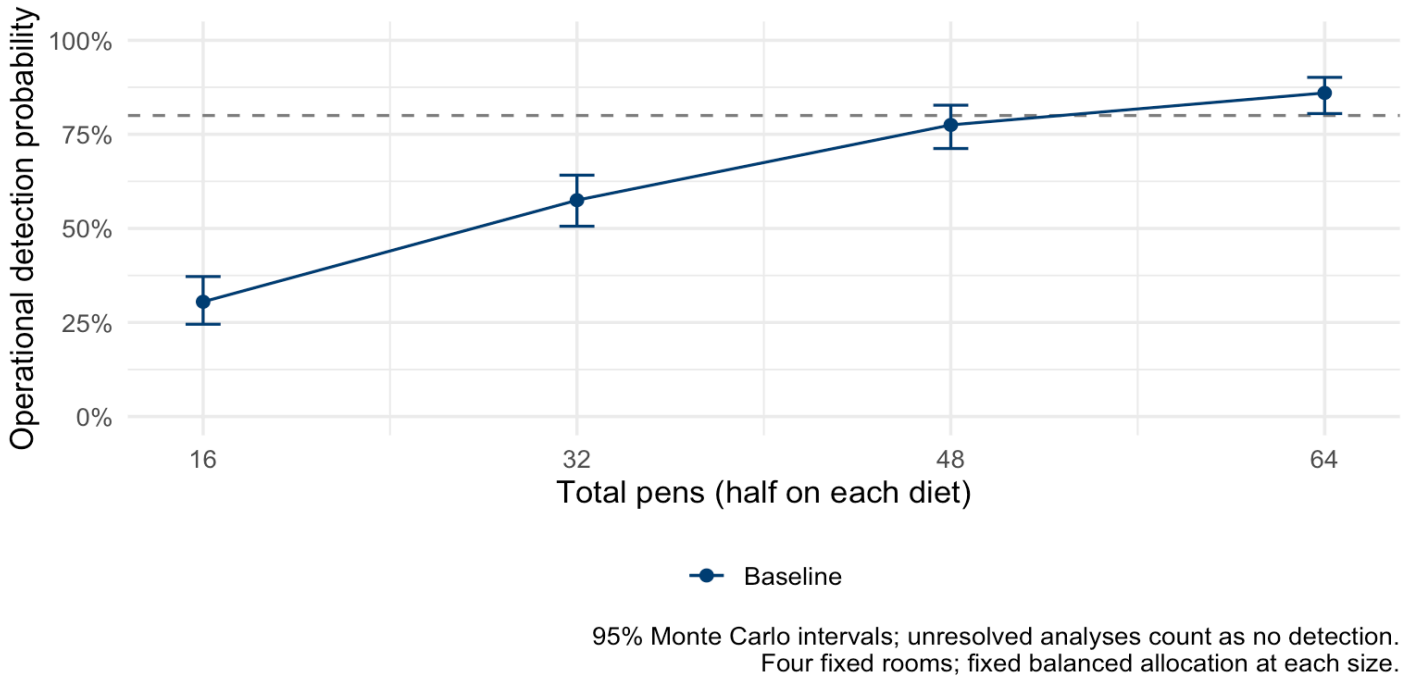
```
# A tibble: 4 × 5
  total_pens detection unresolved_rate lower upper
       <dbl>     <dbl>           <dbl> <dbl> <dbl>
1         16     0.305           0.01  0.245 0.372
2         32     0.575           0.005 0.506 0.641
3         48     0.775           0.005 0.712 0.827
4         64     0.86            0     0.805 0.901
```

These proportions and intervals describe the baseline assumptions.

PLOT THE BASELINE RESULTS

`plot_curve()` plots `detection` with its Monte Carlo interval against `total_pens`.

```
plot_curve(baseline_summary)
```



Read the unresolved rates on the preceding slide alongside this curve.

INTERPRET BEFORE RECOMMENDING

1. What does a point on the curve estimate?
2. Why might adjacent exploratory points be irregular?
3. What uncertainty do the error bars include?
4. Which tested sizes support an 80% target?
5. What if no size reaches it?

Do not invent a minimum sample size between four simulated points.

LOAD THE REFERENCE RESULTS

`reference/power-summary.csv` contains the supplied **2,000-repetition** results. Read it into `reference`, then select Group 3's baseline, nonzero-effect rows.

```
reference <- read_csv("reference/power-summary.csv",
                      show_col_types = FALSE)
reference |>
  filter(group == s$group, scenario == "Baseline", !null) |>
  select(total_pens, attempted, detection, unresolved_rate)
```

```
# A tibble: 4 × 4
  total_pens attempted detection unresolved_rate
       <dbl>     <dbl>     <dbl>           <dbl>
1         16      2000     0.311          0.0065
2         32      2000     0.574          0
3         48      2000     0.738          0.0005
4         64      2000     0.856          0
```

Compare with your 200-repetition results. Use these supplied results if live computation is slow.

CHECK THE NO-EFFECT SCENARIO

The rows with `null == TRUE` set the generating diet effect to zero, keeping baseline, room effects and variation.

```
reference |>
  filter(group == s$group, scenario == "Baseline", null) |>
  select(total_pens, detection, unresolved_rate, lower, upper)
```

```
# A tibble: 4 × 5
  total_pens detection unresolved_rate  lower  upper
       <dbl>     <dbl>           <dbl>  <dbl>  <dbl>
1         16    0.0485          0.018  0.0399 0.0588
2         32    0.048           0.011  0.0395 0.0583
3         48    0.0515          0.009  0.0426 0.0621
4         64    0.0465          0.0085 0.0381 0.0566
```

Here `detection` estimates **false-positive frequency**, not power under a nonzero effect. Compare with the nominal 5% rate.

TWO-MINUTE REPORTS

Each group reports:

1. the response and effect scale
2. the assumed effect and one uncertain assumption
3. one curve finding, with its limitation
4. one credible design or measurement improvement

These outcomes answer different scientific questions. A higher curve does not automatically make an outcome a better choice.

YOUR CONDITIONAL RECOMMENDATION

"Under [generator and parameters], with [design and analysis], the estimated probability of [testing event] at [total pens; pens per diet] is [estimate and interval].

This uses [repetitions] with [unresolved proportion]. We recommend [design or further investigation], conditional on [feasibility and uncertain assumptions]."

Write this after the reports. Save the baseline results with script section 6.

PREDICT A CHANGE TO YOUR POWER CURVE

Imagine repeating the power calculation at the same sample sizes, after making **your group's change** to the data generating process:

Group	Change from the baseline assumptions
Gaussian	Increase SD from 60 to 90 g/day
Binary	Reduce Room 1 control probability from 0.30 to 0.15
Poisson	Reduce Room 1 control mean from 3 to 1.5 events
Gamma	Reduce shape from 25 to 9; keep means fixed

Would you expect your curve to move up, move down, or stay similar? Why?

Predict now; do not run another curve in class.

WHAT STAYS FIXED IN THIS COMPARISON?

For your group's change, retain:

- the diet-effect coefficient and room contrasts
- the sample sizes, four-room design and balanced diet allocation
- the analysis, numerical checks and detection rule

Explain how the changed assumption affects the observations and the chance of detecting the diet effect. A fixed ratio need not mean a fixed difference on the response scale.

Save your prediction. During autumn-break revision, compare the supplied Baseline and Sensitivity results for your group at the **same total pens**. No new simulation is required.

AFTER CLASS AND WEEK 7

Complete the week 42 revision and project-scoping task: **at most 2½ hours per student**, including Unit 6 consolidation.

It replaces the separate 60-minute assignment. The 30-minute revision block includes the supplied sensitivity comparison. No new model fitting.

Week 7 meets **21 October**. Bring your project brief, model map and one question.

INDIVIDUAL EXIT QUESTION

A group reports that **32 pens give 80% power** and proposes increasing the simulation count to improve the experiment.

What information is missing? What would increasing the simulation count actually improve?

OPTIONAL: COMPUTE THE SENSITIVITY CURVE

After class only: create `s_changed` with the assigned sensitivity assumption. For Group 3, Room 1's control mean changes from 3 to 1.5; the count ratio remains 0.65.

```
s_changed <- scenario(3, sensitivity = TRUE)
changed_runs <- run_curve(s_changed, B = 200, seed = 6503)
comparison <- bind_rows(baseline_summary,
                        summarise_runs(changed_runs))
comparison |>
  select(scenario, total_pens, detection, unresolved_rate)
```

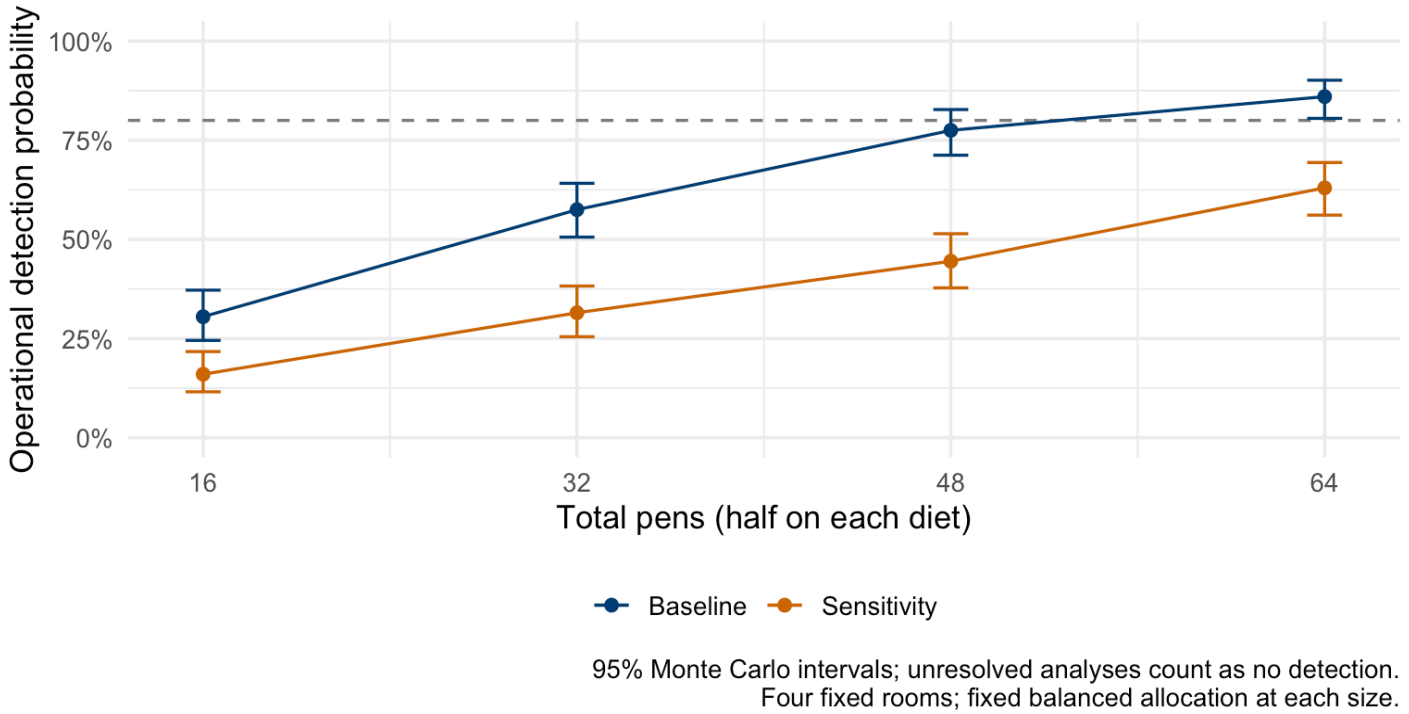
```
# A tibble: 8 × 4
  scenario    total_pens detection unresolved_rate
  <chr>            <dbl>     <dbl>           <dbl>
1 Baseline            16     0.305           0.01
2 Baseline            32     0.575           0.005
3 Baseline            48     0.775           0.005
4 Baseline            64     0.86            0
5 Sensitivity         16     0.16            0.025
6 Sensitivity         32     0.315           0.01
7 Sensitivity         48     0.445           0
8 Sensitivity         64     0.63            0
```

The required follow-up uses supplied results; running this code is optional.

OPTIONAL: COMPARE THE CURVES

The same plotting helper can show both scenarios from `comparison`.

```
plot_curve(comparison)
```



Explain the changed assumption before interpreting the separation between lines.

DETECTION AND PRECISION

A second planning question asks whether the effect can be estimated sufficiently precisely.

For Gaussian gain, consider a 95% interval with **total width at most 60 g/day**. This is an illustrative target, not a biological standard.

An interval can be narrow yet contain zero. It can exclude zero yet be too wide for a useful decision.

GAUSSIAN PRECISION COMPARISON

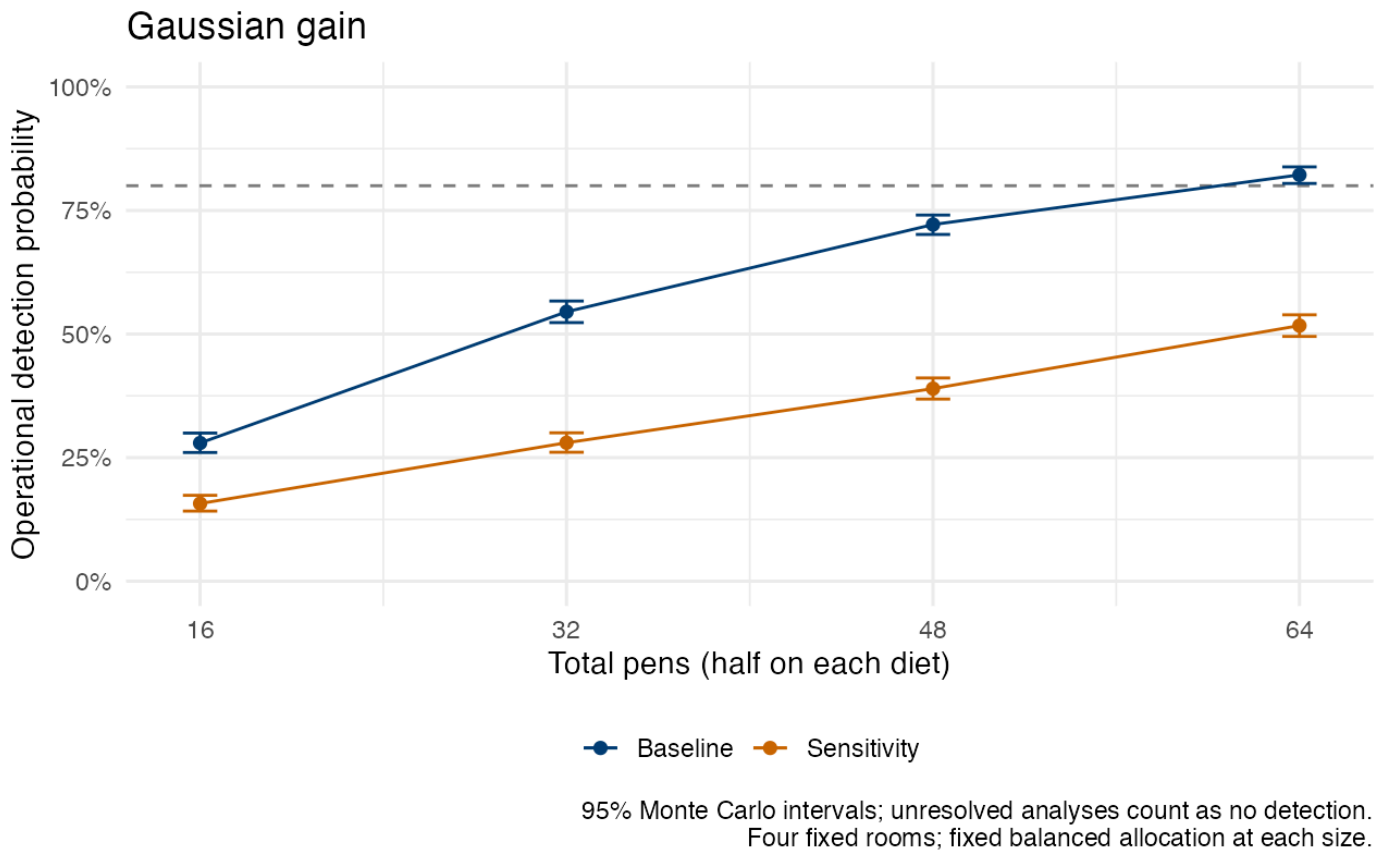
Read the supplied Gaussian results into `precision`. Both columns below are proportions of the same 2,000 attempts.

```
precision <- read_csv("reference/precision-summary.csv",
                      show_col_types = FALSE)
precision |>
  transmute(total_pens, detection = detections / attempted,
            precision_success = probability_precision)
```

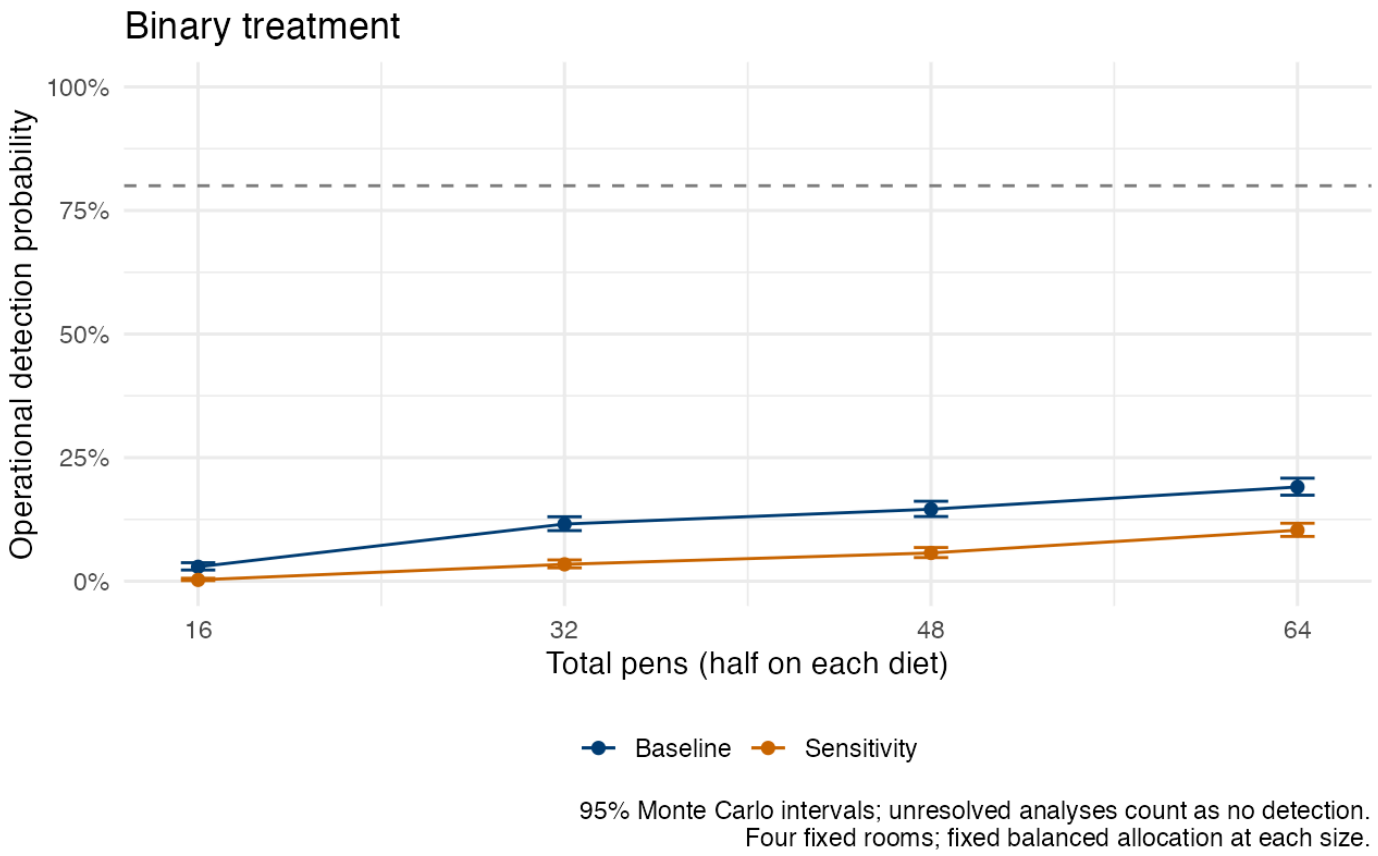
```
# A tibble: 4 × 3
  total_pens detection precision_success
       <dbl>     <dbl>             <dbl>
1         16     0.280            0.0025
2         32     0.545            0.009
3         48     0.722            0.11
4         64     0.822            0.522
```

Precision means a 95% interval at most **60 g/day wide**. The two columns count different events in the same 2,000 simulations.

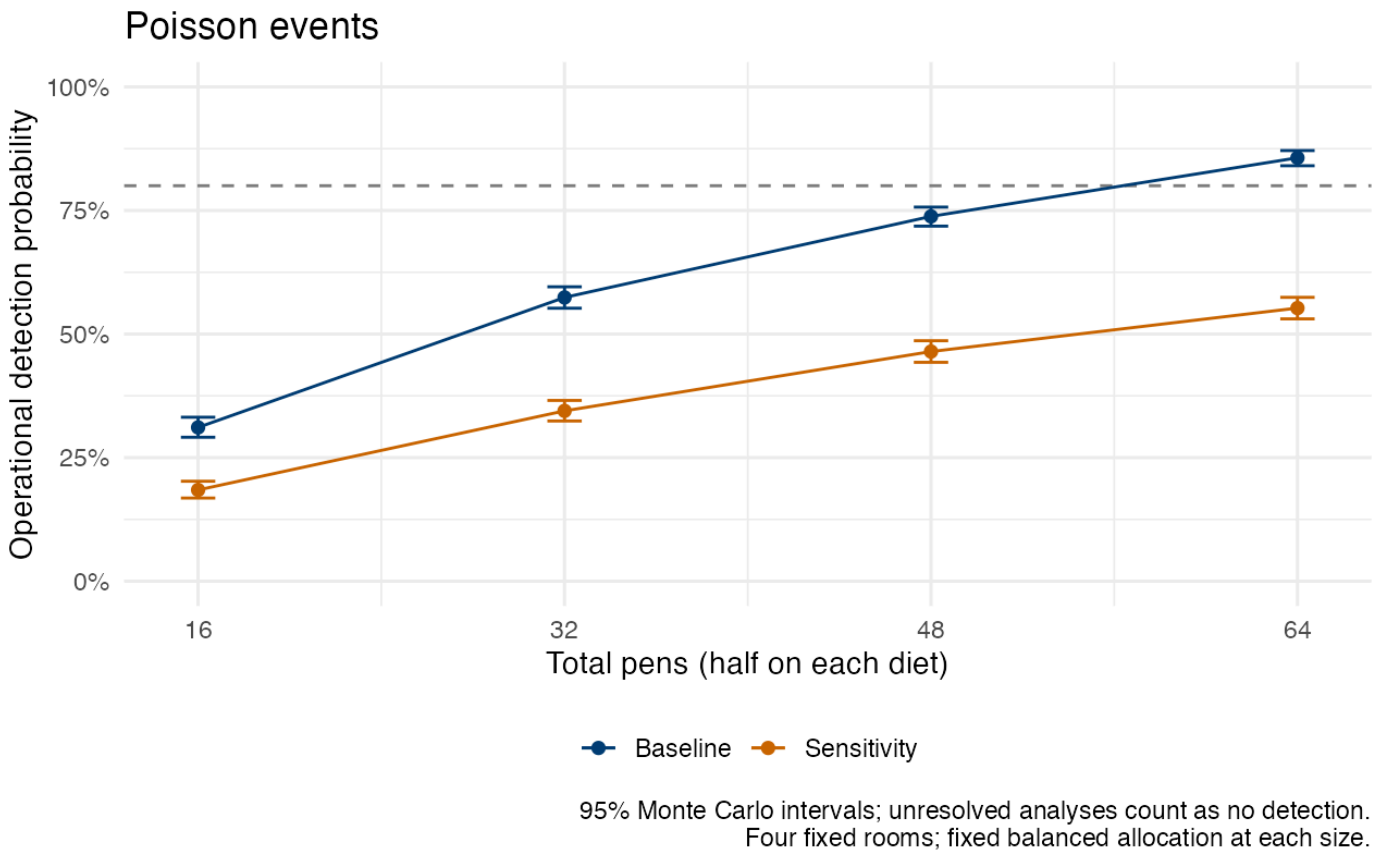
GROUP 1: REFERENCE CURVE



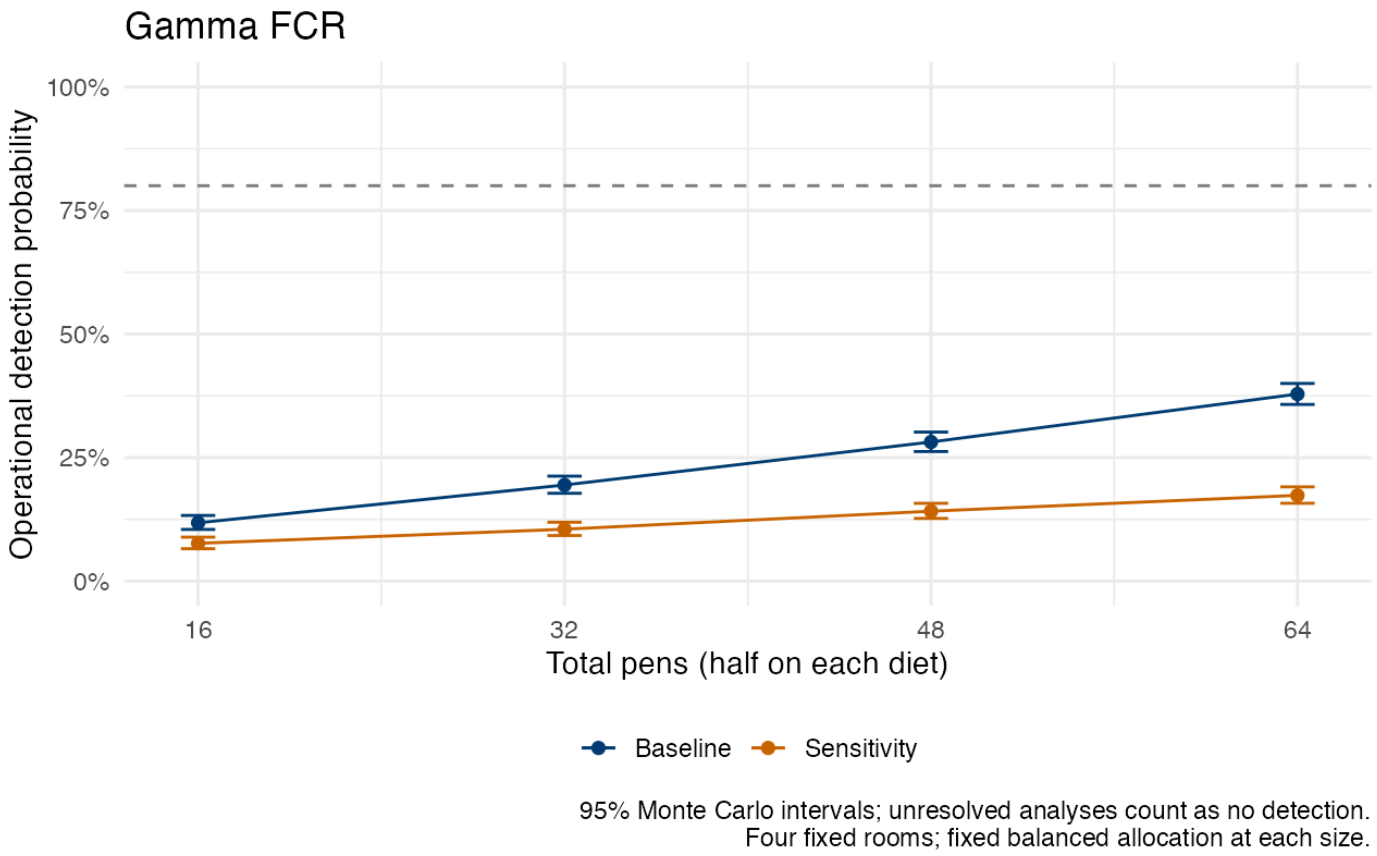
GROUP 2: REFERENCE CURVE



GROUP 3: REFERENCE CURVE



GROUP 4: REFERENCE CURVE



SOURCES

- R Core Team: GLMs and model comparisons.
- Wickham et al. (2019): Welcome to the Tidyverse.
- Colegrave and Ruxton (2020): *Power Analysis: An Introduction for the Life Sciences*. Instructor background.

Reference curves are generated from the supplied teaching scenarios.