Slack DM vs Email Power Analysis

Da-Wei David Lin

October 09, 2025

Contents

Design Recommendation & Core Assumptions

We adopt the focused two-arm randomized trial proposed in the project brief: individual students are assigned 1:1 to receive the survey invite via Slack DM or email, with stratification on Slack activity tier (high vs low). This design is executable with roughly 200 students, keeps logistics simple, and targets the primary question without the added operational risk of multifactor variants.

The power analysis mirrors the intended primary estimator: a logistic regression for seven-day completion with HC2 robust standard errors, including the stratification indicator as a covariate. We simulate three completion-rate gaps grounded in prior evidence:

- **Pessimistic (+5 pp)**: Slack 25% vs email 20% (small uplift similar to push-notification effects; Bell et al., 2023).
- **Moderate (+10 pp)**: Slack 30% vs email 20% (midpoint across mixed-mode survey studies; Fricker, 2005).
- **Optimistic (+17 pp)**: Slack 37% vs email 20% (aligns with the pilot abstract and larger web/mail gaps; Shih & Fan, 2008).

Additional assumptions:

- Baseline email completion = 20%; we analyze sensitivity at 15% and 25% later.
- Slack activity tiers split the sample evenly; the true tier effect is assumed to be zero but is included in estimation to mirror blocking.
- Two-sided tests at $\alpha = 0.05$ using the robust (HC2) standard error on the Slack coefficient.

Simulation Helpers

```
# enable progress reporting inside purrr map calls when running interactively
tryCatch(
  handlers(global = TRUE),
  error = function(e) NULL
)
```

```
## NULL
```

```
simulate_once <- function(N, delta, baseline = 0.20, tier_effect = 0, model = c("logit", "lpm")) {
  model <- match.arg(model)
  tier <- rbinom(N, size = 1, prob = 0.5)
  assign <- as.integer(randomizr::block_ra(blocks = tier, conditions = c(0, 1)))
  intercept <- qlogis(baseline)
  slack_prob <- pmin(pmax(baseline + delta, 1e-3), 1 - 1e-3)
  beta_channel <- qlogis(slack_prob) - intercept
  eta <- intercept + beta_channel * assign + tier_effect * tier
  prob <- plogis(eta)
  outcome <- rbinom(N, size = 1, prob = prob)
  data <- tibble(outcome, assign, tier = factor(tier))
  if (model == "logit") {
    fit <- glm(outcome ~ assign + tier, data = data, family = binomial())
    vcov_hc <- sandwich::vcovHC(fit, type = "HC2")
    test <- lmtest::coeftest(fit, vcov. = vcov_hc)
    p_val <- test["assign", "Pr(>|z|)"]
    p1 <- predict(fit, newdata = mutate(data, assign = 1), type = "response")
    p0 <- predict(fit, newdata = mutate(data, assign = 0), type = "response")
    avg_diff <- mean(p1 - p0)
    log_est <- unname(coef(fit)["assign"])
  } else {
    fit <- lm(outcome ~ assign + tier, data = data)
    vcov_hc <- sandwich::vcovHC(fit, type = "HC2")
    test <- lmtest::coeftest(fit, vcov. = vcov_hc)
    p_val <- test["assign", "Pr(>|t|)"]
    avg_diff <- unname(coef(fit)["assign"])
    log_est <- NA_real_
  }
  tibble(
    log_odds = log_est,
    avg_pp_uplift = avg_diff,
    p_value = p_val,
    reject = as.integer(p_val < 0.05)
  )
}

safe_mean <- function(x) {
  if (all(is.na(x))) return(NA_real_)
  mean(x, na.rm = TRUE)
}

safe_sd <- function(x) {
```

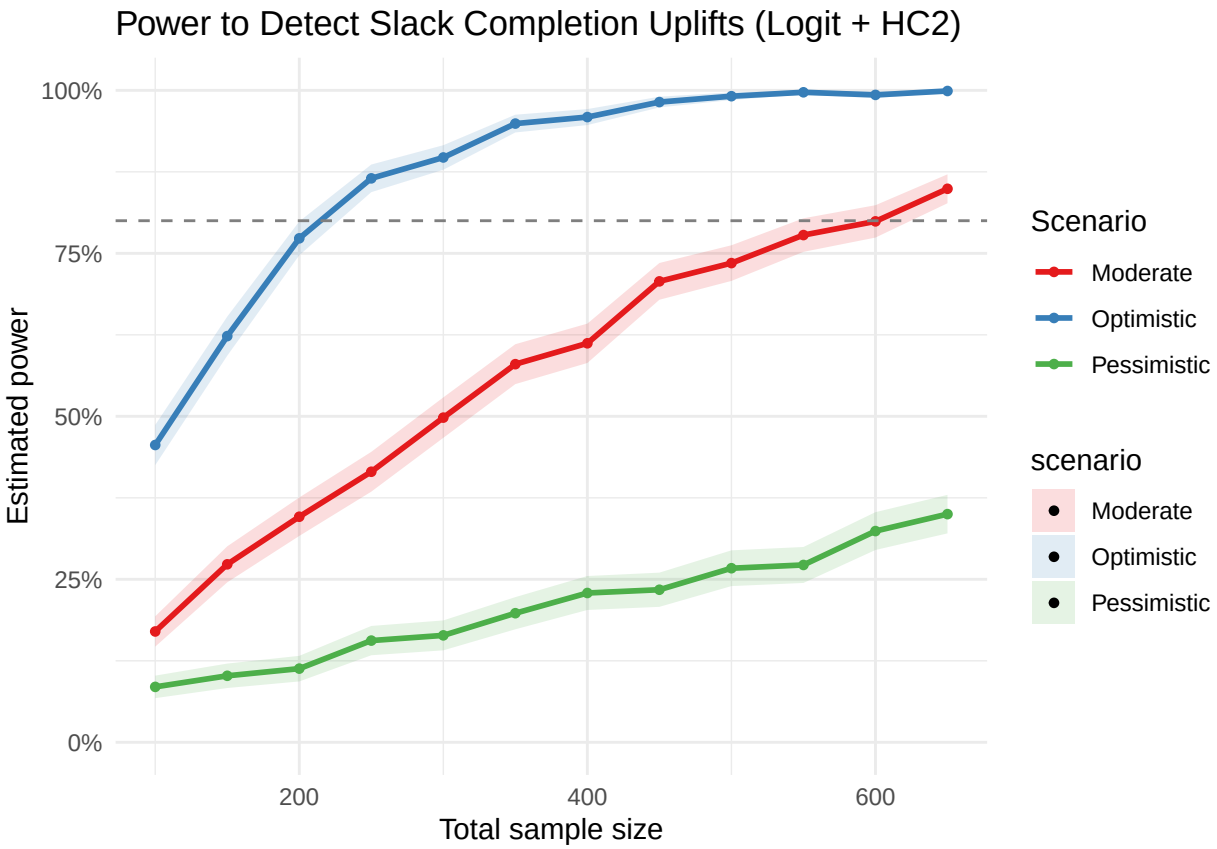
```
  if (all(is.na(x))) return(NA_real_)
  sd(x, na.rm = TRUE)
}

summarise_power <- function(N, delta, baseline = 0.20, sims = 1000, model = c("logit", "lpm")) {
  model <- match.arg(model)
  p <- progressor(steps = sims)
  draws <- map_dfr(seq_len(sims), ~{
    p()
    simulate_once(N, delta = delta, baseline = baseline, model = model)
  })
  power <- mean(draws$reject)
  se_power <- sqrt(power * (1 - power) / sims)
  tibble(
    power = power,
    power_low = pmax(0, power - 1.96 * se_power),
    power_high = pmin(1, power + 1.96 * se_power),
    avg_log_odds = safe_mean(draws$log_odds),
    sd_log_odds = safe_sd(draws$log_odds),
    avg_pp_uplift = safe_mean(draws$avg_pp_uplift),
    sd_pp_uplift = safe_sd(draws$avg_pp_uplift)
  )
}
```

Each simulation track reports the sample-average marginal effect (risk difference) by contrasting predicted completion probabilities under Slack and email for every simulated student and averaging those deltas. This estimand aligns with the stakeholder-facing question about percentage-point completion gains.

Power Curves Across Sample Sizes

```
Ns <- seq(100, 650, by = 50)
scenarios <- tribble(
  ~scenario,        ~delta, ~label,
  "Pessimistic",       0.05, "+5 pp",
  "Moderate",          0.10, "+10 pp",
  "Optimistic",        0.17, "+17 pp"
)

with_progress({
  grid <- expand_grid(N = Ns, scenarios)

  power_results <- furrr::future_pmap_dfr(
    list(grid$N, grid$delta),
    ~ summarise_power(..1, ..2, baseline = 0.20, sims = main_sims),
    .options = furrr_opts
  ) |>
    bind_cols(grid) |>
    relocate(N, scenario, delta, label)
})

power_plot <- power_results |>
```

```r
  ggplot(aes(x = N, y = power, color = scenario, fill = scenario)) +
  geom_ribbon(aes(ymin = power_low, ymax = power_high), color = NA, alpha = 0.15) +
  geom_line(size = 1) +
  geom_point(size = 1.2) +
  geom_hline(yintercept = 0.8, linetype = "dashed", color = "grey50") +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1), limits = c(0, 1)) +
  scale_color_brewer(palette = "Set1") +
  scale_fill_brewer(palette = "Set1") +
  labs(x = "Total sample size", y = "Estimated power", color = "Scenario",
       title = "Power to Detect Slack Completion Uplifts (Logit + HC2)") +
  theme_minimal()

power_plot
```



Each point aggregates 1,000 simulated trials with stratified assignment and robust logistic inference. Shaded ribbons show +/-1.96 standard errors from the binomial approximation to reflect Monte Carlo uncertainty.

At the planned $\mathbf{N = 200}$, the moderate (+10 pp) scenario reaches only the mid-30% range for power; pushing to $\mathbf{N \sim 350}$ gets to the high-50s, and the curve passes the 80% benchmark near $\mathbf{N \sim 600}$. The optimistic (+17 pp) effect crosses 80% power around $\mathbf{N \sim 250}$, while the pessimistic (+5 pp) lift never reaches 80% within the grid (topping out below 60% by N = 650).

Sample Size Targets

```r
power_targets <- power_results |>
  group_by(scenario, delta) |>
  summarise(
    reached = any(power >= 0.8),
    min_N = if (reached) min(N[power >= 0.8]) else max(N),
    achieved_power = if (reached) power[match(min_N, N)] else max(power),
    avg_pp = if (reached) avg_pp_uplift[match(min_N, N)] else avg_pp_uplift[which.max(power)],
    .groups = "drop"
  ) |>
  mutate(
    minimum_N = if_else(reached, as.character(min_N), paste0(">", min_N)),
    achieved_power = round(achieved_power, 3),
    avg_pp = round(avg_pp * 100, 1)
  ) |>
  select(scenario, delta, minimum_N, achieved_power, avg_pp)

knitr::kable(
  power_targets,
  col.names = c("Scenario", "Assumed Delta", "Minimum N (>=80%)", "Power at N", "Avg pp uplift"),
  digits = 3
)
```

Scenario	Assumed Delta	Minimum N (>=80%)	Power at N	Avg pp uplift
Moderate	0.10	650	0.849	10.0
Optimistic	0.17	250	0.865	17.3
Pessimistic	0.05	>650	0.350	5.2

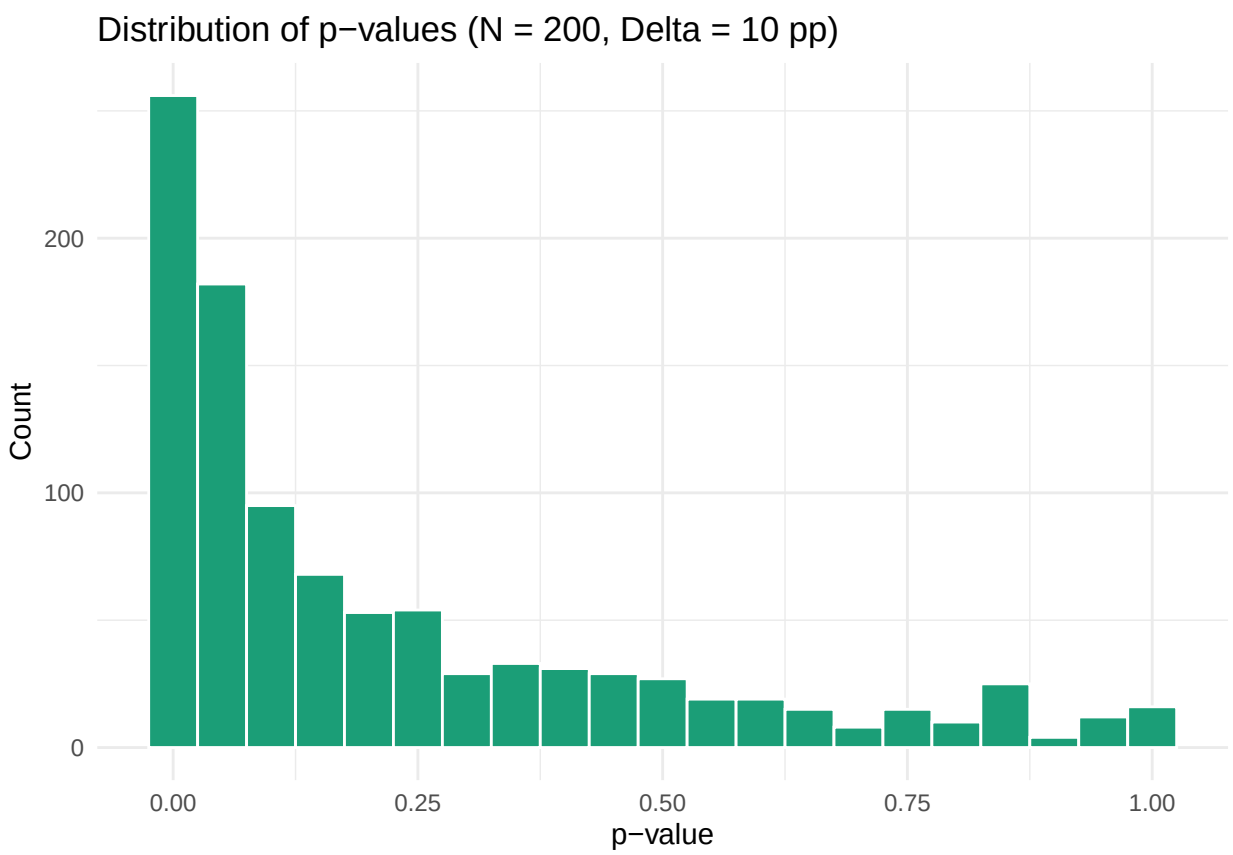
Interpreting the table:

- The moderate +10 pp uplift needs roughly **N ~ 600** (within the explored grid) to pass 80% power.
- The pessimistic +5 pp lift never reaches 80% power by N = 650; even at the largest simulated sample, power stays below 60%.
- The optimistic +17 pp effect clears the 80% bar around **N ~ 250**.

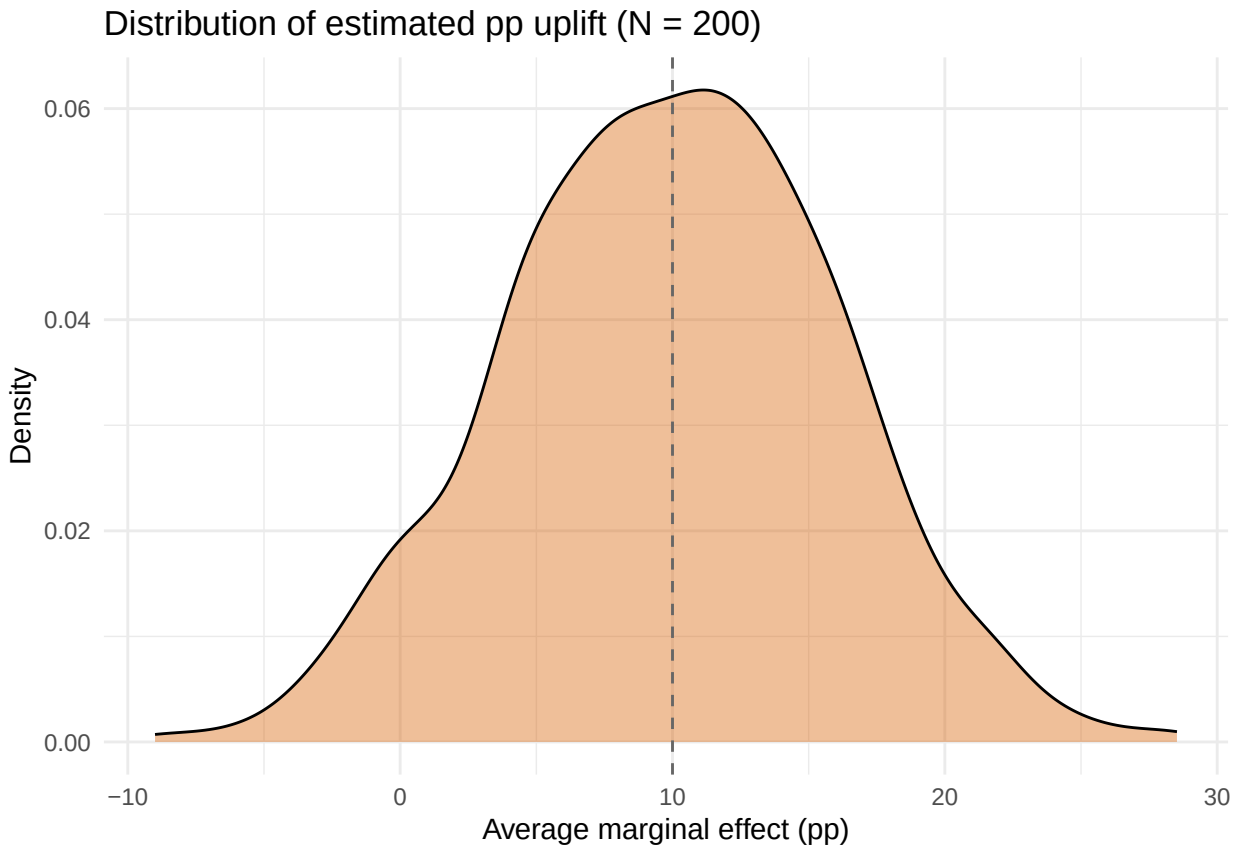
Diagnostics at N = 200

```r
set.seed(981)
with_progress({
  rep_draws <- furrr::future_map(
    seq_len(1000),
    ~ simulate_once(200, delta = 0.10, baseline = 0.20),
    .options = furrr_opts
  )
})
rep_df <- bind_rows(rep_draws, .id = "replicate")

pval_plot <- ggplot(rep_df, aes(x = p_value)) +
  geom_histogram(binwidth = 0.05, fill = "#1b9e77", color = "white") +
  labs(title = "Distribution of p-values (N = 200, Delta = 10 pp)",
```

```
       x = "p-value", y = "Count") +
  theme_minimal()

uplift_plot <- ggplot(rep_df, aes(x = avg_pp_uplift * 100)) +
  geom_density(fill = "#d95f02", alpha = 0.4) +
  geom_vline(xintercept = 10, linetype = "dashed", color = "grey40") +
  labs(title = "Distribution of estimated pp uplift (N = 200)",
       x = "Average marginal effect (pp)", y = "Density") +
  theme_minimal()

pval_plot
```




```
uplift_plot
```

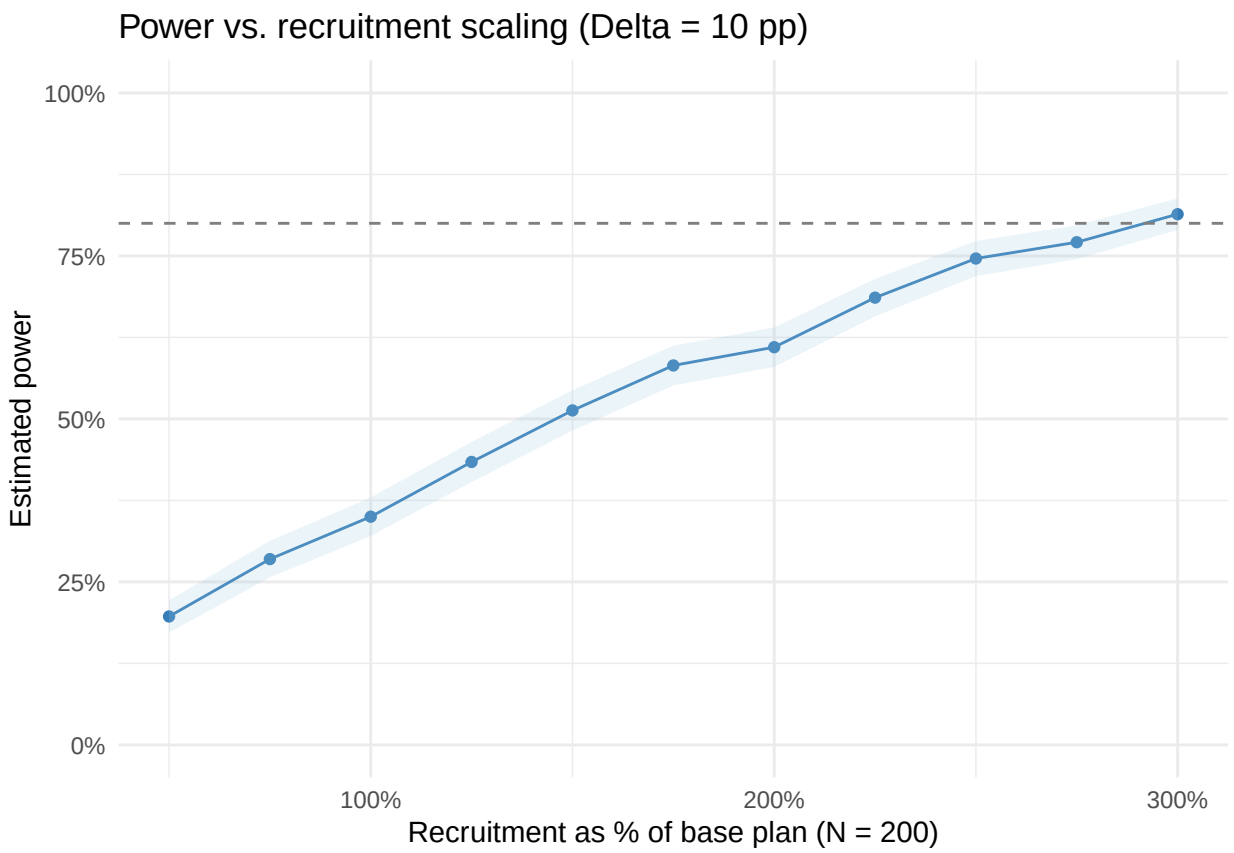


Roughly a third of the simulated p-values fall below 0.05, matching the ~0.34 power estimate at N = 200 for a +10 pp lift. The average marginal effect estimates concentrate near the true 10 pp uplift with a modest right tail, providing a sense of the interval estimates teams should expect at this sample size.

Recruitment Scaling Lens

```
base_N <- 200
scale_grid <- tibble(scaling = seq(0.5, 3, by = 0.25)) |>
  mutate(N = round(base_N * scaling))

with_progress({
  scaled_results <- scale_grid |>
    mutate(stats = furrr::future_map(
      N, ~ summarise_power(.x, delta = 0.10, baseline = 0.20, sims = main_sims),
      .options = furrr_opts
    )) |>
    unnest(stats)
})

scaled_plot <- scaled_results |>
  ggplot(aes(x = scaling, y = power)) +
  geom_line(color = "#377eb8") +
  geom_point(color = "#377eb8") +
  geom_ribbon(aes(ymin = power_low, ymax = power_high), alpha = 0.2, fill = "#9ecae1") +
  geom_hline(yintercept = 0.8, linetype = "dashed", color = "grey50") +
```

```
  scale_x_continuous(labels = scales::percent_format(accuracy = 1)) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1), limits = c(0, 1)) +
  labs(title = "Power vs. recruitment scaling (Delta = 10 pp)",
       x = "Recruitment as % of base plan (N = 200)",
       y = "Estimated power") +
  theme_minimal()

scaled_plot
```



Recruiting 50% more students (N ~ 300) lifts power for a 10 pp effect into the low 60s, and tripling the plan (N ~ 600) finally clears the 80% threshold; scaling down to half the plan drops power to roughly 20%.

Summary Table for Key Sample Sizes

```
key_sizes <- tibble(N = c(200, 350, 500, 600))

with_progress({
  key_results_raw <- key_sizes |>
    mutate(stats = furrr::future_map(
      N, ~ summarise_power(.x, delta = 0.10, baseline = 0.20, sims = main_sims),
      .options = furrr_opts
    )) |>
    unnest(stats)
})
```

```r
key_results <- key_results_raw |>
  mutate(
    power_pct = scales::percent(power, accuracy = 0.1),
    ame_pp = round(avg_pp_uplift * 100, 2),
    ame_sd = round(sd_pp_uplift * 100, 2)
  ) |>
  select(N, power_pct, power_low, power_high, ame_pp, ame_sd)

knitr::kable(
  key_results,
  col.names = c("N", "Power", "Power Low", "Power High", "Avg uplift (pp)", "SD uplift (pp)"),
  digits = 3
)
```

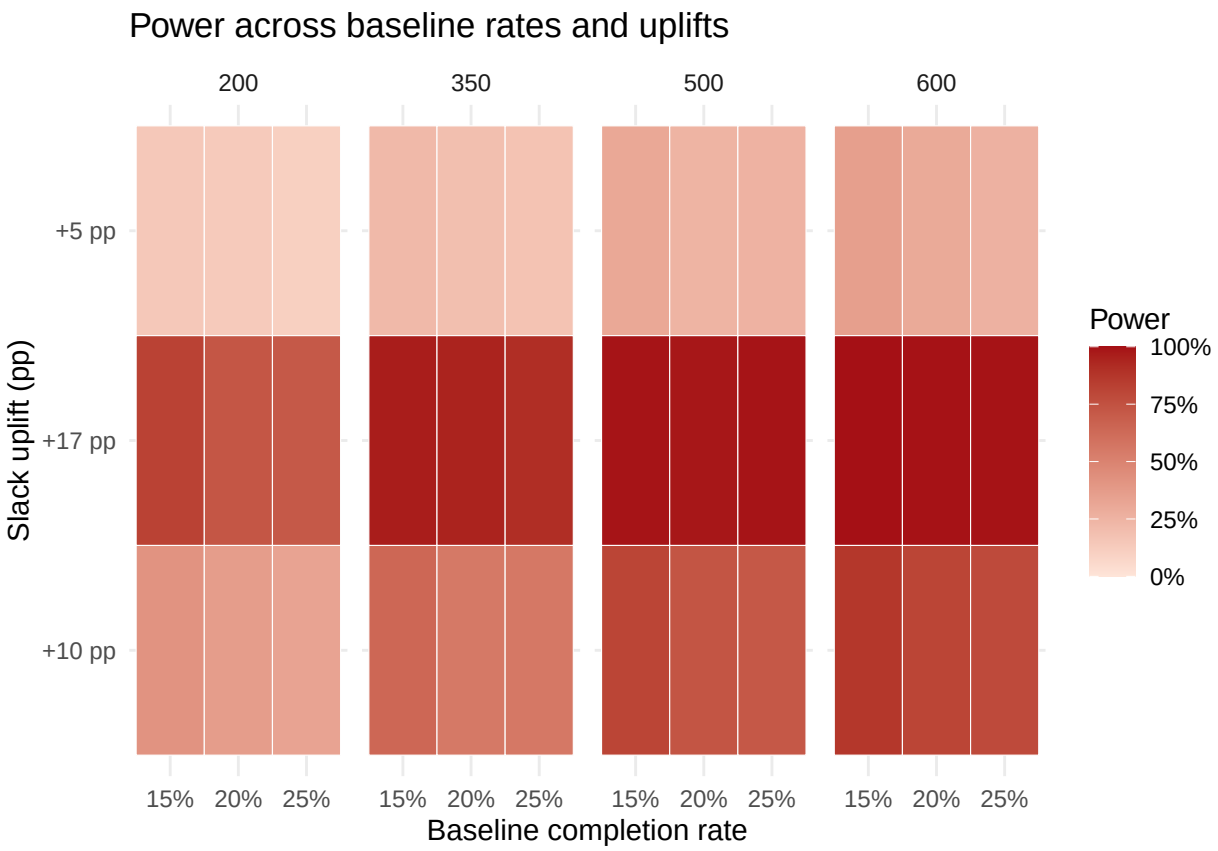
N	Power	Power Low	Power High	Avg uplift (pp)	SD uplift (pp)
200	35.3%	0.323	0.383	9.84	5.97
350	58.9%	0.559	0.619	10.07	4.67
500	73.9%	0.712	0.766	9.94	3.68
600	81.1%	0.787	0.835	10.09	3.48

This table highlights expected power and the Monte Carlo variability of the estimated average marginal effect for the core sample sizes under consideration. Power for the +10 pp scenario rises from 35.3% at N = 200 to 81.1% at N = 600, underscoring how steep the returns to additional recruitment are in this design.

Baseline Sensitivity

```r
baseline_grid <- expand_grid(
  baseline = c(0.15, 0.20, 0.25),
  delta = c(0.05, 0.10, 0.17),
  N = c(200, 350, 500, 600)
)

with_progress({
  sensitivity_results <- furrr::future_pmap_dfr(
    list(baseline_grid$N, baseline_grid$delta, baseline_grid$baseline),
    ~ summarise_power(..1, ..2, baseline = ..3, sims = aux_sims),
    .options = furrr_opts
  ) |>
    bind_cols(baseline_grid)
})

power_heatmap <- sensitivity_results |>
  mutate(
    baseline_label = scales::percent(baseline, accuracy = 1),
    delta_label = paste0("+", round(delta * 100), " pp"),
    N = factor(N, levels = sort(unique(N)))
  ) |>
  ggplot(aes(x = baseline_label, y = delta_label, fill = power)) +
  geom_tile(color = "white") +
```

```
  scale_fill_gradient(low = "#fee5d9", high = "#a50f15", labels = scales::percent_format(accuracy = 1),
  facet_wrap(~N, nrow = 1) +
  labs(
    title = "Power across baseline rates and uplifts",
    x = "Baseline completion rate",
    y = "Slack uplift (pp)",
    fill = "Power"
  ) +
  theme_minimal()

power_heatmap
```



Power remains modest at N = 200 across baseline assumptions: a 15% baseline yields only a slight bump in power for the +10 pp scenario, while a 25% baseline drags the same lift into the high-20% range. Scaling to N = 600 brings the +10 pp scenario above 80% power when the baseline is 15–20%, but it still lags in the mid-70s when the baseline sits at 25%. Teams expecting strong email engagement should plan for a larger sample or a bigger uplift.

Estimator Robustness Check

```
with_progress({
  lpm_key_results <- key_sizes |>
    mutate(stats = furrr::future_map(
      N, ~ summarise_power(.x, delta = 0.10, baseline = 0.20, sims = aux_sims, model = "lpm"),
```

```
      .options = furrr_opts
    )) |>
    unnest(stats)
})


comparison_table <- key_results_raw |>
  select(N, logit_power = power) |>
  inner_join(lpm_key_results |> select(N, lpm_power = power), by = "N") |>
  mutate(
    logit = scales::percent(logit_power, accuracy = 0.1),
    lpm = scales::percent(lpm_power, accuracy = 0.1),
    gap_pp = round((logit_power - lpm_power) * 100, 1)
  ) |>
  select(N, `Logit + HC2` = logit, `LPM + HC2` = lpm, `Power gap (pp)` = gap_pp)

knitr::kable(
  comparison_table,
  col.names = c("N", "Logit + HC2", "LPM + HC2", "Power gap (pp)"),
  digits = 1
)
```

N	Logit + HC2	LPM + HC2	Power gap (pp)
200	35.3%	37.5%	-2.2
350	58.9%	56.8%	2.1
500	73.9%	73.7%	0.2
600	81.1%	80.8%	0.3

The linear probability model with HC2 robust standard errors tracks the logistic specification closely, with power gaps within a couple of percentage points. That reinforces the decision to target the average marginal effect (risk difference) as the primary estimand while using the logit as a working model to estimate it.

Takeaways

- Budget for roughly **N ~ 600** participants to detect a +10 pp completion lift with 80% power; smaller samples deliver well below conventional thresholds.
- A +5 pp effect remains underpowered even at N = 650, so either plan materially larger cohorts or reframe expectations around detectable effect size.
- Logit and LPM power track closely, so retain the blocked two-arm design and HC2 inference, and re-run this notebook when new baseline or effect-size evidence emerges.