Business Data Analytics – Experimentation & Basic Causal Inference

Assoc. Prof. Nguyen Binh Minh, Ph.D.

Learning Objectives (CLO)

- Design trustworthy A/B tests aligned with business goals.
- Understand Type I/II errors, p-values, confidence intervals.
- Compute sample size and reason about power and MDE.
- Apply CUPED for variance reduction and faster reads.
- Recognize causal assumptions and common pitfalls.

Agenda & Timing

1) Experimentation fundamentals (30’)

2) Hypothesis testing & p-values (30’)

3) Power, MDE, sample size (35’)

4) CUPED & variance reduction (30’)

5) Multiple testing & sequential pitfalls (25’)

6) Causal thinking basics (20’)

Why Experiment?

- Measure causal impact, not just correlation.
- Reduce decision risk with randomized controls.
- Build organizational learning via iteration.

Potential Outcomes & Counterfactuals

- Each unit has Y(1) and Y(0); we observe one.
- ATE = E[Y(1) − Y(0)] — randomization estimates it.
- Ignorability via random assignment; SUTVA assumptions.

Treatment, Control & Randomization

Random assignment balances observed/unobserved factors.

Simple vs. stratified/block randomization; cluster designs.

Avoid selection bias and time-varying confounding.

Experiment Lifecycle

- Define: hypotheses, metrics (primary & guardrails), success criteria.
- Design: randomization, sample size, duration, ramp.
- Run: sanity checks, SRM detection, monitoring;
- Analyze: effect, uncertainty, decision;
- Learn: archive, next iteration.

Metric Design

- Primary (KPIs linked to decision), secondary (diagnostics).
- Guardrails (latency, error rate); avoid metric gaming.
- Stable, sensitive, low noise; avoid proxies when possible.

A/A Tests & Sanity Checks

Detect instrumentation issues and SRM before shipping.

Check event counts, conversion baselines, unit balance.

Sample Ratio Mismatch (SRM)

- Observed assignment ratio deviates from design.
- Causes: tracking bugs, bot traffic, randomization drift.
- Stop and investigate; don’t trust results.

Hypotheses & Errors

- H0 (no effect) vs H1 (effect present).
- Type I (false positive, α) and Type II (false negative, β).
- Power = 1 − β; MDE ties effect size to n and noise.

p-values & CIs — Intuition

- p-value: data extremeness under H0, not effect probability.
- Confidence interval: range of plausible effects at 1−α level.
- Report effects with CIs; avoid dichotomous thinking.

Two-Sample Tests — Overview

Proportions (conversion): two-proportion z-test.

Means (revenue): Welch’s t-test; robust to unequal variances.

Nonparametric options: Mann–Whitney when distributions are odd.

Effect Size & MDE

Absolute vs relative lift; standardized effects (Cohen’s h/d).

Define practically significant MDE, not just statistical.

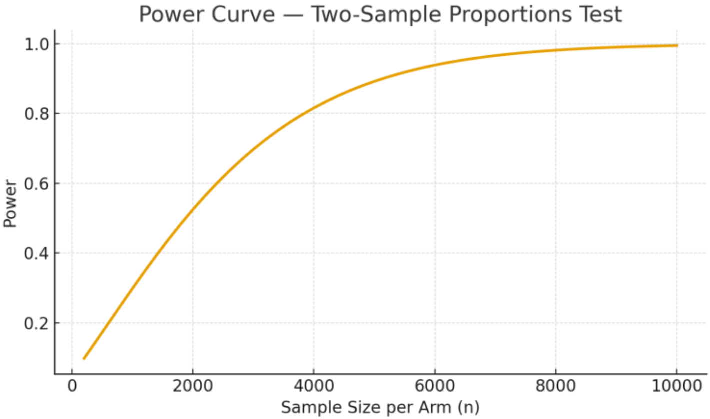
Power & Sample Size — Proportions

Inputs: baseline p0, MDE, α, desired power.

Trade-offs: smaller MDE → larger n; higher power → larger n.

Power Curve (Two-Sample Proportions)

Baseline=0.10, MDE=0.02, alpha=0.05



Sequential Testing & Peeking

- Naïve peeking inflates Type I error.
- Use group-sequential or alpha-spending methods if interim looks are required.
- Alternatively, use Bayesian monitoring with pre-specified rules.

Multiple Testing

- Feature flags and many metrics → multiplicity.
- Control FWER (Bonferroni) or FDR (Benjamini–Hochberg).
- Pre-register primary endpoints to limit garden-of-forking-paths.

Distributional Issues

- Heavy tails in revenue/time-on-site; winsorize or use robust stats.
- Ratio metrics (ARPU) → delta method or Fieller’s theorem.

Variance Reduction — Why

- Reduce noise → smaller required n or shorter test duration.
- Condition on pre-experiment information or covariates.

CUPED — Idea

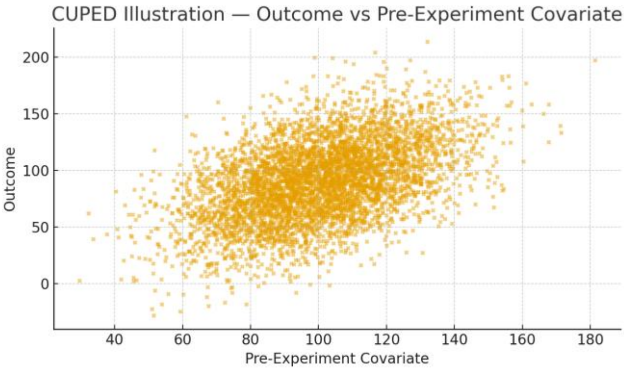
- Use a pre-experiment covariate X highly correlated with outcome Y.
- Define $Y^* = Y - \theta (X - E[X])$; choose θ to minimize $\mathrm{Var}(Y^*)$.
- Keeps effect unbiased under randomization; reduces variance.

CUPED — Estimating θ

- $\theta = \mathrm{Cov}(Y, X) / \mathrm{Var}(X)$ (OLS coefficient of Y on X).
- Compute on pre-assignment data (or training split) to avoid leakage.
- Apply per-arm consistently.

CUPED Illustration — Scatter

Outcome vs Pre-Experiment Covariate



Core Idea & Formula

- $\theta = \mathrm{Cov}(Y, X) / \mathrm{Var}(X)$ (slope from regressing Y on X)
- Adjusted metric: $Y^* = Y - \theta \cdot (X - E[X])$
- Estimate ATE by diff-in-means on Y^* between Treatment and Control.

Variance Reduction

- Correlation $\rho(X,Y) \approx 0.74$.
- Variance factor $\approx (1 - \rho^2)$. Effective sample gain $\approx 1/(1 - \rho^2)$.
- In this example: $1 - \rho^2 \approx 0.45 \Rightarrow$ ~2.23× effective sample size.

Assumptions & Pitfalls

- X must be pre-period or not affected by treatment.
- Proper randomization; check for SRM first.
- Linear relationship Y–X is adequate; for strong nonlinearity consider CUPAC/ML-CUPED.
- Use correct SE/CI (cluster-robust if clustered experiments).

Step-by-step Recipe

- Choose a stable pre-metric X that correlates with Y.
- Compute $\theta = \mathrm{Cov}(Y, X) / \mathrm{Var}(X)$.
- Create adjusted metric $Y^* = Y - \theta \cdot (X - \mathrm{mean}(X))$.
- Run your usual two-sample test on Y*. Report effect, CI, and variance reduction.

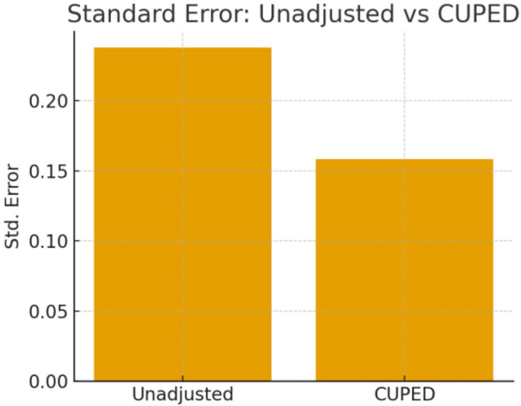
Synthetic Example — Setup

- Control n=5000, Treatment n=5000.
- Post metric Y depends on X plus a true treatment lift ($\delta = 2.0$).
- We compute θ from data and form Y*, then compare T vs C on Y and on Y*.

Results: Effect & Precision

Metric	Effect Estimate	Std. Error	95% CI Low	95% CI High
Unadjusted (Y)	2.104	0.238	1.638	2.57
CUPED (Y*)	1.983	0.158	1.672	2.294

Standard Error: Unadjusted vs CUPED

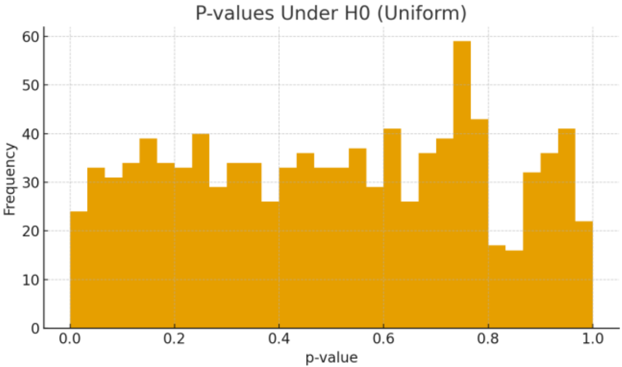


Implementation (Pseudo)

- theta = cov(Y, X)/var(X); Y_adj = Y − theta*(X − mean(X))
- ATE = mean(Y_adj[T=1]) − mean(Y_adj[T=0])
- Or OLS: Y ~ T + X; the coefficient of T equals CUPED effect.

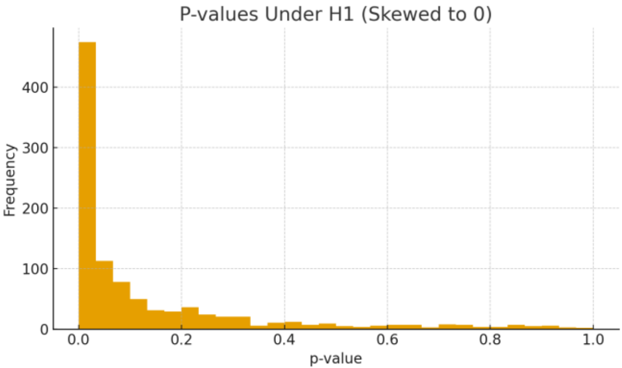
P-values Under H0

Uniform(0,1) under the null



P-values Under H1

Skewed toward 0 under the alternative



Causal Thinking — Basics

- Randomization breaks confounding on average.
- Blocking/stratification improves precision.
- Cluster randomization when interference within clusters is likely.

Reporting Results

- State effect with CI, power achieved, and assumptions.
- Include diagnostics: SRM check, metric stability, outliers.
- Decision and next steps (ship, iterate, or stop).

Mini-Case — Email Promo CTR (Setup)

- Goal: improve CTR by +0.3pp from baseline 3.0%.
- Primary metric: CTR; Guardrails: bounce rate, unsubscribes.
- Design an A/B with stratified randomization by segment.

Mini-Case — Email Promo CTR (Analysis)

- Compute difference in proportions with CI.
- Check p-value, practical significance vs MDE.
- Assess guardrails and make the ship/hold decision.

Hands-on Tasks

- Compute sample size for p0=0.03, MDE=0.003, power=0.85, α=0.05.
- Run a simulation to validate Type I error at α=0.05.
- Apply CUPED with a pre-period open-rate covariate.

Quick Quiz (10)

Define Type I vs Type II error.

What does a p-value actually measure?

How does CUPED reduce variance?

Why is peeking problematic?

When would you prefer DiD to randomized A/B?

Key Takeaways

- Randomized tests turn correlation into causation.
- Power, MDE, and variance reduction govern speed & reliability.
- Report uncertainty; pre-register and avoid p-hacking.

Recommended References

- Kohavi et al. — Trustworthy Online Controlled Experiments.
- Goodman (2019) — What does p-value mean?
- Deng et al. — Improving the Sensitivity of Online Controlled Experiments (CUPED).
- Matplotlib & statsmodels documentation.

Appendix — Proportions Test (Formulae)

- Test statistic: z = (p2 − p1)/SE, with SE from pooled variance under H0.
- CI for difference uses unpooled SE; beware small-sample corrections.

Appendix — Power (Proportions)

- Power depends on p0, p1, α, and n;
- Use normal approximation or exact methods for small n.

Code — Two-Proportion Z-test (p-value)

```
from math import sqrt
from scipy.stats import norm

def two_prop_ztest_pvalue(x1, n1, x2, n2):
    p1_hat = x1 / n1
    p2_hat = x2 / n2
    p_pool = (x1 + x2) / (n1 + n2)
    se = sqrt(p_pool*(1-p_pool)*(1/n1 + 1/n2))
    z = (p2_hat - p1_hat) / se
    return 2*(1 - norm.cdf(abs(z)))  # two-sided
```

Code — CUPED (θ and adjusted outcome)

```
import numpy as np

def cuped_adjust(y, x):
    theta = np.cov(y, x, ddof=1)[0,1] / np.var(x, ddof=1)
    y_adj = y - theta * (x - np.mean(x))
    return theta, y_adj
```