Potencia estadística

Diseño e implementación de experimentos en ciencias sociales

Departamento de Economía (UdelaR)

What is power?

What is power?

- We want to separate signal from noise.
- Power = probability of rejecting null hypothesis, given true effect $\neq 0$.
- In other words, it is the ability to detect an effect given that it exists.
- Formally: (1 - Type II) error rate.
- Thus, power $\in (0, 1)$.
- Standard thresholds: 0.8 or 0.9.

Starting point for power analysis

- Power analysis is something we do *before* we run a study.
  - Helps you figure out the sample you need to detect a given effect size.
  - Or helps you figure out a minimal detectable difference given a set sample size.
  - May help you decide whether to run a study.
- It is hard to learn from an under-powered null finding.
  - Was there an effect, but we were unable to detect it? or was there no effect? We can't say.

Power

- Say there truly is a treatment effect and you run your experiment many times. How often will you get a statistically significant result?
- Some guesswork to answer this question.
  - How big is your treatment effect?
  - How many units are treated, measured?
  - How much noise is there in the measurement of your outcome?

Approaches to power calculation

- Analytical calculations of power
- Simulation

Power calculation tools

- Interactive
  - EGAP Power Calculator
  - rpsychologist
- R Packages
  - pwr
  - DeclareDesign, see also https://declaredesign.org/

Analytical calculations of power

Analytical calculations of power

- Formula:

$$\text{Power} = \Phi\left(\frac{|\tau|\sqrt{N}}{2\sigma} - \Phi^{-1}(1 - \frac{\alpha}{2})\right)$$

- Components:
  - ϕ: standard normal CDF is monotonically increasing
  - τ: the effect size
  - N: the sample size
  - σ: the standard deviation of the outcome
  - α: the significance level (typically 0.05)

Example: Simulation-based power for complete randomization

```
power_calculator <- function(mu_t, mu_c,
    sigma, alpha = 0.05, N) {
    lowertail <- (abs(mu_t - mu_c) * sqrt(N))/(2 *
        sigma)
    uppertail <- -1 * lowertail
    beta <- pnorm(lowertail - qnorm(1 - alpha/2),
        lower.tail = TRUE)
    +1 - pnorm(uppertail - qnorm(1 - alpha/2),
        lower.tail = FALSE)
    return(beta)
}
```

Example: Simulation-based power for complete randomization

```r
power_calculator(mu_t = 1/4, mu_c = 0, sigma = 1,
    alpha = 0.05, N = 100)
```

```
## [1] 0.2388632
```

Example: Simulation-based power for complete randomization

```r
power_calculator(mu_t = 1/4, mu_c = 0, sigma = 1,
    alpha = 0.05, N = 1000)
```

```
## [1] 0.9768629
```

Example: Simulation-based power for complete randomization

```r
power_calculator(mu_t = 1/4, mu_c = 0, sigma = 2,
    alpha = 0.05, N = 1000)
```

```
## [1] 0.5065661
```

Example: Simulation-based power for complete randomization

```
power_calculator(mu_t = 2/4, mu_c = 0, sigma = 1,
    alpha = 0.05, N = 1000)
```

```
## [1] 1
```

Example: Using DeclareDesign

```
library(DeclareDesign)
library(tidyverse)

P0 <- declare_population(N, u0 = rnorm(N))
# declare Y(Z=1) and Y(Z=0)
O0 <- declare_potential_outcomes(Y_Z_0 = 5 +
    u0, Y_Z_1 = Y_Z_0 + tau)
# design is to assign m units to
# treatment
A0 <- declare_assignment(Z = conduct_ra(N = N,
    m = round(N/2)))
# estimand is the average difference
# between Y(Z=1) and Y(Z=0)
estimand_ate <- declare_inquiry(ATE = mean(Y_Z_1 -
    Y_Z_0))
R0 <- declare_reveal(Y, Z)
design0_base <- P0 + A0 + O0 + R0
```

Example: Using DeclareDesign

```r
## For example:
design0_N100_tau25 <- redesign(design0_base,
    N = 100, tau = 0.25)
dat0_N100_tau25 <- draw_data(design0_N100_tau25)
head(dat0_N100_tau25)
```

```
##    ID          u0 Z    Y_Z_0    Y_Z_1        Y
## 1 001  0.44853297 1 5.448533 5.698533 5.698533
## 2 002 -0.20105196 1 4.798948 5.048948 5.048948
## 3 003  1.54239837 0 6.542398 6.792398 6.542398
## 4 004 -0.54998139 1 4.450019 4.700019 4.700019
## 5 005 -1.10680777 1 3.893192 4.143192 4.143192
## 6 006 -0.04259078 0 4.957409 5.207409 4.957409
```

Example: Using DeclareDesign

```r
with(dat0_N100_tau25, mean(Y_Z_1 - Y_Z_0))  # true ATE
```

```
## [1] 0.25
```

```r
with(dat0_N100_tau25, mean(Y[Z == 1]) - mean(Y[Z ==
    0]))  # estimate
```

```
## [1] 0.3374549
```

```r
lm_robust(Y ~ Z, data = dat0_N100_tau25)$coef  # estimate
```

```
## (Intercept)           Z
##   5.0510077   0.3374549
```

Example: Using DeclareDesign

```
E0 <- declare_estimator(Y ~ Z, model = lm_robust,
    label = "t test 1", inquiry = "ATE")
t_test <- function(data) {
    test <- with(data, t.test(x = Y[Z ==
        1], y = Y[Z == 0]))
    data.frame(statistic = test$statistic,
        p.value = test$p.value)
}
T0 <- declare_test(handler = label_test(t_test),
    label = "t test 2")
design0_plus_tests <- design0_base + E0 +
    T0

design0_N100_tau25_plus <- redesign(design0_plus_tests,
    N = 100, tau = 0.25)

## Only repeat the random assignment,
## not the creation of Y0. Ignore
```

Power with covariate adjustment

Covariate adjustment and power

- Covariate adjustment can improve power because it mops up variation in the outcome variable.
  - If prognostic, covariate adjustment can reduce variance dramatically. Lower variance means higher power.
  - If non-prognostic, power gains are minimal.
- All covariates must be pre-treatment. Do not drop observations on account of missingness.
  - See the module on threats to internal validity and the 10 things to know about covariate adjustment.
- Freedman's bias as n of observations decreases and K covariates increases.

Power with covariate adjustment

Covariate adjustment and power

- Covariate adjustment can improve power because it mops up variation in the outcome variable.
  - If prognostic, covariate adjustment can reduce variance dramatically. Lower variance means higher power.
  - If non-prognostic, power gains are minimal.
- All covariates must be pre-treatment. Do not drop observations on account of missingness.
  - See the module on threats to internal validity and the 10 things to know about covariate adjustment.
- Freedman's bias as n of observations decreases and K covariates increases.

Blocking

- Blocking: randomly assign treatment within blocks
  - "Ex-ante" covariate adjustment
  - Higher precision/efficiency implies more power
  - Reduce "conditional bias": association between treatment assignment and potential outcomes
  - Benefits of blocking over covariate adjustment clearest in small experiments

Power for cluster randomization

Power and clustered designs

- Recall the randomization module.
- Given a fixed N, a clustered design is weakly less powered than a non-clustered design.
  - The difference is often substantial.
- We have to estimate variance correctly:
  - Clustering standard errors (the usual)
  - Randomization inference
- To increase power:
  - Better to increase number of clusters than number of units per cluster.
  - How much clusters reduce power depends critically on the intra-cluster correlation (the ratio of variance within clusters to total variance).

A note on clustering in observational research

- Often overlooked, leading to (possibly) wildly understated uncertainty.
  - Frequentist inference based on ratio $\hat{\beta}/\hat{se}$
  - If we underestimate $\hat{se}$, we are much more likely to reject H_0. (Type-I error rate is too high.)
- Many observational designs much less powered than we think they are.

EGAP Power Calculator

- Try the calculator at: https://egap.shinyapps.io/power-app/
- For cluster randomization designs, try adjusting:
  - Number of clusters
  - Number of units per clusters
  - Intra-cluster correlation
  - Treatment effect

Comments

- Know your outcome variable.
- What effects can you realistically expect from your treatment?
- What is the plausible range of variation of the outcome variable?
  - A design with limited possible movement in the outcome variable may not be well-powered.

Conclusion: How to improve your power

1. Increase the N
   - If clustered, increase the number of clusters if at all possible
2. Strengthen the treatment
3. Improve precision
   - Covariate adjustment
   - Blocking
4. Better measurement of the outcome variable