Power Analysis Exercise

Kohei Hayashida

2023-01-04

The purpose of this supplementary document for power analysis is as follows:

- Motivate someone interested in using R for power analysis.
- Explain the connection with power equation and the code.

Review of the basic concepts

- Type 1 Error: (hasty α)
$$\Pr(\text{reject } H_0 \mid H_0 \text{ is true}) = \alpha$$
- Type 2 Error: (careless β)
$$\Pr(\text{accept } H_0 \mid H_1 \text{ is true}) = \beta$$
- Power calculation:
$$n = 2\left(t_\alpha + t_{1-\beta}\right)^2 * \frac{\sigma^2}{\delta^2}$$
  where
- n: sample size
- δ: minimum detectable effect (MDE)
- σ: standard deviation of the effect
- t_α: significance level (usually pre-set at 5%)
- $t_{1-\beta}$: power (pre-set at 80%)

This formula also implies that there is a trade-off between t_α and $t_{1-\beta}$ given n, δ, σ. If you really want to avoid Type 1 error, you need to give up some Type 2 error unless you cannot change other control variables n, σ.

From here, you can do many things!

- To know what sample size n you need given $\delta, \sigma, t_\alpha, t_{1-\beta}$.
- To know what power $t_{1-\beta}$ is used in the study given $n, \delta, \sigma, t_\alpha$.

From Scratch

```
sigma <- 20
delta <- 5
MDES <- delta / sigma
```

Before using the package, it is always a good idea to do it by hand for checking your understanding.

Here, I provide two different ways to do by hand. In your final analysis, please use the R package or PowerUp! because it is using more precise way (but black-box).

Approximation by standard normal:

Since t-distribution depends on n, we need a numerical optimization to find n that satisfies the power equation. Instead, suppose we can approximate it by standard normal distribution (which is independent of n). Note that this only works in the case of large n. This is just for illustration purpose, and you finally need to use the package version which run numerical optimization internally.

```
t_alpha <- qnorm(0.975) # prob. type 1 error
t_beta <- qnorm(0.8) # power

2*(t_alpha + t_beta)^2 * 1/(MDES)^2
```

```
## [1] 251.1642
```

Exact Solution by t-distribution

Recall that t-distribution is a function of n. So, we need to find the root of the power equation. Since t-distribution is complex to solve analytically, it is handy to use numerical root search method as below. This is a version of what `pwr.t.test` is doing internally.

```
power_eq <- function(n) {
  t_alpha <-  qt(0.975, n-1, lower.tail = TRUE)
  t_beta <- qt(0.8, n-1, lower.tail = TRUE)
  n - 2*(t_alpha + t_beta)^2 * 1/(MDES)^2 # This is the power equation under the t-test
}

n <- uniroot(power_eq, interval = c(2+1e-10, 1e+09))$root
print(n)
```

```
## [1] 253.1191
```

Here, we see the almost same number as before. (Also, this is the same answer you get from the PowerUp! in this setting.)

If you are interested in, you can do the same thing by balancing the area of power as is shown in the slide instead of numerically solving the power equation. It just needs more complication in the above codes but doable. Actually, it is the exact way that `pwr.t.test` are implemented. So, from now on, you can rely on `pwr.t.test` with more confidence.

Using Package

Power Calculation in T-test (Differences in Mean)

Suppose we want to test the average effect differences in two groups. Recall that we use t-test in mean difference test. Note that n in the result represents the number in *each* group.

```
pwr.t.test(d = MDES,
           sig.level = 0.05, # Prob. of Type 1 Error
           power = 0.8, # 1 - (Prob. of Type 2 Error)
           type = "two.sample",
           alternative = "two.sided")
```

```
##
##      Two-sample t test power calculation
##
##               n = 252.1275
##               d = 0.25
##       sig.level = 0.05
```

```
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Here, we see the almost same number as before.

Power Calculation in Proportional Test

The details of the power formula in the case of binary outcome, see List, Sadoff, and Wagner (2011, Experimental Economics).

$$\delta = \left(t_{\alpha/2} + t_{\beta}\right)\sqrt{\frac{\sigma_0^2}{n_0} + \frac{\sigma_1^2}{n_1}}$$

The total number you need can be calculated as follows:

$$N^* = \left(\frac{t_{\alpha/2} + t_{\beta}}{\delta}\right)^2 \left(\frac{\sigma_0^2}{\pi_0^*} + \frac{\sigma_1^2}{\pi_1^*}\right)$$

where

$$\pi_0^* = \frac{\sigma_0}{\sigma_0 + \sigma_1}, \pi_1^* = \frac{\sigma_1}{\sigma_0 + \sigma_1}$$

and

$$N = n_0 + n_1, \pi_o + \pi_1 = 1, \pi_0 = \frac{n_0}{n_n + n_1}$$

$$n_0^* = n_1^* = n^* = \left(t_{\alpha/2}\sqrt{2\bar{p}(1-\bar{p})} + t_{\beta}\sqrt{p_0\left(1-p_0\right) + p_1\left(1-p_1\right)}\right)^2 \delta^{-2}$$

where $\bar{p} = \left(p_0 + p_1\right)/2$.

You can code it up by yourself, but let's just use the handy `library(stats)` as follows. Remember that the number you see is the number in each group.

In this example, you can detect the effect at least from 0.7085 with 300 participants each, the usual power and critical points.

```r
library(stats)
power.prop.test(n = 300, # subject number in each group
                p1 = 0.6, # baseline effect
                sig.level = 0.05, # type I error
                power = 0.8, # power
                alternative = "two.sided")
```

```
##
##      Two-sample comparison of proportions power calculation
##
##               n = 300
##              p1 = 0.6
##              p2 = 0.7085789
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Case Study: Application to Online Advertising

Let's do a small case study related to advertising. In class (and also in papers some of you presented), we learned that it is not an easy task to detect the positive effect of advertising. Here, let's consider that from a view point of the power analysis.

From above, we know that precisely estimating the advertising effect is not an easy task because of the following reason.

- High variance in the effect (large uncertainty)
- Small effect size
- Clustered samples with higher intra-cluster correlations

Set the baseline effect 0.001 and say the hike from advertising is 0.0005. Under the standard power and critical point, we need huge sample in each group. So, it would be a good idea to ask if the number of sample size enough when you saw someone is mentioning that they find a positive/negative effect of advertising. Even if it is statistically significant and different from zero, it might be hard to convince you because of lower power design.

```
power.prop.test(p1 = 0.001, # baseline effect
                p2 = 0.0015, # baseline effect + effect size
                sig.level = 0.05, # type I error
                power = 0.8, # power
                alternative = "two.sided")
```

```
##
##      Two-sample comparison of proportions power calculation
##
##               n = 78389.51
##              p1 = 0.001
##              p2 = 0.0015
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Power Calculation with Clustering

$$n = 2\left(t_{\alpha} + t_{1-\beta}\right)^2 * \frac{\sigma^2}{\delta^2} * (1 + (m-1)\rho)$$

where

- m: group size (class size)
- ρ: intracluster correlation coefficient (ICC) with the following definition:

$$\rho = \frac{\text{group variance}}{\text{group variance} + \text{individual variance}}$$

Again, you can code it up by yourself, but let's use the handy package `library(PowerUpR)`.

The following code returns the number of clusters you need to have to preserve the minimum detectable effect size 0.45 when there are 25 subjects in each cluster, and ICC is 0.02.

```
library(PowerUpR)
```

```
mrss.cra2r2(power = 0.80,
            alpha = 0.05, # type 1 error
            es = 0.45, # effect size
```

```
            n = 25, # Average Cluster Size (corresponds to m in the equation)
            rho2 = 0.02) # ICC (corresponds to rho in the equation)
```

```
## J = 11
```

The following code returns the minimum detectable effect size when you have 200 clusters with 20 subjects in each cluster when the ICC is 0.02.

```
mdes.cra2r2(power = 0.80,
            alpha = 0.05, # type 1 error
            J = 200, # Sample Size (# of Clusters)
            n = 25, # Average Cluster Size (corresponds to m in the equation)
            rho2 = 0.02) # ICC (corresponds to rho in the equation)
```

```
##
## Minimum detectable effect size:
## ---------------------------------------
##  0.097 95% CI [0.029,0.165]
## ---------------------------------------
## Degrees of freedom: 198
## Standardized standard error: 0.034
## Type I error rate: 0.05
## Type II error rate: 0.2
## Two-tailed test: TRUE
```

This tells you that if you have more clusters, it is better because you can have smaller minimum detectable effect size.

Simulation-based

The standard analytical formula has a limitation in the sense that it cannot be generalizable in the complex design. You can still use it once you simplify the setting. However, you may be interested in extending the canonical power analysis to fit your own experimental design. In such a case, the simulation-based power analysis would be your friend.

Ordinary case

Panel design