

Poder Estadístico

Antonella Bandiera

13 December, 2021

Efectos reales y estimados

Distribución de las estimaciones

Test de hipótesis

¿Qué es el poder estadístico?

Cálculo analítico de poder

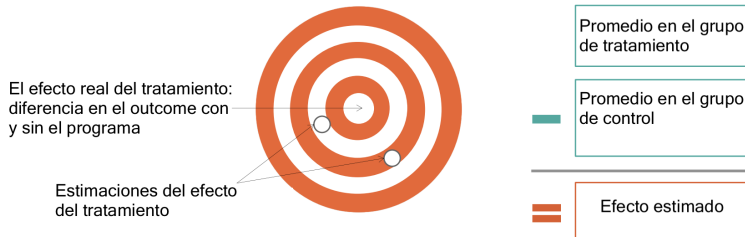
Cálculos de poder usando simulaciones

Poder con controles

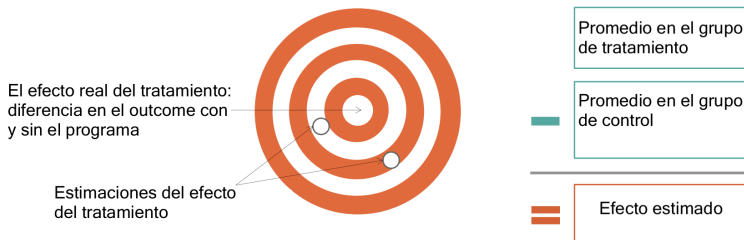
Poder para aleatorización en clusters

Efectos reales y estimados

Estimación del efecto de tratamiento con una muestra experimental



Estimación del efecto de tratamiento con una muestra experimental



- ▶ La variación aleatoria que produce el muestreo implica que el efecto estimado no siempre es igual al efecto del tratamiento. ¿Qué hacemos al respecto?

Estimación del efecto del tratamiento con una muestra experimental

Estimación del efecto del tratamiento

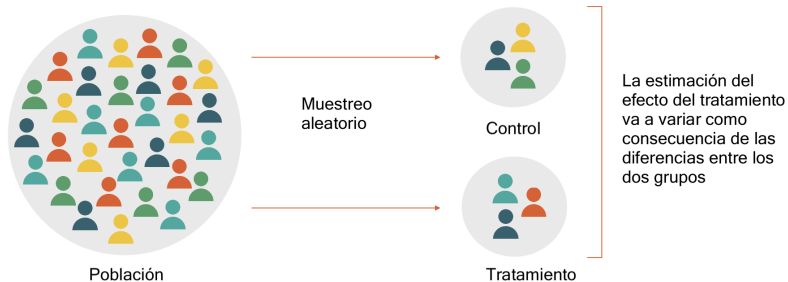
- ▶ En general usamos la diferencia promedio en los outcomes de un grupo de tratamiento y un grupo de control seleccionados aleatoriamente
- ▶ También podemos usar una regresión si vamos a incluir controles o efectos, por ejemplo de bloques
- ▶ El efecto del tratamiento solemos denominarlo $\hat{\beta}$ (donde β es el parámetro o efecto real) o τ y $\hat{\tau}$

Variación muestral y la estimación del efecto del tratamiento

- ▶ La aleatorización garantiza que el grupo de tratamiento y el grupo de control no sean *sistemáticamente* diferentes. Pero eso no implica que no lo sean:

Variación muestral y la estimación del efecto del tratamiento

- ▶ La aleatorización garantiza que el grupo de tratamiento y el grupo de control no sean *sistemáticamente* diferentes. Pero eso no implica que no lo sean:

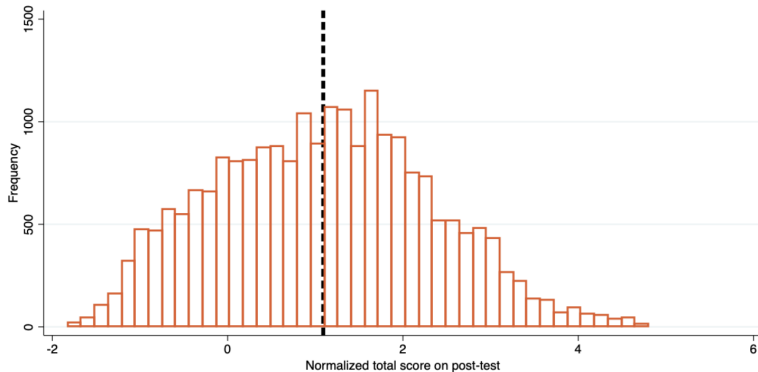


Distribución de las estimaciones

Ejemplo: Evaluación de un programa de educación en India sobre exámenes estandarizados

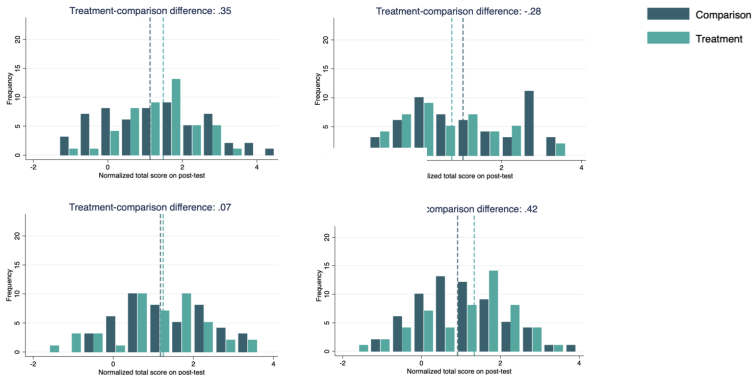
Ejemplo: Evaluación de un programa de educación en India sobre exámenes estandarizados

Una muestra de 23.000 estudiante muestra la variación que hay en en los resultados de los exámenes

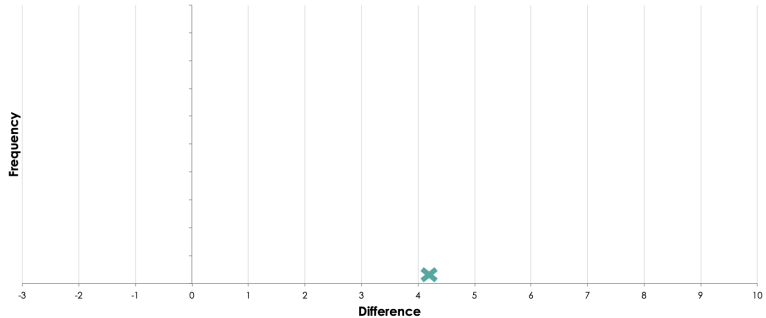


Fuente: Banerjee, A., et al. (2017)

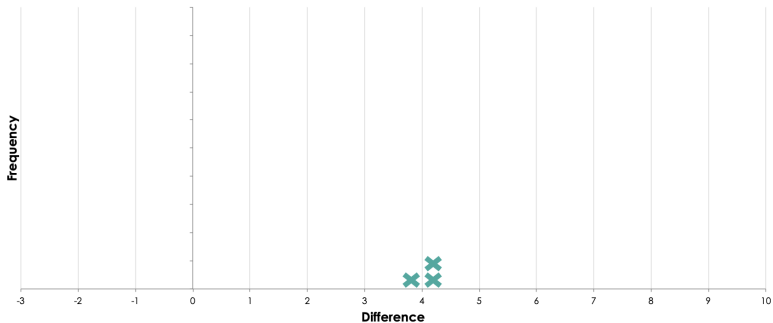
Diferentes muestras aleatorias de la misma población llevan a efectos estimados diferentes



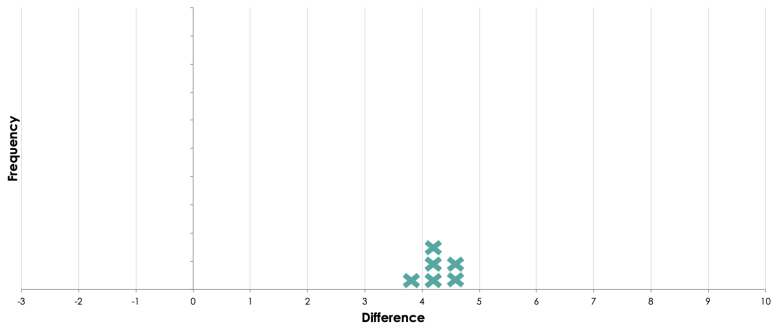
El efecto estimado de un experimento



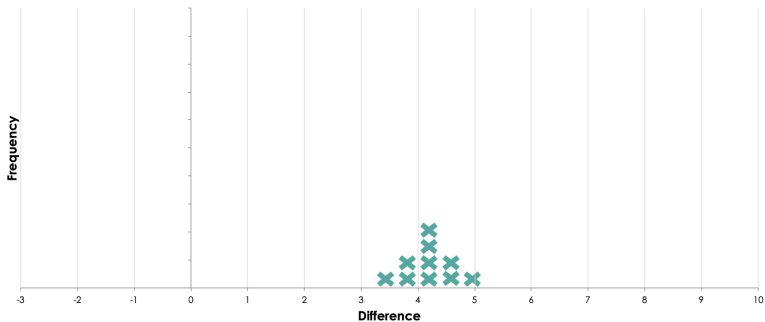
Tres experimentos, tres efectos estimados



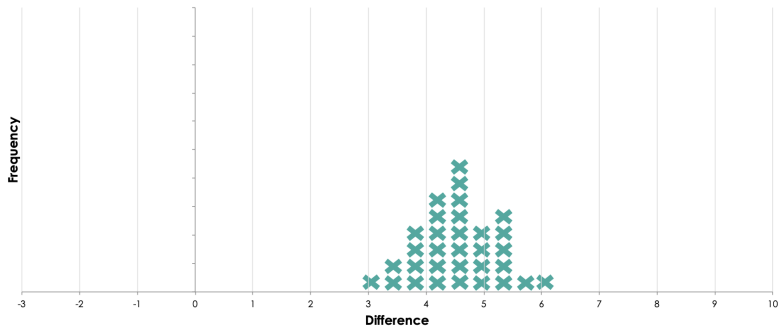
Seis experimentos, seis efectos estimados



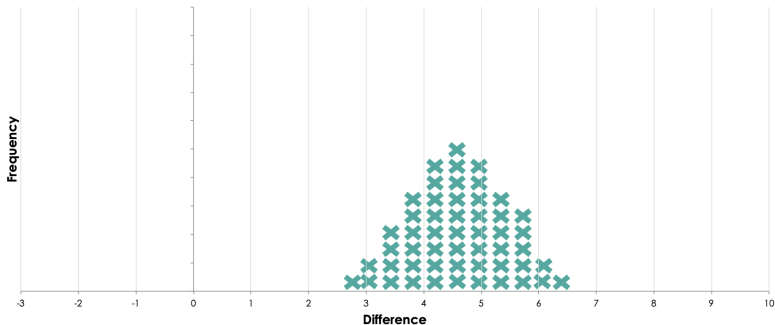
Muchos experimentos: una distribución de estimaciones



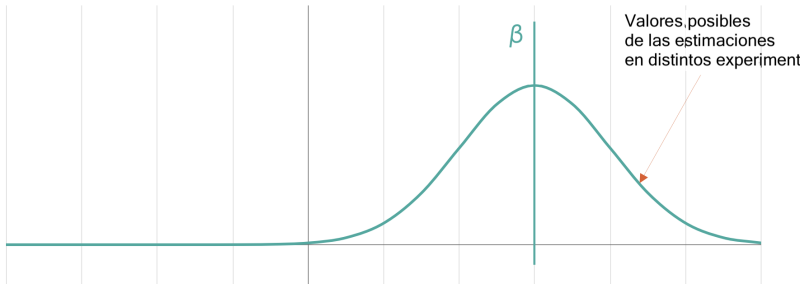
Muchos experimentos: una distribución de estimaciones



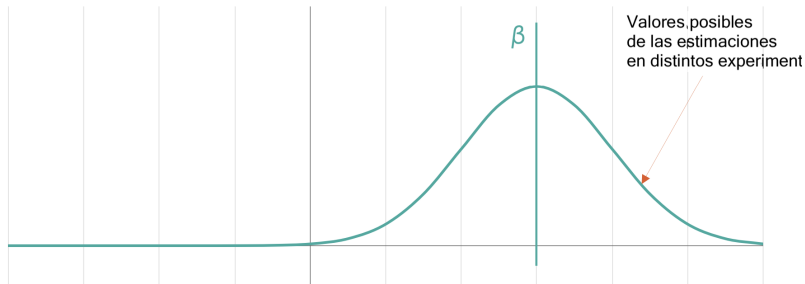
Muchos experimentos: una distribución de estimaciones



La distribución de las estimaciones si el efecto real es β



La distribución de las estimaciones si el efecto real es β



Los efectos estimados $\hat{\beta}$ tendrán una distribución normal centrada en β , si se cumplen algunas condiciones.

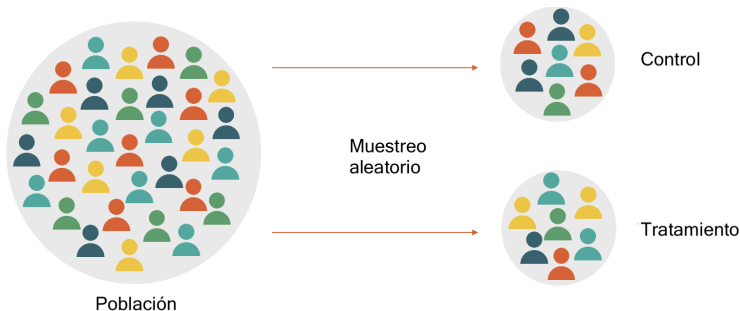
Intución: El tamaño muestral, ¿cómo afecta la distribución de las estimaciones?

Las muestras pueden ser muy diferentes una de otras.

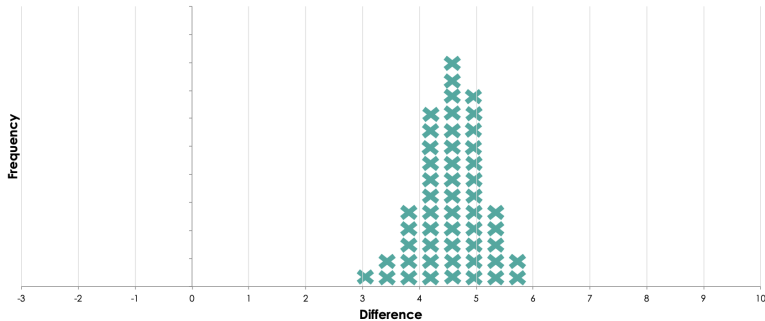


Intución: El tamaño muestral, ¿cómo afecta la distribución de las estimaciones?

Muestras más grandes tenderán a ser más representativas de la población y más parecidas unas a otras



Si el grupo de tratamiento y control son representativos y parecidos entre sí, las estimaciones varían menos



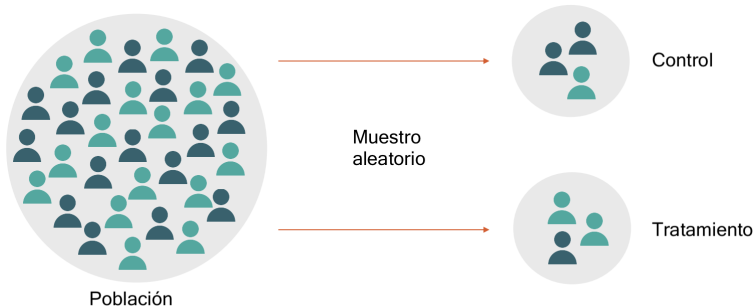
Intiución: La variación en la población, ¿cómo afecta la distribución de las estimaciones?

Muestras que vienen de una población con **mucha variación** pueden verse muy diferentes

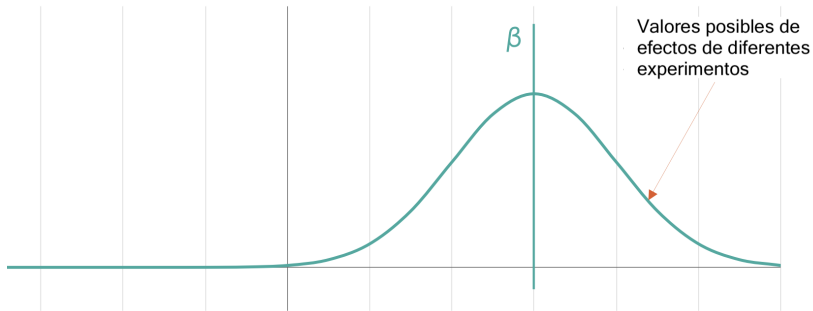


Intiución: La variación en la población, ¿cómo afecta la distribución de las estimaciones?

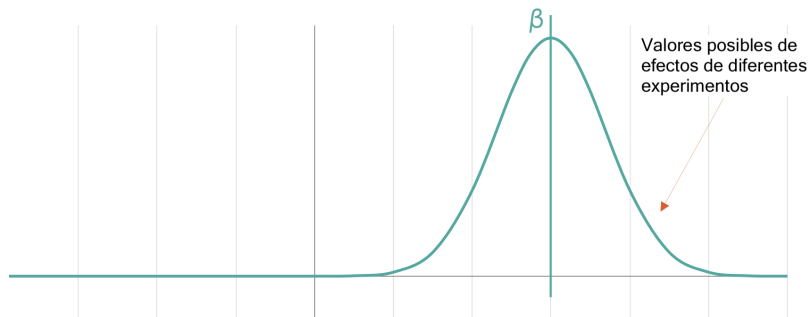
Muestras que vienen de una población que tiene **poca variación** tienden a ser más similares a la población y entre sí



Mientras más grandes y representativos sean los grupos de tratamiento y control, las estimaciones de diferentes muestras variarán menos

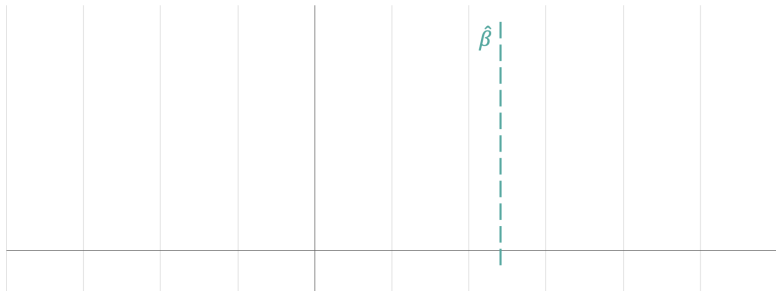


Mientras más grandes y representativos sean los grupos de tratamiento y control, las estimaciones de diferentes muestras variarán menos



Test de hipótesis

Problema: Sólo vemos una estimación con nuestro experimento, ¿qué conclusión sacamos?



Test de hipótesis

Hipótesis nula

La hipótesis nula (H_0) es la que dice que el impacto del programa es cero (i.e. $\beta = 0$)

Test de hipótesis

Hipótesis nula

La hipótesis nula (H_0) es la que dice que el impacto del programa es cero (i.e. $\beta = 0$)

- ▶ Empezamos asumiendo que el programa **no** tuvo ningún impacto

Test de hipótesis

Hipótesis nula

La hipótesis nula (H_0) es la que dice que el impacto del programa es cero (i.e. $\beta = 0$)

- ▶ Empezamos asumiendo que el programa **no** tuvo ningún impacto
- ▶ Experimento mental: **¿qué tan probable es observar un efecto tan extremo como el que observamos ($\hat{\beta}$) en un experimento, si el verdadero efecto es cero?**

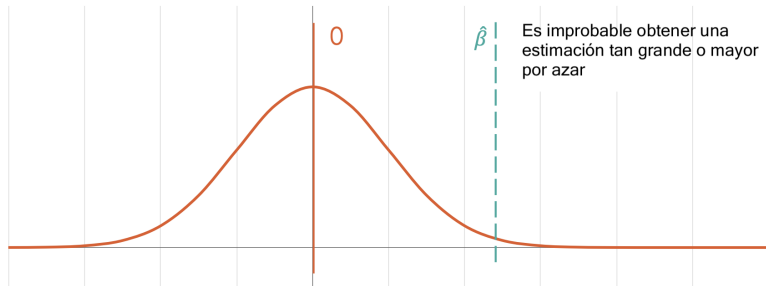
Test de hipótesis

Hipótesis nula

La hipótesis nula (H_0) es la que dice que el impacto del programa es cero (i.e. $\beta = 0$)

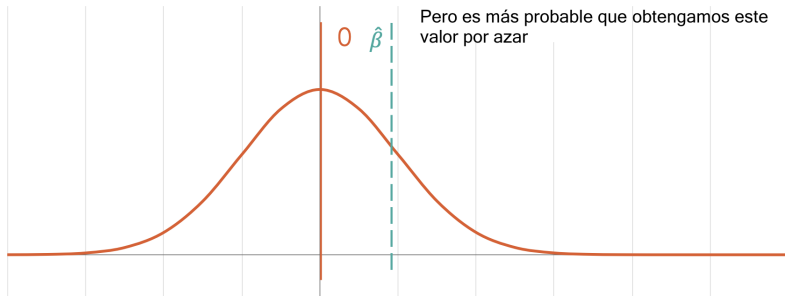
- ▶ Empezamos asumiendo que el programa **no** tuvo ningún impacto
- ▶ Experimento mental: **¿qué tan probable es observar un efecto tan extremo como el que observamos ($\hat{\beta}$) en un experimento, si el verdadero efecto es cero?**
- ▶ Si es muy improbable podemos rechazar la hipótesis nula

Distribución de las estimaciones si el efecto fuera cero



Asumimos un efecto $\beta = 0$

Distribución de las estimaciones si el efecto fuera cero



Asumimos un efecto $\beta = 0$

Definición

Error Tipo I (falso positivo)

La probabilidad de concluir que el efecto del tratamiento es distinto de cero. O la probabilidad de rechazar la hipótesis nula $\beta = 0$, cuando la nula es cierta.

Nivel de significancia

La probabilidad máxima de error Tipo I que estamos dispuestos a aceptar. 5

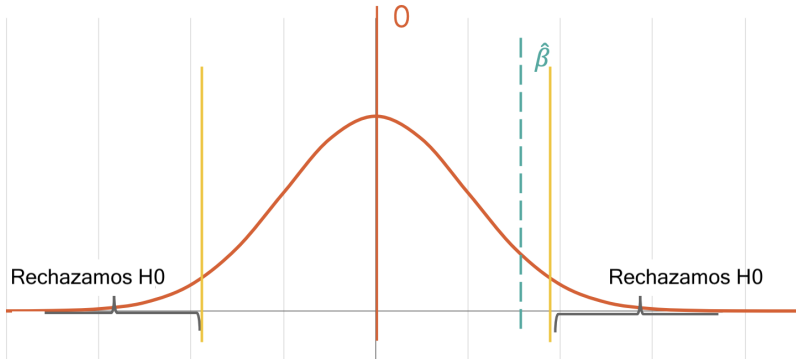
- ▶ Siempre tenemos la posibilidad de cometer un error Tipo I.
- ▶ Decimos que $\hat{\beta}$ es estadísticamente diferente de cero a un nivel de 5%

¿Cuándo $\hat{\beta}$ es estadísticamente distinto de cero al nivel de 5%?

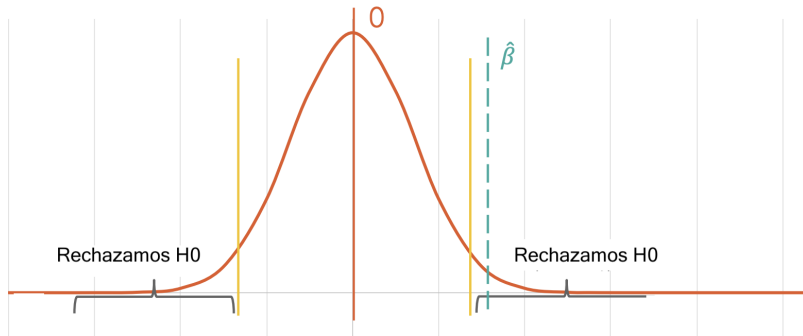


Sólo cuando el estadístico t de $\hat{\beta}$ es lo suficientemente grande o chica podemos rechazar la hipótesis nula de ausencia de efecto.

Un aumento en el tamaño de la muestra



Un aumento en el tamaño de la muestra



Test de hipótesis: Conclusiones

- ▶ Efectos más grandes de $\hat{\beta}$ son menos probables cuando el verdadero efecto es cero.
- ▶ Una muestra más grande aumenta mecánicamente la precisión de nuestras estimaciones: tenemos menos varianza muestral
- ▶ Si es muy improbable que la diferencia que encontramos entre T y C sea por azar, decimos que rechazamos la hipótesis nula.

¿Qué es el poder estadístico?

¿Qué es el poder estadístico?

¿Qué es el poder estadístico?

Error Tipo II

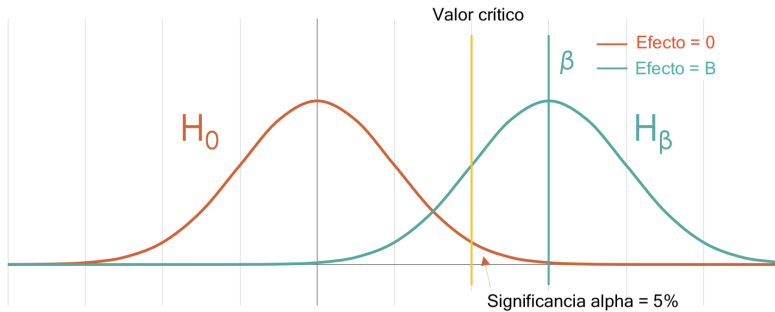
La probabilidad de concluir que no hay un efecto del tratamiento.
La probabilidad de no rechazar la hipótesis nula, cuando la hipótesis nula es falsa.

Poder estadístico

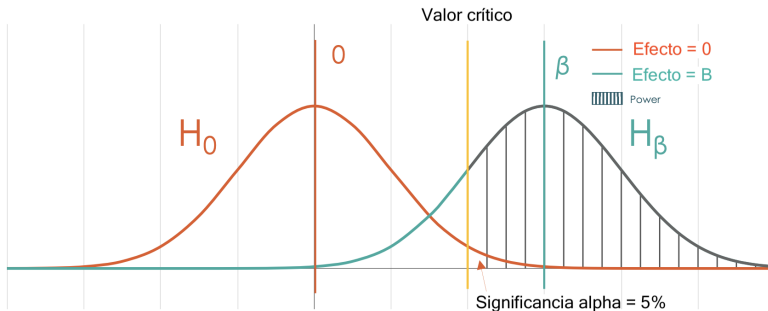
La probabilidad de un positivo correcto. O la probabilidad de evitar un error Tipo II.

- ▶ En otras palabras, el poder estadístico es la capacidad de detectar un efecto, dado que ese efecto existe.
- ▶ Formalmente: $1 - \text{Tasa de error Tipo II}$
- ▶ Entonces, $\text{poder} \in (0, 1)$.
- ▶ Niveles estándar: 0.8 o 0.9.

Si el efecto real fuera β ...



¿qué tan frecuentemente rechazaríamos la hipótesis nula?

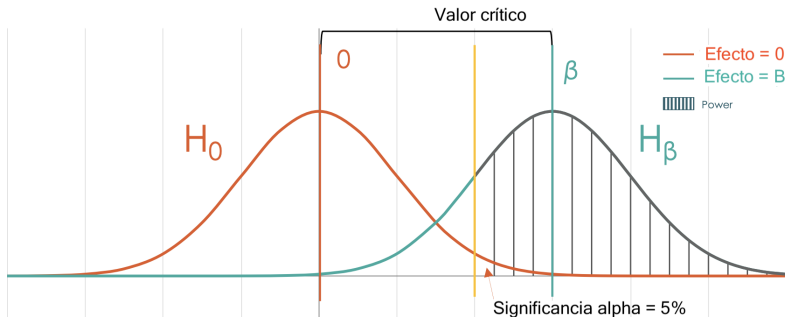


El área sombreada muestra el poder: la probabilidad de obtener una estimación que nos lleve a rechazar la hipótesis nula H_0 cuando el efecto es β

El poder es más bajo cuando tenemos un efecto (real) más pequeño



El poder es mayor con un efecto (real) más grande



El punto de partida

- ▶ Un análisis de poder es algo que hay que hacer **siempre, antes** de un estudio
 - ▶ Nos ayuda a saber el tamaño de muestra que necesitamos
 - ▶ O nos permite conocer el efecto mínimo detectable dado un tamaño muestral
 - ▶ Nos puede ayudar a decidir si vale la pena realizar una intervención
 - ▶ A veces sirve hacerlo ex-post: nos permite saber si teníamos poder para detectar un efecto
- ▶ No aprendemos nada de un efecto nulo que no tiene suficiente poder
 - ▶ Hay un efecto, pero no lo detectamos? O no hay efecto? No hay manera de saber.

Poder

- ▶ Imaginemos que sí hay un efecto del tratamiento en la población y corremos un experimento muchas veces. ¿Qué tan frecuentemente vamos a obtener un resultado significativo?
- ▶ Los elementos del poder estadístico
 - ▶ Magnitud del efecto
 - ▶ Varianza del outcome
 - ▶ Tamaño de la muestra
 - ▶ Balance en el número de unidades en tratamiento y control

Poder y efecto mínimo detectable

- ▶ Los cálculos de poder se pueden usar de tres maneras: a) para estimar el tamaño muestral, dado un nivel de poder y un efecto (también llamado efecto mínimo detectable), b) para calcular el poder, dado un tamaño muestral y un efecto mínimo detectable, c) o para computar un efecto mínimo detectable, dado el poder y el tamaño muestral
- ▶ Calcular el efecto mínimo detectable, dado un nivel de poder y un tamaño muestral nos puede ayudar a entender si vale la pena realizar una intervención
- ▶ Muchas veces no vamos a saber cuál es el efecto, tenemos que usar otras intervenciones, el sentido común
 - ▶ Si no, en términos de desviaciones estándar, 0,2 es un efecto pequeño, 0,5 un efecto moderado y 0,8 un efecto grande.

Enfoques para calcular poder

- ▶ Cálculo analítico de poder
- ▶ Simulaciones

Herramientas para calcular poder

- ▶ Interactivas
 - ▶ EGAP Power Calculator
 - ▶ rpsychologist
- ▶ Paquetes de R
 - ▶ pwr
 - ▶ DeclareDesign, mirar también <https://declaredesign.org/>

Cálculo analítico de poder

Cálculo analítico de poder

- ▶ Fórmula:

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

- ▶ Componentes:

- ▶ ϕ : CDF de la distribución normal estándar
- ▶ τ : magnitud del efecto
- ▶ N : tamaño de la muestra
- ▶ σ : desviación estándar del outcome
- ▶ α : nivel de significancia (típicamente 0.05)

Ejemplo: Cálculo analítico

```
# Análisis de poder para un estudio con 80 observaciones y  
# efecto de tamaño 0.25  
library(pwr)  
pwr.t.test(n = 40, d = 0.25, sig.level = 0.05,  
           power = NULL)
```

```
##  
##      Two-sample t test power calculation  
##  
##              n = 40  
##              d = 0.25  
##      sig.level = 0.05  
##      power = 0.1971831  
##      alternative = two.sided  
##  
## NOTE: n is number in *each* group
```

Ejemplo: Cálculo analítico

```
# Análisis de poder para un estudio con 250 observaciones y  
# efecto de tamaño 0.25  
library(pwr)  
pwr.t.test(n = 150, d = 0.25, sig.level = 0.05,  
           power = NULL)
```

```
##  
##      Two-sample t test power calculation  
##  
##              n = 150  
##              d = 0.25  
##      sig.level = 0.05  
##      power = 0.578543  
##      alternative = two.sided  
##  
## NOTE: n is number in *each* group
```


Ejemplo: Cálculo analítico

```
# Estudio de tamaño muestral dado el poder y efecto  
# de tamaño 0.25  
library(pwr)  
pwr.t.test(n = NULL, d = 0.25, sig.level = 0.05,  
           power = 0.8)
```

```
##  
##      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
```

Limitaciones del cálculo analítico de poder

- ▶ Sólo derivado para algunos estadísticos (diferencia de medias)
- ▶ Hace supuestos específicos sobre el proceso de generación de datos
- ▶ Incompatible con diseños complejos

Cálculos de poder usando simulaciones

Cálculos de poder usando simulaciones

- ▶ Crear datos y simular el diseño
- ▶ Se necesitan supuestos, pero los podemos elegir
- ▶ DeclareDesign hace todo más fácil, mirar <https://declaredesign.org/>

Pasos

- ▶ Definir la muestra y la función de outcomes
- ▶ Definir el tratamiento y el proceso de asignación
- ▶ Crear los datos
- ▶ Asignar el tratamiento, luego estimar los efectos
- ▶ Hacer esto muchas veces

Ejemplos

- ▶ Aleatorización completa
- ▶ Con controles
- ▶ Blocking
- ▶ Con aleatorización en clusters

Ejemplo: Power con simulaciones para aleatorización completa

```
# install.packages("randomizr")
library(randomizr)
library(estimatr)

## Y0 va fijo
## Solo lo generamos una vez:
make_Y0 <- function(N){ rnorm(n = N) }
repeat_experiment_and_test <- function(N, Y0, tau){
  Y1 <- Y0 + tau
  Z <- complete_ra(N = N)
  Yobs <- Z * Y1 + (1 - Z) * Y0
  estimator <- lm_robust(Yobs ~ Z)
  pval <- estimator$p.value[2]
  return(pval)
}
```

Ejemplo: Power con simulaciones para aleatorización completa

```
power_sim <- function(N,tau,sims){  
  Y0 <- make_Y0(N)  
  pvals <- replicate(n=sims,  
    repeat_experiment_and_test(N=N,Y0=Y0,tau=tau))  
  pow <- sum(pvals < .05) / sims  
  return(pow)  
}
```

```
set.seed(12345)  
power_sim(N=80,tau=.25,sims=100)
```

```
## [1] 0.09
```

```
#power_sim(N=80,tau=.25,sims=100)
```


Ejemplo: Usando DeclareDesign

```
library(DeclareDesign)
library(tidyverse)
my_design <- declare_model(N = 100,
  U = rnorm(N),
  Y_Z_0 = U,
  Y_Z_1 = U + rnorm(N, mean = 2, sd = 2)) +
  declare_assignment(Z = complete_ra(N, prob = 0.5)) +
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(Y ~ Z, inquiry = "ATE")
diagnose_design(my_design)
```

```
##
```

```
## Research design diagnosis based on 500 simulations. Diagnosed estimates with bootstrapped standard error
```

```
##
```

```
##      Design Inquiry Estimator Term N Sims   Bias   RMSE   Power Coverage
## my_design      ATE estimator    Z   500   0.01   0.29   1.00    0.98
##                                     (0.01) (0.01) (0.00)  (0.01)
```

```
## Mean Estimate SD Estimate Mean Se Type S Rate Mean Estimand
##           2.02      0.34    0.35      0.00           2.01
##           (0.01)      (0.01) (0.00)      (0.00)           (0.01)
```

Ejemplo: Usando DeclareDesign II

```
my_design2 <- declare_model(N = 100,  
  U = rnorm(N),  
  Y_Z_0 = U,  
  Y_Z_1 = U + rnorm(N, mean = 1, sd = 3)) +  
  declare_assignment(Z = complete_ra(N, prob = 0.5)) +  
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +  
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +  
  declare_estimator(Y ~ Z, inquiry = "ATE")  
diagnose_design(my_design2)
```

```
##
```

```
## Research design diagnosis based on 500 simulations. Diagnosed estimates with bootstrapped standard error
```

```
##
```

```
##      Design Inquiry Estimator Term N Sims   Bias   RMSE   Power Coverage  
## my_design2      ATE estimator      Z   500  0.02  0.36  0.58    0.99  
##                                (0.02) (0.01) (0.02)  (0.01)
```

```
## Mean Estimate SD Estimate Mean Se Type S Rate Mean Estimand  
##      1.03      0.46  0.47      0.00      1.01  
##      (0.02)      (0.02) (0.00)      (0.00)      (0.01)
```

Ejemplo: Usando DeclareDesign II

```
my_design3 <- declare_model(N = 100,  
  U = rnorm(N),  
  Y_Z_0 = U,  
  Y_Z_1 = U + rnorm(N, mean = 1, sd = 3)) +  
  declare_assignment(Z = complete_ra(N, prob = 0.5)) +  
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +  
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +  
  declare_estimator(Y ~ Z, model = lm_robust, inquiry = "ATE")  
diagnose_design(my_design3)
```

```
##
```

```
## Research design diagnosis based on 500 simulations. Diagnoses and estimates with bootstrapped standard errors
```

```
##
```

##	Design	Inquiry	Estimator	Term	N	Sims	Bias	RMSE	Power	Coverage
##	my_design3	ATE	estimator	Z	500	-0.02	0.34	0.53	0.99	
##						(0.01)	(0.01)	(0.02)	(0.00)	

##	Mean Estimate	SD	Estimate	Mean	Se	Type	S	Rate	Mean	Estimand
##	0.96		0.45	0.47			0.00		0.99	
##	(0.02)		(0.01)	(0.00)			(0.00)		(0.01)	

Poder con controles

Controles y poder

- ▶ Incluir controles puede aumentar el poder porque ayuda a reducir la variación en los outcomes
 - ▶ Si los controles predicen el outcome, la reducción de la variación es dramática. Recordemos la fórmula: menor varianza implica más poder
 - ▶ Si no predicen el outcome, la mejora será mínima
- ▶ **Todos los controles deben ser pre-tratamiento.** Nunca tirar observaciones por falta de datos. Imputar media/mediana o multiple imputation.

Blocking

- ▶ Blocking: asignar el tratamiento de manera aleatoria dentro de bloques
 - ▶ Es como incluir controles “ex-ante”
 - ▶ Mayor precisión/eficiencia implica más poder
 - ▶ Beneficios de bloquear sobre incluir controles son evidentes sobre todo en experimento pequeños

Ejemplo: poder con simulación incluyendo un control I

```
## YO va fijo
make_Y0_cov <- function(N){
  u0 <- rnorm(n = N)
  x <- rpois(n=N,lambda=2)
  Y0 <- .5 * sd(u0) * x + u0
  return(data.frame(Y0=Y0,x=x))
}
## X predice YO.
test_dat <- make_Y0_cov(100)
test_lm <- lm_robust(Y0~x,data=test_dat)
summary(test_lm)
```

```
##
## Call:
## lm_robust(formula = Y0 ~ x, data = test_dat)
##
## Standard error type: HC2
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
## (Intercept)  -0.2810    0.15581  -1.803 7.440e-02  -0.5902  0.02821 98
## x              0.5925    0.06346   9.336 3.336e-15   0.4665  0.71838 98
##
## Multiple R-squared:  0.401 , Adjusted R-squared:  0.3949
## F-statistic: 87.17 on 1 and 98 DF, p-value: 3.336e-15
```

Ejemplo: poder con simulación incluyendo un control II

```
## la simulación
repeat_experiment_and_test_cov <- function(N, tau, Y0, x){
  Y1 <- Y0 + tau
  Z <- complete_ra(N = N)
  Yobs <- Z * Y1 + (1 - Z) * Y0
  estimator <- lm_robust(Yobs ~ Z+x,data=data.frame(Y0,Z,x))
  pval <- estimator$p.value[2]
  return(pval)
}

## crear los datos una vez, asignar de manera aleatoria el tratamiento
## reportar la proporción que devuelve p-valor < 0.05
power_sim_cov <- function(N,tau,sims){
  dat <- make_Y0_cov(N)
  pvals <- replicate(n=sims, repeat_experiment_and_test_cov(N=N,
    tau=tau,Y0=dat$Y0,x=dat$x))
  pow <- sum(pvals < .05) / sims
  return(pow)
}
```

```
set.seed(12345)
power_sim_cov(N=80,tau=.25,sims=100)
```

```
## [1] 0.13
```

```
power_sim_cov(N=80,tau=.25,sims=100)
```

```
## [1] 0.21
```


Ejemplo: poder con blocking en DeclareDesign I

```
N = 100
```

```
model <- declare_model(  
  N = N,  
  X = sort(rnorm(N)),  
  u = rnorm(N),  
  potential_outcomes(Y ~ Z - X * Z + u/10)  
)  
inquiry <- declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0))  
blocking <- declare_step(fabricate, couples = rep(1:(N/2), each = 2))  
assignment <- declare_assignment(Z = block_ra(blocks = couples))  
measurement <- declare_measurement(Y = reveal_outcomes(Y ~ Z))  
estimator <- declare_estimator(Y ~ Z)  
  
design_likes <- model + inquiry +  
  blocking + assignment + measurement + estimator  
diagnose_design(design_likes)
```

```
##
```

```
## Research design diagnosis based on 500 simulations. Diagnoses and estimates with bootstrapped standard errors
```

```
##
```

##	Design	Inquiry	Estimator	Term	N	Sims	Bias	RMSE	Power	Coverage
##	design_likes	ATE	estimator	Z	500	0.00	0.02	1.00	1.00	
##						(0.00)	(0.00)	(0.00)	(0.00)	
##	Mean Estimate	SD	Estimate	Mean	Se	Type	S	Rate	Mean	Estimand
##	1.00	0.10	0.14	0.00	1.00					
##	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)				(0.00)	

Poder para aleatorización en clusters

Poder para aleatorización en clusters

- ▶ Dado un N fijo, un diseño con clusters está débilmente menos powered que uno sin clusters
 - ▶ Pero generalmente la diferencia es muy grande
- ▶ Hay que estimar la variación de manera correcta:
 - ▶ Clustering para los errores estándar
 - ▶ Inferencia
- ▶ Para aumentar el poder:
 - ▶ Mejor aumentar el número de clusters que el número de unidades dentro del cluster
 - ▶ En qué medida los clusters reducen el poder depende de la correlación dentro de los clusters (intra-cluster correlation, icc, o la razón entre la varianza dentro del cluster respecto de la varianza total)

Ejemplo: Poder con simulación en un diseño con clusters I

```
## YO va fijo
make_Y0_clus <- function(n_indivs,n_clus){
  # n_indivs numero de individuos en un cluster
  # n_clus numero de clusters
  clus_id <- gl(n_clus,n_indivs)
  N <- n_clus * n_indivs
  u0 <- fabricatr::draw_normal_icc(N=N,clusters=clus_id,ICC=.1)
  Y0 <- u0
  return(data.frame(Y0=Y0,clus_id=clus_id))
}

test_dat <- make_Y0_clus(n_indivs=10,n_clus=100)
# confirmar que esto produce data con 10 en cada uno de los 100 clusters
table(test_dat$clus_id)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
## 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
## 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
## 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
## 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
## 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
## 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
## 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
## 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
## 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
```

```
# confirmar el icc
ICC::ICCbare(y=Y0,x=clus_id,data=test_dat)
```

Ejemplo: Poder con simulación en un diseño con clusters II

```
## [1] 0.08422507
```

```
repeat_experiment_and_test_clus <- function(N, tau, Y0, clus_id){  
  Y1 <- Y0 + tau  
  # randomizamos z al nivel del cluster  
  Z <- cluster_ra(clusters=clus_id)  
  Yobs <- Z * Y1 + (1 - Z) * Y0  
  estimator <- lm_robust(Yobs ~ Z, clusters = clus_id,  
                        data=data.frame(Y0,Z,clus_id), se_type = "CR2")  
  pval <- estimator$p.value[2]  
  return(pval)  
}  
  
power_sim_clus <- function(n_indivs,n_clus,tau,sims){  
  dat <- make_Y0_clus(n_indivs,n_clus)  
  N <- n_indivs * n_clus  
  # aleatorizar el tratamiento sims veces  
  pvals <- replicate(n=sims,  
                    repeat_experiment_and_test_clus(N=N,tau=tau,  
                                                    Y0=dat$Y0,clus_id=dat$clus_id))  
  pow <- sum(pvals < .05) / sims  
  return(pow)  
}
```

```
set.seed(12345)  
power_sim_clus(n_indivs=8,n_clus=100,tau=.25,sims=100)
```

```
## [1] 0.69
```

```
power_sim_clus(n_indivs=8,n_clus=100,tau=.25,sims=100)
```

Ejemplo: Poder con simulación en un diseño con clusters III

```
## [1] 0.74
```

Implicaciones

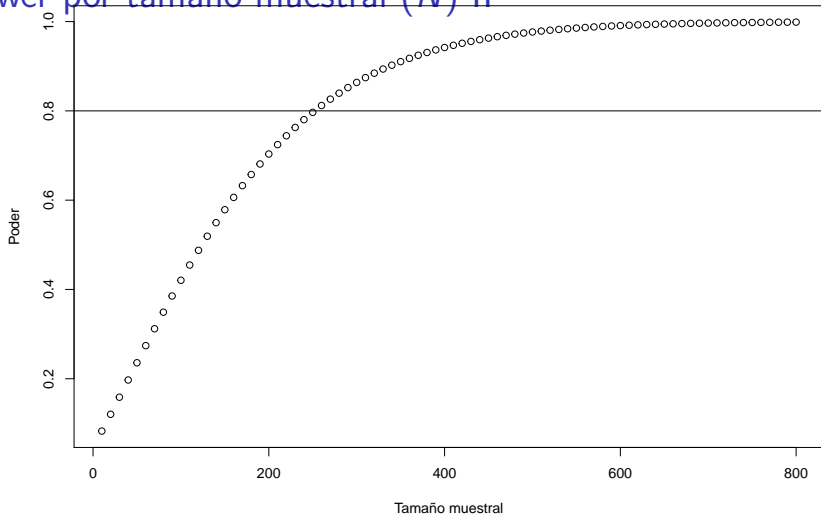
Implicaciones

- ▶ El poder es:
 - ▶ Creciente en N
 - ▶ Creciente en $|\tau|$ (antes $|\beta|$)
 - ▶ Decreciente en σ

Power por tamaño muestral (N) I

```
some_ns <- seq(10,800,by=10)
pow_by_n <- sapply(some_ns, function(then){
  pwr.t.test(n = then, d = 0.25, sig.level = 0.05)$power
})
plot(some_ns,pow_by_n,
     xlab="Tamaño muestral",
     ylab="Poder")
abline(h=.8)
```

Power por tamaño muestral (N) II

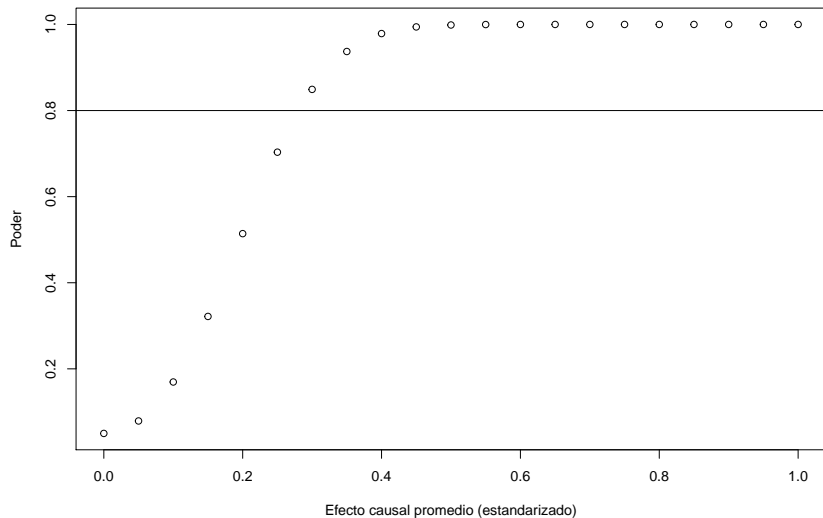


```
## See https://cran.r-project.org/web/packages/pwr/vignettes/pwr-vignette.html  
## for fancier plots  
## ptest <- pwr.t.test(n = NULL, d = 0.25, sig.level = 0.05, power = .8)  
## plot(ptest)
```

Power por magnitud del efecto (τ o β) I

```
some_taus <- seq(0,1,by=.05)
pow_by_tau <- sapply(some_taus, function(thetau){
  pwr.t.test(n = 200, d = thetau, sig.level = 0.05)$power
})
plot(some_taus,pow_by_tau,
     xlab="Efecto causal promedio (estandarizado)",
     ylab="Poder")
abline(h=.8)
```

Power por magnitud del efecto (τ o β) II



Comentarios

- ▶ Conocer el outcome
- ▶ Tratar de establecer efectos realistas
- ▶ Conocer el rango de variación del outcome
 - ▶ Un diseño en el que la variable dependiente no se mueve mucho, posiblemente va a estar underpowered

Conclusiones: ¿Cómo aumentar el poder? I

1. Aumentar el N
 - ▶ Si se usan cluster, aumentar el número de clusters todo lo posible
 - ▶ Si la participación va a ser baja, aumentar el tamaño muestral
2. Fortalecer el tratamiento
3. Mejorar precisión
 - ▶ Controles pre-tratamiento
 - ▶ Blocking
4. Mediciones precisas del outcome
5. Mantener balance entre tratamiento y control