# Aleatorización en R

Lily Medina
University of California, Berkeley

23-09-2021

# Aleatorización

# Buenas prácticas

# Aleatorización

# Repaso

- El problema fundamental de la inferencia causal:

# Repaso

- El problema fundamental de la inferencia causal:
  - No podemos medir el efecto causal a nivel individual, porque no podemos observar $Y_i(d_i = 1)$ y $Y_i(d_i = 0)$ al mismo tiempo.

# Repaso

- Pero podemos estimar el efecto causal promedio (ATE, por sus siglas en inglés):

# Repaso

- Pero podemos estimar el efecto causal promedio (ATE, por sus siglas en inglés):
  - $ATE = \mathrm{E}[Yi(1) - Yi(0)] = \mathrm{E}[Yi(1)] - \mathrm{E}[Yi(0)]$

# Repaso

- Pero podemos estimar el efecto causal promedio (ATE, por sus siglas en inglés):
  - $ATE = \mathrm{E}[Yi(1) - Yi(0)] = \mathrm{E}[Yi(1)] - \mathrm{E}[Yi(0)]$
- Necesitamos hacer dos supuestos para poder estimar el ATE:

# Repaso

- Pero podemos estimar el efecto causal promedio (ATE, por sus siglas en inglés):
  - $ATE = \mathrm{E}[Yi(1) - Yi(0)] = \mathrm{E}[Yi(1)] - \mathrm{E}[Yi(0)]$
- Necesitamos hacer dos supuestos para poder estimar el ATE:
  - $0 < \Pr(d_i = 1) < 1$

# Repaso

- Pero podemos estimar el efecto causal promedio (ATE, por sus siglas en inglés):
  - $ATE = \mathrm{E}[Yi(1) - Yi(0)] = \mathrm{E}[Yi(1)] - \mathrm{E}[Yi(0)]$
- Necesitamos hacer dos supuestos para poder estimar el ATE:
  - $0 < \Pr(d_i = 1) < 1$
  - $[Y_i(1), Y_i(0)] \perp d_i$

# Repaso

- Pero podemos estimar el efecto causal promedio (ATE, por sus siglas en inglés):
  - $ATE = \mathrm{E}[Yi(1) - Yi(0)] = \mathrm{E}[Yi(1)] - \mathrm{E}[Yi(0)]$
- Necesitamos hacer dos supuestos para poder estimar el ATE:
  - $0 < \Pr(d_i = 1) < 1$
  - $[Y_i(1), Y_i(0)] \perp d_i$
- La asignación aleatoria al tratamiento hace que estos dos supestos se cumplan

# Ejemplos de asignación aleatoria

Asignación aleatoria simple

Asignación aleatoria completa

Asignación aleatoria en bloques

Asignación aleatoria por conglomerados

# Aleatorización simple (lanzamiento de moneda)

- Para cada unidad, se “lanza una moneda” para ver si esta será tratada. Luego se miden las variable de resultados distinguiendo entre los valores de la moneda.

# Aleatorización simple (lanzamiento de moneda)

- Para cada unidad, se "lanza una moneda" para ver si esta será tratada. Luego se miden las variable de resultados distinguiendo entre los valores de la moneda.
- Las monedas no tienen que estar equilibradas (50-50), pero se debe conocer la probabilidad de asignación al tratamiento.

# Aleatorización simple (lanzamiento de moneda)

- Para cada unidad, se "lanza una moneda" para ver si esta será tratada. Luego se miden las variable de resultados distinguiendo entre los valores de la moneda.
- Las monedas no tienen que estar equilibradas (50-50), pero se debe conocer la probabilidad de asignación al tratamiento.
- No se puede garantizar un número específico de unidades tratadas y unidades de control.

# Ejemplo: aleatorización simple I

```r
# Definir un número de semilla aleatoria  para asegurarse
# que el código es replicable
set.seed(12345)

# Definir un tamaño de muestra
N <- 200

# Crear la asignación aleatoria simple
# (Tengan en cuenta que en un experimento se
# tiene solo un intento y por lo tanto size=1)
# Llamamos simple.ra a nuestro objeto
# con N personas en total
simple.ra <- rbinom(n = N, size = 1, prob = .5)

# 112 personas fueron asignadas al tratamiento
sum(simple.ra)
```

```
[1] 112
```

# Ejemplo: aleatorización simple I

```r
set.seed(12345)
N <- 200

# Creemos una base de datos artificial
datos <- data.frame(
  id = 1:N,
  edad = sample(20:50, size = N, replace = TRUE),
  genero = sample(0:1, size = N, replace = TRUE)
)

# Aquí sucede la asignación:
datos$d <- rbinom(n = N, size = 1, prob = .5)

# veámos las primeras filas de nuestros datos
head(datos)
```

```
  id edad genero d
1  1   33      1 0
2  2   38      0 1
3  3   35      0 0
4  4   45      0 1
5  5   47      0 1
6  6   43      0 0
```

# Ejemplo: Aleatorización simple II

```
# También pueden utilizar el paqueter randomizr
# install.packages(randomizr)
library(randomizr)

# Para que sea replicable
set.seed(23456)
# Para hacer aleatorización simple
# utilizamos  la función simple_ra
# Nuestro objecto con N personas en total
# se llama treatment
treatment <- simple_ra(
  N    = N, # total sample size
  prob = 0.5 # probability of receiving treatment
)
head(treatment)
```

```
[1] 0 1 1 1 1 0
```

```
sum(treatment)
```

```
[1] 96
```

# Aleatorización completa (sacando bolas de una urna)

- Se asigna al tratamiento un número fijo $m$ de $N$ unidades.

# Aleatorización completa (sacando bolas de una urna)

- Se asigna al tratamiento un número fijo $m$ de $N$ unidades.
- La probabilidad de que se asigne una unidad al tratamiento es $m/N$.

# Aleatorización completa (sacando bolas de una urna)

- Se asigna al tratamiento un número fijo $m$ de $N$ unidades.
- La probabilidad de que se asigne una unidad al tratamiento es $m/N$.
- Esto es como tener una urna o tazón con $N$ bolas de las cuales $m$ se marcan como tratamiento y $N - m$ como control. Las loterías públicas utilizan este método.

# Ejemplo: Aleatorización completa I

```r
# Defina el tamaño de la muestra  N
N <- 200
# Defina la cantidad de unidades trataatadas m
m <- 100

# Cree un vector de m 1's y N-m 0's
complete.ra <- c(rep(1, m), rep(0, N - m))

# y después reorganícelo utilizando sample()
# Por defecto la función hace muestreo sin reemplazo

set.seed(12345) # para que sea replicable
complete.ra <- sample(complete.ra)

sum(complete.ra)
```

```
[1] 100
```

# Ejemplo: Aleatorización completa II

```r
# También puede usar el paquete randomizr
library(randomizr)

# para replicar
set.seed(23456)

# Asignación utilizando aleatorización completa
treatment <- complete_ra(
  N = 200, # tamaño de la muestra
  m = 100
) # No. de unidades asignadas al
# tratamiento

sum(treatment)
```

```
[1] 100
```

# Aleatorización en bloques (o estratificada) I

- Creamos bloques de unidades y seleccionamos unidades al azar dentro de cada bloque por separado. Es como si hicieramos mini-experimentos en cada bloque.

# Aleatorización en bloques (o estratificada) I

- Creamos bloques de unidades y seleccionamos unidades al azar dentro de cada bloque por separado. Es como si hicieramos mini-experimentos en cada bloque.
  - Ejemplo: bloque = distrito, unidades = comunidades. Aleatorizamos el tratamiento al nivel de la comunidad **dentro de un distrito** y también medimos nuestras variables de resultado al nivel de la comunidad.

# Aleatorización en bloques (o estratificada) I

- Creamos bloques de unidades y seleccionamos unidades al azar dentro de cada bloque por separado. Es como si hicieramos mini-experimentos en cada bloque.
  - Ejemplo: bloque = distrito, unidades = comunidades. Aleatorizamos el tratamiento al nivel de la comunidad **dentro de un distrito** y también medimos nuestras variables de resultado al nivel de la comunidad.
- Los bloques que representan un subgrupo sustancialmente significativo pueden ayudarnos a entender cómo los efectos pueden diferir por subgrupo.

# Aleatorización en bloques (o estratificada) I

- Creamos bloques de unidades y seleccionamos unidades al azar dentro de cada bloque por separado. Es como si hicieramos mini-experimentos en cada bloque.
  - Ejemplo: bloque = distrito, unidades = comunidades. Aleatorizamos el tratamiento al nivel de la comunidad **dentro de un distrito** y también medimos nuestras variables de resultado al nivel de la comunidad.
- Los bloques que representan un subgrupo sustancialmente significativo pueden ayudarnos a entender cómo los efectos pueden diferir por subgrupo.
  - Al controlar el número de sujetos por subgrupo nos aseguramos de tener suficientes sujetos en cada grupo.

# Aleatorización en bloques (o estratificada) I

- Creamos bloques de unidades y seleccionamos unidades al azar dentro de cada bloque por separado. Es como si hicieramos mini-experimentos en cada bloque.
  - Ejemplo: bloque = distrito, unidades = comunidades. Aleatorizamos el tratamiento al nivel de la comunidad **dentro de un distrito** y también medimos nuestras variables de resultado al nivel de la comunidad.
- Los bloques que representan un subgrupo sustancialmente significativo pueden ayudarnos a entender cómo los efectos pueden diferir por subgrupo.
  - Al controlar el número de sujetos por subgrupo nos aseguramos de tener suficientes sujetos en cada grupo.
  - Esto es especialmente útil cuando se tiene un grupo atípico: por simple chance puede que resulten muy pocas unidades de ese grupo en el tratamiento o en el control, incluso si hacemos asignación aleatoria (o puede que haya algún desbalance).

# Ejemplo de aleatorización por bloques

```r
set.seed(2)
N <- 20

# Creemos una base de datos artificial
datos <- data.frame(
  id = 1:N,
  edad = sample(20:50, size = N, replace = TRUE),
  # Definimos la misma cantidad de mujeres y hombres en la muestra
  genero = c(rep(0, N / 2), rep(1, N / 2))
)

datos$d_simple <- simple_ra(
  N = N,
  prob = 0.5
)
table(genero = datos$genero, d = datos$d_simple)
```

```
      d
genero 0 1
     0 8 2
     1 3 7
```

# Ejemplo de aleatorización por bloques

```r
# Aquí sucede la asignación:
datos$d_blocks <- block_ra(blocks = datos$genero, prob = 0.5)
table(genero = datos$genero, d = datos$d_blocks)
```

```
      d
genero 0 1
     0 5 5
     1 5 5
```

# Aleatorización por conglomerados I

- Un conglomerado es un **grupo de unidades**. En un estudio aleatorizado por conglomerados, todas las unidades del conglomerado se asignan al mismo estado de tratamiento.

# Aleatorización por conglomerados I

- Un conglomerado es un **grupo de unidades**. En un estudio aleatorizado por conglomerados, todas las unidades del conglomerado se asignan al mismo estado de tratamiento.
- Se debe usar la aleatorización por conglomerados si la intervención se lleva a cabo al nivel de conglomerados.

# Aleatorización por conglomerados I

- Un conglomerado es un **grupo de unidades**. En un estudio aleatorizado por conglomerados, todas las unidades del conglomerado se asignan al mismo estado de tratamiento.
- Se debe usar la aleatorización por conglomerados si la intervención se lleva a cabo al nivel de conglomerados.
  - Por ejemplo, si la intervención tiene que ver con los patios de recreo de la escuela, entonces la escuela es la unidad de asignación, incluso si la salud de los estudiantes es una variable de interés medida al nivel de los estudiantes.

# Aleatorización por conglomerados I

```
set.seed(23456)
# Creemos una base de datos artificial
datos <- data.frame(
  id = 1:N,
  edad = sample(20:50, size = N, replace = TRUE),
  # Definimos la misma cantidad de mujeres y hombres en la muestra
  genero = c(rep(0, N / 2), rep(1, N / 2)),
  escuela = rep(1:20, 5)
)


datos$d <- cluster_ra(clusters = datos$escuela)
head(table(datos$escuela, datos$d))
```

```
   0 1
1  0 5
2  0 5
3  5 0
4  5 0
5  0 5
6  5 0
```

# Buenas prácticas

# Buenas prácticas: replicabilidad

- Guía de métodos de EGAP sobre aleatoriedad (https://egap.org/resource/10-things-to-know-about-randomization/)

# Buenas prácticas: replicabilidad

- Guía de métodos de EGAP sobre aleatoriedad (https://egap.org/resource/10-things-to-know-about-randomization/)
- Definir una semilla (seed) y guardar el código y la columna con la asignación aleatoria.

# Buenas prácticas: replicabilidad

- Guía de métodos de EGAP sobre aleatoriedad (https://egap.org/resource/10-things-to-know-about-randomization/)
- Definir una semilla (seed) y guardar el código y la columna con la asignación aleatoria.
- Verificar/ Revisar

# Buenas Prácticas: balance

```
set.seed(23456)
N <- 200
# Creemos una base de datos artificial
datos <- data.frame(
  id = 1:N,
  edad = sample(20:50, size = N, replace = TRUE),
  genero = c(rep(0, N / 2), rep(1, N / 2))
)

datos$d <- simple_ra(
  N = N,
  prob = 0.5
)
```

# Buenas Prácticas: balance

```r
# Explorando descriptivamente el balance entre grupos.
# Nota: que los grupos parezcan balanceados a primera vista
# no es igual que si están estadísticamente balanceados.
# Aun así es una exploración que vale la pena hacer
datos %>%
  group_by(tratamiento = d) %>%
  summarise(
    m_edad = mean(edad),
    hombres = sum(genero == 0),
    mujeres = sum(genero == 1)
  )
```

```
# A tibble: 2 x 4
  tratamiento m_edad hombres mujeres
        <int>  <dbl>   <int>   <int>
1           0   33.5      56      54
2           1   33.6      44      46
```

# Buenas Prácticas: balance

- Revisar el balance general del estudio con una prueba D cuadrado (D-square test) utilizando la función `xBalance` en el paquete `RItools` (Hansen and Bowers (2008))(inferencia de aleatorización con muestras grandes):

```
---Overall Test---
        chisquare df p.value
unstrat     0.084  2    0.96
---
Signif. codes:  0 '***' 0.001 '** ' 0.01 '*  ' 0.05 '.  ' 0.1 '   ' 1
```

# Buenas prácticas: balance

- La asignación aleatoria nos da, en valor esperado, **balance general** en las distintas covariables. Esto no garantiza que todas las relaciones entre el tratamiento y las covariables sean cero. De hecho, en un experimento pequeño, la magnitud del desbalance puede llegar ser alta, incluso si la aleatorización se produjo perfectamente.

# Referencias

Hansen, Ben B., and Jake Bowers. 2008. "Covariate Balance in Simple, Stratified and Clustered Comparative Studies." *Statistical Science* 23 (2): 219–36.