

# Tarea 9

Dalia Camacho

## 9 - Inferencia gráfica, tamaño de muestra, bootstrap paramétrico.

```
suppressMessages(library(tidyverse))
library(nullabor)
library(knitr)
set.seed(578)
```

### Inferencia gráfica

Los datos marg\_diabetes incluyen información de marginación y diabetes en México: \* `ent`, `id_ent`, `mun`, `id_mun`, `cvegeo`: corresponden al estado, municipio y sus códigos de identificación. \* `n_causa` es el número de muertes de adultos mayores a 65 años a causa de diabetes en 2015, y `tasa_mun` la tasa correspondiente por cada 10,000 habitantes. \* `tasa_alf` (porcentaje de población alfabetizada), `ind_des_hum` (índice de desarrollo humano), `conapo` (índice de marginación).

Utiliza los datos para explorar gráficamente la relación entre algunas de las variables, utiliza el protocolo `lineup` para hacer inferencia gráfica.

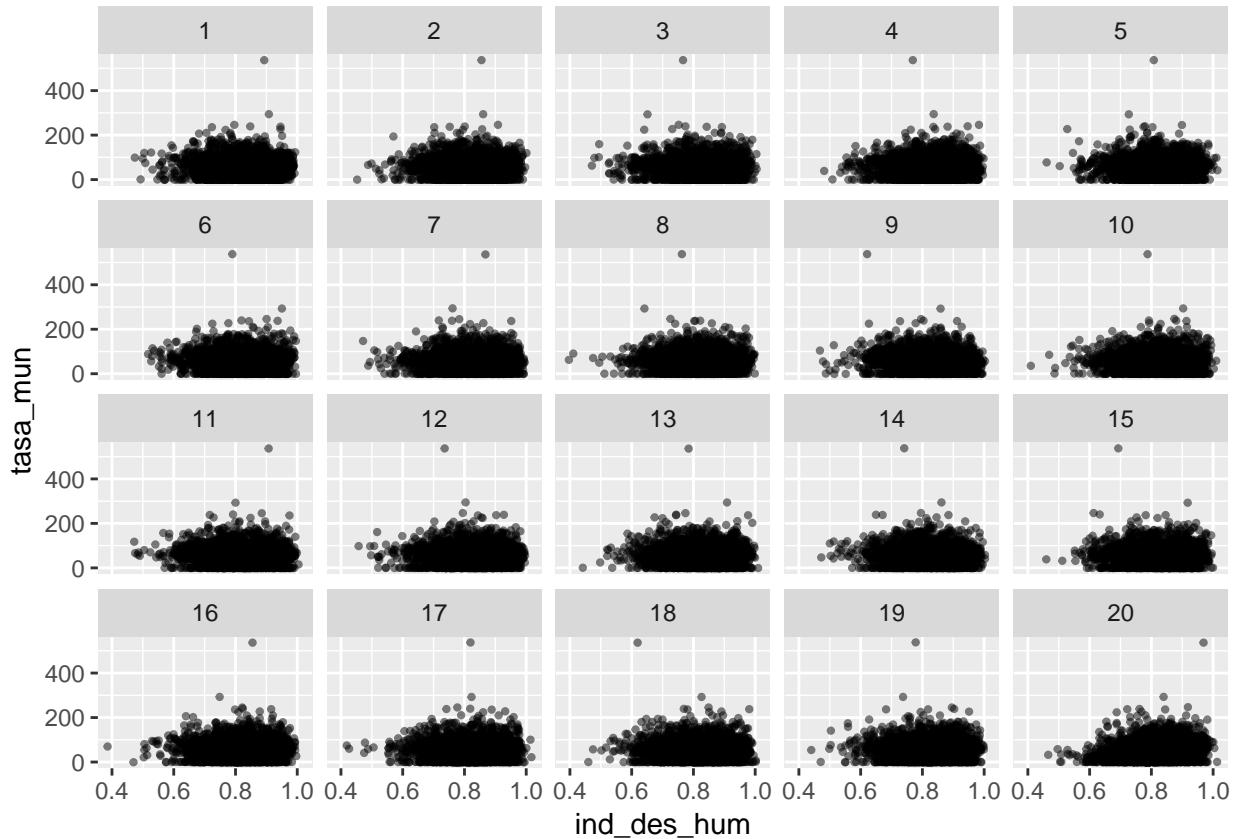
```
suppressMessages(Diabetes <- read_csv("https://raw.githubusercontent.com/tereom/est-computacional-2018/r/lineup/Diabetes"))
glimpse(Diabetes)
```

```
## Observations: 2,456
## Variables: 9
## $ id_ent      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, ...
## $ id_mun      <int> 1, 2, 3, 4, 10, 5, 6, 7, 11, 8, 9, 1, 2, 5, 3, 4, ...
## $ ent         <chr> "Aguascalientes", "Aguascalientes", "Aguascaliente...
## $ mun         <chr> "Aguascalientes", "Asientos", "Calvillo", "Cosío", ...
## $ pob          <dbl> 50112.4050, 2940.0092, 4606.0462, 964.7360, 1208.2...
## $ n_causa      <int> 324, 23, 16, 11, 12, 34, 13, 24, 8, 2, 11, 160, 43...
## $ tasa_mun    <dbl> 64.65465, 78.23105, 34.73695, 114.02083, 99.31408, ...
## $ ind_des_hum <dbl> 0.899, 0.884, 0.901, 0.889, 0.880, 0.898, 0.897, 0...
## $ conapo       <int> 5, 3, 4, 3, 3, 5, 4, 4, 4, 4, 3, 5, 5, 5, 5, 4, ...
```

Evaluamos la relación entre la tasa de muertes por diabetes por cada 10,000 habitantes con el índice de desarrollo humano.

```
permutInd <- lineup(null_permute("tasa_mun"), Diabetes)
```

```
## decrypt("y9RE o5G5 JL rYiJGJYL OX")
ggplot(permutInd, aes(x = ind_des_hum, y = tasa_mun)) +
  facet_wrap(~ .sample) +
  geom_jitter(position = position_jitter(width = 0.1, height = 1),
              size = 0.8, alpha = 0.5)
```

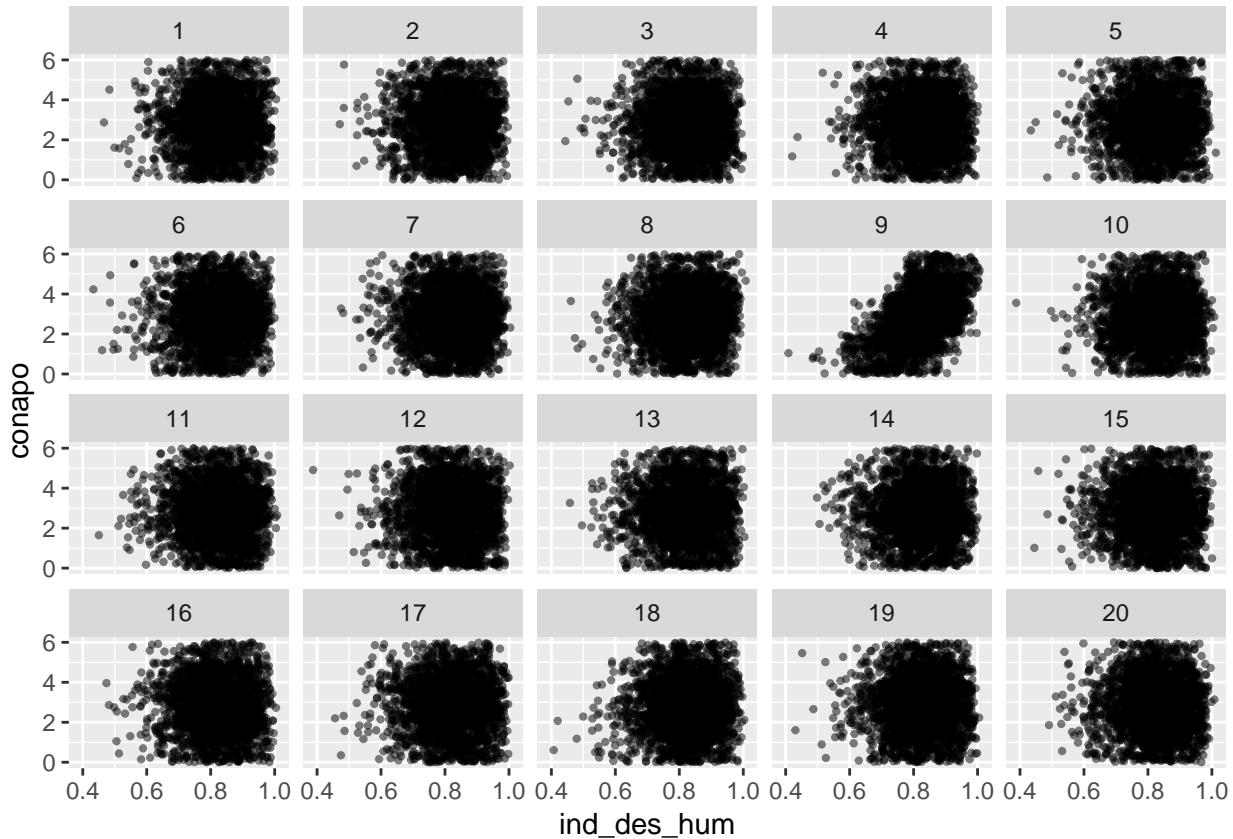


El conjunto de los datos verdaderos no es distinguible, por lo que no hay una relación real entre muertes por diabetes e índice de desarrollo humano

ahora evaluamos la relación entre el índice de marginación con el índice de desarrollo humano.

```
permutInd <- lineup(null_permute("conapo"), Diabetes)

## decrypt("y9RE o5G5 JL rYiJGJYL Ts")
ggplot(permutInd, aes(x = ind_des_hum, y = conapo)) +
  facet_wrap(~ .sample) +
  geom_jitter(position = position_jitter(width = 0.1, height = 1),
             size = 0.8, alpha = 0.5)
```



En este caso el conjunto 5 es distingüible de los demás conjuntos, por lo que sí existe relación entre índice de marginación e índice de desarrollo humano.

### Simulación para calcular tamaños de muestra

Supongamos que queremos hacer una encuesta para estimar la proporción de hogares donde se consume refresco de manera regular, para ello se diseña un muestreo por conglomerados donde los conglomerados están dados por conjuntos de hogares de tal manera que todos los conglomerados tienen el mismo número de hogares. La selección de la muestra se hará en dos etapas:

1. Seleccionamos  $J$  conglomerados de manera aleatoria.
2. En cada conglomerado seleccionamos  $n/J$  hogares para entrevistar.

El estimador será simplemente el porcentaje de hogares del total de la muestra. Suponemos que la verdadera proporción es cercana a 0.50 y que la media de la proporción de interés a lo largo de los conglomerados tiene una desviación estándar de 0.1.

1. Supongamos que la muestra total es de  $n = 1000$ . ¿Cuál es la estimación del error estándar para la proporción estimada si  $J = 1, 10, 100, 1000$ ?

```
# Definimos los parámetros para simular el error estándar
nsim <- 1000
n     <- 1000
J      <- c(1, 10, 100, 1000)
mup   <- 0.5
sdp   <- 0.1

Mean_p <- function(n,j){
```

```

muestra <- c()
for(i in 1:j){
  p           <- rnorm(1, mup, sdp)
  muestra <- c(muestra, rbinom(n/j, 1, p))
}
mean(muestra)

for (j in J) {
  assign(paste0("SE_", j), rerun(nsim, Mean_p(n,j)) %>%
         flatten_dbl() %>% sd())
}

df <- data.frame((c("J=1"=SE_1, "J=10"=SE_10, "J=100"=SE_100, "J=1000"=SE_1000)))
names(df) <- "SE"
kable(df)

```

	SE
J=1	0.1013060
J=10	0.0344051
J=100	0.0183638
J=1000	0.0162814

2. El objetivo es estimar la proporción que consume refresco en la población con un error estándar de a lo más 2%. ¿Que valores de  $J$  y  $n$  debemos elegir para cumplir el objetivo al menor costo? Los costos del levantamiento son:

- 50 pesos por encuesta.
  - 500 pesos por conglomerado







```

Less_0.02 <- which(MatSE < 0.02)
Optvars   <- Less_0.02 [which.min(MatCost[Less_0.02 <- which(MatSE < 0.02)])]
j          <- floor(Optvars/60)
i          <- Optvars-j*60

```

El menor costo se obtiene con 2750 personas y con 28 conglomerados y el costo es `MatCost[i,j]`.

## Bootstrap paramétrico

- Sean  $X_1, \dots, X_n \sim N(\mu, 1)$ . Sea  $\theta = e^\mu$ , crea una base de datos usando  $\mu = 5$  que consista de  $n = 100$  observaciones.

```

n      <- 100
mu    <- 5
sigma <- 1
Datos <- rnorm(n,mu, sigma)

```

- Usa el método delta para estimar  $\hat{\theta}$  y crea un intervalo del 95% de confianza. Usa bootstrap paramétrico para crear un intervalo del 95%.

```

# Método delta
mu_est     <- mean(Datos)
SE_mu_est <- 1/sqrt(n)

g <- function(tau){exp(tau)}
g_prima <- g
theta_est <- g(mu_est)
SE_est    <- g_prima(mu_est)*SE_mu_est
SE_est

## [1] 15.14235

IC_low  <- theta_est + SE_est*qnorm(0.025)
IC_up   <- theta_est + SE_est*qnorm(0.975)
c(IC_low, IC_up)

## [1] 121.745 181.102

```

Usa bootstrap no paramétrico para crear un intervalo del 95%. Compara tus respuestas.

```

bootEstimates <- function(){
  Bsample <- rnorm(n, mu_est, SE_mu_est)
  mu_B    <- mean(Bsample)
  theta_B <- g(mu_B)
}
theta_Boots <- rerun(nsims, bootEstimates()) %>% flatten_dbl()

SE_boot <- sqrt(1/(nsims-1)*sum(theta_Boots-theta_est)^2)

IC_low  <- theta_est + SE_boot*qnorm(0.025)
IC_up   <- theta_est + SE_boot*qnorm(0.975)
c(IC_low, IC_up)

## [1] 149.3252 153.5218

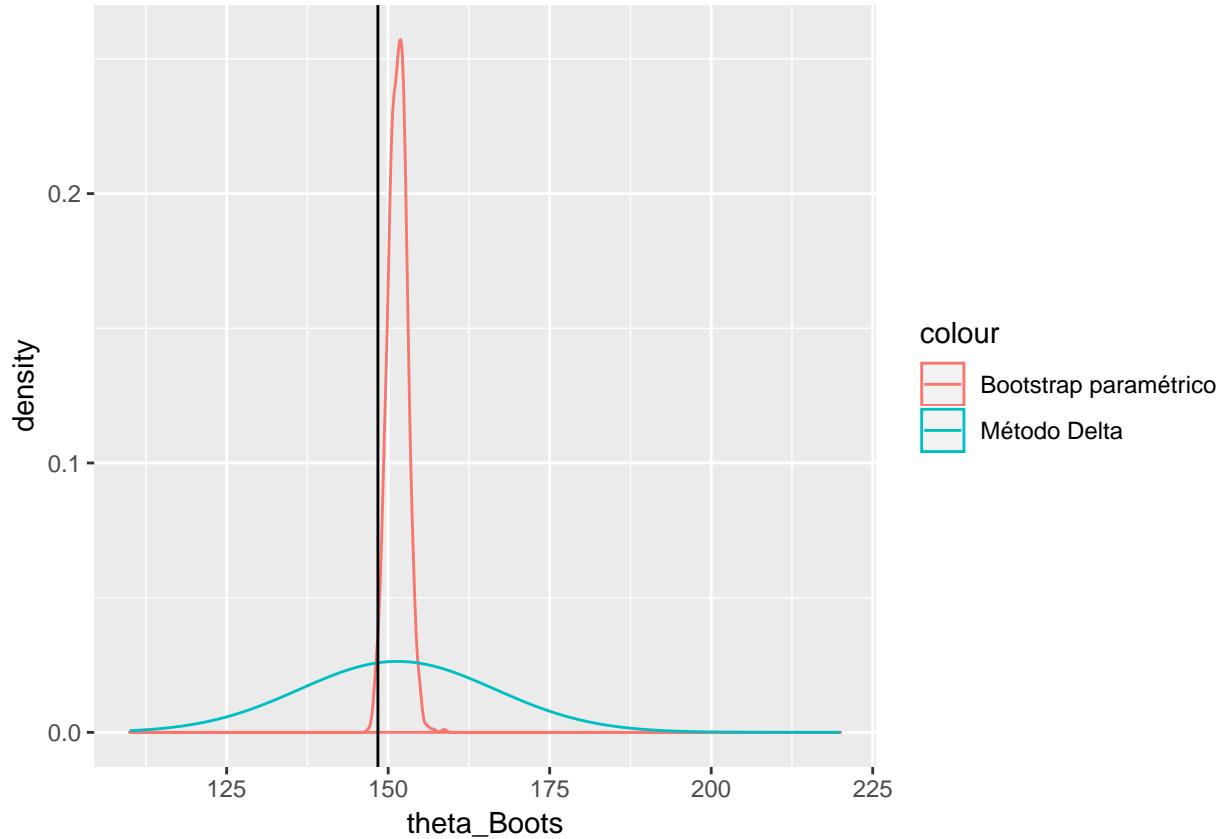
```

Los intervalos obtenidos por el método Delta son más grandes que los obtenidos por bootstrap paramétrico. Pero los intervalos del bootstrap paramétrico no contienen al verdadero valor.

- Realiza un histograma de replicaciones bootstrap para cada método, estas son estimaciones de la distribución de  $\hat{\theta}$ . El método delta también nos da una aproximación a esta distribución:  $Normal(\hat{\theta}, \hat{se}^2)$ . Comparalos con la verdadera distribución de  $\hat{\theta}$  (que puedes obtener vía simulación). ¿Cuál es la aproximación más cercana a la verdadera distribución?

```
x <- seq(110, 220)
y <- dnorm(x, theta_est, SE_est)
ggplot()+
  geom_density(aes(theta_Boots, color= "Bootstrap paramétrico"),
               alpha=0.5, bins=30) +
  geom_line(aes(x,y, color= "Método Delta"))+
  geom_vline(aes(xintercept=exp(mu)))
```

## Warning: Ignoring unknown parameters: bins



En este caso la distribución del bootstrap paramétrico se encuentra alejada de la verdadera distribución.

Pista:  $se(\hat{\mu}) = 1/\sqrt{n}$