EST-24107: Simulación

Profesor: Alfredo Garbuno Iñigo — Otoño, 2022 — *Bootstrap*.
Objetivo: En esta sección estudiaremos los métodos de remuestreo que permiten cuantificar incertidumbre en situaciones donde nuestro estimador se construye con una sola muestra y donde no hay acceso al sistema que genera los datos. Esto contrasta con los métodos anteriores pues antes hemos estudiado bajo el supuesto de poder tener acceso al generador de números aleatorios correctos. En esta ocasión sólo tenemos una muestra y querríamos cuantificar incertidumbre en nuestros estimadores.
Lectura recomendada: El libro de Chihara and Hesterberg [1] presenta una discusión del tema bajo el esquema de análisis estadístico. El libro Efron and Tibshirani [3] es una referencia clásica para *bootstrap*. El capítulo 8 de Wasserman [4] contiene una discusión condensada de *bootstrap*. El capítulo 10 de Efron and Hastie [2] establece el método *bootstrap* como parte del *set* de herramientas del estadístico moderno.

1. INTRODUCCIÓN

El remuestreo se refiere a un conjunto de técnicas estadísticas, computacionalmente intensivas, que `estiman` la *distribución de una población* basadas en `muestreo aleatorio con reemplazo` de una muestra observada.

Se considera una muestra aleatoria $X_1, \dots, X_N$ como si fuera una población finita y se generan muestras aleatorias de la misma muestra para estimar características poblacionales y hacer inferencia de la población muestreada.

Las técnicas de remuestreo permiten calcular medidas de ajuste (en términos de sesgo, varianza, intervalos de confianza, errores de predicción o de algunas otras medidas) a los estimados basados en muestras.

Estas técnicas son usualmente no paramétricas, y varias son tan antiguas como la estadística misma. Por ejemplo, las técnicas de permutación son de Fisher (1935) y Pitmann (1937); la validación cruzadas fue propuesta por Kurtz en 1948, y el *Jackknife* fue propuesto por Maurice Quenouille en 1949 aunque fue John Tukey en 1958 quién le dio el nombre a la técnica.

1.1. Contexto histórico

Bradley Efron introdujo el `bootstrap` en 1979. El término 'bootstrapping' se refiere al concepto de *pulling oneself up by one's bootstraps*, frase que aparentemente se usó por primera vez en:

- Raspe, R. E. (1786). *Gulliver Revived: Or the Singular Travels, Campaigns, Voyages, and Adventures of Baron Munikhouson, Commonly Called Munchausen.*

1.2. Idea general

El objetivo del remuestreo es estimar alguna característica poblacional, representada por θ (tal como media, mediana, desviación estándar, coeficientes de regresión, matriz de covarianza, etc.) `basada sólo en los datos observados`.

También interesan las propiedades de la `distribución de estimador`, sin hacer supuestos restrictivos sobre la forma de la distribución de los datos originales.

Para una muestra aleatoria $X_1, \dots, X_N$, la `distribución de remuestreo` es la distribución empírica $\hat{\mathbb{P}}_N$, que asigna probabilidad $1/N$ a cada una de las observaciones de la muestra.

1.3. Ejemplo

Consideremos una muestra de 6 parejas. La variable de interés es la diferencia del ingreso de los miembros de cada pareja (en miles de pesos al mes).

i	$P_i^{(1)}$	$P_i^{(2)}$	$d_i = P_i^{(1)} - P_i^{(2)}$
1	24	18	6
2	14	17	-3
3	40	35	5
4	44	41	3
5	35	37	-2
6	45	45	0

Definamos θ como el promedio de las diferencias de ingreso poblacional. Podemos estimar θ con

$$\hat{\theta}_N = \frac{6 - 3 + 5 + 3 - 2 + 0}{6} = 1.5\,. \tag{1}$$

¿Cómo calculamos la variabilidad de nuestro estimador? Es decir, ¿cómo calculamos la variabilidad de $\hat{\theta}_n$?

1.3.1. ***Ejercicio:*** Escribe la fórmula del error estándar bajo los siguientes supuestos:

1. La diferencia tiene una distribución $d_i \sim \mathsf{N}(\theta, \sigma^2)$.
2. La varianza σ^2 es conocida.

1.4. `Observaciones:`

- Suponer que la diferencia de ingresos d_i se comporta como una variable normal puede no estar *tan* errado. Pues con un número suficiente de muestras podríamos suponer que el resultado del TLC se cumple. Entonces, ¿qué hacemos si no conocemos la distribución de las observaciones?
- Si no conocemos σ^2 lo podemos estimar con la muestra. Por ejemplo, podemos utilizar intervalos de confianza derivados de una distribución t_{N-1}.
- Si nos interesa otro parámetro de la población podemos construir estimadores diferentes. Por ejemplo, nos podría interesar la `mediana` de una población $q_{0.5} = \mathbb{P}^{-1}(1/2)$. Para este caso, podemos estimar dicho parámetro por medio de

$$\hat{q}_{0.5} = \begin{cases} X_{(\frac{n+1}{2})} & \text{si } N \text{ es impar} \\ \frac{X_{(n/2)} + X_{(n/2+1)}}{2} & \text{si } N \text{ es par} \end{cases}\,. \tag{2}$$

En Fig. 1 la estimación de la mediana en distintos grupos acompañados de su estimación de incertidumbre.

1.5. La distribución de muestreo

Hasta ahora lo que hemos hecho es estimar $\hat{\pi}_N^{\mathsf{MC}}(f) \approx \pi(f) = \int f(x)\,\pi(x)\,\mathrm{d}x$ por medio de muestras de la densidad $\pi(\cdot)$. Es decir, por medio de

$$X_1, \ldots, X_N \overset{\mathsf{iid}}{\sim} \pi\,. \tag{3}$$

Hemos considerado la noción `frecuentista` de medir nuestra incertidumbre en nuestro estimador por medio del **error estándar** de nuestro estimador. Donde éste último está definido como

$$\mathsf{ee}\left(\hat{\pi}_N^{\mathsf{MC}}(f)\right) = \left(\mathbb{V}(\hat{\pi}_N^{\mathsf{MC}}(f))\right)^{1/2}\,, \tag{4}$$

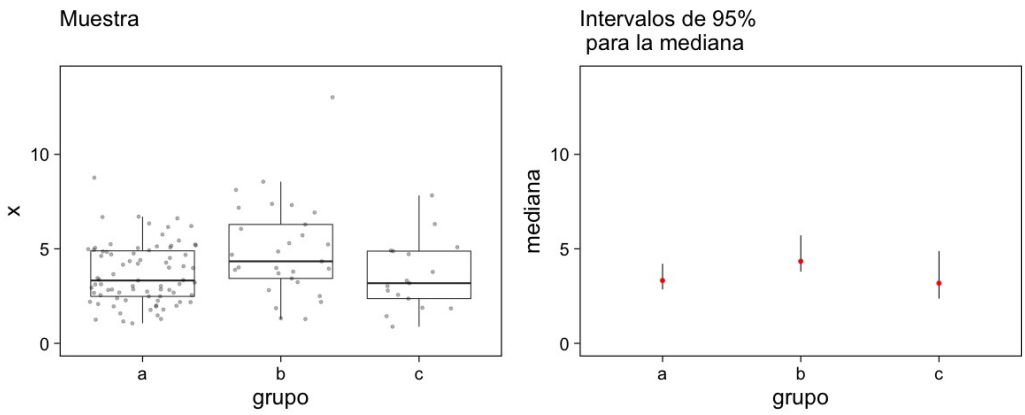


FIGURA 1. *Estimación de mediana (panel izquierdo) con intervalos de incertidumbre (panel derecho).*

y la varianza es con respecto a la variabilidad que *nace* por haber observado distintas muestras.

Es decir, estamos considerando la situación en que podemos replicar el proceso de muestreo tantas veces como queramos (o recursos computacionales tengamos). Denotemos por B el número de réplicas que podemos realizar y denotemos por

$$X_1^{(b)}, \ldots, X_N^{(b)} \overset{\text{iid}}{\sim} \pi\,, \qquad b = 1, \ldots, B\,, \tag{5}$$

las réplicas que generamos.

Notemos que es a través de este proceso de crear réplicas que podemos construir una distribución para $\hat{\pi}_N^{\mathsf{MC}}(f)$ y notemos, además, que nuestro estimador es el resultado de aplicar una función a la muestra dada

$$\hat{\pi}_{N,b}^{\mathsf{MC}}(f) = t(X_1^{(b)}, \ldots, X_N^{(b)})\,, \qquad b = 1, \ldots, B\,. \tag{6}$$

La distribución resultante de nuestro estimador $\hat{\pi}_N^{\mathsf{MC}}(f)$ —derivada de haber observado un conjunto de datos distinto—es lo que en sus cursos de estadística le llamamos `distribución de muestreo` del estimador.

Nota que en esta situación asumimos que podemos generar tantas muestras como queramos de la distribución de interés π. En esta sección del curso estudiaremos un mecanismo para cuando no podemos hacer eso (generar muestras de una población) y sólo tenemos acceso a una muestra—que asumimos aleatoria—de la población que nos interesa.

2. LA IDEA DEL BOOTSTRAP

Como explicamos, el problema que tenemos ahora es que normalmente sólo tenemos una muestra, así que no es posible calcular las distribuciones de muestreo como hicimos arriba y evaluar qué tan preciso es nuestro estimador. Sin embargo, podemos hacer lo siguiente:

Supongamos que tenemos una muestra $X_1, X_2, \ldots, X_N$ de alguna población desconocida y un estimador $\hat{\theta}_N = t(X_1, \ldots, X_N)$.

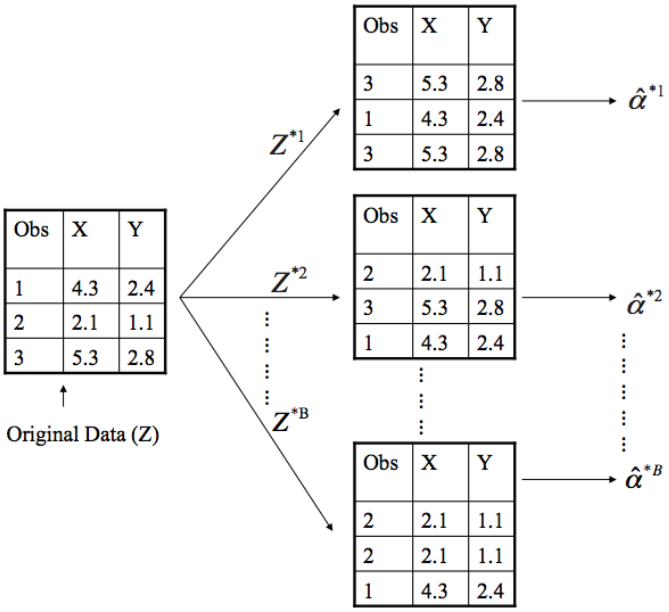
2.1. Mundo poblacional

1. Si tuviéramos la distribución poblacional, simulamos muestras iid para aproximar la distribución de muestreo de nuestro estimador, y así entender su variabilidad.
2. Pero **no** tenemos la distribución poblacional.
3. **Sin embargo, podemos estimar la distribución poblacional con nuestros valores muestrales**.

2.2. Mundo bootstrap

1. Si usamos la estimación del inciso 3, entonces usando el inciso 1 podríamos tomar muestras de nuestros datos muestrales, como si fueran de la población, y usando el mismo tamaño de muestra. El muestreo lo hacemos con reemplazo de manera que produzcamos muestras independientes de la misma "población estimad", que es la muestra.
2. Evaluamos nuestra estadística en cada una de estas remuestras.
3. A la distribución resultante le llamamos **distribución *bootstrap*** o **distribución de remuestreo** del estimador.
4. Usamos la distribución *bootstrap* de la muestra para estimar la variabilidad en nuestra estimación con **la muestra original**.

El esquema de esta estrategia lo podemos representar con la figura siguiente



Veamos que sucede para un ejemplo concreto, donde nos interesa estimar la media de los precios de venta de una población de casas. Tenemos nuestra muestra:

```
set.seed(2112)
poblacion_casas ← read_csv("data/casas.csv")
muestra ← sample_n(poblacion_casas, 200, replace = TRUE) ▷
  select(id, nombre_zona, area_habitable_sup_m2, precio_miles)
```

```
# A tibble: 6 × 4
     id nombre_zona area_habitable_sup_m2 precio_miles
  <dbl> <chr>                       <dbl>        <dbl>
1   502 Somerst                     164.          227.
2    79 Sawyer                      164.          136.
3   440 Edwards                     111.          110
4   524 Edwards                     434.          185.
5  1442 CollgCr                      78.8         149.
6   769 CollgCr                     171.          217.
```

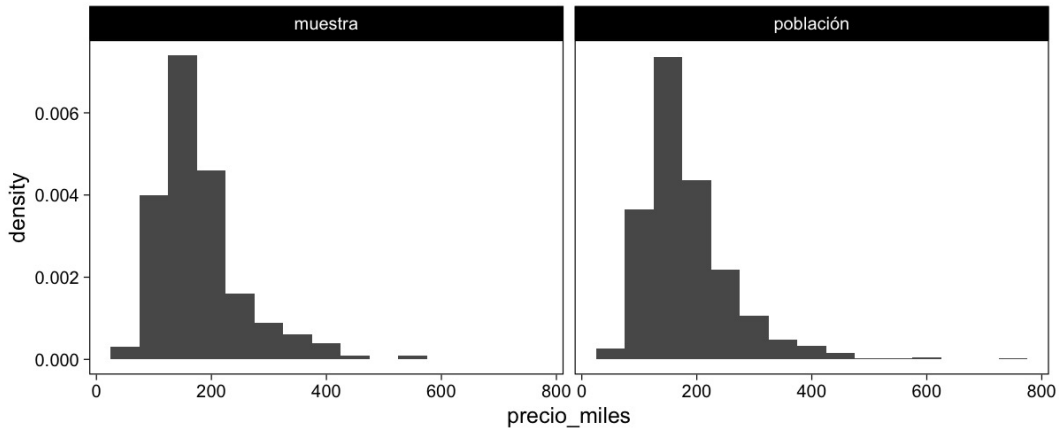
```
[1] "Hay 1144 casas en total, tomamos muestra de 200"
```

Esta muestra nos da nuestro estimador de la distribución poblacional.

```
mean(muestra$precio_miles)
```

```
[1] 179.96
```

Por ejemplo, podemos fijarnos en un gráfico con histogramas:



Y vemos que la aproximación es razonable en las partes centrales de la distribución.

Ahora supongamos que nos interesa cuantificar la precisión de nuestra estimación de la media poblacional de precios de casas, y usaremos la media muestral para hacer esto. Para nuestra muestra, nuestra estimación puntual es:

```
media ← mean(muestra$precio_miles)
media
```

```
[1] 179.96
```

Y recordamos que para aproximar la distribución de muestreo podíamos muestrear repetidamente la población y calcular el valor del estimador en cada una de estas muestras. Aquí no tenemos la población, **pero tenemos una estimación de la población**: la muestra obtenida.

Así que para evaluar la variabilidad de nuestro estimador, entramos en el mundo boostrap, y consideramos que la población es nuestra muestra.

Podemos entonces extraer un número grande de muestras con reemplazo de tamaño 200 **de la muestra**: el muestreo debe ser análogo al que se tomó para nuestra muestra original. Evaluamos nuestra estadística (en este caso la media) en cada una de estas remuestras:

```
## paso 1: define el estimador
calcula_estimador ← function(data){
  data ▷
    summarise(media_precio = mean(precio_miles), .groups = "drop")
}
```

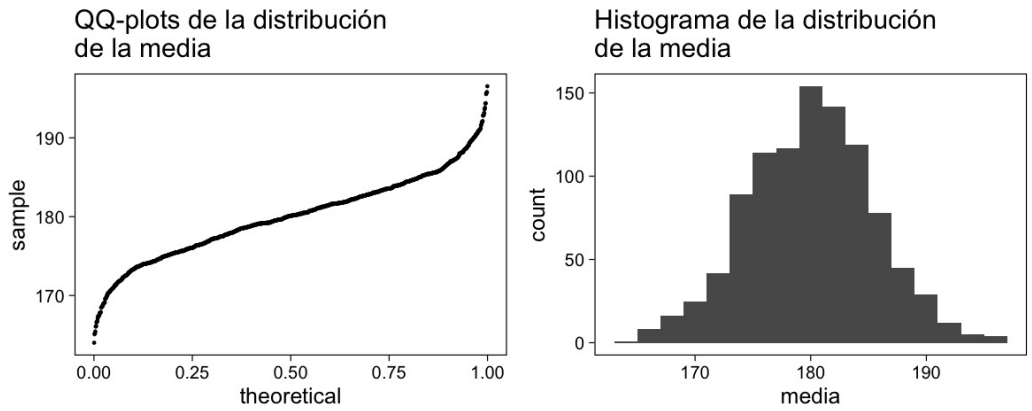
```
## paso 2: define el proceso de remuestreo
genera_remuestras ← function(data, n = 200){
  data ▷
    sample_n(200, replace = TRUE)
}
```

```
## paso 3: definimos el paso bootstrap
paso_bootstrap ←function(id){
  muestra ▷
  genera_remuestras() ▷
    calcula_estimador() ▷
    pull(media_precio)
}
```

```
## paso 4: aplica el procedimiento bootstrap
media_muestras ← map_dbl(1:1000, paso_bootstrap)
media_muestras[1:10]
```

```
[1] 176.38 175.96 175.91 176.35 177.77 186.78 179.23 181.62 175.07 177.13
```

Y nuestra estimación de la distribución de muestreo para la media es entonces:



A esta le llamamos la distribución de remuestreo de la media, que definimos más abajo. Ahora podemos calcular un intervalo de confianza del 90 % simplemente calculando los cuantiles de esta distribución (no son los cuantiles de la muestra original!):

```
limites_ic ← quantile(media_muestras, c(0.05,  0.95)) ▷ round(4)
limites_ic
```

```
    5%    95%
171.05 188.93
```

Otra cosa que podríamos hacer para describir la dispersión de nuestro estimador es calcular el error estándar de remuestreo, que estima el error estándar de la distribución de muestreo:

```
ee_boot ← sd(media_muestras)
round(ee_boot, 2)
```

```
[1] 5.4
```

2.2.1. ***Definición*** *[La distribución de remuestreo]:* Sea $X_1, X_2, \ldots, X_N$ una muestra independiente y idénticamente distribuida (iid), y $\hat{\theta}_N = s(X_1, X_2, \ldots, X_N)$ una estadística. Supongamos que los valores que obervamos son $x_1, x_2, \ldots, x_N$. La **distribución de remuestreo** de $\hat{\theta}_N$ es la distribución de $\theta^*_N = s(X^*_1, X^*_2, \ldots X^*_N)$, donde cada X^*_i se obtiene tomando al azar uno de los valores de $x_1, x_2, \ldots, x_N$.

Otra manera de decir esto es que la remuestra $X^*_1, X^*_2, \ldots, X^*_N$ es una muestra con reemplazo de los valores observados $x_1, x_2, \ldots, x_N$.

2.2.2. ***Ejemplo:*** Si observamos la muestra

```
muestra ← sample(1:20, 5)
muestra
```

```
[1] 15 16  8  6  2
```

Una remuestra se obtiene:

```
sample(muestra, size = 5, replace = TRUE)
```

```
[1]  2 15  2 15  2
```

Nótese que algunos valores de la muestra original pueden aparecer varias veces, y otros no aparecen del todo.

2.3. Nota

La muestra original es una aproximación de la población de donde fue extraída. Así que remuestrear la muestra aproxima lo que pasaría si tomáramos muestras de la población. La **distribución de remuestreo** de una estadística, que se construye tomando muchas remuestras, aproxima la distribución de muestreo de la estadística.

Y el proceso que hacemos es:

2.3.1. ***Remuestreo para una población:*** Dada una muestra de tamaño n de una población,

1. Obtenemos una remuestra de tamaño n con reemplazo de la muestra original
2. Repetimos este remuestreo muchas veces (por ejemplo $10,000$).
3. Construímos la distribución *bootstrap*, y examinamos sus características (dónde está centrada, dispersión y forma).

3. EL PRINCIPIO DE PLUG-IN

La idea básica detrás del *bootstrap* es el principio de *plug-in* para estimar parámetros poblacionales: si queremos estimar una cantidad poblacional, calculamos esa cantidad poblacional con la muestra obtenida. Es un principio común en estadística.

Por ejemplo, si queremos estimar la media o desviación estándar poblacional, usamos la media muestral o la desviación estándar muestral. Si queremos estimar un cuantil de la población usamos el cuantil correspondiente de la muestra, y así sucesivamente.

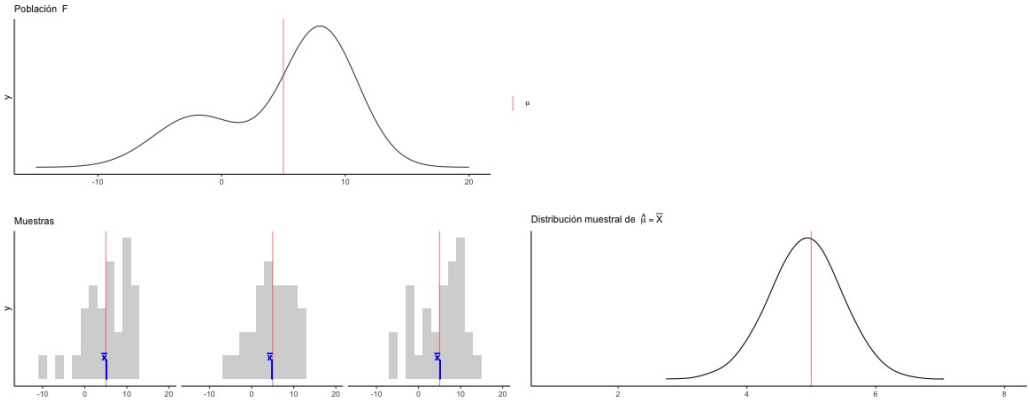
En todos estos casos, lo que estamos haciendo es:

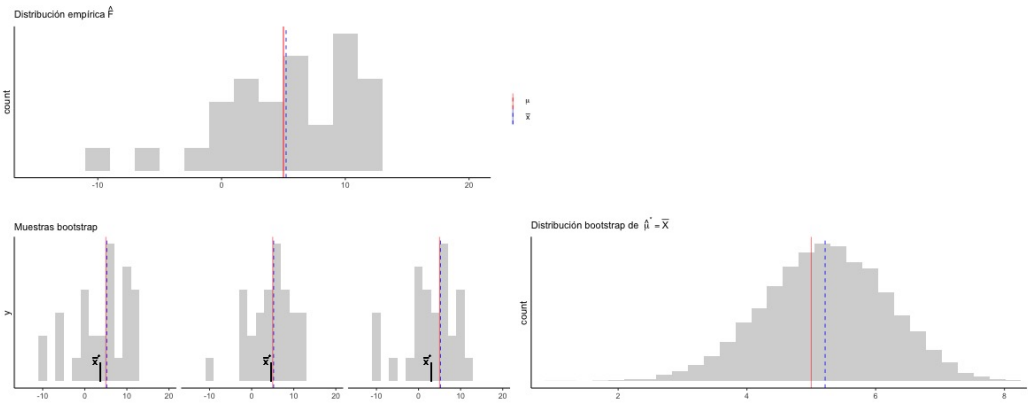
- Tenemos una fórmula para la cantidad poblacional de interés en términos de la distribución poblacional.
- Tenemos una muestra, que usamos para estimar la cantidad poblacional. La distribución que da una muestra se llama distribución **empírica**.
- Contruimos nuestro estimador "enchufando" la distribución empírica de la muestra en la fórmula del estimador.

En el *bootstrap* aplicamos este principio simple a la **distribución de muestreo**:

- **Si tenemos la población**, podemos **calcular** la distribución de muestreo de nuestro estimador tomando muchas muestras de la **población**.
- Estimamos la **poblacion** con la **muestra** y enchufamos en la frase anterior:
- Podemos **estimar** la distribucion de muestreo de nuestro estimador tomando muchas muestras de la **muestra** (*bootstrap*).

Nótese que el proceso de muestreo en el último paso **debe ser el mismo** que se usó para tomar la muestra original. Estas dos imágenes simuladas con base en un ejemplo de [1] muestran lo que acabamos de describir:





3.1. Observación

Veremos ejemplos más complejos, pero nótese que si la muestra original son observaciones independientes obtenidas de la distribución poblacional, entonces logramos esto en las remuestras tomando aleatoriamente observaciones con reemplazo de la muestra. Igualmente, las remuestras deben ser del mismo tamaño que la muestra original.

3.1.1. ***Ejercicio:***

- ¿Porqué no funcionaría tomar muestras sin reemplazo? Piensa si hay independencia entre las observaciones de la remuestra, y cómo serían las remuestras sin reemplazo.
- ¿Por qué no se puede hacer bootstrap si no conocemos cómo se obtuvo la muestra original?

3.2. Notación

- Denotamos por $\mathbb{P}$ la función de distribución acumulada de la población y su estimador, que es la función empírica $\hat{\mathbb{P}}_n$, como en [3].
- Denotamos por $X_1, \ldots, X_N \overset{\text{iid}}{\sim} \mathbb{P}$ una muestra aleatoria de la distribución $\mathbb{P}$ y por $\theta = t(\mathbb{P})$ una cantidad poblacional de interés y que queremos estimar.

3.2.1. Aclaración: La notación hace enfásis en que la característica de la distribución que nos interesa es resultado de aplicar un procedimiento numérico $t(\cdot)$ a nuestra distribución $\mathbb{P}$. Esto no contradice nuestra notación. Pues nosotros hemos considerado

$$\theta = t(\mathbb{P}) = \mathbb{P}(f)\,, \tag{7}$$

donde $\mathbb{P}(f)$ es el procedimiento asociado a realizar la integral de la función f ponderada por $\mathbb{P}$. Es decir, $\int f(u)\mathrm{d}\mathbb{P}(u)$.

3.2.2. ***Definición*** *[El principio de* plug-in *]:* Es un método de estimación de parámetros que utiliza muestras para el procedimiento de estimación. El estimador *plug-in* del parámetro $\theta = t(\mathbb{P})$ está definido como

$$\hat{\theta} = t(\hat{\mathbb{P}}_n)\,. \tag{8}$$

Es decir, para estimar θ por medio del procedimiento $t(\cdot)$ utilizamos la aproximación la distribución de acumulación empírica como si fuera la distribución real.

3.2.3. Aclaración:

- Hemos visto que nuestras aproximaciones a $\pi(f)$ son de la forma

$$\hat{\pi}_N^{(\cdot)}(f) = \frac{1}{N}\sum_{n=1}^{N} f(x_n)\,, \tag{9}$$

donde las x_n son realizaciones aleatorias de la distribución π.
- Noten que nuestros estimadores de la familia de métodos Monte Carlo también son estimadores *plug-in* pues tomamos la distribución

$$\pi_N^{(\cdot)}(x) = \frac{1}{N}\sum_{n=1}^{N} \delta(x - x_n)\,, \tag{10}$$

como la aproximación de $\pi(\cdot)$ para obtener valores de estimador que nos interesa $\theta = \pi(f)$.
- Esto nos permite escribir nuestros estimadores como

$$\hat{\theta} = t\left(\hat{\pi}_N^{(\cdot)}\right)\,. \tag{11}$$

- La distribución empírica $\hat{\mathbb{P}}_n$ es un estimador *razonable* de la distribución poblacional $\mathbb{P}$ pues por el teorema de Glivenko-Cantelli ([4], o aquí), $\hat{\mathbb{P}}_n$ converge a $\mathbb{P}$ cuando el tamaño de muestra $n \to \infty$, lo cual es intuitivamente claro.
- Nuestros estimadores *bootstrap* son estimadores

$$\hat{\theta}_N^* = t(X_1^*, \ldots, X_N^*)\,, \tag{12}$$

donde $X_1^*, \ldots, X_N^*$ son muestras escogidas por `muestreo aleatorio simple` de la muestra original.
- Es decir, podemos pensar en $X_1^*, \ldots, X_N^* \overset{\text{iid}}{\sim} \hat{\mathbb{P}}_N$. Por lo que nuestros estimadores *bootstrap* son de la forma

$$\hat{\theta}_N^* = t\left(\hat{\mathbb{P}}_N\right)\,. \tag{13}$$

3.3. Ejemplo

En el siguiente ejemplo (tomadores de té), podemos estimar la proporción de tomadores de té que prefiere el té negro usando nuestra muestra:

```
te ← read_csv("data/tea.csv") ▷
  rowid_to_column() ▷
  select(rowid, Tea, sugar)
```

```
te ▷
  mutate(negro = ifelse(Tea == "black", 1, 0)) ▷
  summarise(prop_negro = mean(negro), n = length(negro), .groups = "drop")
```

```
# A tibble: 1 × 2
  prop_negro     n
       <dbl> <int>
1      0.247   300
```

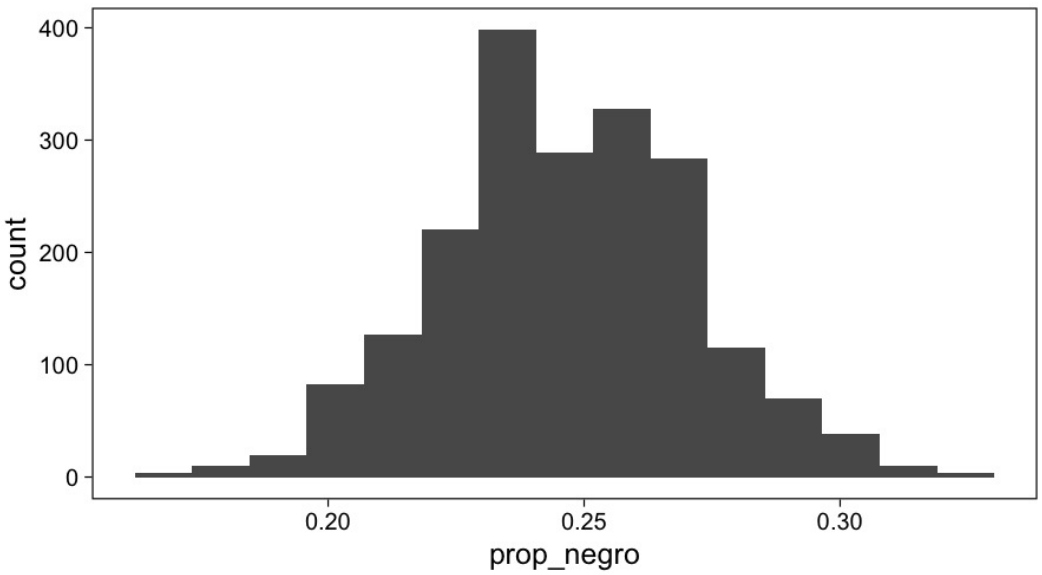
¿Cómo evaluamos la precisión de este estimador? Supondremos que el estudio se hizo tomando una muestra aleatoria simple de tamaño 300 de la población de tomadores de té que nos interesa. Podemos entonces usar el boostrap:

```
## paso 1: define el estimador
calc_estimador ← function(datos){
  prop_negro ← datos ▷
    mutate(negro = ifelse(Tea == "black", 1, 0)) ▷
    summarise(prop_negro = mean(negro), n = length(negro), .groups = "drop") ▷
    pull(prop_negro)
  prop_negro
}
```

```
## paso 2: define el proceso de remuestreo
muestra_boot ← function(datos){
  ## tomar muestra con reemplazo del mismo ñtamao
  sample_n(datos, size = nrow(datos), replace = TRUE)
}
```

```
## paso 3: definimos el paso bootstrap
paso_bootstrap ← function(id){
  muestra_boot(datos = te) ▷
    calc_estimador()
}
```

```
## paso 4: aplica el procedimiento bootstrap
prop_negro_tbl ← map_dbl(1:2000, paso_bootstrap ) ▷
  as_tibble() ▷
  rename( prop_negro = value)
```



Y podemos evaluar varios aspectos, por ejemplo dónde está centrada y qué tan dispersa es la distribución *bootstrap*:

```
prop_negro_tbl ▷
  summarise(
    cuantil_25 = quantile(prop_negro, 0.25),
    cuantil_75 = quantile(prop_negro, 0.75),
    media = mean(prop_negro),
    ee = sd(prop_negro)/sqrt(muestra.obs$n),
    sesgo = mean(prop_negro) - muestra.obs$media,
    .groups = "drop") ▷
  mutate(across(where(is.numeric), round, 4))
```

```
# A tibble: 1 × 5
  cuantil_25 cuantil_75 media     ee   sesgo
       <dbl>      <dbl> <dbl>  <dbl>   <dbl>
1       0.23      0.263 0.246 0.0014 -0.0002
```

4. PROPIEDADES DISTRIBUCIÓN BOOTSTRAP

Uasremos la distribución *bootstrap* principalmente para evaluar la variabilidad de nuestros estimadores (y también otros aspectos como sesgo) estimando la dispersión de la distribución de muestreo. Sin embargo, es importante notar que **no** la usamos, por ejemplo, para saber dónde está centrada la distribución de muestreo, o para "mejorar" la estimación remuestreando.

4.1. Ejemplo

En nuestro ejemplo, podemos ver varias muestras (por ejemplo 20) de tamaño 200. Podemos calcular las distribuciones de remuestreo para cada muestra bootstrap y compararlas con la distribución de muestreo real. El procedimiento es como sigue.

```
set.seed(911)
## Generamos 20 conjuntos de datos observados
muestras ← map(1:16, function(x) {
  muestra ← sample_n(poblacion_casas, 200, replace = F) ▷
    mutate(rep = x, tipo = "muestras")
}) ▷ bind_rows()
## Agregamos las columnas tipo y rep
dat_pob ← poblacion_casas ▷ mutate(tipo = "ópoblacin", rep = 1)
## Pegamos las tablas
datos_sim ← bind_rows(dat_pob, muestras)
```

```
## paso 1: define el estimador
calc_estimador ← function(datos){
  media_precio ← datos ▷
    summarise(media = mean(precio_miles), .groups = "drop") ▷
    pull(media)
  media_precio
}
```

```
## paso 2: define el proceso de remuestreo
muestra_boot ← function(datos, n = NULL){
  ## tomar muestra con reemplazo del mismo ñtamao
  if(is.null(n)){
```

```
      m ← sample_n(datos, size = nrow(datos), replace = TRUE)}
  else {
      m ← sample_n(datos, size = n, replace = TRUE)
    }
  m
}
```

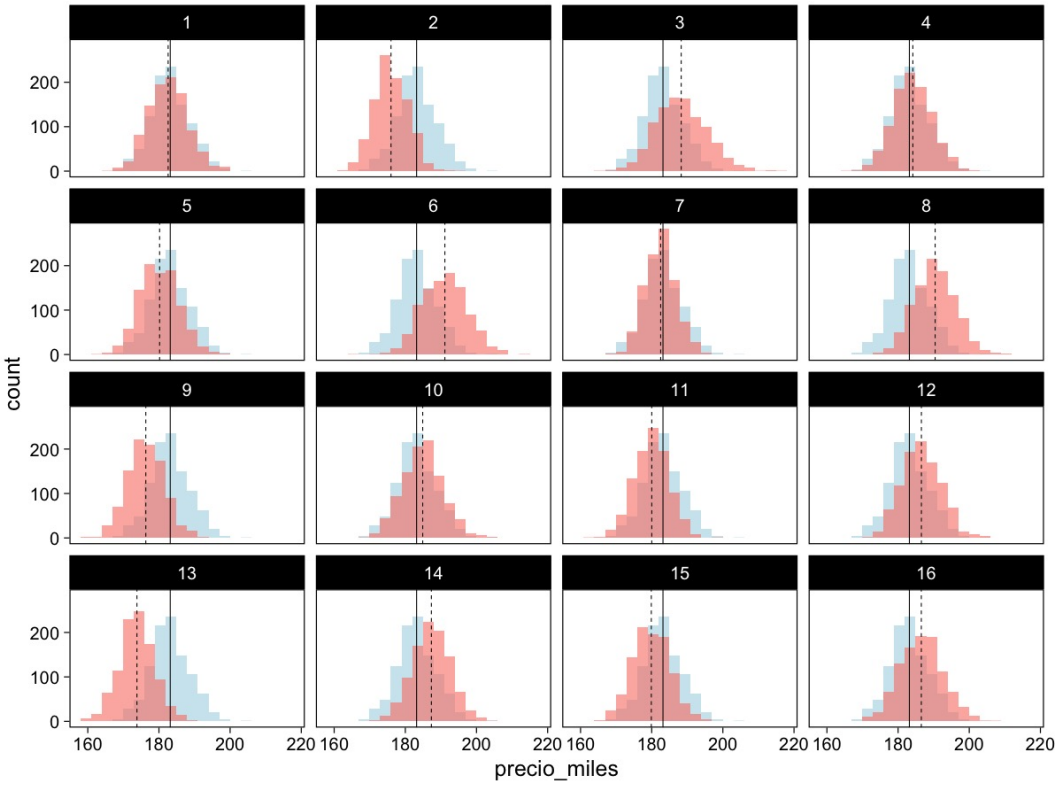
```
## paso 3: definimos el paso bootstrap
paso_bootstrap ← function(data, n = NULL){
  data ▷
    muestra_boot(n) ▷
    calc_estimador()
}
```

```
## paso 4: define el procedimiento bootstrap
procedimiento_bootstrap ← function(data){
  tibble(precio_miles = rerun(1000, paso_bootstrap(data)))
}
```

```
## paso 5: aplica el procedimiento bootstrap
dist_boot ← datos_sim ▷
  filter(tipo == "muestras") ▷
  select(precio_miles, rep) ▷
  group_by(rep) ▷ nest() ▷
  mutate(precio_miles =  map(data, procedimiento_boostrap)) ▷
  select(rep, precio_miles) ▷
  unnest(precio_miles) ▷
  mutate(precio_miles = unlist(precio_miles))

write_rds(dist_boot, "cache/sims_boot_precios.rds")
```

```
## extra: comparamos contra distribucion de muestreo
dist_muestreo ← datos_sim ▷
  filter(tipo == "ópoblacin") ▷
  group_by(rep) ▷ nest() ▷
  mutate(precio_miles =  map(data, function(data){
    tibble(precio_miles = rerun(1000, paso_bootstrap(data, n = 200)))
  })) ▷
  select(rep, precio_miles) ▷
  unnest(precio_miles) ▷
  mutate(precio_miles = unlist(precio_miles))
write_rds(dist_muestreo, "cache/sims_muestreo_precios.rds")
```



Obsérvese que:

- En algunos casos la aproximación es mejor que en otros (a veces la muestra tiene valores ligeramente más altos o más bajos).
- La dispersión de cada una de estas distribuciones *bootstrap* es similar a la de la verdadera distribución de muestreo (en rojo), pero puede está desplazada dependiendo de la muestra original que utilizamos.
- Adicionalmente, los valores centrales de la distribución de *bootstrap* tiende cubrir el verdadero valor que buscamos estimar, que es:

```
poblacion_casas ▷
  summarise(media = mean(precio_miles), .groups = "drop")
```

```
# A tibble: 1 × 1
  media
  <dbl>
1  183.
```

4.2. Variación en distribución bootstrap

En el proceso de estimación *bootstrap* hay dos fuentes de variación pues:

- La muestra original se selecciona con aleatoriedad de una población.
- Las muestras *bootstrap* se seleccionan con aleatoriedad de la muestra original. Esto es, la estimación *bootstrap* ideal es un resultado asintótico $B = \infty$, en esta caso $\hat{\mathsf{ee}}_B$ iguala la estimación *plug-in* $\mathsf{ee}_{\mathbb{P}_n}$.

En el proceso de *bootstrap* podemos controlar la variación del segundo aspecto, conocida como **implementación de muestreo Monte Carlo**, y la variación Monte Carlo decrece conforme incrementamos el número de muestras.

Podemos eliminar la variación Monte Carlo si seleccionamos todas las posibles muestras con reemplazo de tamaño n, hay $\binom{2n-1}{n}$ posibles muestras y si seleccionamos todas obtenemos $\hat{\text{ee}}_\infty$ (*bootstrap* ideal), sin embargo, en la mayor parte de los problemas no es factible proceder así.

4.2.1. `Ejercicio:` ¿Cuántas remuestras posibles existen para nuestro ejemplo introductorio de la diferencia de nivel de ingreso?

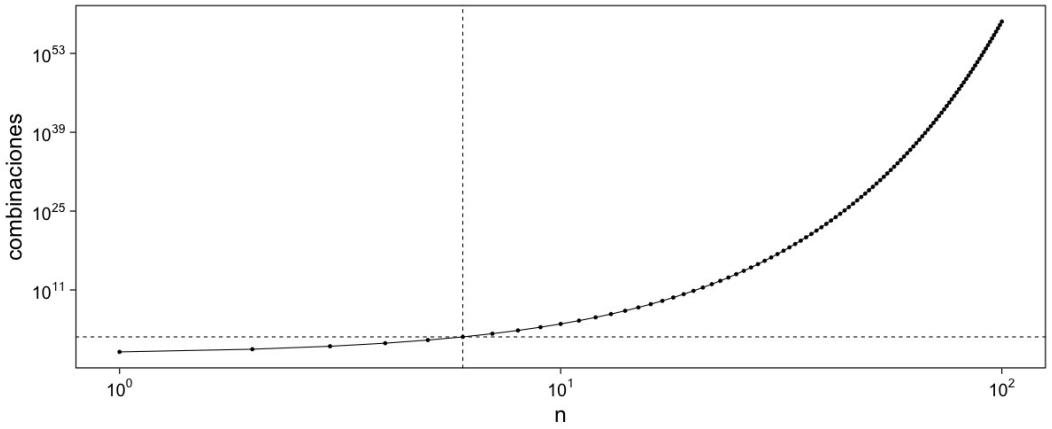
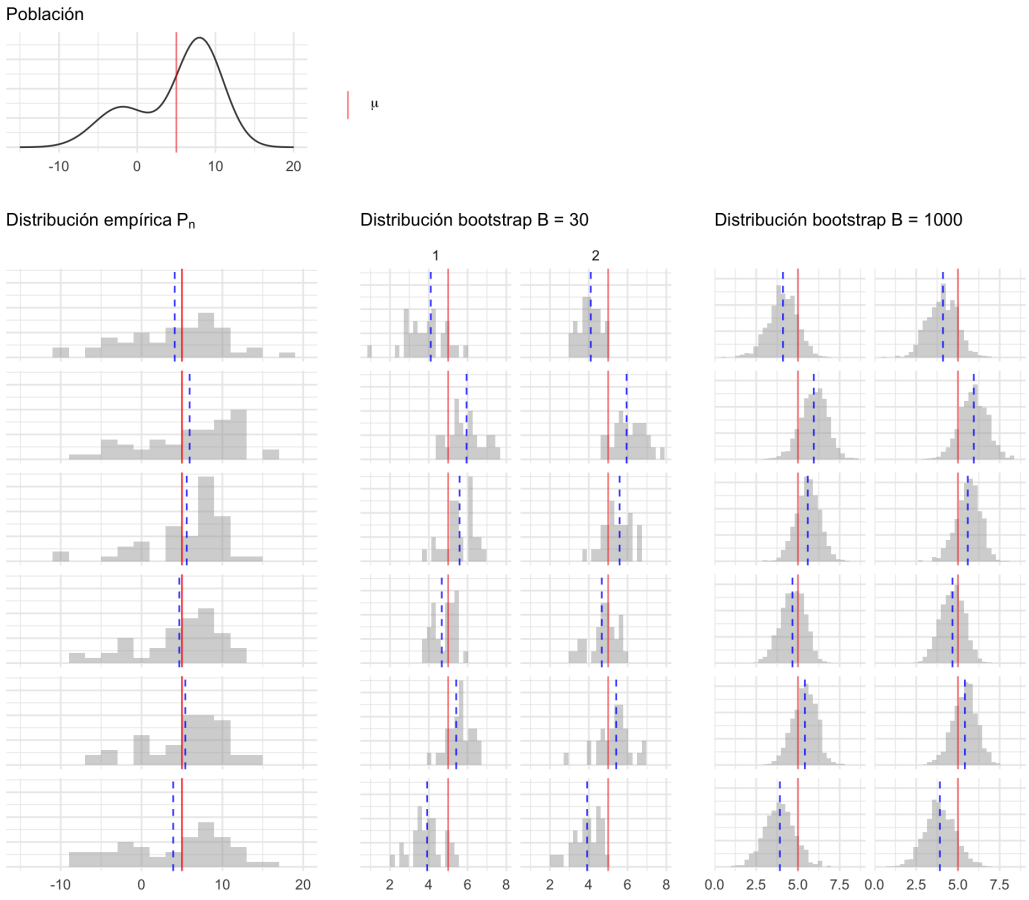


FIGURA 2. *Número de remuestras posibles como función del tamaño de muestra.*

En la siguiente gráfica mostramos 6 posibles muestras de tamaño 50 simuladas de la población, para cada una de ellas se graficó la distribución empírica y se se realizan histogramas de la distribución bootstrap con $B = 30$ y $B = 1000$, en cada caso hacemos dos repeticiones, notemos que cuando el número de muestras bootstrap es grande las distribuciones bootstrap son muy similares (para una muestra de la población dada), esto es porque disminuimos el erro Monte Carlo. También vale la pena recalcar que la distribución *bootstrap* está centrada en el valor observado en la muestra (línea azúl punteada) y no en el valor poblacional sin embargo la forma de la distribución es similar a lo largo de las filas.



Entonces, ¿cuántas muestras bootstrap?

1. Incluso un número chico de replicaciones bootstrap, digamos $B = 25$ es informativo, y $B = 50$ con frecuencia es suficiente para dar una buena estimación de $\mathsf{ee}_P(\hat{\theta})$ ([3]).
2. Cuando se busca estimar error estándar ([1]) recomienda $B = 1000$ muestras, o $B = 10,000$ muestras dependiendo la precisión que se busque.

4.3. Error estándar bootstrap

La variación de nuestro estimador es lo que conocemos como error estándar y lo denotamos por

$$\mathsf{ee}\left(\hat{\theta}_N\right) = \left(\mathbb{V}(\hat{\theta}_N)\right)^{1/2}. \tag{14}$$

El estimador *plug-in* de esta cantidad (el error estándar) es

$$\mathsf{ee}_{\hat{\mathbb{P}}_N}\left(\hat{\theta}^*_N\right), \tag{15}$$

donde $\hat{\theta}^*_N$ es un estimador *bootstrap* con $\hat{\theta}^*_N = s(X^*_1, \ldots, X^*_N)$ donde $X^*_1, \ldots, X^*_N \overset{\text{iid}}{\sim} \hat{\mathbb{P}}_N$.

4.3.1. **Definición** *[El error estándar* bootstrap *]:* El estimador *bootstrap* del error estándar se calcula por medio de remuestras $X^{(b)}_1, \ldots, X^{(b)}_N \overset{\text{iid}}{\sim} \hat{\mathbb{P}}_N$ con $b = 1, \ldots, B$ de acuerdo a

$$\hat{\mathsf{ee}}_B(\hat{\theta}^*) = \frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}^{(b)} - \hat{\theta}^{(\cdot)}\right)^2, \tag{16}$$

donde $\hat{\theta}^{(b)}$ denota el estimador de θ utilizando la remuestra b y $\hat{\theta}^{(\cdot)}$ es la media de las remuestras. Es decir,

$$\hat{\theta}^{(\cdot)} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^{(b)}\,. \tag{17}$$

Nota que

$$\lim_{B\to\infty}\hat{\mathsf{ee}}_B\left(\hat{\theta}^*\right) = \mathsf{ee}_{\hat{\mathbb{P}}_N}\left(\hat{\theta}^*\right)\,. \tag{18}$$

4.4. Estimación de sesgo

En tareas de estimación nos interesa cuantificar el sesgo de nuestros procedimientos. Por ejemplo, si nos interesa evaluar un estimador $\hat{\theta} = s(X_{1:N})$ de una característica $\theta = t(\mathbb{P})$ el sesgo del estimador está definido como

$$\mathsf{sesgo}_{\mathbb{P}} = \mathsf{sesgo}_{\mathbb{P}}(\hat{\theta},\theta) = \mathbb{E}_{\mathbb{P}}[\hat{\theta}] - t(\mathbb{P})\,. \tag{19}$$

Los estimadores *plug-in* usualmente tienen un sesgo pequeño. Pero si nos interesa poder cuantificarlo. Para esto utilizamos

$$\mathsf{sesgo}_{\hat{\mathbb{P}}} = \mathsf{sesgo}_{\hat{\mathbb{P}}}(\hat{\theta},\theta) = \mathbb{E}_{\hat{\mathbb{P}}}[\hat{\theta}^*] - t(\hat{\mathbb{P}})\,. \tag{20}$$

Nota que

$$\hat{\theta} = t(\hat{\mathbb{P}}) = s(X_{1:N})\,, \tag{21}$$

$$\mathbb{E}_{\hat{\mathbb{P}}}[\hat{\theta}^*] \approx \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^{(b)}\,, \qquad \text{donde} \qquad \hat{\theta}^{(b)} = s(X_1^{(b)},\ldots,X_N^{(b)})\,. \tag{22}$$

4.4.1. ***Observación:*** El sesgo y el error estándar están relacionados por medio del error cuadrático medio (MSE), el cual definimos como

$$\mathsf{MSE} = \mathsf{MSE}(\hat{\theta},\theta) = \mathbb{E}\left[(\hat{\theta}-\theta)^2\right]\,. \tag{23}$$

5. JACKKNIFE

Es una técnica originalmente propuesta para medir sesgo (Quenouille, 1949) y errores estándar (Tukey, 1958). Para una muestra $X_1,\ldots,X_N \overset{\text{iid}}{\sim} \mathbb{P}$ de observaciones se consideran N remuestreos de la forma

$$X_{(i)} = (X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_N)\,. \tag{24}$$

Se utilizan dichas remuestras para calcular nuestra colección de estimadores

$$\hat{\theta}_{(i)} = s\left(X_{(i)}\right)\,. \tag{25}$$

5.0.1. ***Nota:*** para estimadores *plug-in* de la forma $\hat{\theta} = t(\hat{\mathbb{P}})$, tenemos

$$\hat{\theta}_{(i)} = t\left(\hat{\mathbb{P}}_{(i)}\right)\,, \tag{26}$$

donde $\hat{\mathbb{P}}_{(i)}$ denota la función de distribución empírica de las $N-1$ observaciones.

5.1. Estimación de sesgo usando jackknife

El estimador del sesgo de *jackknife* se define como

$$\widehat{\mathsf{sesgo}}_{\mathsf{jack}} = (N-1)\left(\hat{\theta}_{(\cdot)} - \hat{\theta}\right), \tag{27}$$

donde

$$\hat{\theta}_{(\cdot)} = \frac{1}{N}\sum_{i=1}^{N}\hat{\theta}_{(i)}\,. \tag{28}$$

> Mas adelante discutiremos sobre el factor $N-1$ adelante de la estimación de sesgo. Pero por el momento puedes pensar en que los estimadores *jackknife* son promedios de $N-1$ observaciones, mientras que el estimador observado es producto de N observaciones. Por lo tanto, se necesite un factor que permita comparar ambas cantidades.

El uso de la estimación de sesgo es para poder presentar un estimador con una corrección por sesgo. Es decir, el estimador *jackknife* entonces es

$$\hat{\theta}_{\mathsf{jack}} = \hat{\theta} - \widehat{\mathsf{sesgo}}_{\mathsf{jack}}\,, \tag{29}$$

donde si utilizamos la sustitución adecuada obtenemos

$$\hat{\theta}_{\mathsf{jack}} = N\,\hat{\theta} - (N-1)\,\hat{\theta}_{(\cdot)}\,. \tag{30}$$

5.1.1. ***Ejercicio:*** Considera la situación de estimar la varianza a partir de una muestra $X_1, \ldots, X_N \overset{\mathsf{iid}}{\sim} \mathbb{P}$ por medio del estimador *plug-in*:

$$\hat{\theta} = \frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X}_N)^2\,. \tag{31}$$

Verifica que el estimador *jackknife* corregido por sesgo es el estimador insesgado usual

$$\hat{\theta}_{\mathsf{jack}} = \hat{\theta} - \widehat{\mathsf{sesgo}}_{\mathsf{jack}} = \frac{1}{N-1}\sum_{i=1}^{N}\left(X_i - \bar{X}_N\right)^2\,. \tag{32}$$

El sesgo del estimador *plug-in* satisface la siguiente ecuación:

$$\mathsf{sesgo}(\hat{\theta}) = \mathbb{E}(\hat{\theta} - \sigma^2) = \frac{N-1}{N}\sigma^2 - \sigma^2 = -\frac{\sigma^2}{N}\,. \tag{33}$$

El método *jackknife* calcula términos con sesgo igual a

$$\mathsf{sesgo}(\hat{\theta}_{(i)}) = -\frac{\sigma^2}{N-1}\,. \tag{34}$$

Por lo que, tenemos

$$\begin{aligned}
\mathbb{E}\left(\hat{\theta}_{(i)} - \hat{\theta}\right) &= \mathbb{E}\left(\hat{\theta}_{(i)} - \theta\right) - \mathbb{E}(\hat{\theta} - \theta) && (35)\\
&= \mathsf{sesgo}\left(\hat{\theta}_{(i)}\right) - \mathsf{sesgo}(\hat{\theta}) && (36)\\
&= -\frac{\sigma^2}{N-1} - \left(-\frac{\sigma^2}{N}\right) && (37)\\
&= -\frac{\sigma^2}{N(N-1)} = \frac{\mathsf{sesgo}(\hat{\theta})}{N-1}\,. && (38)
\end{aligned}$$

Lo que implica que el estimador *jackknife* con factor $N-1$ nos da una estimación correcta del sesgo de nuestros estimadores.

5.2. Estimación de errores estándar usando jackknife

El estimador del error estándar usando las remuestras *jackknife* se define como

$$\widehat{\text{ee}}_{\text{jack}} = \left[\frac{N-1}{N} \sum_{i=1}^{N} \left(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)} \right)^2 \right]^{1/2} . \tag{39}$$

5.2.1. Observación sobre el estimador usando jackknife Nota que necesitamos un `factor de inflación` para los términos cuadráticos. Esto es por que, intuitivamente, los términos son cercanos entre si y no caracterizan las desviaciones que observaríamos con promedios de N observaciones.

6. BOOSTRAP Y OTRAS ESTADÍSTICAS

El *bootstrap* es una técnica versátil. Un ejemplo son **estimadores de razón**, que tienen la forma

$$\hat{r} = \frac{\overline{y}}{\overline{x}} . \tag{40}$$

Por ejemplo, ¿cómo haríamos estimación para el procentaje de área area habitable de las casas en relación al tamaño del lote? Una manera de estimar esta cantidad es dividiendo la suma del área habitable de nuestra muestra y dividirlo entre la suma del área de los lotes de nuestra muestra, como en la fórmula anterior. Esta fórmula es más difícil pues tanto numerador como denominador tienen variabilidad, y estas dos cantidades no varían independientemente.

Con el *bootstrap* podemos atacar estos problemas.

6.1. Estimadores de razón

Nuestra muestra original es:

```
set.seed(250)
casas_muestra ← sample_n(poblacion_casas, 200)
casas_muestra ▷ as.data.frame() ▷ str()
```

```
'data.frame':   200 obs. of  46 variables:
 $ id                   : num  1166 855 579 1158 882 ...
 $ tipo_zona            : chr  "RL" "RL" "FV" "RL" ...
 $ frente_lote          : num  79 102 34 34 44 81 70 78 64 61 ...
 $ calle                : chr  "Pave" "Pave" "Pave" "Pave" ...
 $ forma_lote           : chr  "IR1" "Reg" "Reg" "IR1" ...
 $ nombre_zona          : chr  "NridgHt" "Sawyer" "Somerst" "NridgHt" ...
 $ tipo_edificio        : chr  "1Fam" "1Fam" "TwnhsE" "Twnhs" ...
 $ estilo               : chr  "1Story" "1Story" "2Story" "1Story" ...
 $ calidad_gral         : num  7 5 7 7 7 6 5 6 6 5 ...
 $ condicion_gral       : num  5 4 5 5 5 5 5 6 5 7 ...
 $ ñao_construccion     : num  2009 1955 2007 2007 1990 ...
 $ calidad_exteriores   : chr  "Gd" "TA" "Gd" "Gd" ...
 $ material_exteriores  : chr  "VinylSd" "Wd Sdng" "VinylSd" "VinylSd" ...
 $ condicion_exteriores : chr  "TA" "TA" "TA" "TA" ...
 $ calidad_sotano       : chr  "Gd" "TA" "Gd" "Gd" ...
 $ condicion_sotano     : chr  "TA" "TA" "TA" "TA" ...
 $ tipo_sotano          : chr  "Unf" "ALQ" "Unf" "GLQ" ...
 $ calefaccion          : chr  "GasA" "GasA" "GasA" "GasA" ...
 $ calidad_calefaccion  : chr  "Ex" "TA" "Ex" "Ex" ...
 $ aire_acondicionado   : chr  "Y" "Y" "Y" "Y" ...
```

```
 $ ñbaos_completos      : num  2 1 2 2 2 1 1 2 2 2 ...
 $ ñbaos_medios         : num  0 1 0 0 1 0 0 0 1 0 ...
 $ recamaras_sup        : num  3 3 2 2 3 3 3 3 3 3 ...
 $ calidad_cocina       : chr  "Gd" "TA" "Gd" "Gd" ...
 $ cuartos_sup          : num  7 6 5 6 7 5 6 7 7 5 ...
 $ tipo_garage          : chr  "Attchd" "Attchd" "Detchd" "Attchd" ...
 $ terminado_garage     : chr  "RFn" "Unf" "Unf" "RFn" ...
 $ num_coches           : num  2 2 2 2 2 0 0 2 2 2 ...
 $ calidad_garage       : chr  "TA" "TA" "TA" "TA" ...
 $ condicion_garage     : chr  "TA" "TA" "TA" "TA" ...
 $ ñao_venta            : num  2009 2006 2008 2009 2007 ...
 $ mes_venta            : num  9 7 2 7 4 5 12 6 2 9 ...
 $ tipo_venta           : chr  "New" "WD" "WD" "WD" ...
 $ condicion_venta      : chr  "Partial" "Abnorml" "Abnorml" "Normal" ...
 $ lat                  : num  42.1 42 42.1 42.1 42 ...
 $ long                 : num  -93.7 -93.7 -93.6 -93.7 -93.6 ...
 $ area_sotano_m2       : num  140 164 64 122 107 ...
 $ area_1er_piso_m2     : num  139.5 165.3 65.3 122.1 110.3 ...
 $ area_2o_piso_m2      : num  0 0 64 0 49.2 ...
 $ area_habitable_sup_m2: num  140 165 129 122 160 ...
 $ area_garage_m2       : num  59.8 42.2 50.2 58.2 37.2 ...
 $ area_lote_m2         : num  886 1665 335 465 1278 ...
 $ precio_miles         : num  233 170 146 230 188 ...
 $ valor_misc_miles     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ precio_m2_miles      : num  1.67 1.03 1.13 1.88 1.18 ...
 $ precio_m2            : num  1671 1029 1129 1884 1175 ...
```

El estimador de interés es:

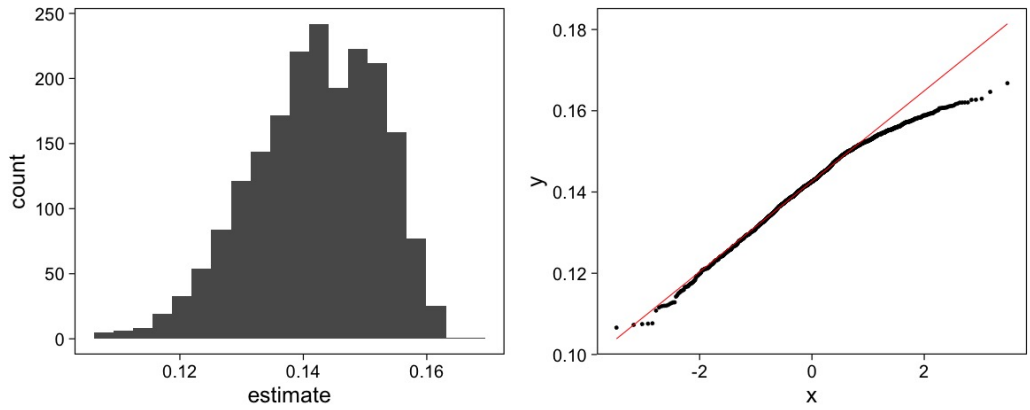
```
estimador_razon ← function(split, ...){
  muestra ← analysis(split)
  muestra ▷
    summarise(estimate = sum(area_habitable_sup_m2) / sum(area_lote_m2),
              .groups = "drop") ▷
    mutate(term = "area del lote construida")
}
```

Y nuestra estimación puntual es

```
estimador ← casas_muestra ▷
  summarise(estimate = sum(area_habitable_sup_m2) / sum(area_lote_m2))
estimador
```

```
# A tibble: 1 × 1
  estimate
     <dbl>
1    0.141
```

Es decir que en promedio, un poco menos del 15 % del lote total es ocupado por área habitable. Ahora hacemos bootstrap para construir un intervalo:



En este caso la cola derecha parece tener menos dispersión que una distribución normal. Usamos un intervalo de percentiles para obtener:

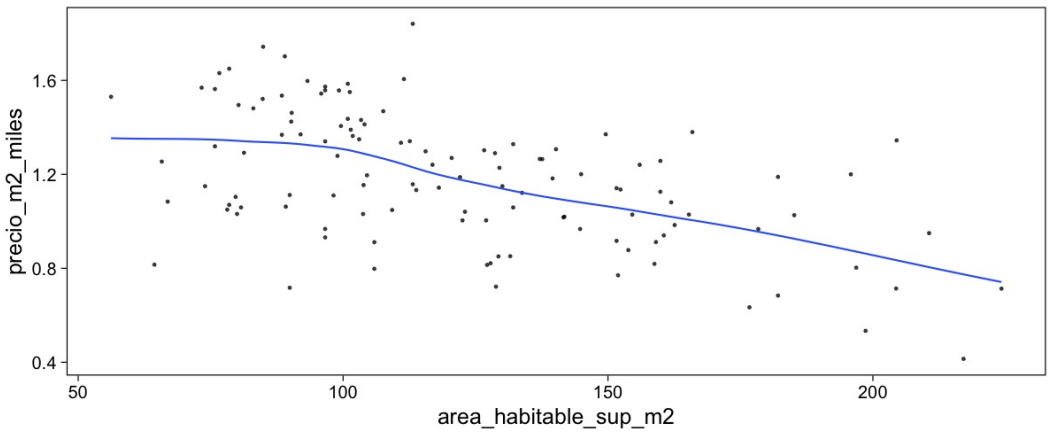
```
dist_boot ▷ int_pctl(res_boot) ▷
  mutate(estimador = estimador$estimate) ▷
  rename(media_boot = .estimate) ▷
  mutate(sesgo = media_boot - estimador) ▷
  select(-.method, -term)
```

```
# A tibble: 1 × 6
  .lower media_boot .upper .alpha estimador   sesgo
   <dbl>      <dbl>  <dbl>  <dbl>     <dbl>   <dbl>
1  0.121      0.142  0.159   0.05     0.141 0.00101
```

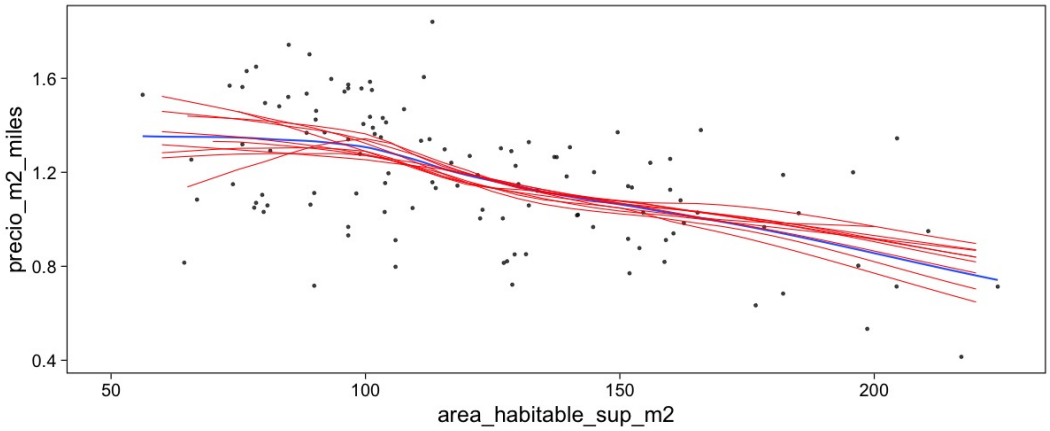
Nótese que el sesgo es bajo. De modo que en esta zona, entre 12 % y 16 % de toda el área disponible es ocupada por área habitable: estas son casas que tienen jardines o terrenos, garage relativamente grandes.

6.2. Suavizadores

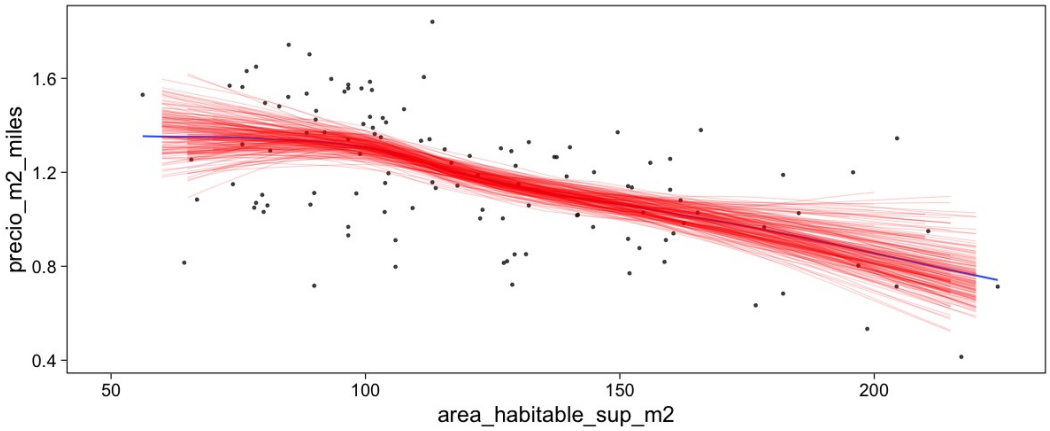
Podemos usar el *bootstrap* para juzgar la variabilidad de un suavizador, que consideramos como nuestra estadística:



Podemos hacer bootstrap para juzgar la estabilidad del suavizador:



Donde vemos que algunas cambios de pendiente del suavizador original no son muy interpretables (por ejemplo, para áreas chicas) y alta variabilidad en general en los extremos. Podemos hacer más iteraciones para calcular bandas de confianza:



Donde observamos cómo tenemos incertidumbre en cuanto al nivel y forma de las curvas en los extremos de los datos (casas grandes y chicas), lo cual es natural. Aunque podemos resumir para hacer bandas de confianza, mostrar remuestras de esta manera es informativo: por ejempo: vemos cómo es probable también que para casas de menos de 70 metros cuadrados el precio por metro cuadrado no cambia tanto (líneas constantes).

REFERENCIAS

[1] L. M. Chihara and T. C. Hesterberg. *Mathematical Statistics with Resampling and R*. John Wiley & Sons, Inc., Hoboken, NJ, USA, aug 2018. ISBN 978-1-119-50596-9 978-1-119-41654-8. . 1, 8, 16

[2] B. Efron and T. Hastie. *Computer Age Statistical Inference*. Institute of Mathematical Statistics Monographs. Cambridge University Press, 2016. ISBN 978-1-107-14989-2. 1

[3] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Springer US, Boston, MA, 1993. ISBN 978-0-412-04231-7 978-1-4899-4541-9. . 1, 9, 16

[4] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer Texts in Statistics. Springer New York, New York, NY, 2004. ISBN 978-1-4419-2322-6 978-0-387-21736-9. . 1, 10