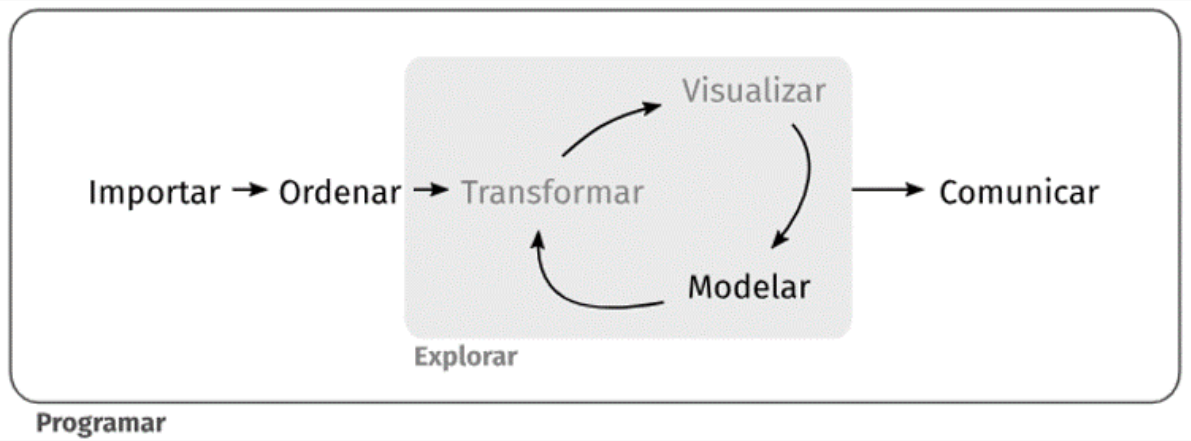
Ciencia de Datos para Políticas Públicas

Módulo 2 - Clase 4: Intervalo de confianza / Prueba de hipótesis

Pablo Aguirre Hormann
06/07/2021

¿Qué veremos hoy?

- Visualización de datos
- Manejo de datos
- Transformación de datos
- **Inferencia Estadística/Econometría**
  - Intervalos de confianza
  - Prueba de hipótesis



Describir vs Inferir

- La estadística descriptiva es una rama que apunta a **resumir información** de la mejor manera.
- Es decir, reducir datos a un par de indicadores y/o visualizaciones tratando de perder la menor cantidad de información.
- La estadística descriptiva **no conlleva incertidumbre** ya que no buscamos extrapolar estas descripciones a otros datos.
- Por otro lado, la estadística inferencial busca **obtener conclusiones/aprendizajes** sobre algún fenómeno usando muestras.
- En otras palabras, usamos información de una **muestra** para concluir algo sobre una **población**.
- Dos de las herramientas más importantes en la estadística inferencial son los **intervalos de confianza** y las **pruebas de hipótesis**.

Pie para la próxima clase

```
Call:
lm(formula = ROLL ~ UNEM + HGRAD + INC, data = datavar)

Residuals:
      Min        1Q    Median        3Q       Max
-1148.840  -489.712    -1.876   387.400  1425.753

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.153e+03  1.053e+03  -8.691 5.02e-09 ***
UNEM         4.501e+02  1.182e+02   3.809 0.000807 ***
HGRAD        4.065e-01  7.602e-02   5.347 1.52e-05 ***
INC          4.275e+00  4.947e-01   8.642 5.59e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 670.4 on 25 degrees of freedom
Multiple R-squared: 0.9621, Adjusted R-squared: 0.9576
F-statistic: 211.5 on 3 and 25 DF,  p-value: < 2.2e-16
```

Teorema del Límite Central (TLC)

(y Ley de los Grandes Números - LGN)

TLC/LGN

Si ciertas condiciones se cumplen, las estimaciones muestrales se distribuirán de forma normal con media igual al parámetro poblacional. La dispersión será inversamente proporcional al tamaño muestral

Población vs Muestra

- **Población**, N: grupo bien definido de sujetos (por ejemplo, población de un país).
- **Muestra**, n: Subconjunto de individuos provenientes de una población que se obtienen a través de algún procedimiento de muestreo (ej. muestreo aleatorio simple).

Censo

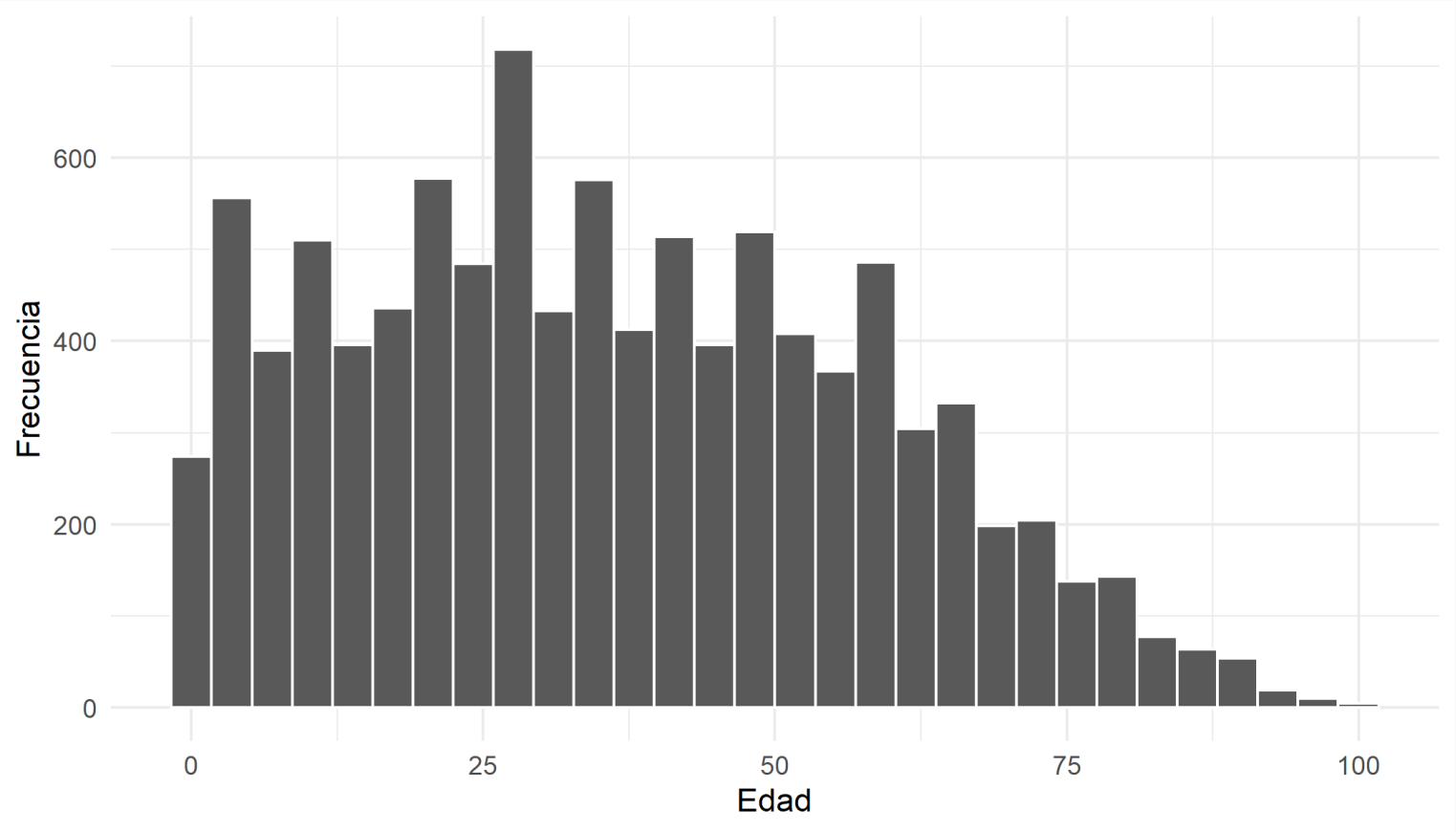
Asumamos que estos datos corresponden al total del país ($N = 10.000$).

```
censo ← read_csv("../datos/muestra_censo_2017.csv")
str(censo)
```

```
## tibble [10,000 x 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ x           : num [1:10000] 1 2 3 4 5 6 7 8 9 10 ...
##  $ edad        : num [1:10000] 35 37 59 29 76 65 25 16 49 35 ...
##  $ sexo        : chr [1:10000] "M" "H" "H" "M" ...
##  $ p_originario: chr [1:10000] "no" "no" "no" "no" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   x = col_double(),
##   ..   edad = col_double(),
##   ..   sexo = col_character(),
##   ..   p_originario = col_character()
##   .. )
```

Analicemos la variable `edad`.

Distribución de edad



Claramente no sigue una distribución normal

Sabemos μ y σ

Debido a que tenemos la información de todas las personas sabemos cual es el promedio y la desviación estándar de la población.

```
(datos_poblacion ← censo %>%
  summarise(promedio = mean(edad, na.rm = TRUE),
            sd = sd(edad, na.rm = TRUE)))
```

```
##   promedio       sd
## 1  36.0179 22.14836
```

Pero **practicamente nunca sabremos estos parámetros poblacionales**, y lo que tenemos que hacer es sacar conclusiones desde **muestras** (hacer estimaciones).

Si sacamos una muestra desde una población y estimamos un parámetro, por ejemplo la media llamamos a esto una **estimación puntual**. A esta estimación la llamaremos $\hat{\mu}$.

Muestras

Una muestra aleatoria de 100 observaciones ($n = 100$)

```
censo %>% sample_n(100) %>%
  summarise(promedio = mean(edad))
```

```
##   promedio
## 1    39.08
```

$\hat{\mu} = 39.08$

Una segunda muestra aleatoria de 100 observaciones ($n = 100$)

```
censo %>% sample_n(100) %>%
  summarise(promedio = mean(edad))
```

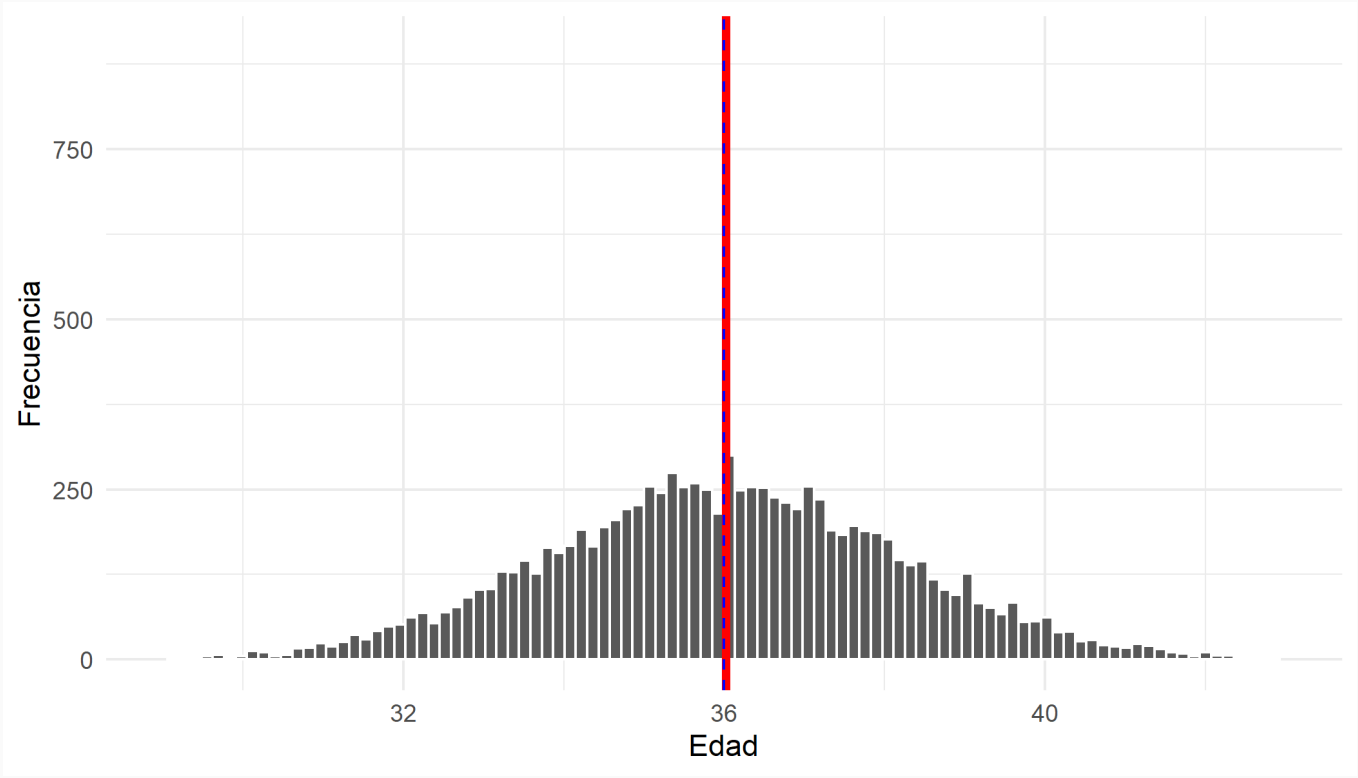
```
##   promedio
## 1    36.15
```

$\hat{\mu} = 36.15$

¿Qué pasa si repetimos esto muchas veces?
¿Cómo se distribuyen todas las estimaciones?

10.000 muestras con n=100

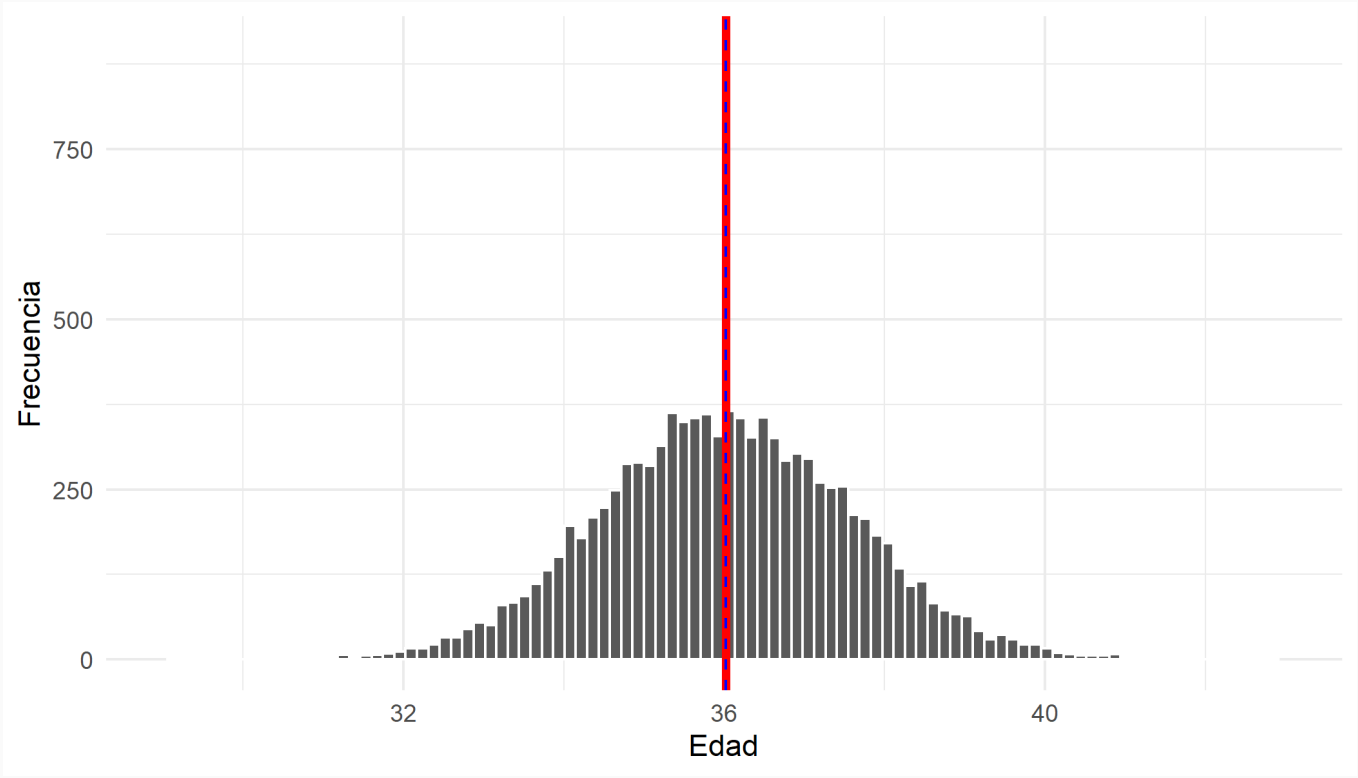
Sacamos **10.000 muestras aleatorias de 100 individuos**, calculamos el promedio de edad para cada muestra, y graficamos la distribución de cada uno de los promedios calculados:



¡La distribución de las estimaciones de cada muestra aproximan una distribución normal con media igual a la media poblacional! Decimos entonces, que nuestra estimación es insesgada.

10.000 muestras con n=200

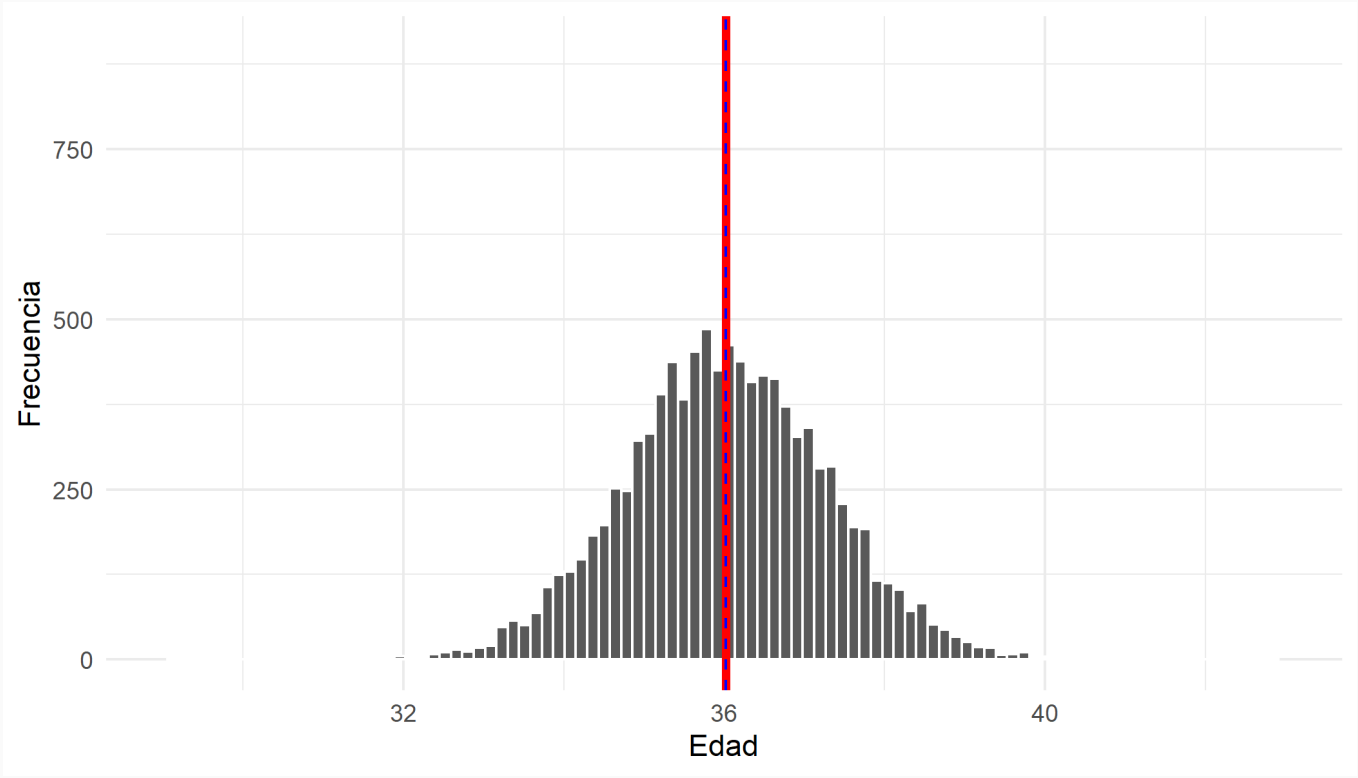
Sacamos **10.000 muestras aleatorias de 200 individuos**, calculamos el promedio de edad para cada muestra, y graficamos la distribución de cada uno de los promedios calculados:



¡La distribución de las estimaciones de cada muestra aproximan una distribución normal con media igual a la media poblacional! Decimos entonces, que nuestra estimación es insesgada. **¡Y a mayor tamaño muestral menor la dispersión!**

10.000 muestras con n=300

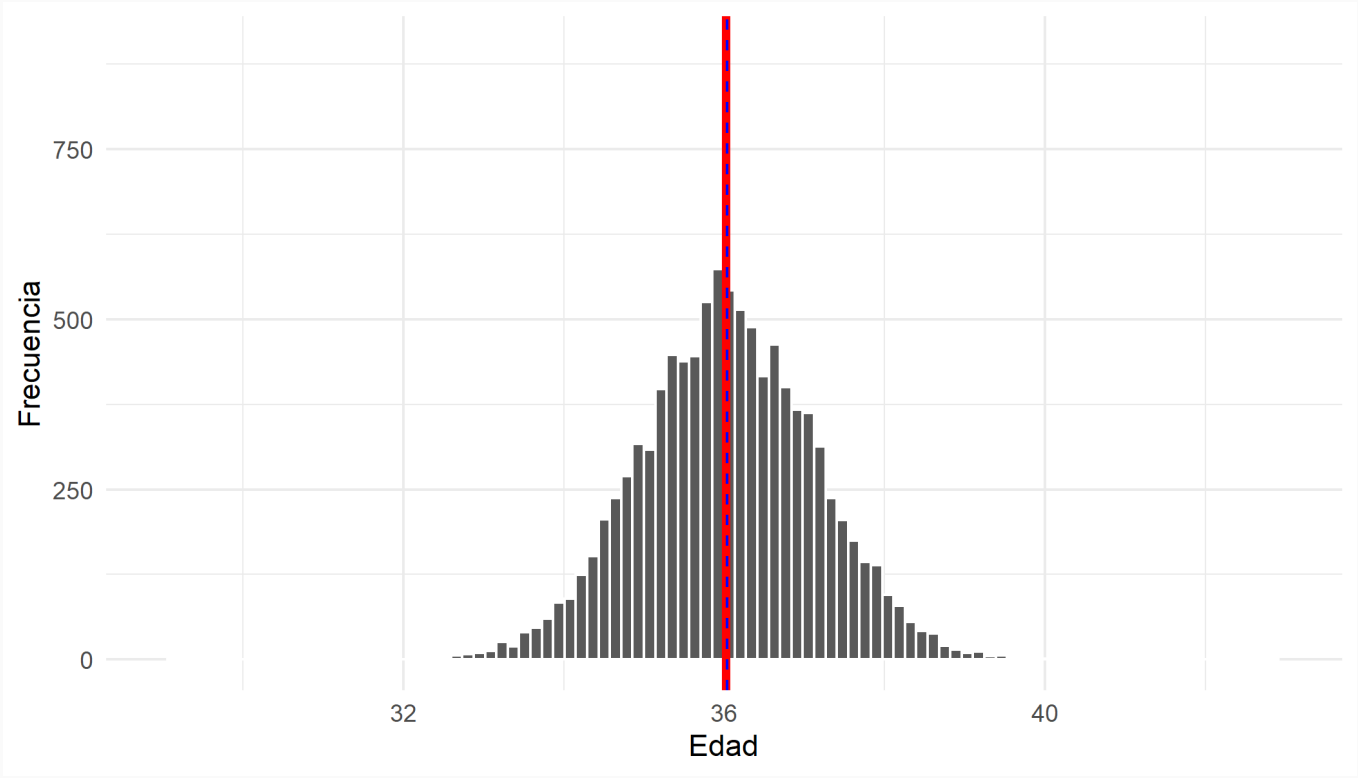
Sacamos **10.000 muestras aleatorias de 300 individuos**, calculamos el promedio de edad para cada muestra, y graficamos la distribución de cada uno de los promedios calculados:



¡La distribución de las estimaciones de cada muestra aproximan una distribución normal con media igual a la media poblacional! Decimos entonces, que nuestra estimación es insesgada. **¡Y a mayor tamaño muestral menor la dispersión!**

10.000 muestras con n=400

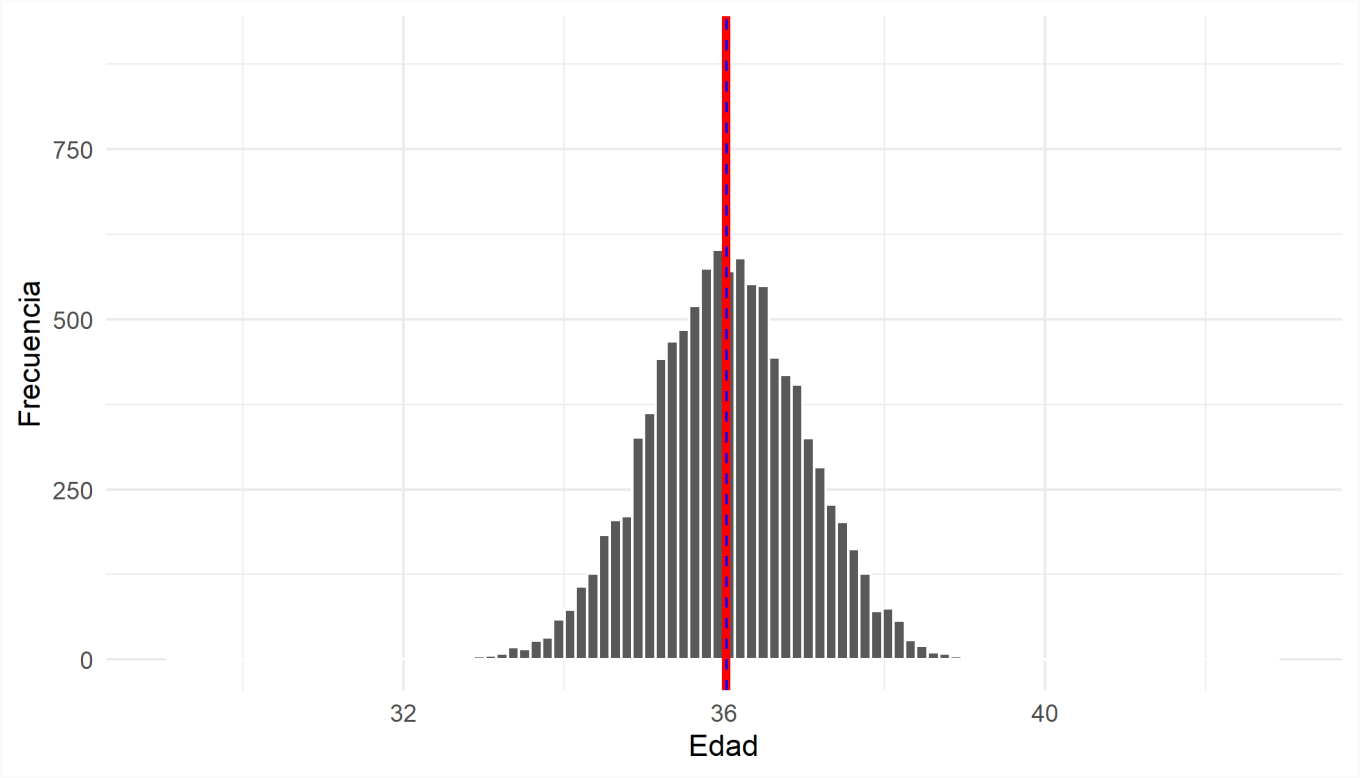
Sacamos **10.000 muestras aleatorias de 400 individuos**, calculamos el promedio de edad para cada muestra, y graficamos la distribución de cada uno de los promedios calculados:



¡La distribución de las estimaciones de cada muestra aproximan una distribución normal con media igual a la media poblacional! Decimos entonces, que nuestra estimación es insesgada. **¡Y a mayor tamaño muestral menor la dispersión!**

10.000 muestras con n=500

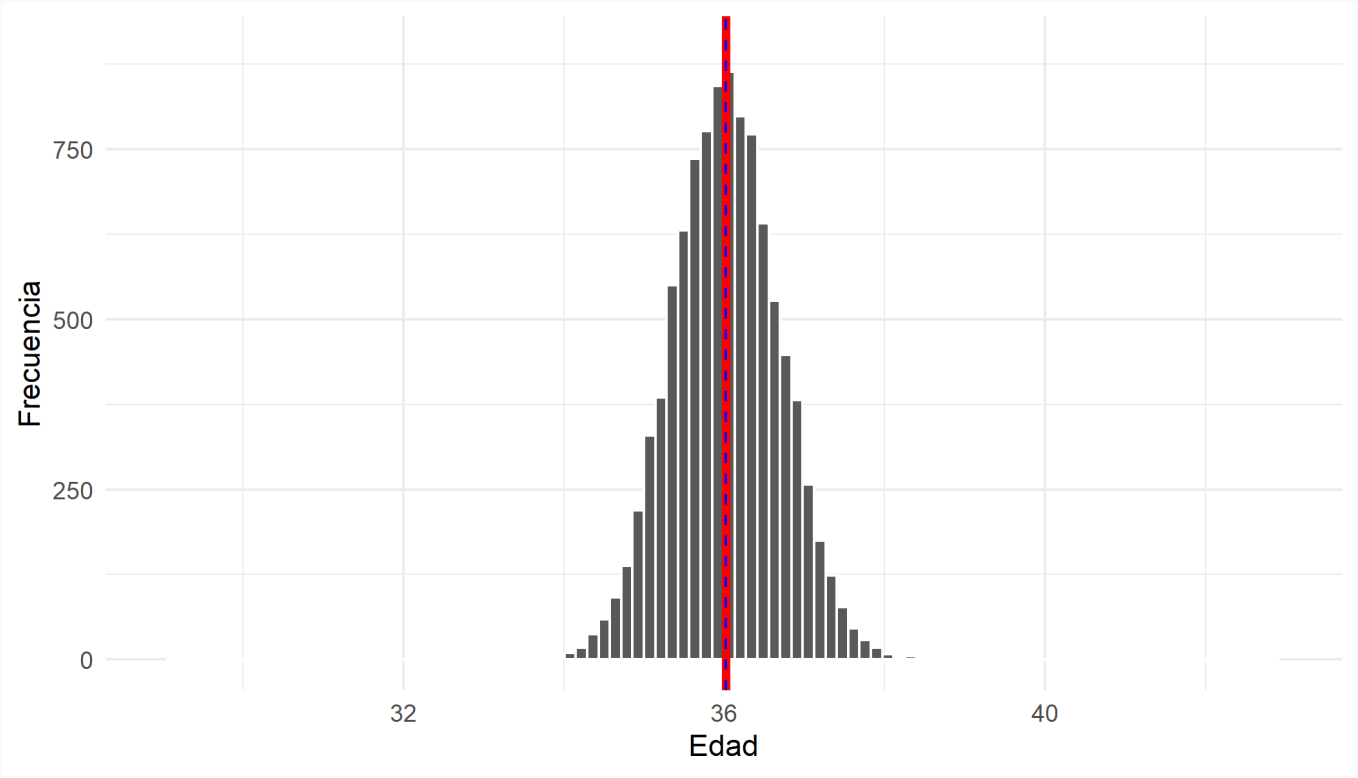
Sacamos **10.000 muestras aleatorias de 500 individuos**, calculamos el promedio de edad para cada muestra, y graficamos la distribución de cada uno de los promedios calculados:



¡La distribución de las estimaciones de cada muestra aproximan una distribución normal con media igual a la media poblacional! Decimos entonces, que nuestra estimación es insesgada. **¡Y a mayor tamaño muestral menor la dispersión!**

10.000 muestras con n=1000

Sacamos **10.000 muestras aleatorias de 1000 individuos**, calculamos el promedio de edad para cada muestra, y graficamos la distribución de cada uno de los promedios calculados:



¡La distribución de las estimaciones de cada muestra aproximan una distribución normal con media igual a la media poblacional! Decimos entonces, que nuestra estimación es insesgada. **¡Y a mayor tamaño muestral menor la dispersión!**

Comparar distribuciones muestrales

$n = 100$

```
mean(guardar_mediacenso)
## [1] 35.99654
sd(guardar_mediacenso)
## [1] 2.212659
```

$n = 200$

```
mean(guardar_mediacenso2)
## [1] 36.02233
sd(guardar_mediacenso2)
## [1] 1.565967
```

$n = 300$

```
mean(guardar_mediacenso3)
## [1] 36.02401
sd(guardar_mediacenso3)
## [1] 1.248814
```

$n = 400$

```
mean(guardar_mediacenso4)
## [1] 36.03845
sd(guardar_mediacenso4)
## [1] 1.091304
```

$n = 500$

```
mean(guardar_mediacenso5)
## [1] 36.02631
sd(guardar_mediacenso5)
## [1] 0.9569096
```

$n = 1000$

```
mean(guardar_mediacenso6)
## [1] 36.0231
sd(guardar_mediacenso6)
## [1] 0.6589096
```

TLC/LGN

Si ciertas condiciones se cumplen, las estimaciones muestrales se distribuirán de forma normal con media igual al parámetro poblacional. La dispersión será inversamente proporcional al tamaño muestral

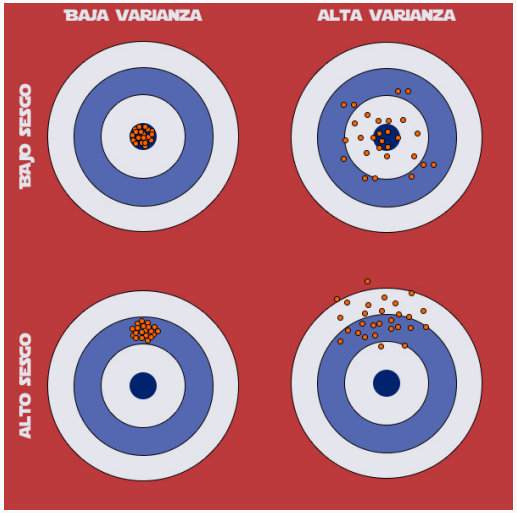
$$\hat{\mu} \sim aproximadamente\, \mathcal{N}(\mu, \frac{\sigma^2}{n})$$

$\sqrt{\frac{\sigma^2}{n}} = \frac{\sigma}{\sqrt{n}} = error\ est\acute{a}ndar$

A tener en cuenta

En general, si sacamos una **muestra aleatoria** de tamaño n desde una población N, entonces:

- La muestra es **insesgada** y **representativa** de la población. Y a mayor n menor dispersión.
- Los resultados basados en la muestra podrían ser **generalizados a la población**.
- La estimación puntual, $\hat{\mu}$, es una **"buena suposición"** del parametro poblacional desconocido, μ.
- Entonces, en vez de hacer un censo (costoso en muchos sentidos), podemos hacer **inferencia sobre una población usando muestreo**.



Intervalo de confianza

¿Qué acabamos de hacer?

- Al hacer muestras repetitivas desde una población, obtenemos la **distribución muestral de la media de** `edad` y podemos ver que se distribuye normalmente.
- Al tomar distintas muestras pudimos ver que las estimaciones puntuales variaban. Esto lo denominamos **variación muestral** y se puede cuantificar usando el **error estándar**. A mayor n (tamaño muestral), menor error estándar (o estimaciones más precisas).
- Ahora bien, en "la vida real" no podremos hacer lo que mostramos ya que generalmente **contaremos con una muestra (que ojalá sea lo más grande posible)**. Además, no sabremos el valor del parámetro real que queremos estimar.
- ¿Cómo podemos considerar los efectos de la variación muestral si -usualmente- tenemos solo una muestra?
- Para esto ocuparemos un **método de remuestreo** conocido como **bootstrapping**. Esto nos permitirá también obtener un rango de valores posibles para nuestro parámetro. Este rango de valores es lo que conocemos como **intervalos de confianza**.

Asumamos que no sabemos μ

¿Cuál es la edad promedio en Chile?

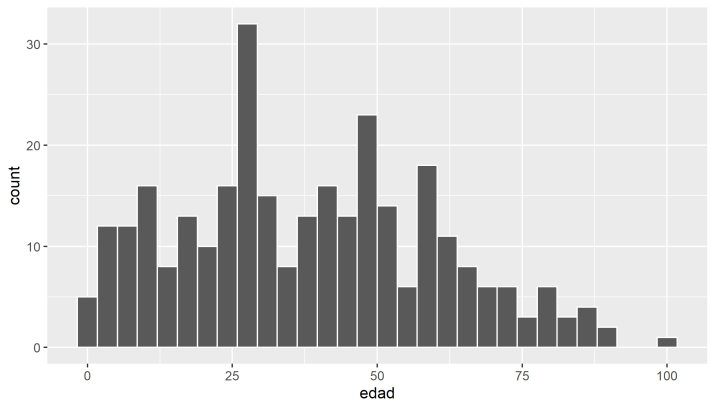
- Para responder esta pregunta podríamos entrevistar a todas las personas del país y preguntarles su edad (censo).
- Claramente esto es algo muy costoso por lo que normalmente lo que haríamos es **tomar una muestra de la población**. Digamos que en este caso solo contamos con una muestra, n, de **300 personas** obtenida desde la población, N, de 10.000.

```
set.seed(1) # para tener los mismos resultados
(muestra_censo ← censo %>%
    sample_n(300) %>%
    mutate(Id = row_number()) %>%
    select(Id, edad))
```

```
##       Id edad
## 1      1   89
## 2      2   51
## 3      3   48
## 4      4    8
## 5      5   38
## 6      6   17
## 7      7   18
## 8      8   23
## 9      9   38
## 10    10   27
## 11    11    9
## 12    12   91
## 13    13   86
## 14    14   28
## 15    15   59
## 16    16   48
## 17    17   76
## 18    18   14
## 19    19    3
## 20    20   29
## 21    21   11
## 22    22    3
## 23    23   53
```

Explorar la muestra

```
muestra_censo %>%
  ggplot(aes(x = edad)) +
  geom_histogram(color = "white")
```



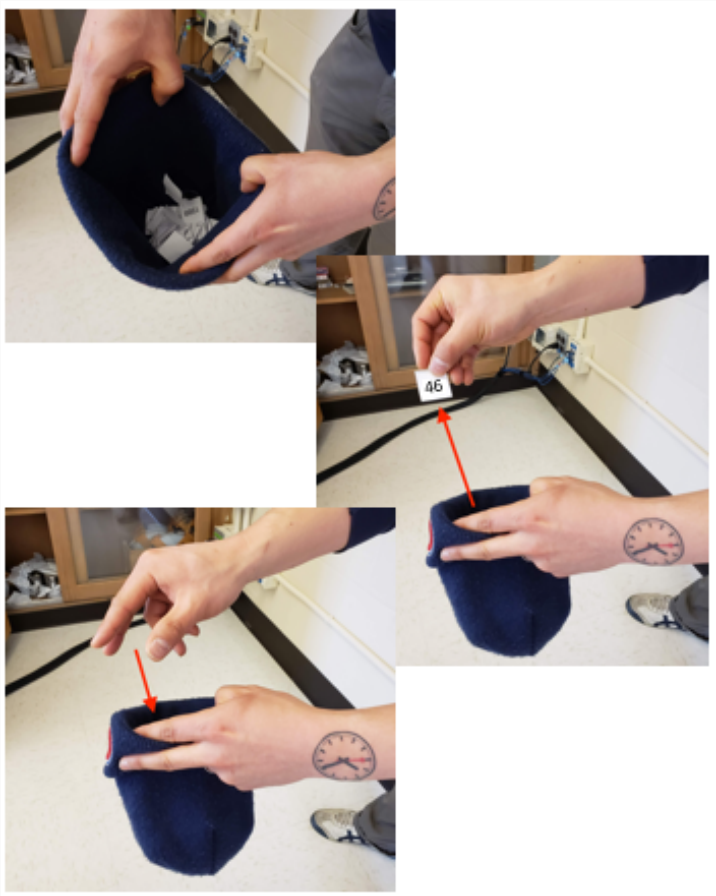

```
(edad_promedio_muestra ← muestra_censo %>%
  summarise(promedio_edad = mean(edad,
                                 na.rm = TRUE)))
```

```
##   promedio_edad
## 1      38.12667
```

- Si estamos dispuestos a asumir que `muestra_censo` es una muestra representativa de nuestra población, entonces **una "buena suposición" de la edad promedio** de Chile sería **38.13**.
- $\hat{\mu} = 38.13$ es nuestra estimación de μ (que en la práctica sería desconocido).
- Antes calculamos los efectos de la variación muestral sacando muchas muestras repetitivamente pero **ahora solo contamos con una muestra**.

Bootstrapping

1. Consideremos nuestra muestra de $n = 300$ observaciones/personas.
2. Imaginemos que ponemos 300 papeles con las edades de nuestra muestra en un gorro.
3. Sacaremos una observación, registraremos su valor (ej. edad = 46) y **pondremos de vuelta el papel** en el gorro
4. **Repetiremos el paso 3.** tantas veces como sea nuestro n. En este caso 300 veces.
5. Terminaremos con una **remuestra** con $n = 300$ creada a partir de nuestra única muestra original, `muestra_censo`



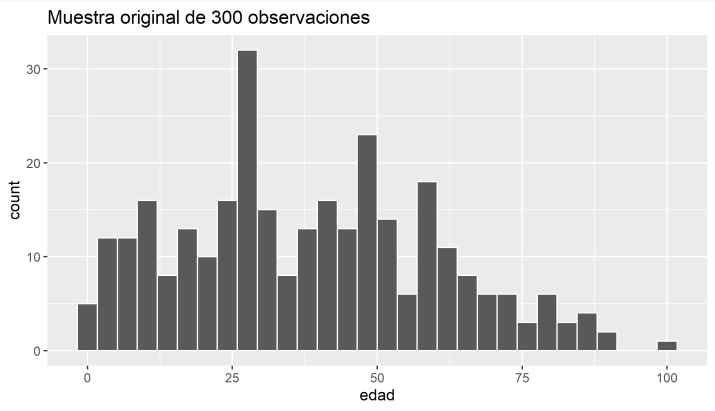
Bootstrapping

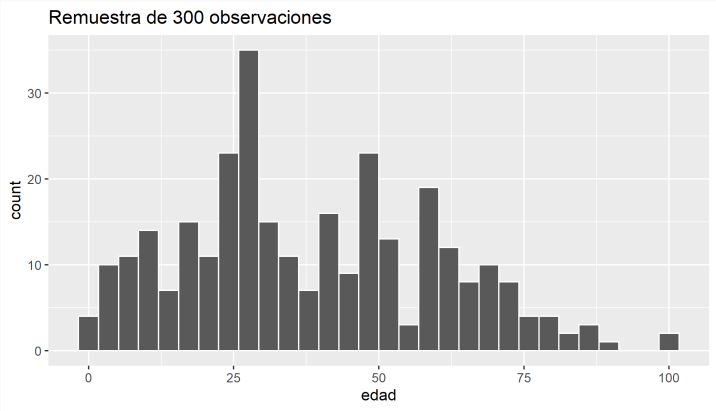
- Lo que acabamos de hacer es una **remuestra** desde la muestra original. No estamos yendo a la población, N, a buscar otras $n = 300$ personas.
- **¿Por qué volvemos a "poner en el gorro" cada valor remuestrado?** Porque de no hacerlo terminaríamos con exactamente la misma muestra original. Hacer el acto de "devolver" cada papel nos intruduce **variación muestral**.
- En otras palabras, lo que hacemos es un **muestreo con reemplazo** desde la muestra original de 300 observaciones.

```
set.seed(5)
censo_muestra_r1 ← muestra_censo %>%
  sample_n(300,
           replace = TRUE)
```

Analizar la remuestra

Si observamos como se distribuye nuestra muestra original, `muestra_censo`, y la remuestra que acabamos de hacer, `censo_muestra_r1`, vemos que son similares (no idénticas).






```
##   promedio_edad
## 1      38.12667
```

```
##   promedio_edad
## 1      38.22667
```

Obtenemos también un promedio distinto al calculado originalmente y esta **variación es debido al remuestreo con reemplazo** que hicimos.

¿Qué pasaría si repetimos este ejercicio de remuestreo muchas veces? **Ojo, esto si es algo factible "en la vida real"**.

Muchas remuestras

1. Consideremos nuestra muestra de $n = 300$ observaciones/personas.
2. Imaginemos que ponemos 300 papeles con las edades de nuestra muestra en un gorro.
3. Sacaremos una observación, registraremos su valor (ej. edad = 46) y pondremos de vuelta el papel en el gorro
4. Repetiremos el paso **3.** tantas veces como sea nuestro n. En este caso 300 veces.
5. Terminaremos con una **remuestra** con $n = 300$ creada a partir de nuestra única muestra original, `muestra_censo`
6. **Hacemos lo anterior 1.000 veces**.

Muchas remuestras

1. Consideremos nuestra muestra de $n = 300$ observaciones/personas.
2. Imaginemos que ponemos 300 papeles con las edades de nuestra muestra en un gorro.
3. Sacaremos una observación, registraremos su valor (ej. edad = 46) y pondremos de vuelta el papel en el gorro
4. Repetiremos el paso **3.** tantas veces como sea nuestro n. En este caso 300 veces.
5. Terminaremos con una **remuestra** con $n = 300$ creada a partir de nuestra única muestra original, `muestra_censo`
6. **Hacemos lo anterior 1.000 veces**.

```
(guardar_remuestras ← read_csv("../datos/remuestras_edad.csv"))
```

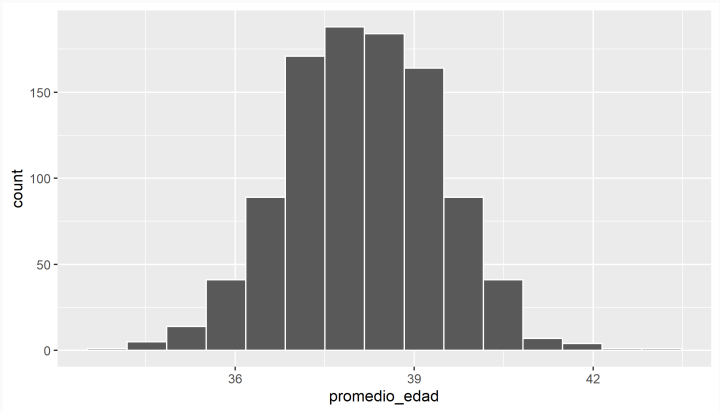
```
## # A tibble: 300,000 x 2
##     edad remuestra
##    <dbl>     <dbl>
##  1    35         1
##  2    37         1
##  3    26         1
##  4     6         1
##  5    24         1
##  6    49         1
##  7    60         1
##  8    14         1
##  9    24         1
## 10    23         1
## #  ... with 299,990 more rows
```

Muchas remuestras

```
(promedio_remuestras ←
   guardar_remuestras %>%
      group_by(remuestra) %>%
      summarise(promedio_edad = mean(edad,
                                     na.rm = TRUE)))
```

```
## # A tibble: 1,000 x 2
##    remuestra promedio_edad
##        <dbl>         <dbl>
##  1         1          38.8
##  2         2          38.4
##  3         3          37.7
##  4         4          39.1
##  5         5          36.7
##  6         6          36.1
##  7         7          37.1
##  8         8          37.2
##  9         9          37.5
## 10        10          39.3
## # … with 990 more rows
```

```
promedio_remuestras %>%
  ggplot(aes(x = promedio_edad)) +
  geom_histogram(bins = 15, color = "white")
```



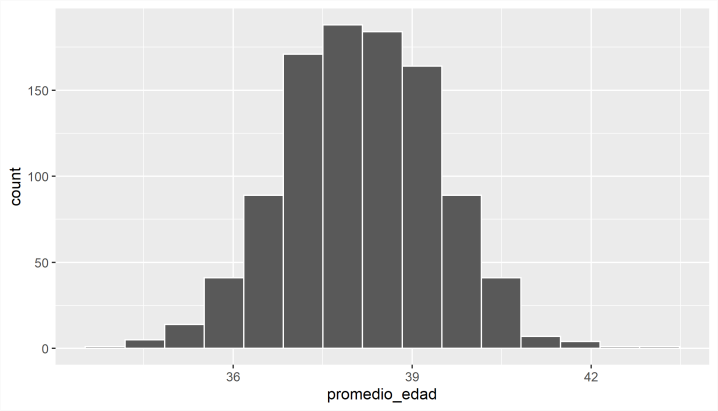
¿Qué hicimos?

- Usamos **bootstrap como una forma de representar la variación muestral** vista anteriormente.
- La distribución que vimos recién se denomina **distribución bootstrap** y es una **aproximación de la distribución muestral** de la media.
- La distribución bootstrap probablemente **no tendrá el mismo "centro"** que la distribución muestral. En otras palabras, **bootstrap no nos permite mejorar la "calidad" de nuestra estimación**.
- Pero, la distribución bootstrap si **tendrá una forma y dispersión similar a la distribución muestral**. Entonces, si nos da una **buena estimación del error estándar**.
- Este último punto nos permitirá construir **intervalos de confianza**.

Entendiendo intervalos de confianza

- Podemos pescar tanto con una **caña** como con una **red**. La red probablemente te permite pescar más pescados que la caña.
- Digamos que μ, el parámetro a estimar, es un pescado.
- Una estimación puntual a partir de una muestra, $\hat{\mu}$, para representar μ sería como una caña.
- ¿Cómo sería una red? Tratemos de ver entre que dos valores de `edad` se encuentra el mayor número de estimaciones. ¿Entre 37 y 41? ¿36.5 y 41.5?
- Esta última idea es lo que llamaremos un intervalo de confianza. El **intervalo de confianza nos da un rango de valores posibles**.

Distribución bootstrap



¿Qué necesitamos para construir un I.C.?

- Una **distribución bootstrap**.
- Un **nivel de confianza** (90%, **95%**, 99%).
    - A mayor nivel de confianza, los intervalos serán más amplios.
    - Normalmente trabajaremos con un nivel de confianza de 95%.
- Construiremos intervalos de confianza a través de dos métodos:
    - **método de percentiles**.
    - **método del error estándar**.

Método de percentiles

Características de `promedio_edad` en `promedio_remuestras`.

```
promedio_remuestras %>%
  select(promedio_edad) %>%
  summary()
```

```
##  promedio_edad
##  Min.   :33.95
##  1st Qu.:37.28
##  Median :38.13
##  Mean   :38.15
##  3rd Qu.:39.08
##  Max.   :43.24
```

Si queremos saber **entre que dos valores** están, por ejemplo, **el 95% de las observaciones**, necesitamos saber **bajo qué valor están el 2.5% inferior de los datos** y **sobre qué valor están el 2.5% superior de los datos**.

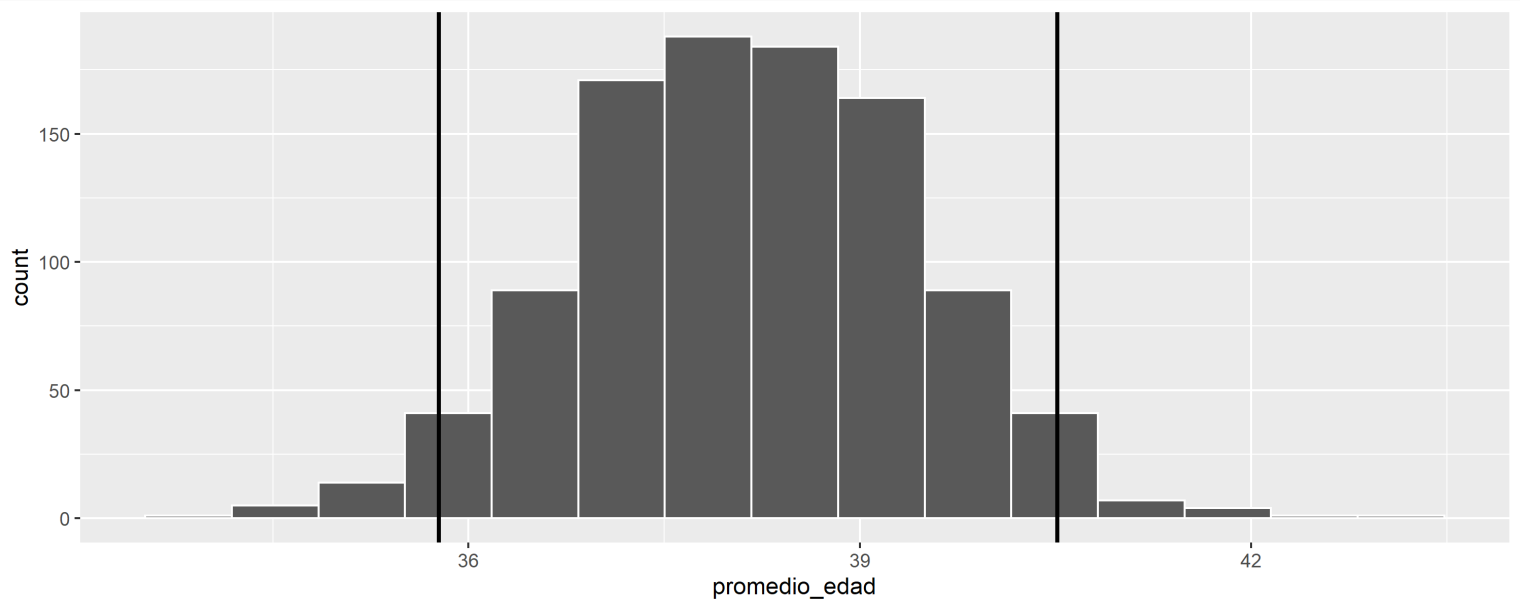
```
promedio_remuestras %>%
  arrange(-promedio_edad) %>% # Ordenamos los datos
  slice(25, 975) %>% # Tenemos 1.000 valores
  select(2)
```

```
## # A tibble: 2 x 1
##   promedio_edad
##           <dbl>
## 1          40.5
## 2          35.8
```

Método de percentiles

```
(metodo_percentiles ← promedio_remuestras %>%
  summarise(percentil_2.5 = quantile(promedio_edad, 0.025), # Calcular percentil 2.5
            percentil_97.5 = quantile(promedio_edad, 0.975))) # Calcular percentil 97.5
```

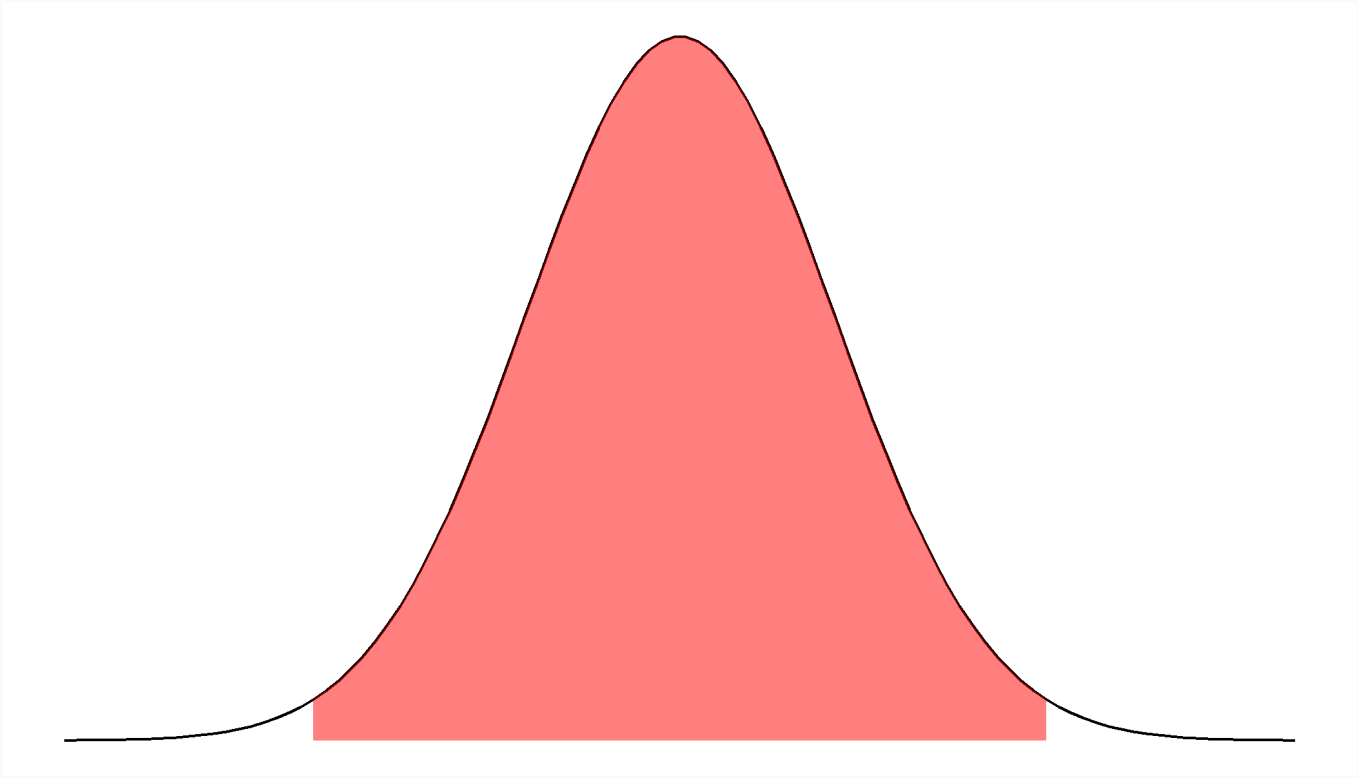
```
## # A tibble: 1 x 2
##   percentil_2.5 percentil_97.5
##           <dbl>          <dbl>
## 1          35.8           40.5
```



Método error estándar

```
pnorm(1.96, mean = 0, sd = 1) - pnorm(-1.96, mean = 0, sd = 1)
```

```
## [1] 0.9500042
```

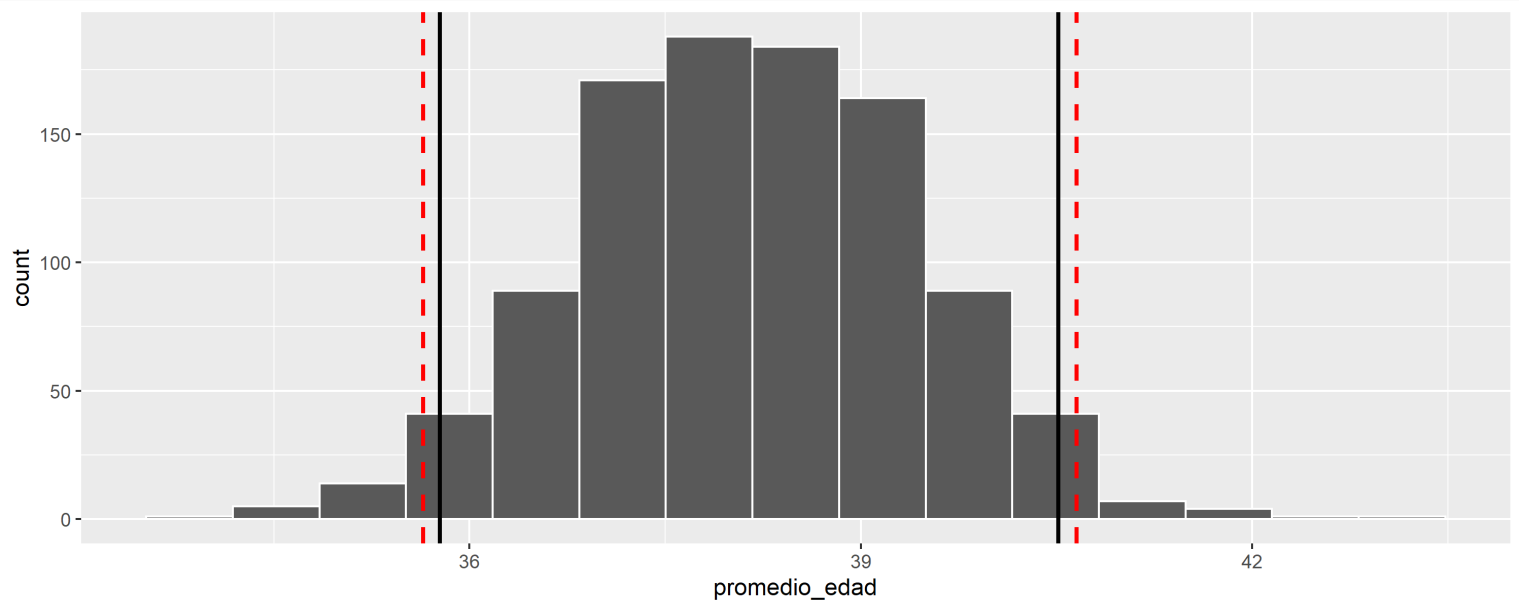


$$\hat{\mu} \pm 1.96 \times EE$$

Método error estándar

```
(metodo_ee ← promedio_remuestras %>%
  summarise(promedio = mean(promedio_edad), # Calcular promedio estándar
            EE = sd(promedio_edad)) %>%  # Calcular error
  mutate(lim_inf = promedio - (1.96*EE), # Calcular límite inferior I.C.
         lim_sup = promedio + (1.96*EE))) # Calcular límite superior I.C.
```

```
## # A tibble: 1 x 4
##   promedio    EE lim_inf lim_sup
##      <dbl> <dbl>   <dbl>   <dbl>
## 1     38.1  1.28    35.6    40.7
```



Paquete **infer**

- `infer` es un paquete para inferencia estadística.
- Relacionado al `tidyverse`

```
muestra_censo %>%
  summarise(promedio = mean(edad))
```

```
##   promedio
## 1 38.12667
```

```
muestra_censo %>%
  specify(response = edad) %>%
  calculate(stat = "mean")
```

```
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1  38.1
```

- El cálculo usando `infer` es más largo. ¿Para qué entonces?
- Nos presenta "verbos" más ligados a la estadística.
- Será útil cuando veamos prueba de hipótesis.
- Es más flexible para cuando queremos hacer inferencia para más de una variable.

Paquete **infer**

```
muestra_censo %>%
  specify(response = edad)
```

```
## Response: edad (integer)
## # A tibble: 300 x 1
##     edad
##    <int>
##  1    89
##  2    51
##  3    48
##  4     8
##  5    38
##  6    17
##  7    18
##  8    23
##  9    38
## 10    27
## #  ... with 290 more rows
```

- `specify` permite identificar la variable (o variables) sobre la cuál haremos los cálculos.
- Noten como en la práctica no cambia nada en el `data.frame`. En ese sentido es similar a `group_by`.

Paquete **infer**

```
muestra_censo %>%
  specify(response = edad) %>%
  generate(reps = 1000,
           type = "bootstrap")
```

```
## Response: edad (integer)
## # A tibble: 300,000 x 2
## # Groups:   replicate [1,000]
##    replicate  edad
##        <int> <int>
##  1         1    59
##  2         1    37
##  3         1    68
##  4         1    23
##  5         1    47
##  6         1    15
##  7         1    21
##  8         1    56
##  9         1    27
## 10         1    21
## #  ... with 299,990 more rows
```

- `generate` nos permite generar las 1.000 remuestras bootstrap.
- El resultado tiene 300.000 filas debido a que son 1.000 remuestras de tamaño igual a 300.
- Se genera una columna "replicate" correspondiente a cada una de las 1.000 remuestras.

Paquete **infer**

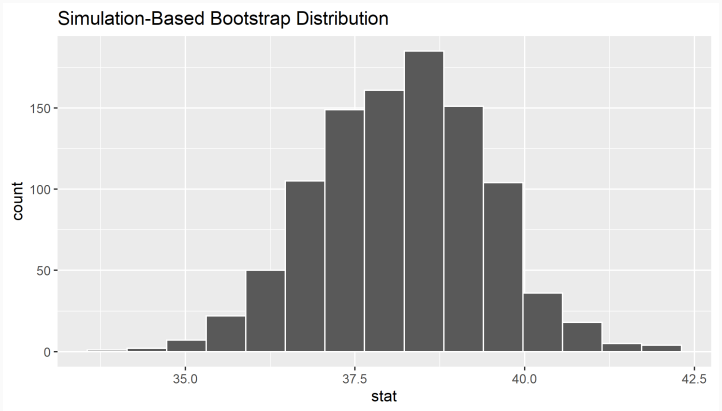
```
muestra_censo %>%
  specify(response = edad) %>%
  generate(reps = 1000,
           type = "bootstrap") %>%
  calculate(stat = "mean")
```

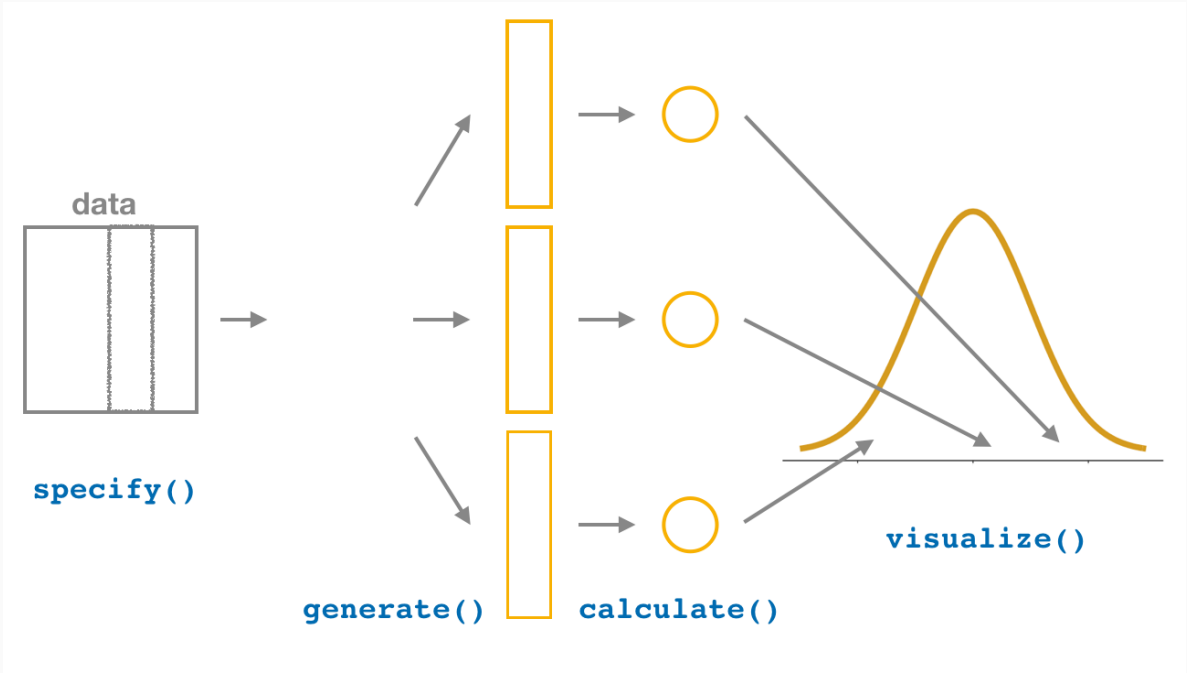
```
## # A tibble: 1,000 x 2
##    replicate  stat
##        <int> <dbl>
##  1         1  36.7
##  2         2  38.4
##  3         3  39.3
##  4         4  39.5
##  5         5  39.3
##  6         6  38.7
##  7         7  38.9
##  8         8  38.8
##  9         9  37.3
## 10        10  39.2
## # ... with 990 more rows
```

- Con `calculate` transformamos cada una de las 1.000 remuestras de 300 observaciones, en 1.000 medias.
- Noten que el resultado son 1.000 filas con una columna correspondiente a cada "replica" y la otra con el cálculo hecho.

Paquete **infer**

```
muestra_censo %>%
  specify(response = edad) %>%
  generate(reps = 1000,
           type = "bootstrap") %>%
  calculate(stat = "mean") %>%
  visualise()
```





Construir I.C. con **infer**

```
set.seed(1)
guardar_remuestras_i ← muestra_censo %>%
  specify(response = edad) %>%
  generate(reps = 1000,
           type = "bootstrap") %>%
  calculate(stat = "mean")
```

Método percentiles

```
guardar_remuestras_i %>%
  get_confidence_interval(level = 0.95,
                          type = "percentile")
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     35.8     40.5
```

Método error estándar

```
guardar_remuestras_i %>%
  get_confidence_interval(level = 0.95,
                          type = "se",
                          point_estimate = edad_promedio_mu
```

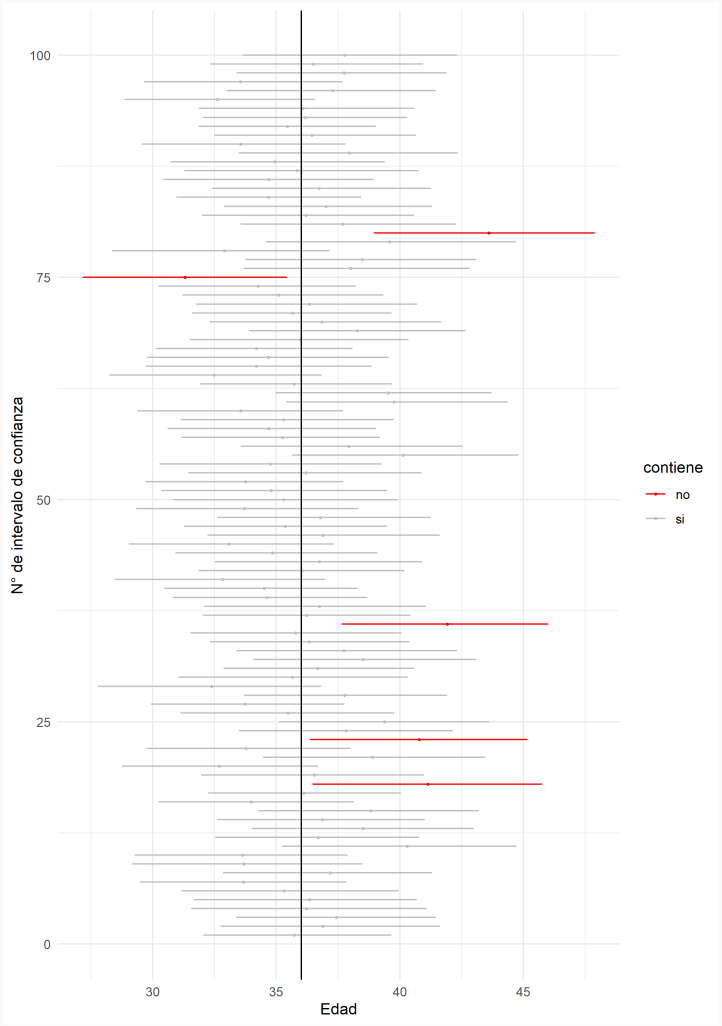
```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     35.6     40.6
```

Interpretar I.C.

- Ya pudimos construir intervalos de confianza a partir de una muestra tomada desde una población. Ahora podemos **evaluar su efectividad**.
- La efectividad de un intervalo de confianza se juzga según si este **contiene o no el verdadero valor del parámetro poblacional**. *¿Capturó la red al pescado?*
- En nuestro ejemplo, ¿nuestros intervalos de confianza, $[35.8, 40.5]$ o $[35.6, 40.6]$, capturan el verdadero promedio de `edad`, $\mu = 36.0179$?
- **¡Sí!** nuestros intervalos construídos con un 95% de nivel de confianza a partir de una muestra con $n = 300$ incluyen al valor real del parámetro poblacional. **¿Ocurrirá esto para todas las muestras que tomemos?**

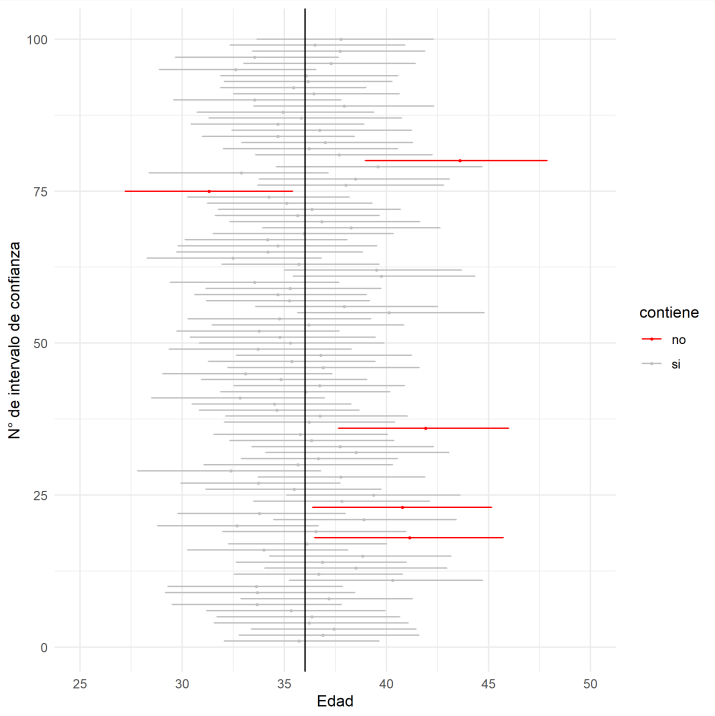
Interpretar I.C.

- 100 intervalos de confianza a partir de 100 muestras aleatorias distintas y considerando un nivel de confianza de 95%.
- La linea negra corresponde al valor real del parámetro poblacional, μ (edad de la población).
- Las lineas horizontales corresponden a los intervalos de confianza y son de color gris si el intervalo incluye al valor real y rojas si no.
- De los 100 intervalos, 95 incluyen el valor real del parámetro. En otras palabras, un nivel de confianza de 95% significa que de cada 100 intervalos **esperamos** que 95 incluyan μ.

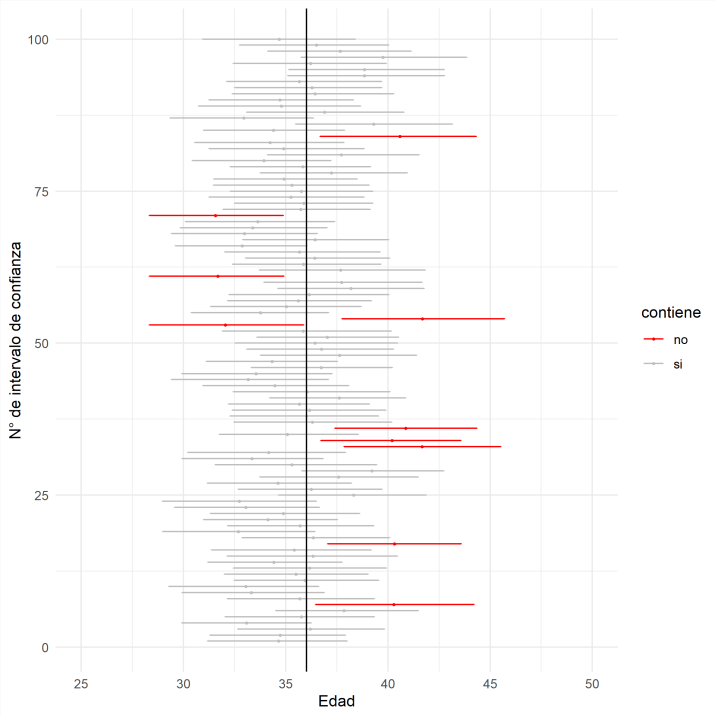


Distintos niveles de confianza

95% nivel de confianza



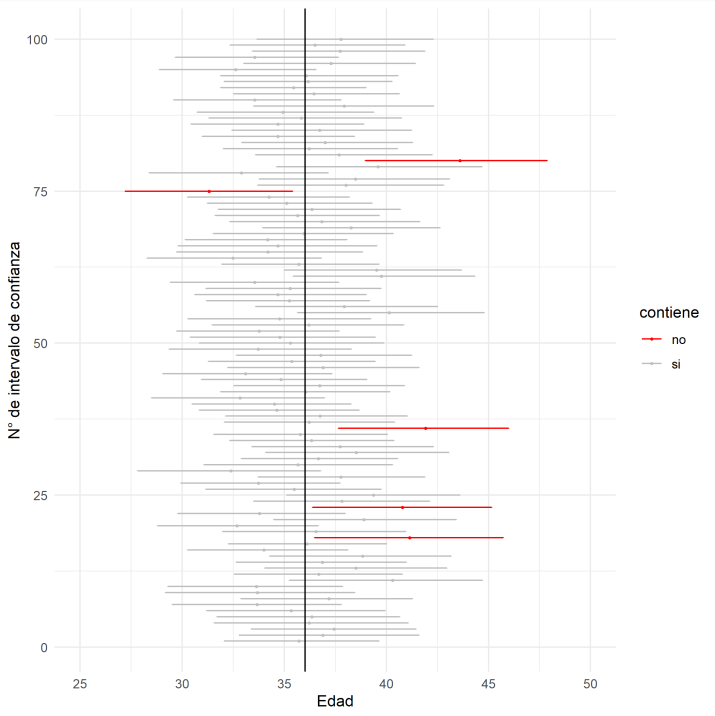
90% nivel de confianza



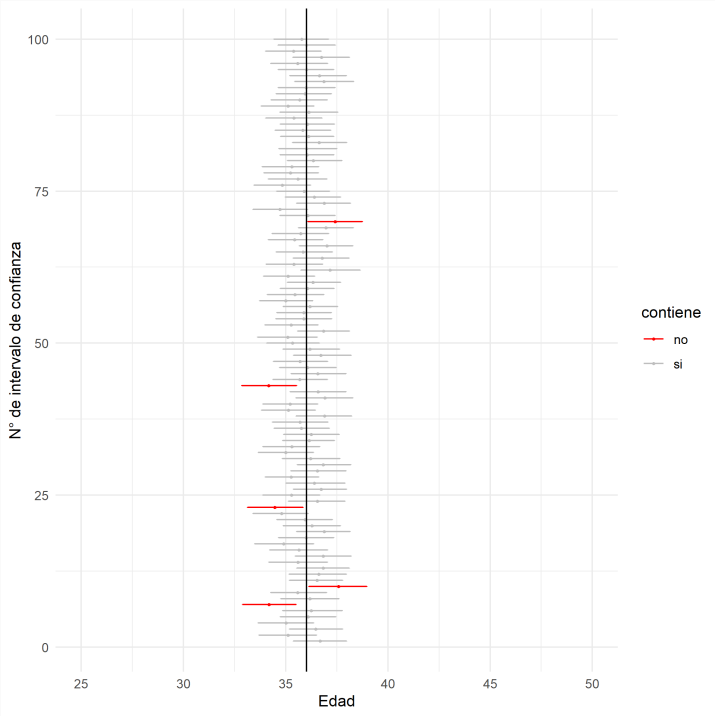
Mayores niveles de confianza llevan a intervalos más amplios

Distintos n

$n = 100$



$n = 1000$



Mayores tamaños muestrales llevan a intervalos más angostos

Resumiendo...

- Como dijimos, "un nivel de confianza de 95% significa que de cada 100 intervalos que pudieramos construir, **esperaríamos** que 95 incluyan μ."
- No es lo mismo que decir "hay un 95% de probabilidad de que el intervalo de confianza contenga a μ".

I.C "basado en teoría"

- Hasta este momento nuestros I.C. se construyeron usando el método de percentiles o el de error estándar **haciendo simulaciones**.
- Otro método es simplemente ocupar una fórmula "basada en teoría" del tipo:
  $I.C. = Estimación\ Puntual \pm (Valor\ Crítico \times Error\ Estándar)$
- **Esta fórmula es una aproximación**, usando solo la información de una muestra, a lo que obtenemos a través de simulaciones.
- Por ende, ya que no realizaremos remuestras en este caso, **necesitamos una fórmula para estimar el error estándar**.

Error estándar basado en teoría

- Para el caso de una media: $\frac{s}{\sqrt{n}}$. Donde s es la desviación estándar de la muestra.
- Para una proporción: $\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$. Donde $\hat{p}$ es la proporción calculada desde la muestra.

Error estándar de formas distintas

Usando la dispersión de las remuestras (bootstrap)

```
promedio_remuestras %>%
  summarise(sd = sd(promedio_edad, na.rm = TRUE)) %>% transmute(error_est = sd)
```

Usando fórmula con la desviación estándar de la población, $\frac{\sigma}{\sqrt{n}}$

```
censo %>%
  summarise(sd = sd(edad, na.rm = TRUE)) %>% transmute(error_est = sd/sqrt(300))
```

Usando fórmula con la desviación estándar de la muestra, $\frac{s}{\sqrt{n}}$

```
muestra_censo %>%
  summarise(sd = sd(edad, na.rm = TRUE)) %>% transmute(error_est = sd/sqrt(300))
```

Comparar

```
## # A tibble: 1 x 3
##   bootstrap_ee poblacion_ee formula_ee
##          <dbl>        <dbl>      <dbl>
## 1         1.28         1.28       1.28
```

I.C "basado en teoría"

$$I.C. = Estimación\ Puntual \pm (Valor\ Crítico \times Error\ Estándar)$$
$$I.C. = \hat{\mu} \pm (1.96 \times \frac{s}{\sqrt{n}})$$

```
muestra_censo %>%
  summarise(mu = mean(edad, na.rm = TRUE),
            sd = sd(edad, na.rm = TRUE),
            formula_ee = sd/sqrt(300),
            IC_1 = mu - (1.96*formula_ee),
            IC_2 = mu + (1.96*formula_ee))
```

```
##         mu      sd formula_ee     IC_1     IC_2
## 1 38.12667 22.1389    1.27819 35.62141 40.63192
```

$$I.C. = 38.13 \pm (1.96 \times 1.278)$$
$$I.C. = 38.13 \pm (2.5)$$
$$I.C. = [35.6, 40.6]$$

Al igual que en los otros I.C. que construimos, el promedio poblacional de `edad`, $\mu = 36.0179$, también está incluído en este intervalo.

Ejercicio

Ejercicio

- `EjercicioIntervaloConfianza.R`

Prueba de hipótesis

Un caso real

- Un estudio de 1974 analizó el **efecto que el sexo (masculino/femenino) en las posibilidades de ser ascendido/a** en un trabajo.
- A 48 supervisores de una industria se les pidió que asumieran el rol de un director de RRHH y se les entregó **un CV** para que **decidieran si es que el o la candidato/a debiera ser ascendido/a**.
- Los **48 CV eran exactamente iguales con excepción del nombre**. A 24 de las personas se les dieron CVs con solo nombres "típicos de hombres" y al otro grupo de 24 solo con nombres "típicos de mujeres".
- Considerando la asignación aleatoria y la posibilidad de que el CV sea "hombre" o "mujer" solamente, este experimento serviría como una aproximación para aislar el efecto del sexo de una persona en ser o no ascendido/a.

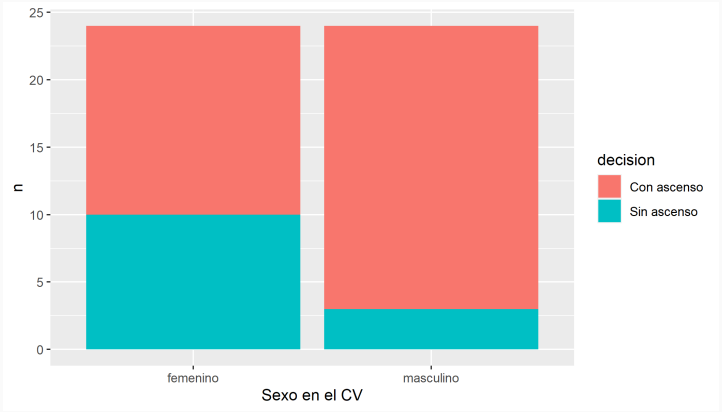
```
ascensos ← read_csv("../datos/ascensos.csv")
glimpse(ascensos)
```

```
## Rows: 48
## Columns: 3
## $ id       <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, ...
## $ decision <chr> "Con ascenso", "Con ascenso", "Con ascenso", "Con ascenso" ...
## $ sexo     <chr> "masculino", "masculino", "masculino", "masculino", "mascu ...
```

Resultados

- 35 personas fueron seleccionadas para ser ascendidas.
- De los 24 CVs con nombres de **hombre**, **21 fueron ascendidos** (87.5%).
- De los 24 CVs con nombres de **mujer**, **14 fueron ascendidas** (58.3%).
- **29.2 puntos % de diferencia** entre hombres y mujeres.

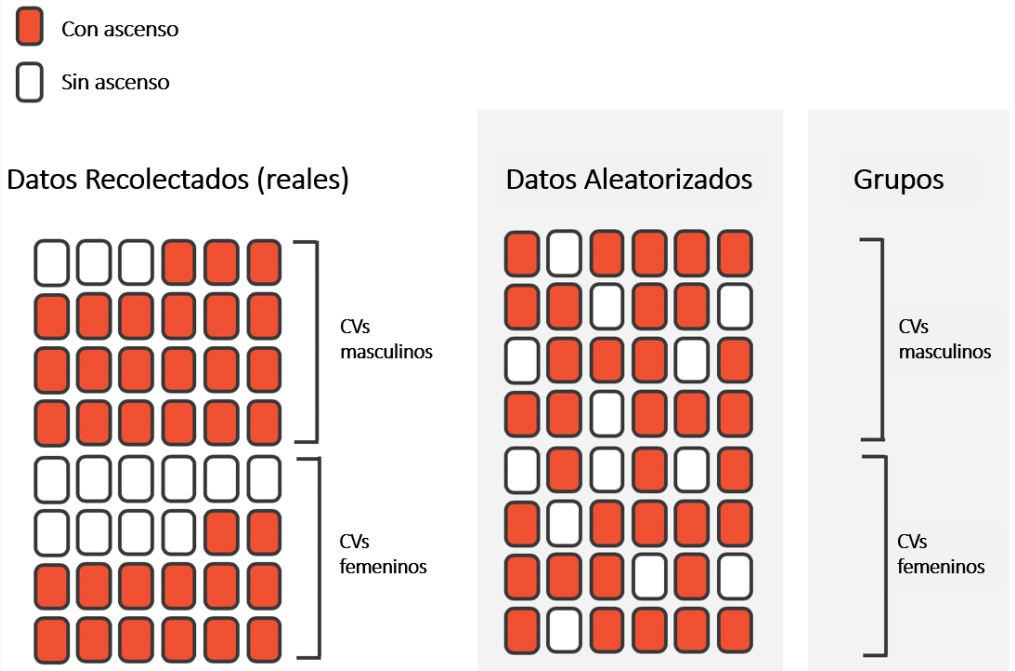
```
ascensos %>%
  group_by(sexo, decision) %>%
  summarise(n = n()) %>%
  ggplot(aes(x = sexo, y = n, fill = decision)) +
  geom_col() +
  labs(x = "Sexo en el CV")
```



- *¿Es esta evidencia concluyente de que en esta industria existe discriminación contra las mujeres a la hora de realizar ascensos laborales?*
- *¿Podría ser esta diferencia solo "por casualidad" en un mundo hipotético donde no existe discriminación* ***o es una diferencia significativa****?*

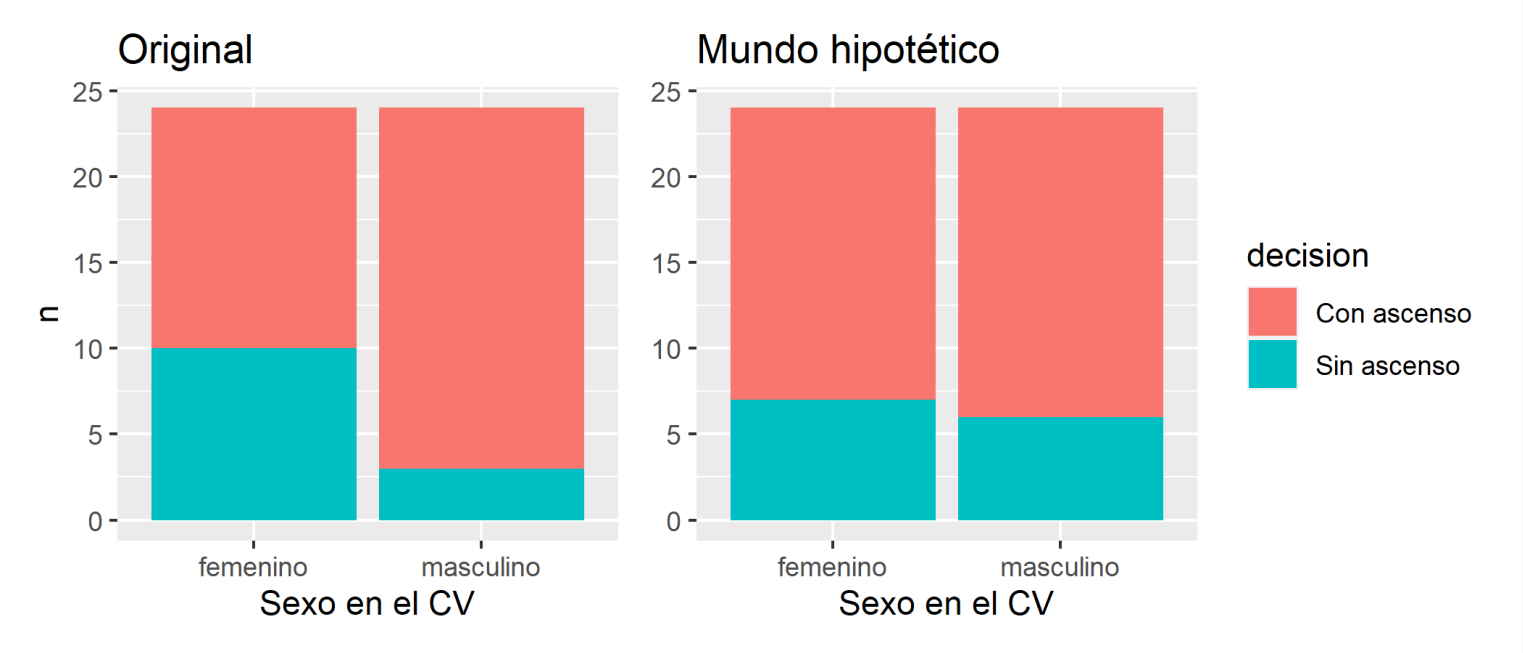
Mundo hipotético

- Imaginemos un mundo donde la discriminación laboral hacia las mujeres no existe. Entonces, **en este mundo el ser mujer u hombre no tiene ninguna influencia** en si alguien es o no ascendido/a.
- En términos de nuestros datos, `ascensos`, la variable `sexo` sería irrelevante. Puesto de otra forma, podríamos reordenar nuestros datos para que los **35 ascensos se repartan aleatoriamente entre los 24 `masculino` y 24 `femenino`**.



Mundo hipotético

- Imaginemos un mundo donde la discriminación laboral hacia las mujeres no existe. Entonces, **en este mundo el ser mujer u hombre no tiene ninguna influencia** en si alguien es o no ascendido/a.
- En términos de nuestros datos, `ascensos`, la variable `sexo` sería irrelevante. Puesto de otra forma, podríamos reordenar nuestros datos para que los **35 ascensos se repartan aleatoriamente entre los 24** `masculino` **y 24** `femenino`.



Mundo hipotético

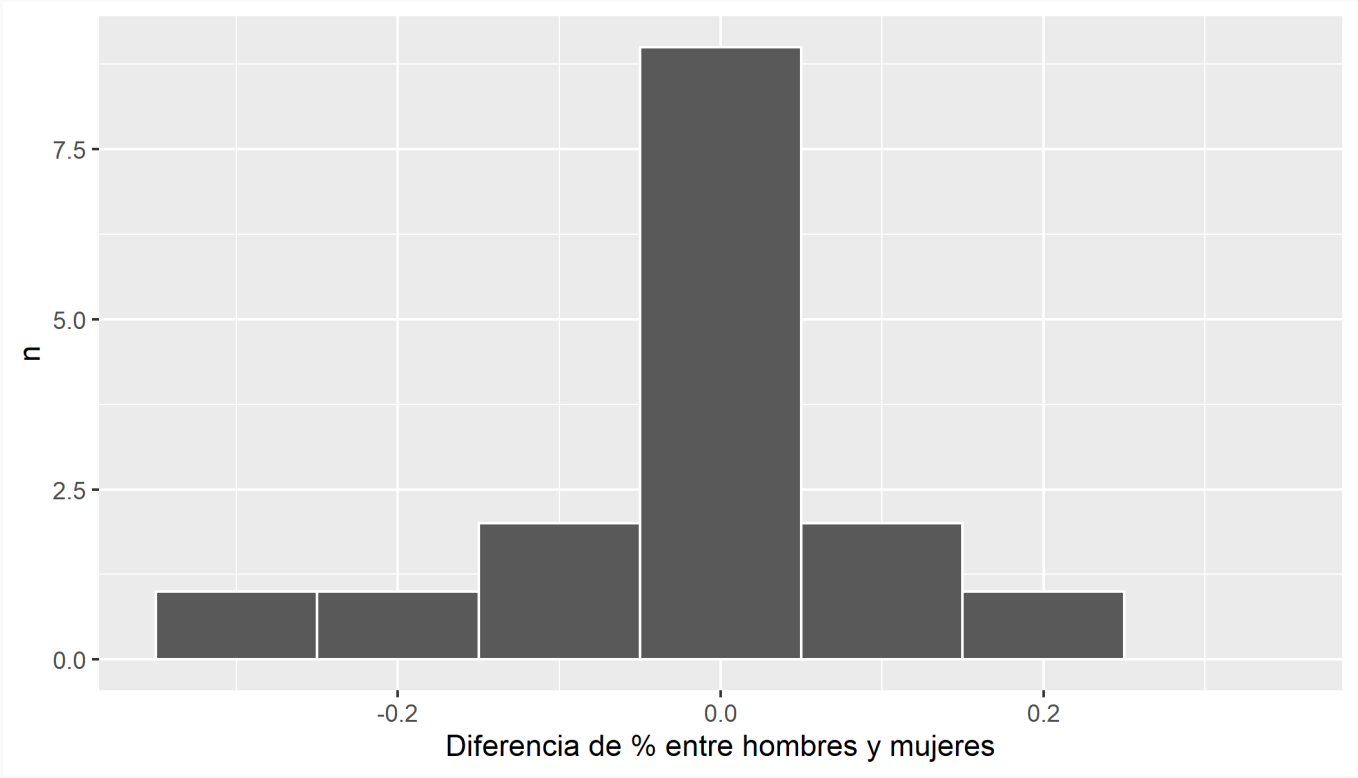
```
ascensos_reordenado %>%
  group_by(sexo, decision) %>%
  summarise(n = n()) %>%
  pivot_wider(names_from = sexo, values_from = n)
```

```
## # A tibble: 2 x 3
##   decision    femenino masculino
##   <chr>          <int>     <int>
## 1 Con ascenso       17        18
## 2 Sin ascenso        7         6
```

- En este mundo hipotético de no discriminación, **18 de 24 hombres** fueron ascendidos (75%) y **17 de 24 mujeres** también (70.8%).
- La diferencia que en nuestra muestra original era de 29.2 puntos % **en este caso es de 4.2 puntos %**.
- Ahora, esta asignación aleatoria es solo un ejemplo. Podríamos repetir este proceso algunas veces más y tener una **distribución de diferencias**.

16 aleatorizaciones

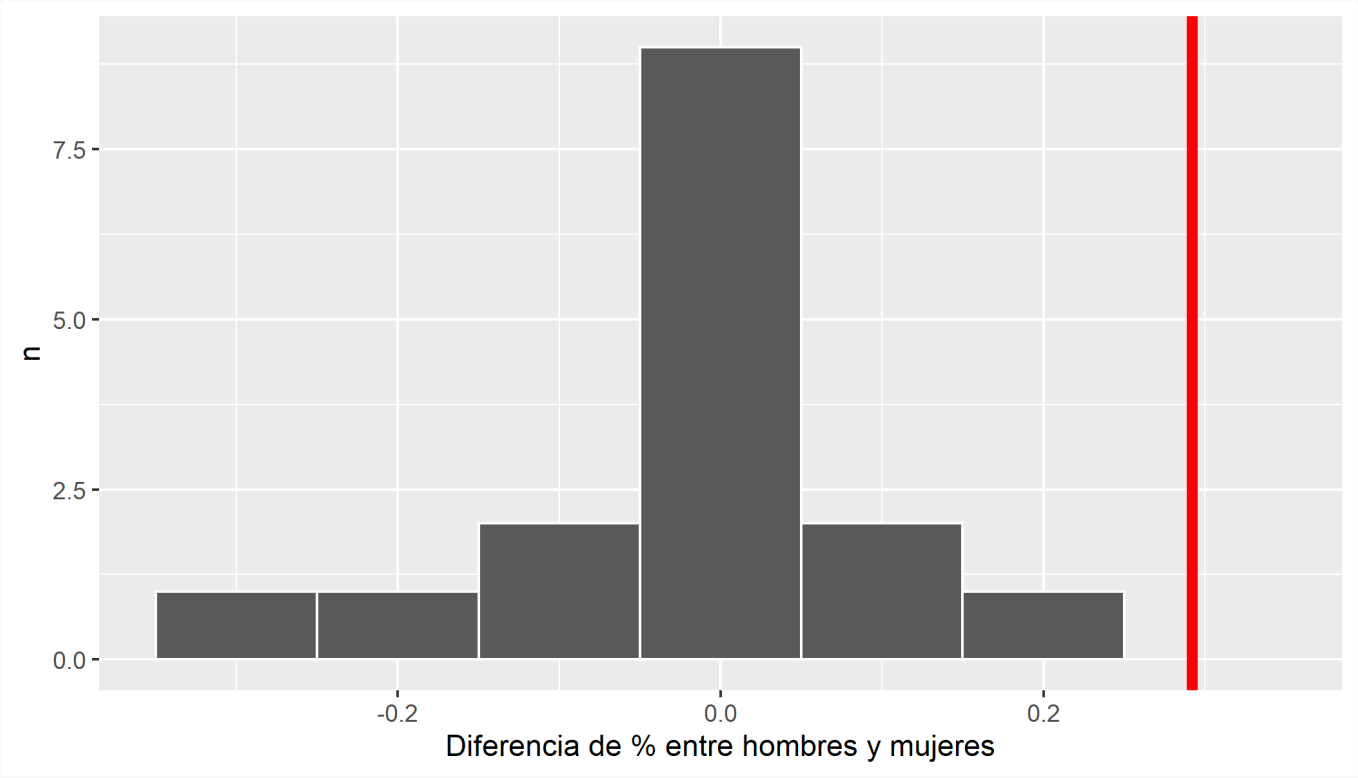
Distribución de 16 veces reordenar nuestros datos para que los **35 ascensos se repartan aleatoriamente entre los 24 `masculino` y 24 `femenino`**.



¿Cómo se vería nuestro cálculo inicial (real) en esta distribución?

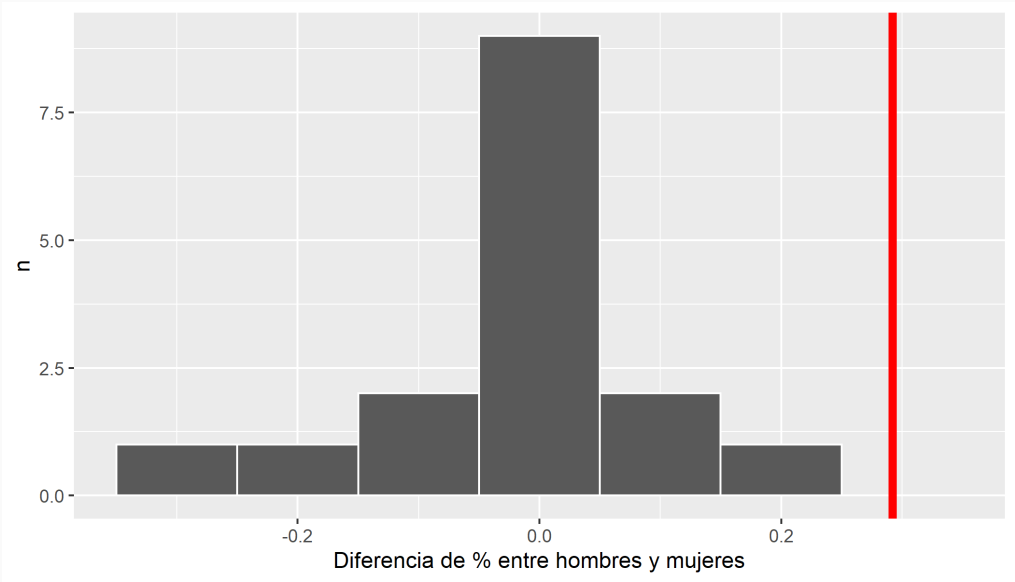
16 aleatorizaciones

Distribución de 16 veces reordenar nuestros datos para que los **35 ascensos se repartan aleatoriamente entre los 24** `masculino` **y 24** `femenino`.



¿Qué significa esto?

- Recordemos que este histograma representa la distribución de la diferencia de ascensos entre hombres y mujeres en un **mundo hipotético de no discriminación**.
- De hecho, **la distribución esta centrada en cero**. Sin embargo, en este mundo hipotético igual pueden haber diferencias en ascensos debido a la **variación muestral**.
- Teniendo todo eso en consideración, **¿qué tan factible es una diferencia de 29.2 puntos porcentuales de más ascensos para hombres que mujeres en un mundo de no discriminación?**



¿Qué acabamos de hacer?

- El procedimiento que hicimos se conoce como ***prueba de hipótesis* usando *permutaciones***. En este caso el acto de permutar fue aleatorizar los ascensos entre `masculino` y `femenino`.
- Las permutaciones son otra forma de **remuestreo** como el *bootstrap* que ya aplicamos. Mientras que el *bootstrap* fue un remuestreo con reemplazo, la ***permutación* es remuestreo sin reemplazo**.
- En este caso usamos las permutaciones para simular un mundo de no discriminación y el resultado fue que **la realidad es poco probable bajo ese mundo supuesto**.
- En otras palabras, con la evidencia disponible tenderíamos a **rechazar que este mundo hipotético es factible** y que, por ende, **hay evidencia de que existe discriminación**.
- Nuestra estimación de **29.2 puntos porcentuales de diferencia parece ser claramente mayor a 0**, sugiriendo discriminación en contra de las mujeres. Pero, **¿es esta diferencia significativamente diferente a 0?**
- Las pruebas de hipótesis nos permitirán responder esto.

Prueba de hipótesis: Algunos conceptos

Una hipótesis **es una declaración respecto al valor de un parámetro poblacional desconocido** (por ej. μ). Para el caso que vimos recién, este parámetro sería la diferencia entre la proporción de "CV masculinos" con ascenso menos la proporción de "CVs femeninos" con ascenso, $p_m - p_f$.

Una prueba de hipótesis consiste en una prueba entre **dos hipótesis contrarias**: *(i)* una **hipótesis nula**, H_0, versus *(ii)* una **hipótesis alternativa**, H_A.

- Generalmente **la hipótesis nula es una declaración de que no hay efecto** o no hay diferencia. Se habla de que la **hipótesis nula representa el status quo** o una situación de "no interés".
- Por otro lado, **la hipótesis nula sería la declaración que se quiere establecer** y que se probará a través de la evidencia.

$$H_0: \textit{ los hombres y las mujeres son ascendidos a tasas similares}$$

$$H_A: \textit{ los hombres son ascendidos a tasas más altas que las mujeres}$$

Prueba de hipótesis: Algunos conceptos

Una hipótesis **es una declaración respecto al valor de un parámetro poblacional desconocido** (por ej. μ). Para el caso que vimos recién, este parámetro sería la diferencia entre la proporción de "CV masculinos" con ascenso menos la proporción de "CVs femeninos" con ascenso, $p_m - p_f$.

Una prueba de hipótesis consiste en una prueba entre **dos hipótesis contrarias**: *(i)* una **hipótesis nula**, H_0, versus *(ii)* una **hipótesis alternativa**, H_A.

- Generalmente **la hipótesis nula es una declaración de que no hay efecto** o no hay diferencia. Se habla de que la **hipótesis nula representa el status quo** o una situación de "no interés".
- Por otro lado, **la hipótesis nula sería la declaración que se quiere establecer** y que se probará a través de la evidencia.

$$H_0 : p_m - p_f = 0$$

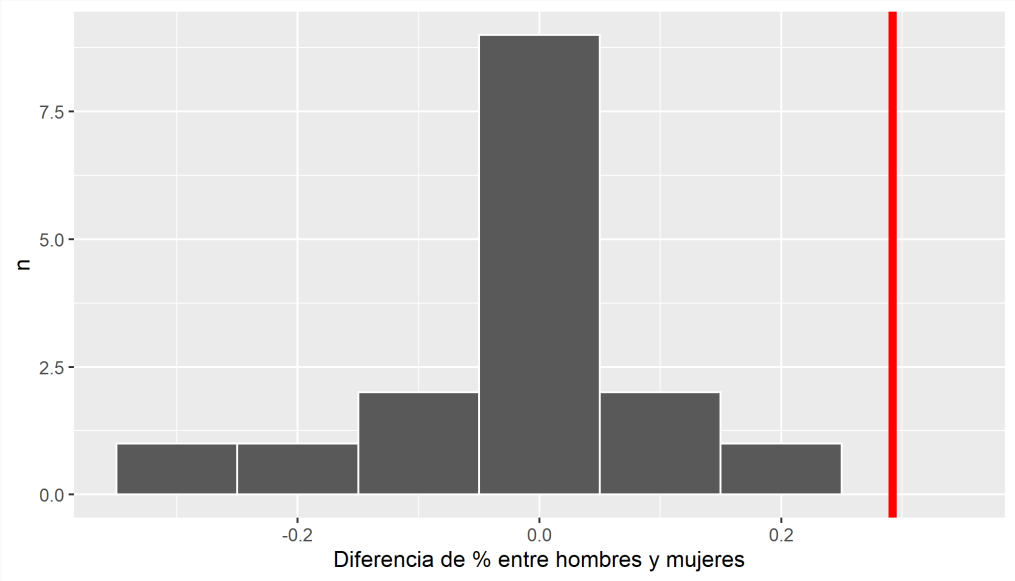
$$H_A : p_m - p_f > 0$$

Prueba de hipótesis: Algunos conceptos

Un estadístico es una formula para una estimación/cálculo. Antes hablabamos de la estimación de la media muestral como $\hat{\mu}$ (promedio muestral). Para este ejemplo **nuestro estadístico sería la diferencia de proporciones**, $\hat{p}_m - \hat{p}_f$.

El estadístico observado es el valor calculado/observado. En nuestro ejemplo sería **29.2 puntos porcentuales**.

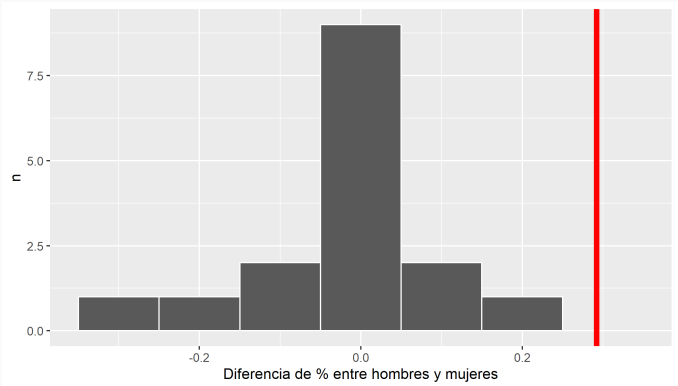
La distribución nula es la distribución del estadístico **asumiendo** que la hipótesis nula, H_0, es cierta. No decimos que H_0 sea necesariamente cierto sino que asumimos esto para ver **que tan factible es nuestro estadístico observado bajo este supuesto**.



Prueba de hipótesis: Algunos conceptos

El **valor p o p-value** es la probabilidad de observar un estadístico tan o más "extremo" que el que tenemos, **asumiendo que la hipótesis nula, H_0, es cierta**.

- Podríamos decir que es una cuantificación de "sorpresa" de los resultados. En nuestro ejemplo, **¿qué tan "sorprendidos" estamos de que la diferencia en ascensos fuera 29.2 puntos % dada la distribución nula?**
- Puesto de otra forma, ¿que proporción de los datos son más "extremos" que este resultado?



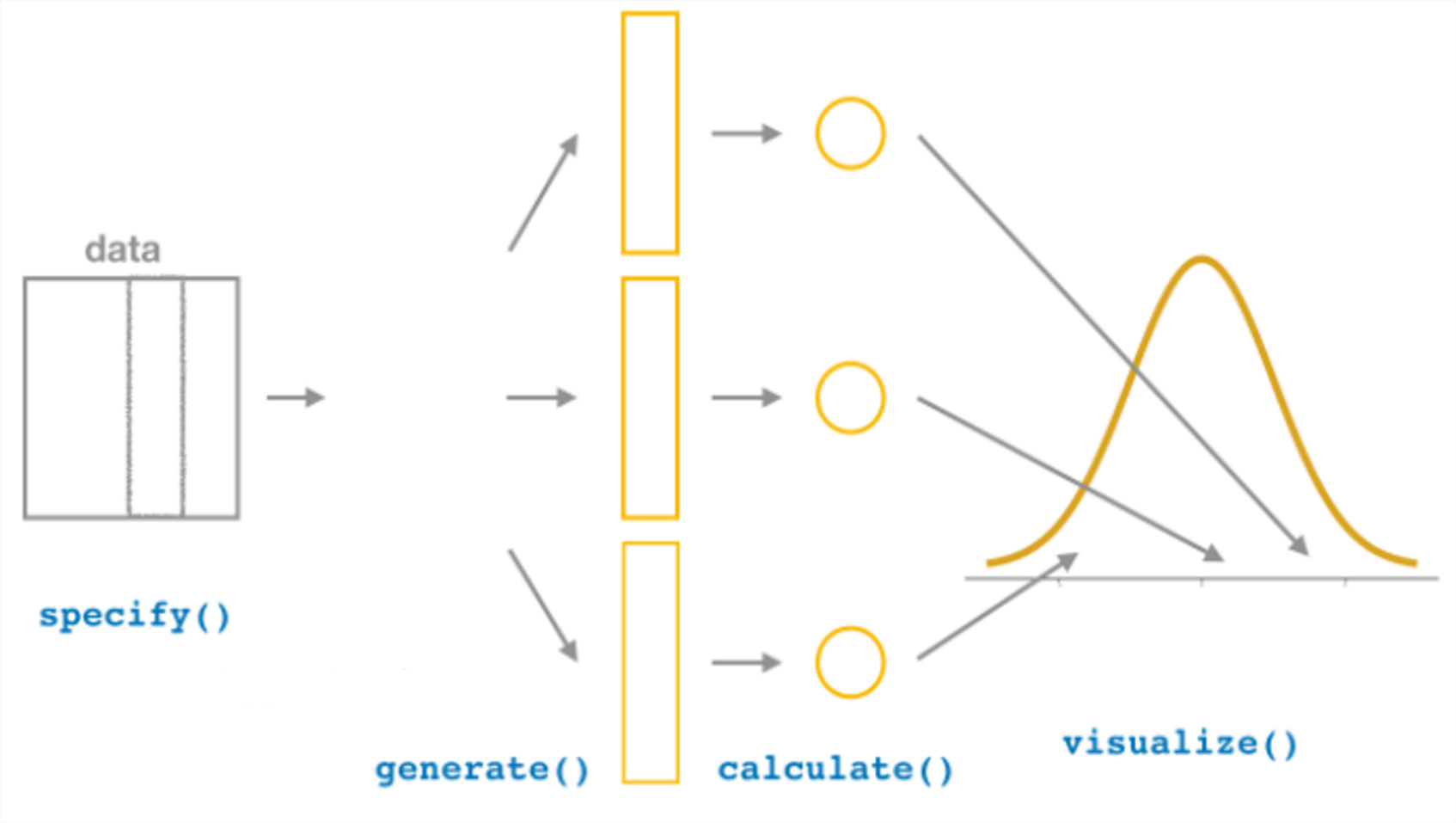
En este caso, **la respuesta es 0**. Por ende lo que observamos es **muy sorprendente** bajo la hipótesis nula, H_0. Tan raro es lo que vemos que lo más sensato sería **rechazar la hipótesis nula** de que no existe diferencia.

Prueba de hipótesis: Algunos conceptos

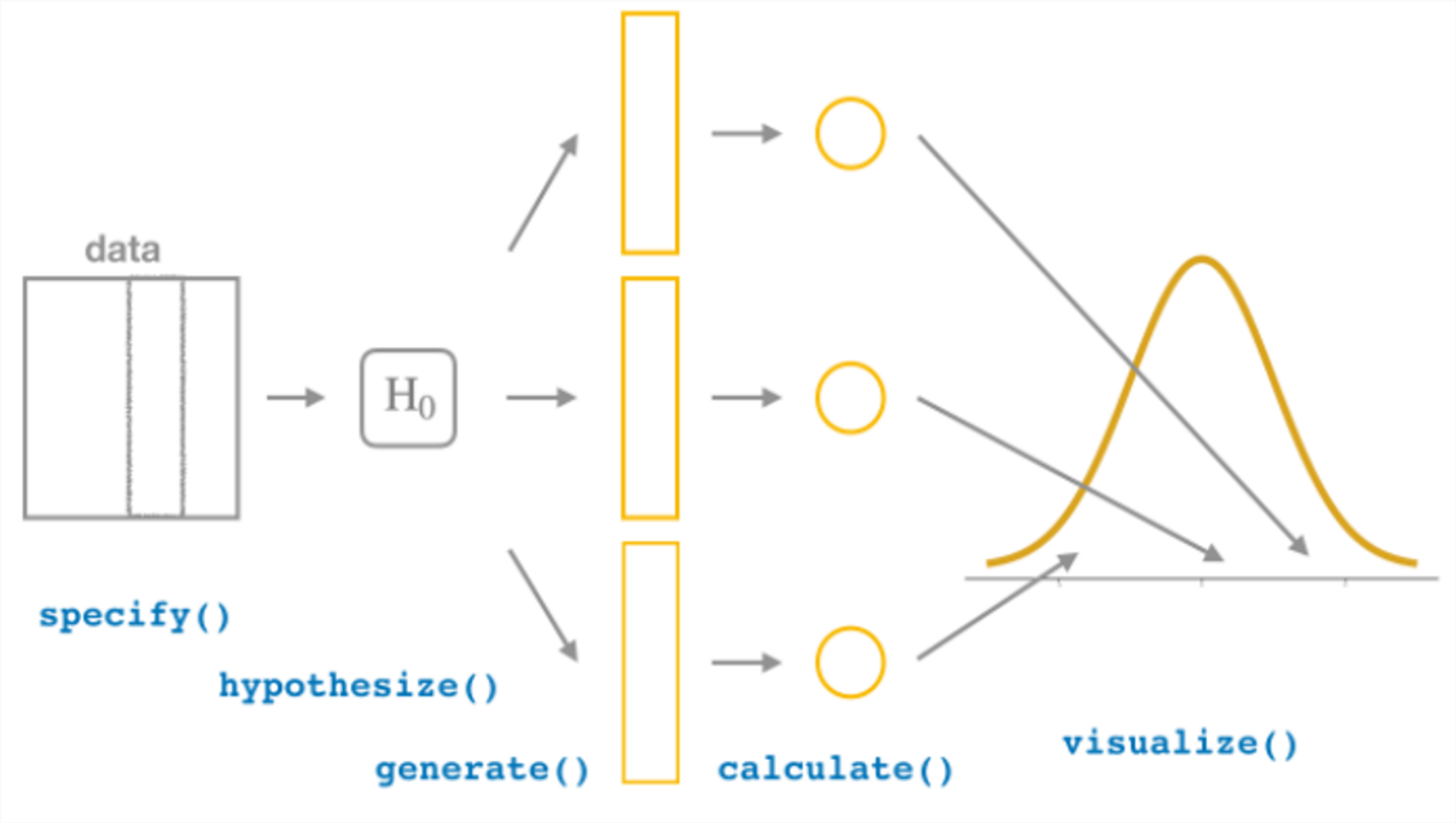
El nivel de significancia es un punto de corte que establecemos para el p-value. Normalmente se le denomina α.

- Rechazaremos la hipótesis nula, H_0, si es que $p - value < \alpha$
- En caso contrario, $p - value > \alpha$, decimos que "no rechazamos H_0" o "fallamos en rechazarla".
- Valores comúnmente usados para α son 0.1 (10%), **0.05 (5%)**, y 0.01 (1%).

Prueba de hipótesis con **infer**



Prueba de hipótesis con **infer**



Prueba de hipótesis con **infer**

```
ascensos %>%
  specify(formula = decision ~ sexo,
          success = "Con ascenso")
```

```
## Response: decision (factor)
## Explanatory: sexo (factor)
## # A tibble: 48 x 2
##    decision    sexo
##    <fct>       <fct>
##  1 Con ascenso masculino
##  2 Con ascenso masculino
##  3 Con ascenso masculino
##  4 Con ascenso masculino
##  5 Con ascenso masculino
##  6 Con ascenso masculino
##  7 Con ascenso masculino
##  8 Con ascenso masculino
##  9 Con ascenso masculino
## 10 Con ascenso masculino
## # ... with 38 more rows
```

- A diferencia del ejemplo anterior con `infer`, ahora queremos ver el efecto de una **variable explicatoria**, `sexo`, en una **variable respuesta**, `decision`.
- También le indicamos a la función que en este caso nos queremos enfocar en los casos "Con ascenso".

Prueba de hipótesis con **infer**

```
ascensos %>%
  specify(formula = decision ~ sexo,
          success = "Con ascenso") %>%
  hypothesise(null = "independence")
```

```
## Response: decision (factor)
## Explanatory: sexo (factor)
## Null Hypothesis: independence
## # A tibble: 48 x 2
##    decision    sexo
##    <fct>       <fct>
##  1 Con ascenso masculino
##  2 Con ascenso masculino
##  3 Con ascenso masculino
##  4 Con ascenso masculino
##  5 Con ascenso masculino
##  6 Con ascenso masculino
##  7 Con ascenso masculino
##  8 Con ascenso masculino
##  9 Con ascenso masculino
## 10 Con ascenso masculino
## # ... with 38 more rows
```

- Este es un nuevo paso que no vimos para los intervalos de confianza.
- `null` tiene dos opciones: `point` o `independence`. El primero es para cuando trabajamos con "una muestra" y el segundo cuando tenemos dos.
- En este caso nuestras "dos muestras" son los CVs con nombres masculinos y los CVs con nombres femeninos.

Prueba de hipótesis con **infer**

```
ascensos %>%
  specify(formula = decision ~ sexo,
          success = "Con ascenso") %>%
  hypothesise(null = "independence") %>%
  generate(reps = 1000,
           type = "permute")
```

```
## Response: decision (factor)
## Explanatory: sexo (factor)
## Null Hypothesis: independence
## # A tibble: 48,000 x 3
## # Groups:   replicate [1,000]
##    decision    sexo      replicate
##    <fct>       <fct>         <int>
##  1 Con ascenso masculino         1
##  2 Con ascenso masculino         1
##  3 Sin ascenso masculino         1
##  4 Sin ascenso masculino         1
##  5 Con ascenso masculino         1
##  6 Con ascenso masculino         1
##  7 Con ascenso masculino         1
##  8 Con ascenso masculino         1
##  9 Con ascenso masculino         1
## 10 Sin ascenso masculino         1
## # ... with 47,990 more rows
```

- Acá generamos la aleatorización asumiendo que la hipótesis nula es cierta. Antes hicimos 16 veces esto, ahora 1.000.
- A diferencia de los intervalos de confianza, donde usamos `"bootstrap"`, ahora usamos `type = "permute"`.

Prueba de hipótesis con **infer**

```
ascensos %>%
  specify(formula = decision ~ sexo,
          success = "Con ascenso") %>%
  hypothesise(null = "independence") %>%
  generate(reps = 1000,
           type = "permute") %>%
  calculate(stat = "diff in props",
            order = c("masculino", "femenino"))
```

```
## # A tibble: 1,000 x 2
##    replicate    stat
##        <int>   <dbl>
##  1         1  0.125
##  2         2  0.125
##  3         3 -0.0417
##  4         4 -0.208
##  5         5  0.125
##  6         6 -0.208
##  7         7 -0.0417
##  8         8  0.292
##  9         9  0.125
## 10        10  0.0417
## # ... with 990 more rows
```

- Teniendo las 1.000 replicas, ahora podemos **calcular el estadístico**.
- En este caso sería la diferencia de proporción con ascensos para CVs "masculinos" y "femeninos", $\hat{p}_m - \hat{p}_f$

Prueba de hipótesis con **infer**

```
(distribucion_nula ← ascensos %>%
  specify(formula = decision ~ sexo,
          success = "Con ascenso") %>%
  hypothesise(null = "independence") %>%
  generate(reps = 1000,
           type = "permute") %>%
  calculate(stat = "diff in props",
            order = c("masculino", "femenino")))
```

```
## # A tibble: 1,000 x 2
##    replicate    stat
##        <int>   <dbl>
##  1         1  0.125
##  2         2  0.125
##  3         3 -0.0417
##  4         4 -0.208
##  5         5  0.125
##  6         6 -0.208
##  7         7 -0.0417
##  8         8  0.292
##  9         9  0.125
## 10        10  0.0417
## # ... with 990 more rows
```

- Crearemos un objeto `distribucion_nula` para guardar los 1.000 resultados de $\hat{p}_m - \hat{p}_f$.

¿Cuál era el estadístico observado?

Podemos hacer este cálculo usando `infer`

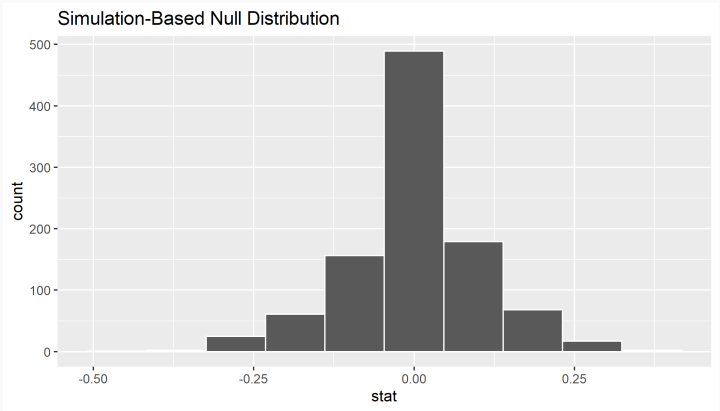
```
(dif_observada ← ascensos %>%
  specify(formula = decision ~ sexo,
          success = "Con ascenso") %>%
  calculate(stat = "diff in props",
            order = c("masculino", "femenino")))
```

```
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1 0.292
```

29.2 puntos porcentuales más de ascensos para hombres que para mujeres.

Prueba de hipótesis con **infer**

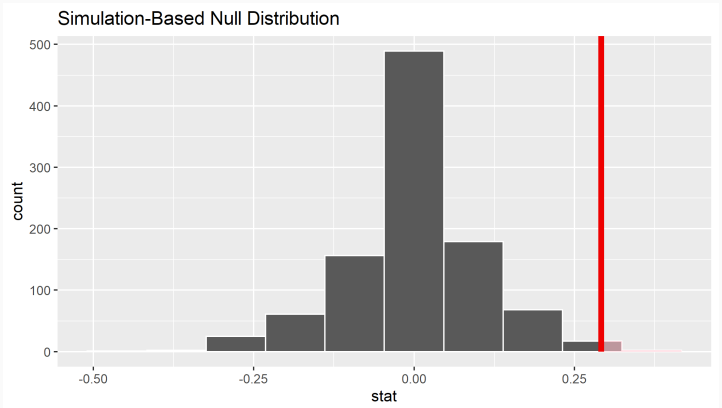
```
distribucion_nula %>%
  visualise(bins = 10)
```



- Visualicemos la distribución nula creada a partir de los 1.000 estadísticos calculados, $\hat{p}_m - \hat{p}_f$.

Prueba de hipótesis con **infer**

```
distribucion_nula %>%
  visualise(bins = 10) +
  shade_p_value(obs_stat = dif_observada,
                direction = "right")
```



- Visualicemos la distribución nula creada a partir de los 1.000 estadísticos calculados, $\hat{p}_m - \hat{p}_f$.
- Y agreguemos el estadístico observado que guardamos como `dif_observada` (línea roja).
- El argumento `direction = "right"` es debido a que $H_A : p_m - p_f > 0$
- El área sombreada a la derecha de la línea roja corresponde al **p-value**.
- Pareciera que, **asumiendo como cierta la hipótesis nula, ver el resultado de 29.2 es poco probable**. ¿Qué tan poco?

Prueba de hipótesis con **infer**

```
distribucion_nula %>%
  get_p_value(obs_stat = dif_observada,
              direction = "right")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.019
```

- **La probabilidad de observar una diferencia en ascensos al menos tan grande como 29.2** puntos porcentuales, dado que la distribución nula sea cierta, es $0.019 = 1.9\%$.
- Ya que este **p-value** es menor al nivel de significancia planteado inicialmente, $\alpha = 0.05$, **rechazamos la hipótesis nula**, $H_0 : p_m - p_f = 0$.
- En otras palabras, hay evidencia para hacernos cambiar de opinión sobre el *status quo*, H_0, y pensar que si hay discriminación.
- Noten que la conclusión depende del nivel de α definido.

Prueba de hipótesis vs I.C.

Una ventaja del paquete `infer` es que podemos pasar rápidamente de hacer una prueba de hipótesis a construir un intervalo de confianza.

```
distribucion_nula <- ascensos %>%
  specify(formula = decision ~ sexo,
          success = "Con ascenso") %>%
  hypothesise(null = "independence") %>%
  generate(reps = 1000,
           type = "permute") %>%
  calculate(stat = "diff in props",
            order = c("masculino", "femenino"))
```

```
distribucion_bootstrap <- ascensos %>%
  specify(formula = decision ~ sexo,
          success = "Con ascenso") %>%
  #hypothesise(null = "independence") %>%
  generate(reps = 1000,
           type = "bootstrap") %>%
  calculate(stat = "diff in props",
            order = c("masculino", "femenino"))
```

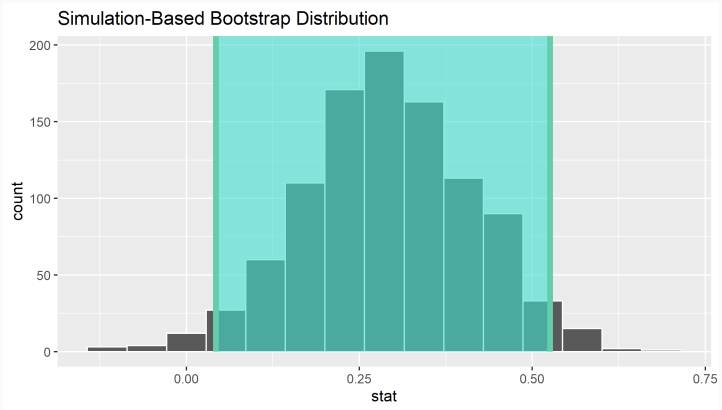
Prueba de hipótesis vs I.C.

Calculemos el intervalo de confianza (95%)

```
(ic_percentil ← distribucion_bootstrap %>%
  get_confidence_interval(level = 0.95,
                          type = "percentile"))
```

```
distribucion_bootstrap %>%
  visualise() +
  shade_confidence_interval(endpoints = ic_percentil)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   0.0423    0.525
```



- Noten que en este intervalo de confianza **un valor que NO cae dentro es 0**.
- Prácticamente toda la distribución está sobre 0, sugiriendo que la diferencia de ascensos es en favor de los hombres.

Prueba de hipótesis basada en teoría

- Al igual como construimos I.C. a partir de formulas, podemos hacer lo mismo con las pruebas de hipótesis.
- Nuevamente, estos métodos se desarrollaron hace mucho tiempo y se convirtieron en norma ante la imposibilidad de poder hacer miles de cálculos de forma simple y eficiente.
- A través de las formulas derivadas hace tiempo se logra aproximar lo que hoy podemos hacer a través de simulaciones.

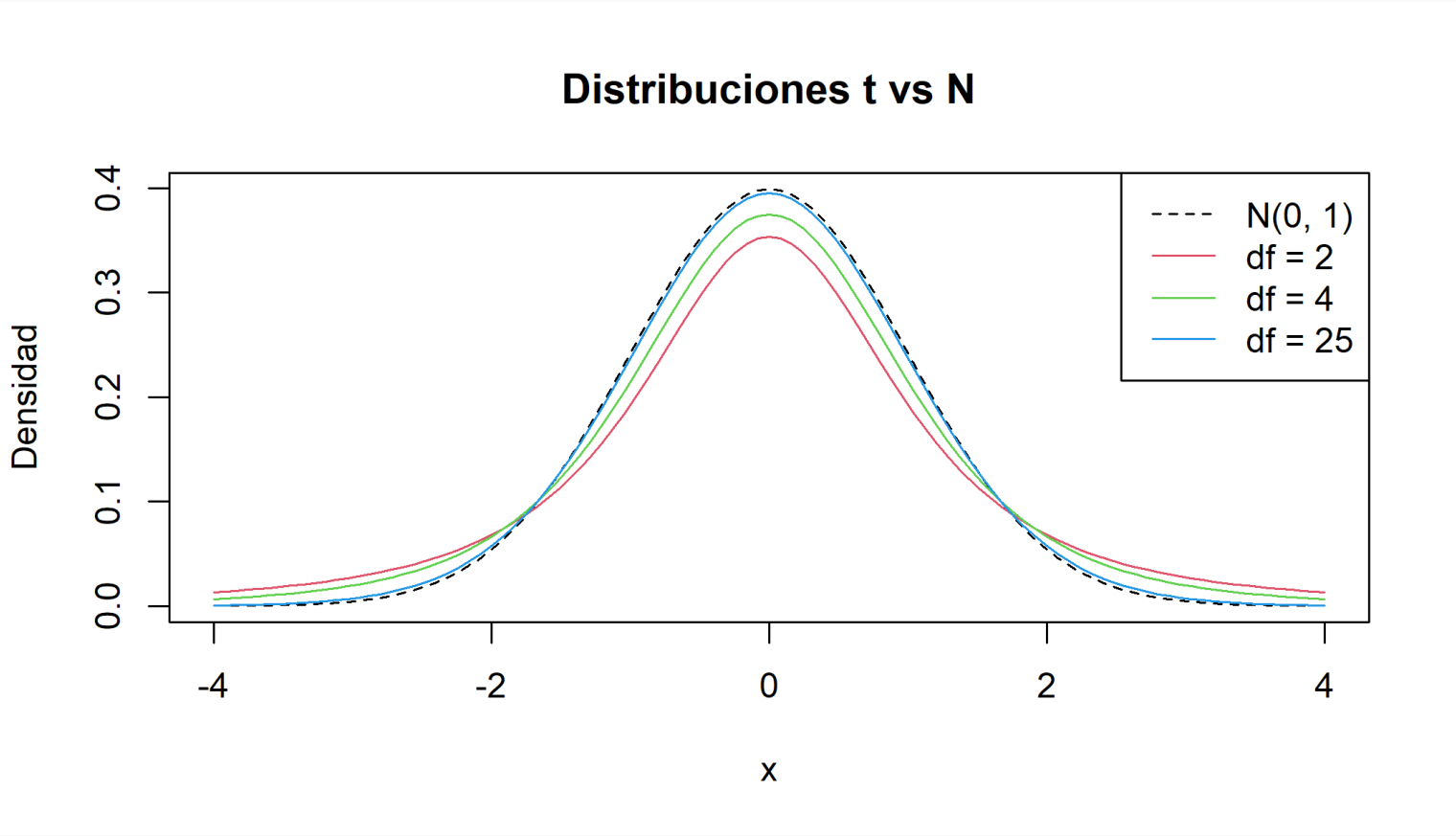
Recordemos: Z-score

$$Z = \frac{X - \mu}{\sigma}$$

- El **Z-score** es una estandarización de una variable aleatoria, X, en términos de la media poblacional, μ, y su desviación estándar, σ.
- Esto resulta en que cada valor de X estandarizado ahora representa a cuantas desviaciones estándar de la media se encuentra ese valor.
- Al estandarizar podemos comparar variables.
- **Un valor estandarizado que ocuparemos normalmente en inferencia será el estadístico-t**:

$$\frac{\hat{\mu} - \mu}{\frac{s}{\sqrt{n}}} \sim t_{n-1}$$

Distribución t



$$PDF = \frac{\Gamma(\frac{n}{2})}{\sqrt{(n-1)\pi}\Gamma(\frac{n-1}{2})}(1+\frac{x^2}{n-1})^{-\frac{n}{2}}$$

¿Cómo usar esto en hipótesis?

Ejemplo: Promedio de edad

$$H_0 = 35;\ H_A \neq 35$$

```
muestra_censo %>%
  summarise(promedio_muestra = mean(edad),
            s = sd(edad))
```

```
##   promedio_muestra       s
## 1         38.12667 22.1389
```

$$t = \frac{\hat{\mu} - \mu}{\frac{s}{\sqrt{n}}} = \frac{38.127 - 35}{\frac{22.1389}{\sqrt{300}}} = 2.446$$


```
pt(2.446, df = 299, lower.tail = FALSE)*2
```

```
## [1] 0.01502134
```

$pvalue < \alpha = 0.015 < 0.05$, se rechaza H_0

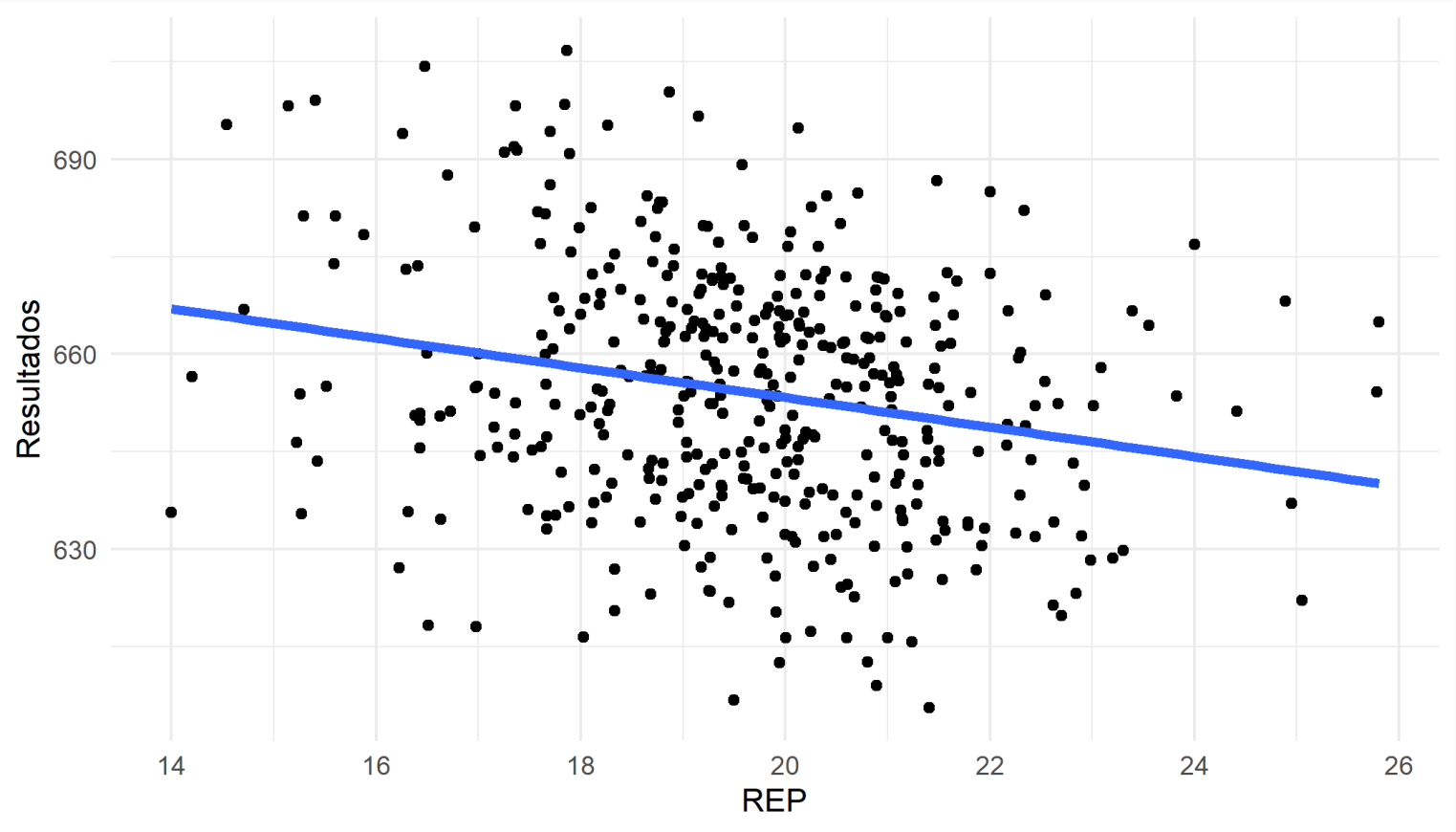
Ejercicio

Ejercicio

- `EjercicioPruebaHipotesis.R`

Próxima clase

- Regresiones



Próxima clase

```
Call:
lm(formula = ROLL ~ UNEM + HGRAD + INC, data = datavar)

Residuals:
      Min        1Q    Median        3Q       Max
-1148.840  -489.712    -1.876   387.400  1425.753

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.153e+03  1.053e+03  -8.691 5.02e-09 ***
UNEM         4.501e+02  1.182e+02   3.809 0.000807 ***
HGRAD        4.065e-01  7.602e-02   5.347 1.52e-05 ***
INC          4.275e+00  4.947e-01   8.642 5.59e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 670.4 on 25 degrees of freedom
Multiple R-squared: 0.9621, Adjusted R-squared: 0.9576
F-statistic: 211.5 on 3 and 25 DF,  p-value: < 2.2e-16
```