06-tcl-bootstrap.R

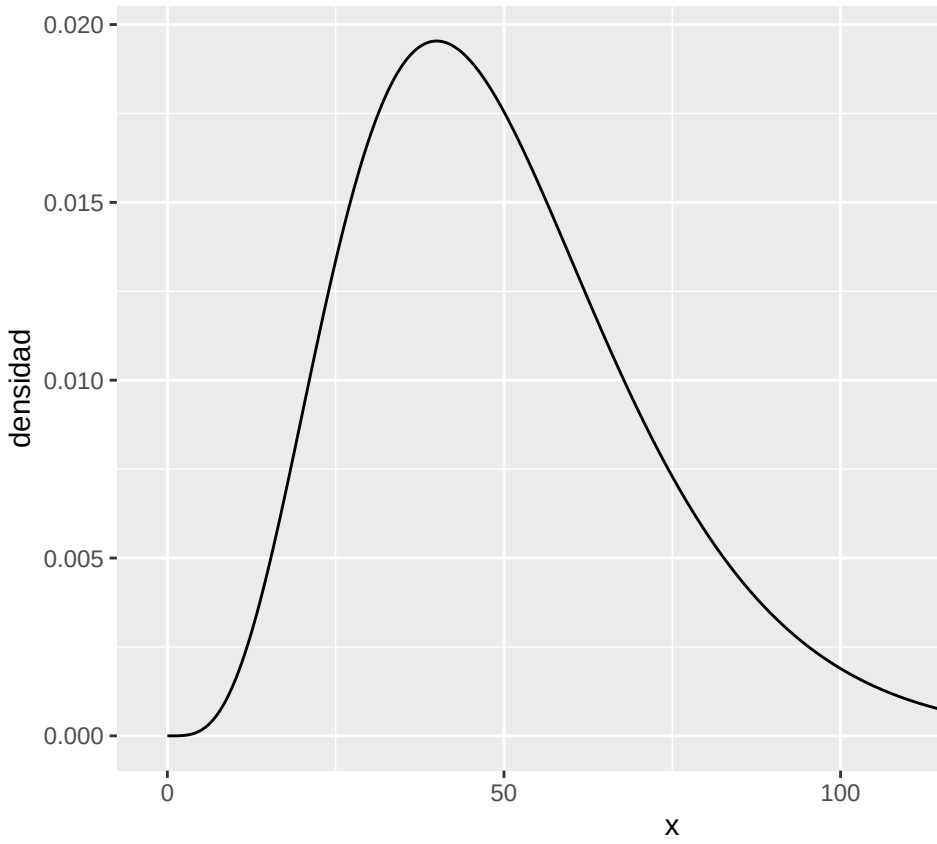
Enviar reporte con respuestas y código

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.1
## v tidyr   1.1.1     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

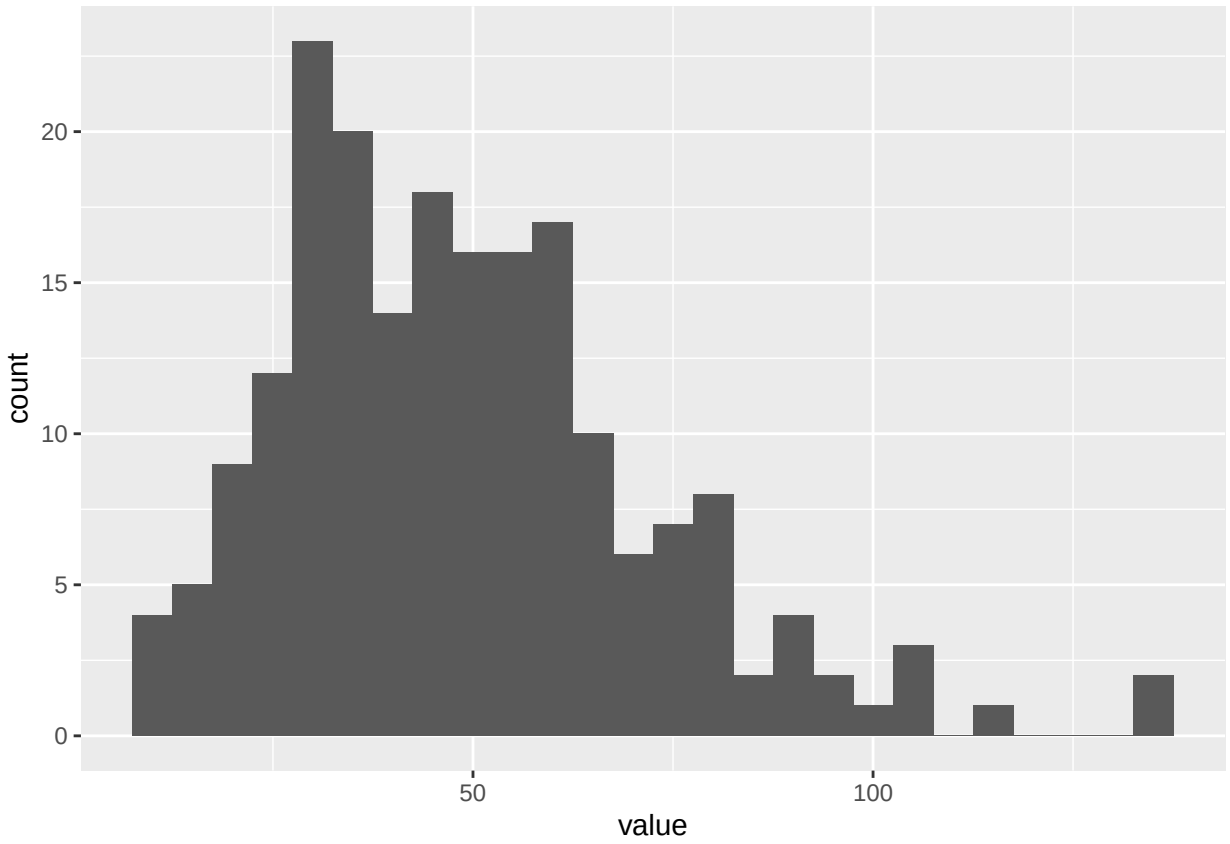
```
library(patchwork)
```

```
#### Ejemplo 1 ####
# Consideramos la distribución gamma con
# a = 5, tasa lambda = 0.1. Su media teórica es 50 = 5/0.1
# cuya densidad teórica es

x <- seq(0, 150, 0.01)
tibble(x = x) %>%
  mutate(densidad = dgamma(x, 5, 0.1)) %>%
  ggplot(aes(x = x, y = densidad)) + geom_line()
```



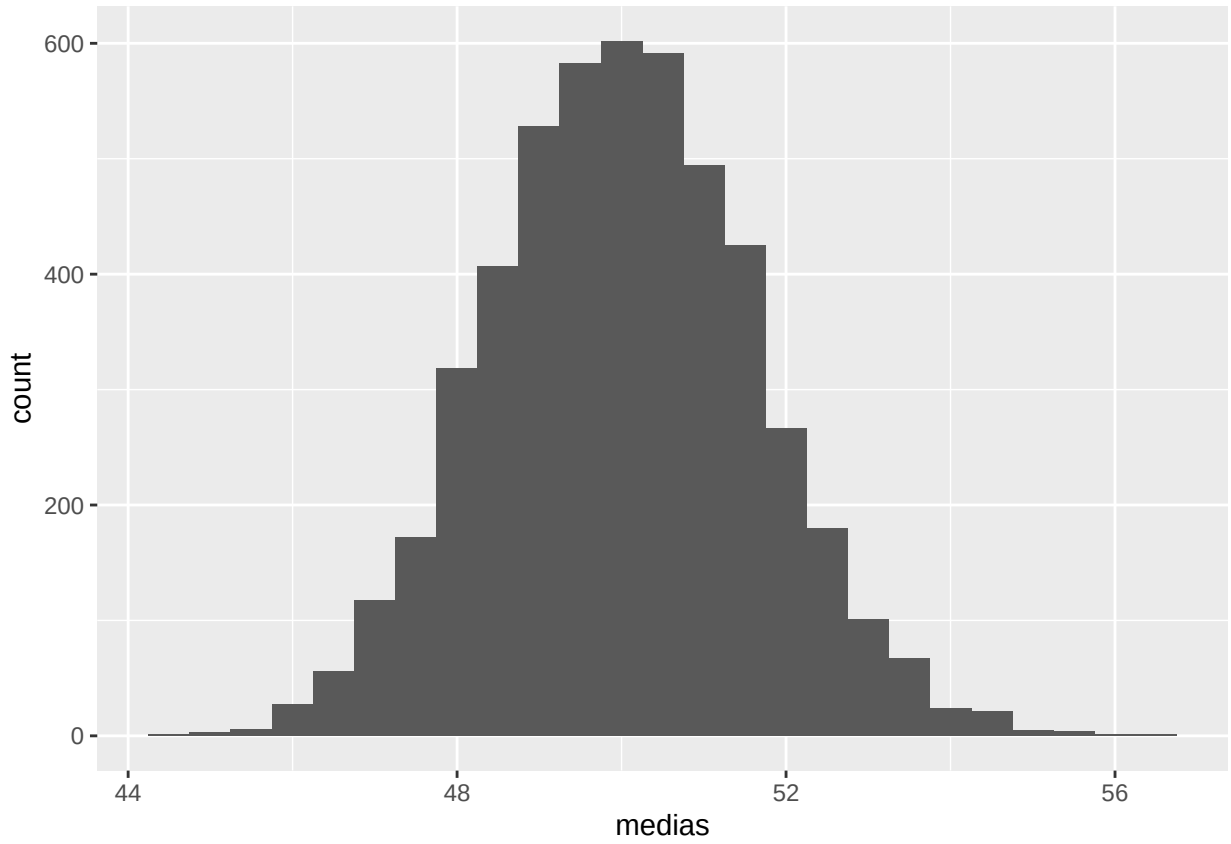
Ejercicios: teorema central del límite

```
# tomamos una muestra:
set.seed(232)
n <- 200
muestra <- rgamma(n, 5, 0.1) %>%
  as_tibble

## La distribución de los datos se ve como sigue
ggplot(
  muestra,
  aes(x = value)
) +
  geom_histogram(binwidth = 5)
```



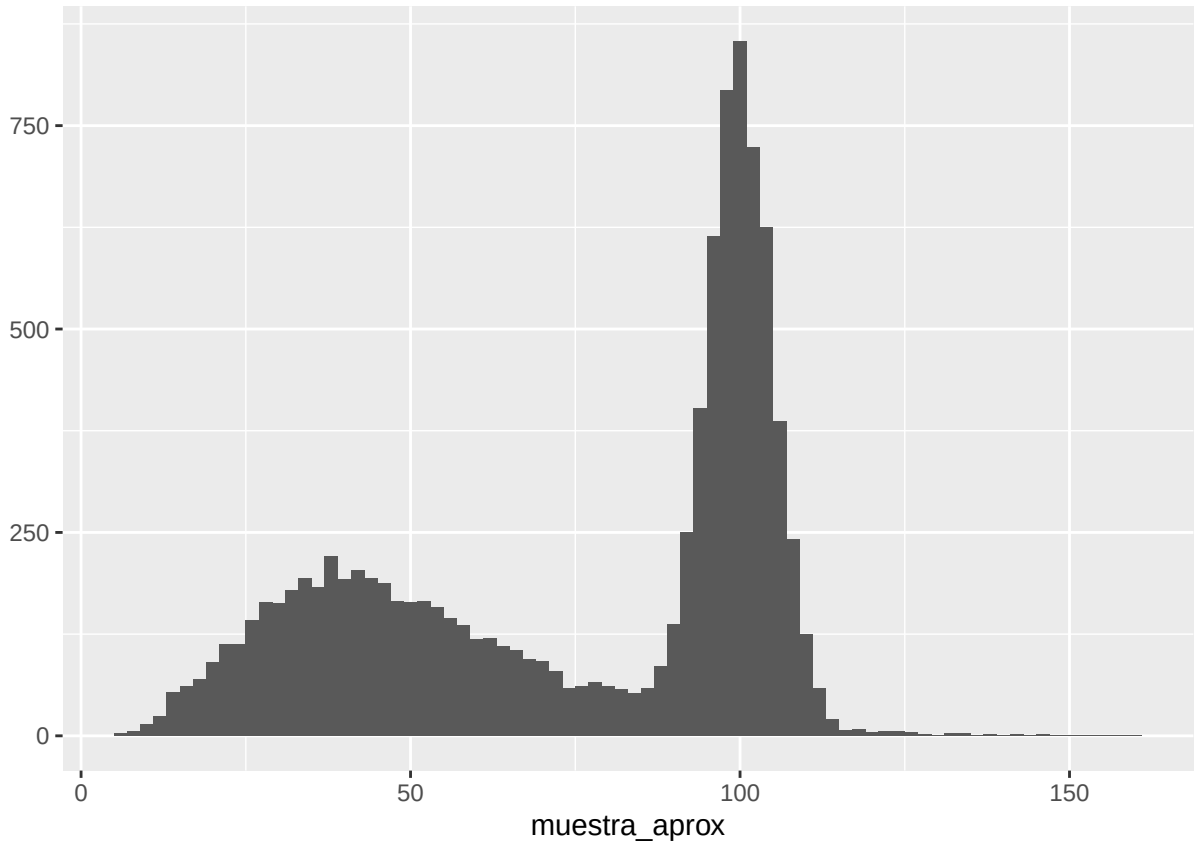
¿Parece tener distribución normal?

```
# Ahora consideramos la distribución de muestreo de
# la media de esta distribución, con tamaño de muestra
# fijo n
medias <- map_dbl(1:5000,  ~ mean(rgamma(n, 5, 0.1)))
medias_gamma <- tibble(medias = medias)

ggplot(
  medias_gamma,
  aes(
    x = medias
  )
) +
  geom_histogram(binwidth = 0.5)
```

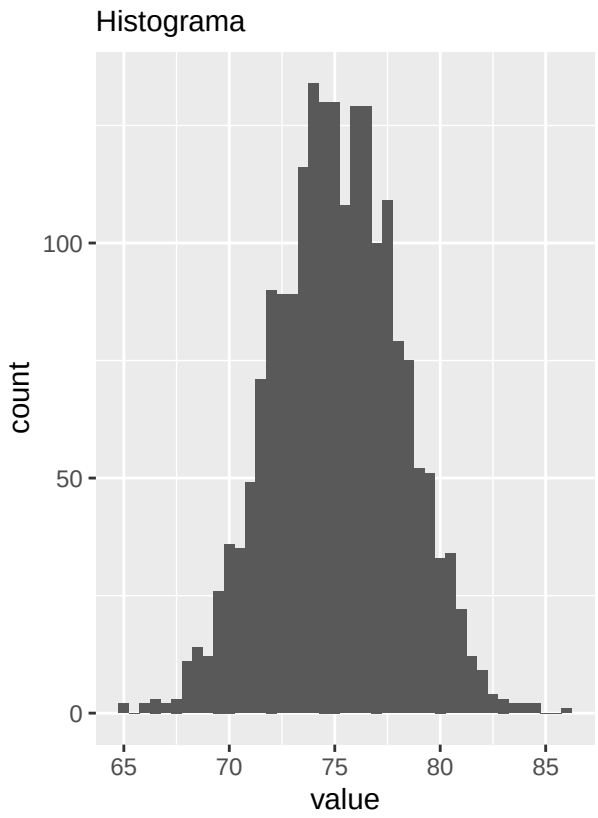


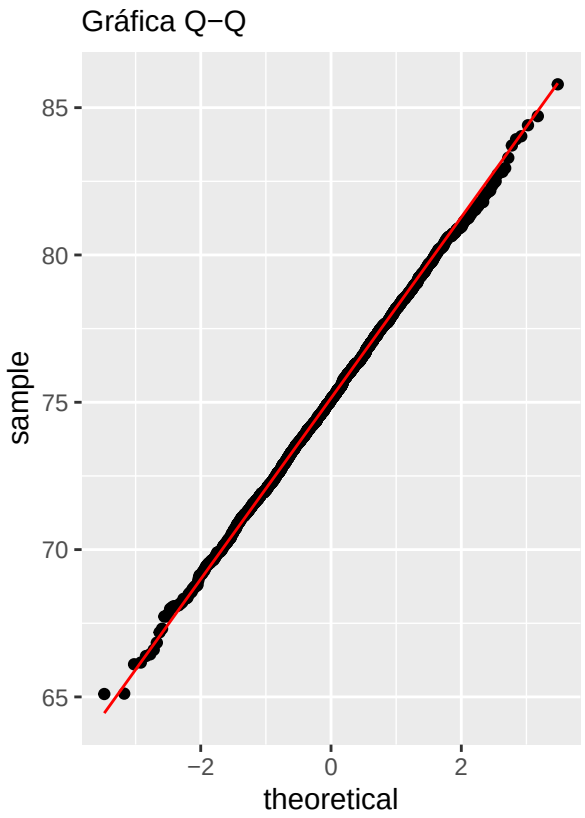

```
####### Ejemplo: mezcla de distribuciones
# Este ejemplo es más complicado. Imaginemos
# que nuestro modelo teórico es una mezcla
# de dos poblaciones, una gamma y una normal
muestrear_pob <- function(n){
  u <- runif(n) # número aleatorio
  map_dbl(u, ~ ifelse(.x < 1/2, rgamma(1, 5, 0.1), rnorm(1, 100, 5)))
}
```

```
# El modelo teórico se puede graficar, pero también podemos
# obetener una aproximación buena haciendo una cantidad grande
# de simulaciones
muestra_aprox <- muestrear_pob(10000)
qplot(muestra_aprox, binwidth= 2)
```




```r
## Ahora consideramos estimar la media de esta
## distribución con un muestra de tamaño 100
## ¿Cómo se ve la distribución de muestreo de la media?
medias <- map_dbl(1:2000,  ~ mean(muestrear_pob(100))) %>%
  as_tibble()


## grafica un histograma
g1 <- ggplot(
  medias,
  aes(x = value)
) +
  geom_histogram(binwidth = 0.5) +
  labs(subtitle = "Histograma")


# gráfica cuantil-cuantil normal
g2 <- ggplot(
  medias,
  aes(sample = value)
) +
  geom_qq(distribution = stats::qnorm) +
  geom_qq_line(colour = "red") +
  labs(subtitle = "Gráfica Q-Q")

g1 + g2
```





¿Cómo se ve la distribución de muestreo de la media?

```
#### Ejemplo discreto ####
# Tomaremos muestra de unos y ceros
set.seed(1212)
n_volados <- 200
muestra <- rbinom(n_volados, 1, prob = 0.7)
head(muestra)
```

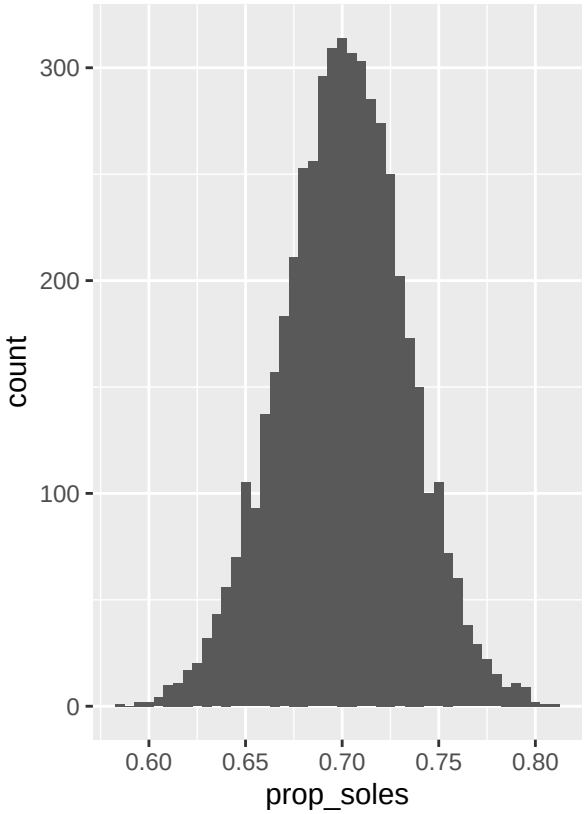
```
## [1] 1 1 0 1 1 1
```

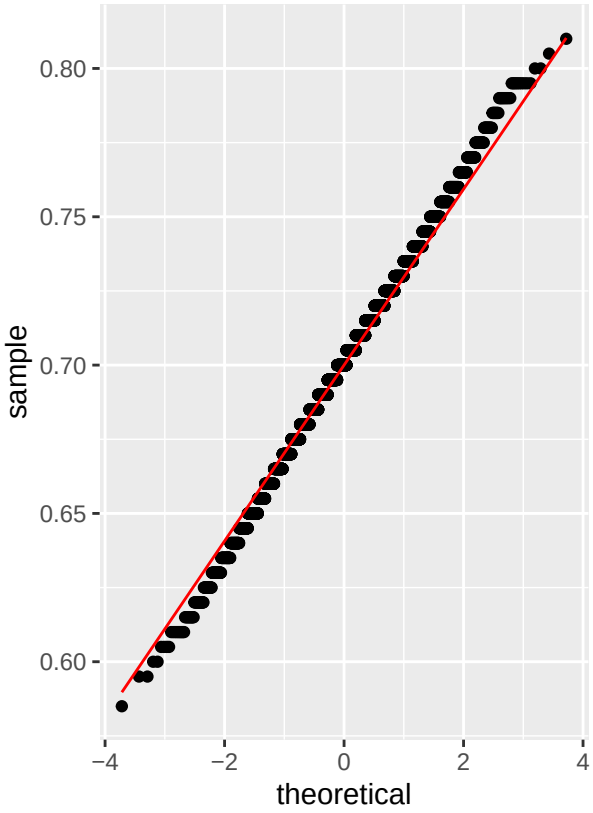
```
# la media es la proporción de unos en la muestra,
# o la proporción de "soles":
mean(muestra)
```

```
## [1] 0.785
```

```
## ¿Cuál es la distribución de muestreo para la proporción
# de soles en la muestra?
prop_soles <- map_dbl(1:5000,  ~ mean(rbinom(n_volados, 1, prob = 0.7)))
prop_soles_tbl <- tibble(prop_soles = prop_soles)
```

```
## checa un histograma, ¿se ve normal? También ve una gráfica qq
g1 <- ggplot(prop_soles_tbl, aes(x = prop_soles)) + geom_histogram(binwidth = 0.005)
g2 <- ggplot(prop_soles_tbl, aes(sample = prop_soles)) + geom_qq(distribution = stats::qnorm) + geom_qq
g1 + g2
```





checa un histograma, ¿se ve normal? También ve una gráfica qq

```
#### Error estándar e intervalos bootstrap normales

## Ejemplo 1: error estándar de una media
# Retomaremos el ejemplo de la prueba ENLACE de la tarea anterior
# Para cada tamaño de muestra n = 10, 100, 1000
# i) Selecciona una muestra y utilizala para estimar la media de las
#   calificaciones de español 3o de primaria
# ii) Utiliza bootstrap para calcular el error estándar de tu estimador
# iii) Grafica la distribución bootstrap


##Lectura de los datos
en_data <- read_csv("enlace_15.csv")
```

```
## Parsed with column specification:
## cols(
##   id = col_double(),
##   cve_ent = col_double(),
##   turno = col_character(),
##   tipo = col_character(),
##   esp_3 = col_double(),
##   esp_6 = col_double(),
##   n_eval_3 = col_double(),
##   n_eval_6 = col_double()
## )
```

```
## Toma de muestras
s10 <- sample_n(en_data, 10)
```

```r
s100 <- sample_n(en_data, 100)
s1000 <- sample_n(en_data, 1000)


## Estimación de la media de las calificaciones con las distintas muestras
sprintf("Media con base en muestra de tamaño 10: %0.2f", mean(s10$esp_3))
```

```
## [1] "Media con base en muestra de tamaño 10: 557.10"
```

```r
sprintf("Media con base en muestra de tamaño 100: %0.2f", mean(s100$esp_3))
```

```
## [1] "Media con base en muestra de tamaño 100: 542.44"
```

```r
sprintf("Media con base en muestra de tamaño 1000: %0.2f", mean(s1000$esp_3))
```

```
## [1] "Media con base en muestra de tamaño 1000: 552.08"
```

```r
print("")
```

```
## [1] ""
```

```r
## Cálculo del error estándar de nuestro estimador

#### Simlulación de muestreo sobre las distintas muestras (10, 100 y 1000 elementos)
sim_mean_s <- function(mother_sample, sim_sample){
  map_dbl(1:500, ~ mother_sample %>%
    sample_n(sim_sample, replace = T) %>%
    summarise(media_cal = mean(esp_3), .groups = "drop") %>%
    pull(media_cal))
}
sim_mean_s10 <- sim_mean_s(s10, 10)
sim_mean_s100 <- sim_mean_s(s100, 100)
sim_mean_s1000 <- sim_mean_s(s1000, 1000)

#### Cálculo del error estándar de remuestreo e impresión de resultados
sprintf("Muestra tamaño 10 -> Media: %0.2f    Error estándar: %0.2f", mean(s10$esp_3), sd(sim_mean_s10))
```

```
## [1] "Muestra tamaño 10 -> Media: 557.10    Error estándar: 14.90"
```

```r
sprintf("Muestra tamaño 100 -> Media: %0.2f    Error estándar: %0.2f", mean(s100$esp_3), sd(sim_mean_s1
```

```
## [1] "Muestra tamaño 100 -> Media: 542.44    Error estándar: 6.23"
```

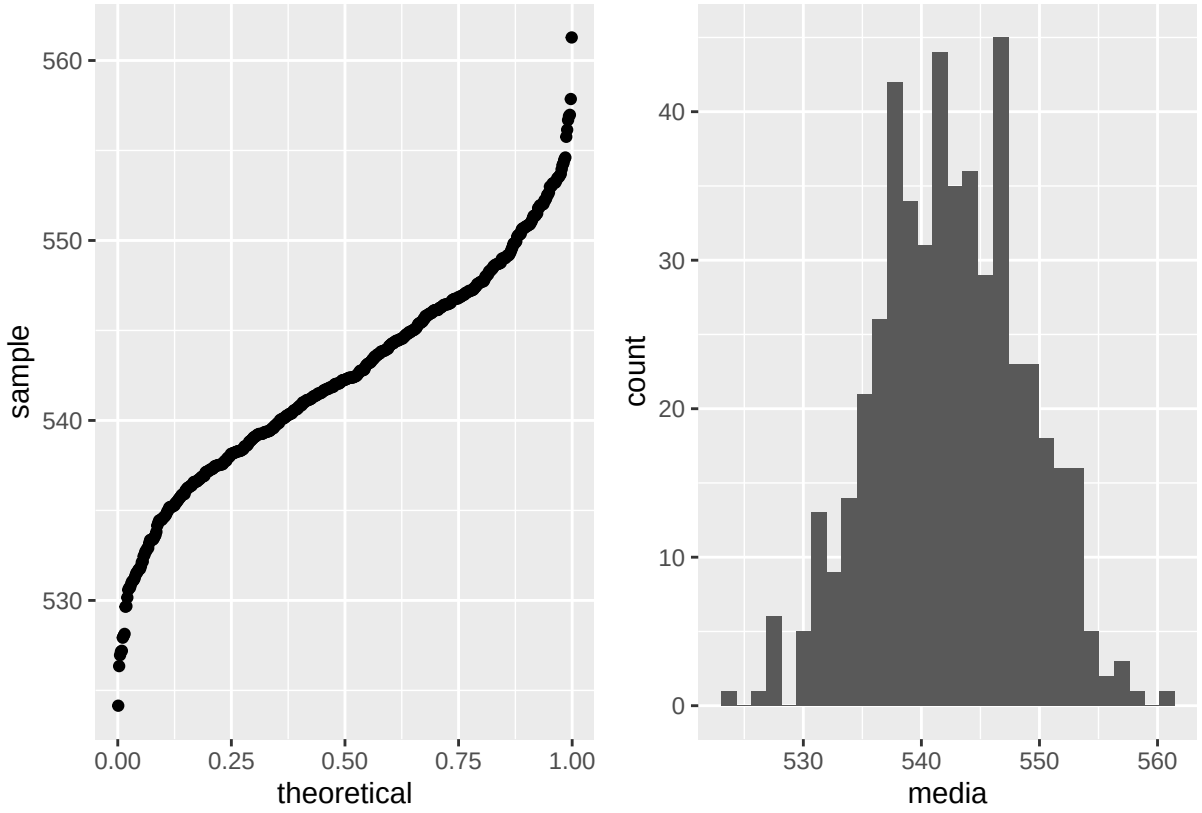
```r
sprintf("Muestra tamaño 1000 -> Media: %0.2f    Error estándar: %0.2f", mean(s1000$esp_3), sd(sim_mean_
```

```
## [1] "Muestra tamaño 1000 -> Media: 552.08    Error estándar: 1.77"
```

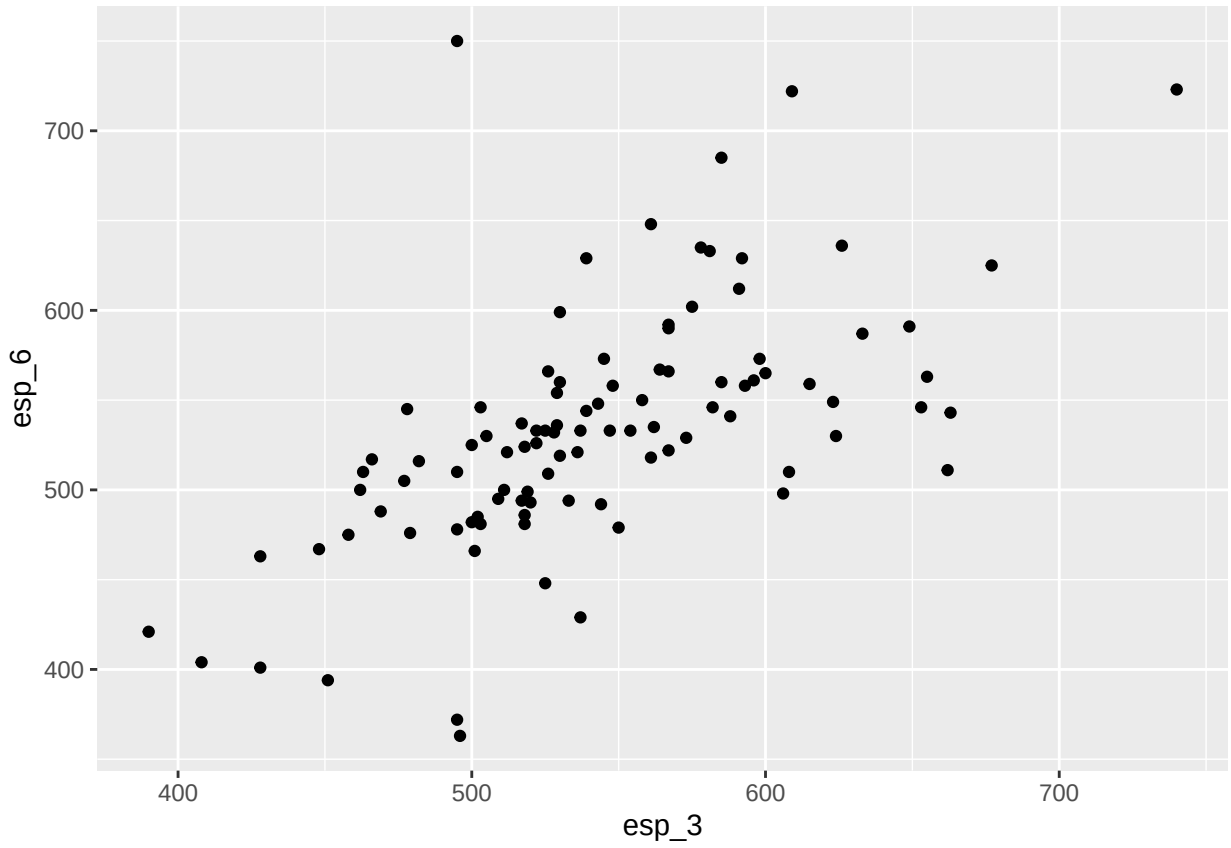
```r
## Gráfica de la distribución de bootstrap
g_bs <- function(sample_mean) {
  plt_data <- tibble(media = sample_mean)
  g1 <- ggplot(
    plt_data,
    aes(sample = media)
  ) +
    geom_qq(distribution = stats::qunif)
  g2 <- ggplot(
    plt_data,
    aes(x = media)
  ) +
```

```
    geom_histogram()

  g1 + g2
}

g_bs(sim_mean_s100)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Retoma la muestra de tamaño 100, y calcula la correlación entre las
# calificaciones de español 3o y 6o de primaria
# Utiliza bootstrap para calcular el error estandar

s100 %>%
  ggplot(
    aes(x = esp_3, y = esp_6)
  ) +
  geom_point()
```


700
600
500
400
esp_6
400
500
600
700
esp_3