Tarea 05 Distribución Muestral

Rodrigo Alan Garcia Perez

2024-09-18

1. **Proporciones.** Usaremos datos de reincidencia en conducta criminal del estado de Iowa, este estado sigue a los delincuentes por un periodo de 3 años y registra el número de días hasta reincidencia para aquellos que son readmitidos en prisión. El departamento de correcciones utiliza los datos de reincidencia para evaluar sus programas de prevención de recaída en conducta criminal.

Los datos Recidivism contienen información de todos los delincuentes condenados por dos tipos de delito durante 2010 (*Recid* indica si recayeron en conducta criminal).

- De éstos 31.6% reincidieron y volvieron a prisión. Utiliza simulación para aproximar la simulación muestral de $\hat{p}$, la proporción de delincuentes que reincidieron para muestras de tamaño

  25.

- Calcula el error estándar de $\hat{p}$, y compáralo con el teórico $\sqrt{p(1-p)/n}$.

- Repite para muestras de tamaño 250 y compara.

```
set.seed(123)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
recidivism <- read_csv("Recidivism.csv")
```

```
## Rows: 17022 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): Gender, Age, Age25, Race, Offense, Recid, Type
## dbl (1): Days
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(recidivism)
```

```
## Rows: 17,022
## Columns: 8
## $ Gender  <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M~
## $ Age     <chr> "Under 25", "55 and Older", "25-34", "55 and Older", "25-34", ~
```

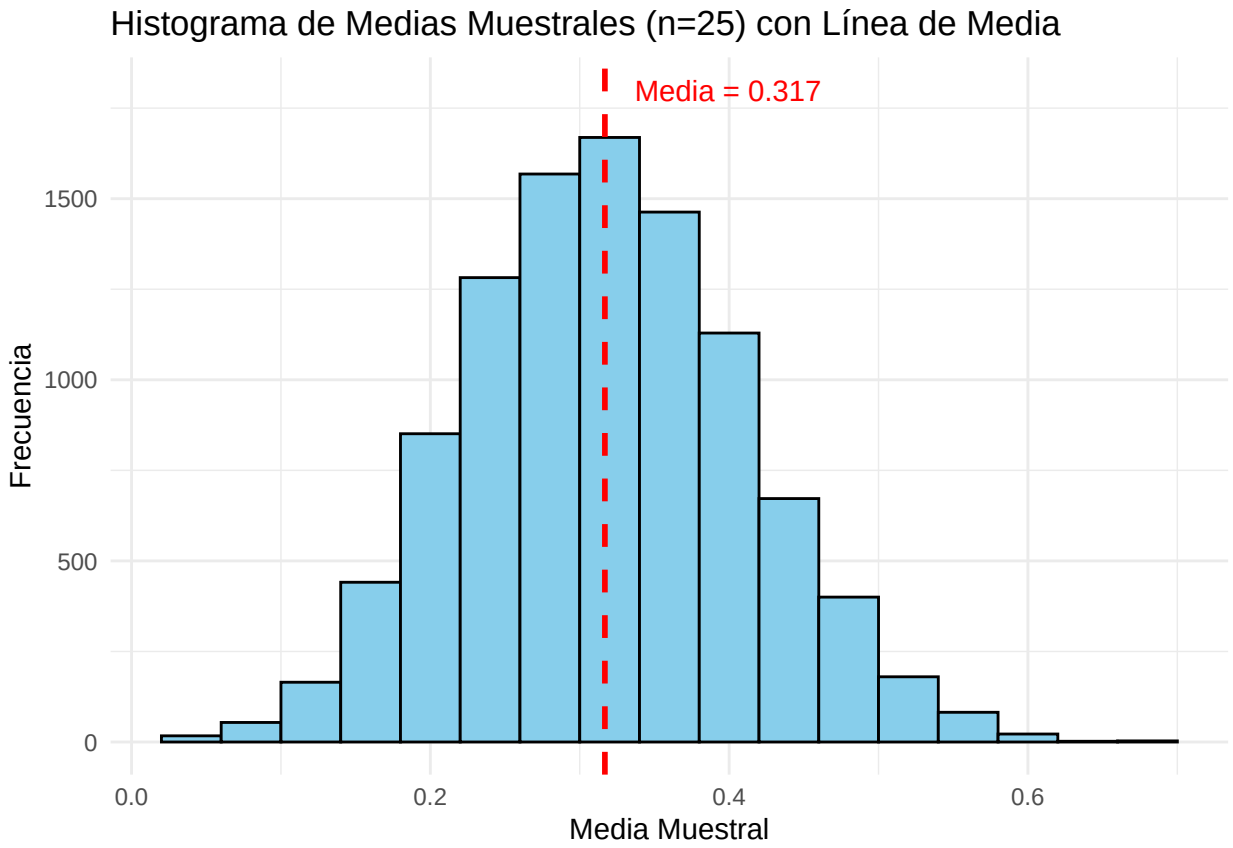
```
## $ Age25   <chr> "Under 25", "Over 25", "Over 25", "Over 25", "Over 25", "Under~
## $ Race    <chr> "White-NonHispanic", "White-NonHispanic", "White-NonHispanic",~
## $ Offense <chr> "Felony", "Felony", "Felony", "Felony", "Felony", "Felony", "M~
## $ Recid   <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes",~
## $ Type    <chr> "Tech", "Tech", "Tech", "Tech", "Tech", "Tech", "New", "Tech",~
## $ Days    <dbl> 16, 19, 22, 25, 26, 27, 28, 41, 44, 46, 48, 49, 49, 51, 51, 53~
```

```r
# Necesitamos cambiar Recid a un formato numerico
recidivism$Recid <- ifelse(recidivism$Recid == "Yes", 1, 0)
glimpse(recidivism)
```

```
## Rows: 17,022
## Columns: 8
## $ Gender  <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M~
## $ Age     <chr> "Under 25", "55 and Older", "25-34", "55 and Older", "25-34", ~
## $ Age25   <chr> "Under 25", "Over 25", "Over 25", "Over 25", "Over 25", "Under~
## $ Race    <chr> "White-NonHispanic", "White-NonHispanic", "White-NonHispanic",~
## $ Offense <chr> "Felony", "Felony", "Felony", "Felony", "Felony", "Felony", "M~
## $ Recid   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ Type    <chr> "Tech", "Tech", "Tech", "Tech", "Tech", "Tech", "New", "Tech",~
## $ Days    <dbl> 16, 19, 22, 25, 26, 27, 28, 41, 44, 46, 48, 49, 49, 51, 51, 53~
```

```r
# Porcentaje de reincidencia conocida
p <- 0.316

# Número de simulaciones
n_sim <- 10000

# Tamaño de muestra para la simulación
sample_size_25 <- 25

# Realizar simulaciones para tamaño de muestra 25
simulations_25 <- replicate(n_sim, {
  sample_data <- sample(recidivism$Recid, sample_size_25, replace = TRUE)
  mean(sample_data)
})

# Calcular la media de simulations_25
mean_simulation_25 <- mean(simulations_25)

# Crear el histograma de las medias muestrales y agregar la línea roja
ggplot(data.frame(simulations_25), aes(x = simulations_25)) +
  geom_histogram(binwidth = 0.04, fill = "skyblue", color = "black") +
  geom_vline(aes(xintercept = mean_simulation_25), color = "red", linetype = "dashed", size = 1) +
    annotate("text", x = mean_simulation_25 + 0.02, y = 1800, label = paste("Media =", round(mean_simula
           color = "red", hjust = 0) +
  labs(title = "Histograma de Medias Muestrales (n=25) con Línea de Media",
       x = "Media Muestral",
       y = "Frecuencia") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
# Calcular el error estándar (p-hat) para muestras de tamaño 25
error_estandar_25 <- sd(simulations_25)
error_estandar_25
```

```
## [1] 0.09430503
```

```
# Calcular el error teorico
# Se dice teorico porque conocemos p
error_teorico_25 <- sqrt(p * (1 - p) / sample_size_25)
error_teorico_25
```

```
## [1] 0.09298258
```

```
cat("Error estándar empírico para muestra de tamaño 25:", error_estandar_25, "\n")
```

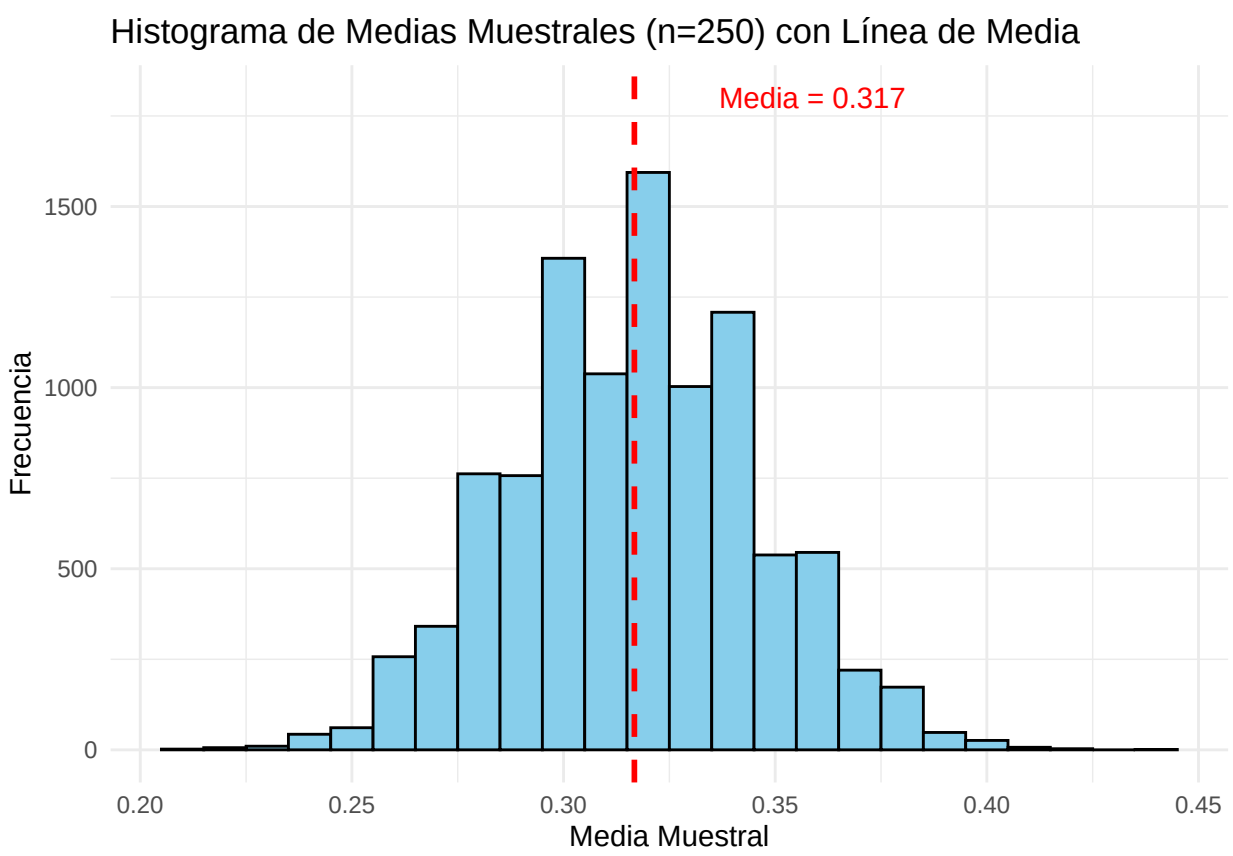
```
## Error estándar empírico para muestra de tamaño 25: 0.09430503
```

```
cat("Error estándar teórico para muestra de tamaño 25:", error_teorico_25, "\n")
```

```
## Error estándar teórico para muestra de tamaño 25: 0.09298258
```

```
# Repetir el proceso para muestras de tamaño 250
sample_size_250 <- 250

# Realizar simulaciones para tamaño de muestra 250
simulations_250 <- replicate(n_sim, {
  sample_data <- sample(recidivism$Recid, sample_size_250, replace = TRUE)
  mean(sample_data)
})
```

```r
# Calcular la media de simulations_250
mean_simulation_250 <- mean(simulations_250)

# Crear el histograma de las medias muestrales y agregar la línea roja
ggplot(data.frame(simulations_250), aes(x = simulations_250)) +
  geom_histogram(binwidth = 0.01, fill = "skyblue", color = "black") +
  geom_vline(aes(xintercept = mean_simulation_250), color = "red", linetype = "dashed", size = 1) +
    annotate("text", x = mean_simulation_250 + 0.02, y = 1800, label = paste("Media =", round(mean_simul
           color = "red", hjust = 0) +
  labs(title = "Histograma de Medias Muestrales (n=250) con Línea de Media",
       x = "Media Muestral",
       y = "Frecuencia") +
  theme_minimal()
```




```r
# Calcular el error estándar empírico de la proporción muestral (p-hat) para muestras de t
empirical_se_250 <- sd(simulations_250) # Calcular el error estándar teórico
theoretical_se_250 <- sqrt(p * (1 - p) / sample_size_250)

# Mostrar los resultados
cat("Error estándar empírico para muestra de tamaño 250:", empirical_se_250, "\n")
```

```
## Error estándar empírico para muestra de tamaño 250: 0.02915214
```

```r
cat("Error estándar teórico para muestra de tamaño 250:", theoretical_se_250, "\n")
```

```
## Error estándar teórico para muestra de tamaño 250: 0.02940367
```

2. **Mezcla de distribuciones**. Imaginemos que nuestro modelo teórico es una mezcla de dos poblaciones,

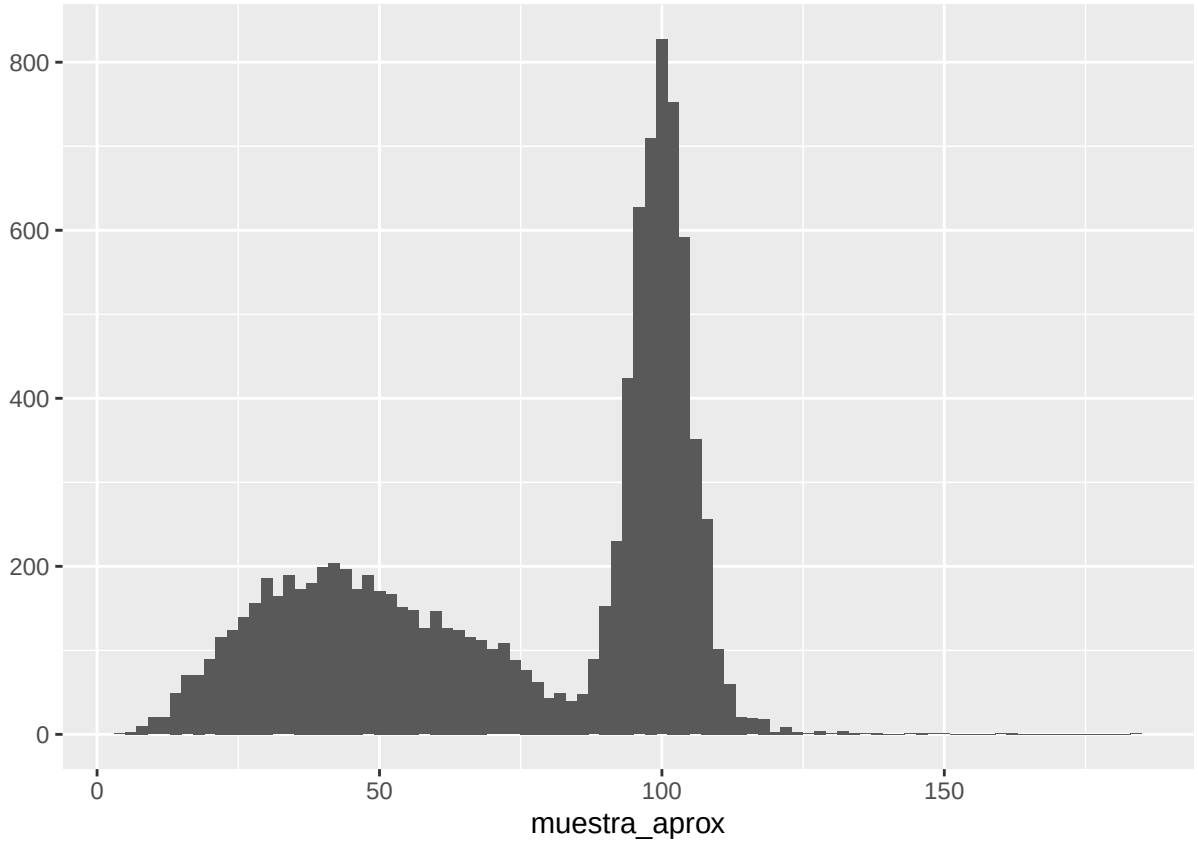
una gamma y una normal

```
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

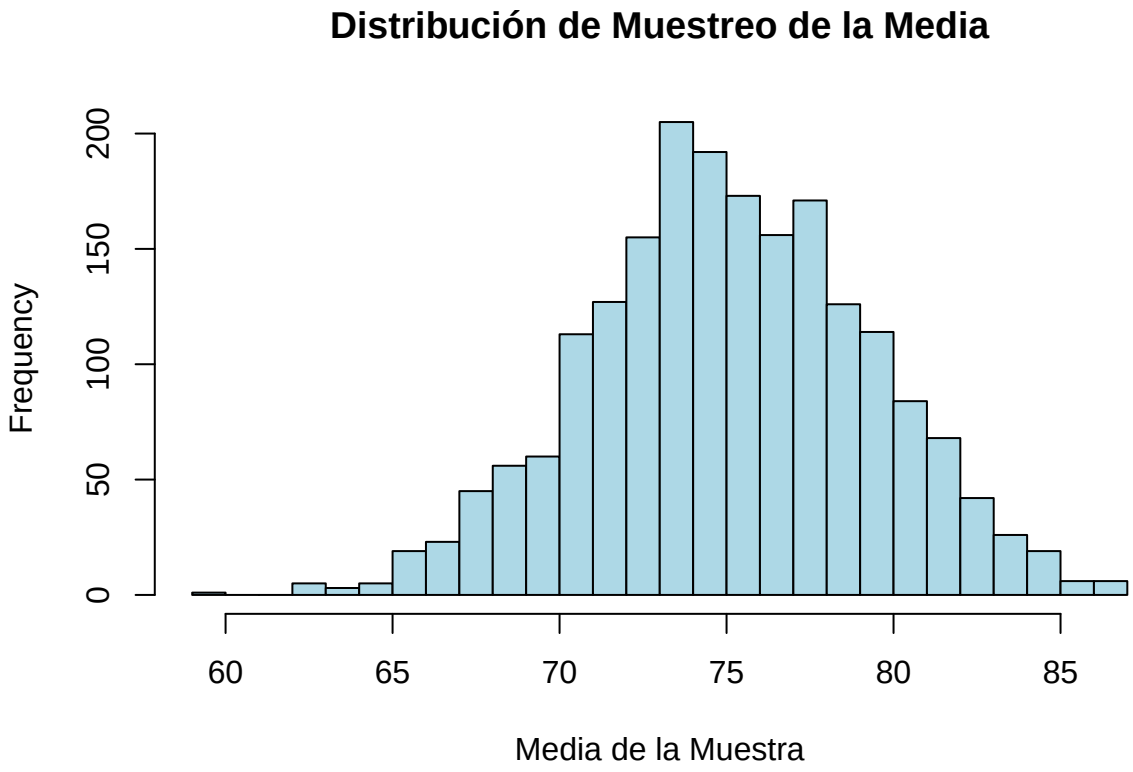
```
muestra_aprox <- muestrear_pob(10000)
qplot(muestra_aprox, binwidth= 2)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

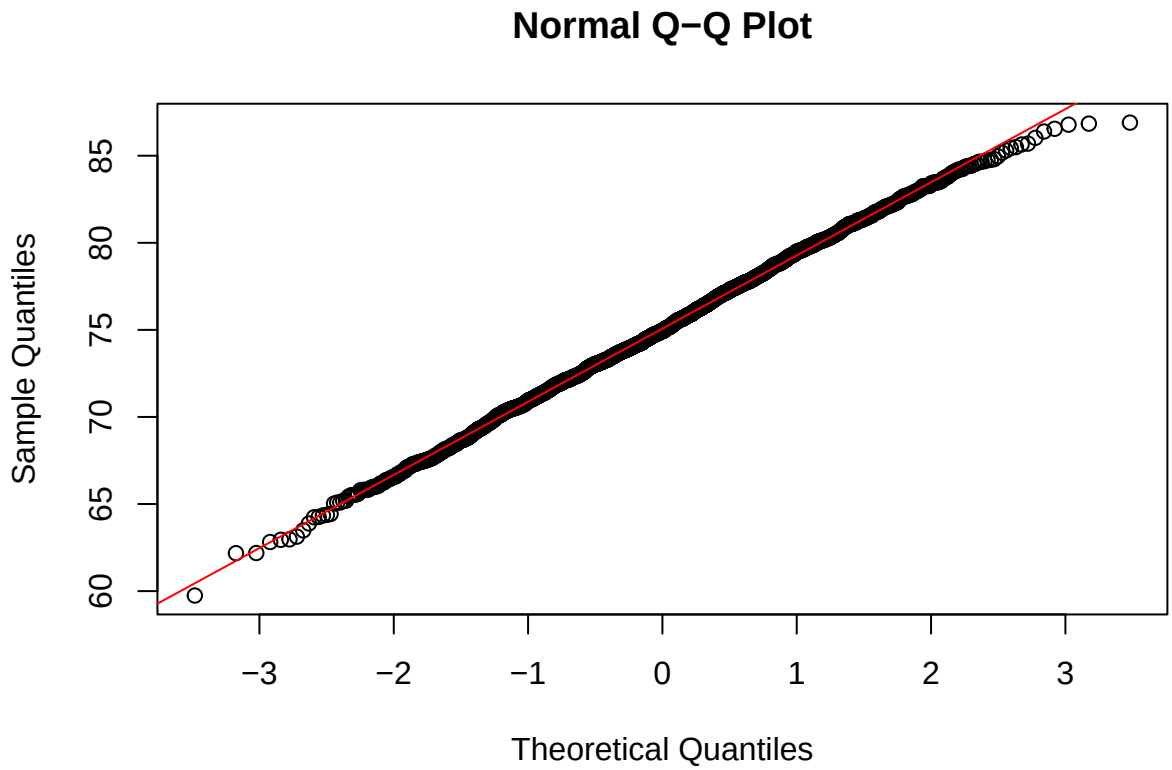


Ahora consideramos estimar la media de esta distribución con un muestra de tamaño 50. ¿Cómo se ve la distribución de muestreo de la media? Grafica un histograma y una gráfica cuantil-cuantil normal.

```
# Generar 2000 muestras de tamaño 50 y calcular sus medias
medias <- map_dbl(1:2000, ~ mean(muestrear_pob(50)))


# Crear un histograma de las medias
hist(medias, breaks = 30, main = "Distribución de Muestreo de la Media",
     xlab = "Media de la Muestra", col = "lightblue", border = "black")
```




```r
# Crear una gráfica cuantil-cuantil (QQ plot) para verificar la normalidad
qqnorm(medias)  # Esta línea crea el QQ plot
qqline(medias, col = "red")  # Esta línea agrega la línea de referencia normal al QQ plot
```



3. **El error estándar de una media.** Supongamos que x es una variable aleatoria que toma valores en

los reales con distribución de probabilidad F. Denotamos por μ y σ^2 la media y varianza de F,

$$\mu = E(x),$$
$$\sigma^2 = var(x) = E[(x-\mu)^2]$$

Ahora, sea $(X_1, ..., X_n)$ una muestra aleatoria de F, de tamaño n, la media de la muestra $\bar{X} = \sum_{i=1}^{n} X_i/n$ tiene:

- esperanza μ,
- varianza σ^2/n.

En palabras: la esperanza de $\bar{X}$ es la misma que la esperanza de x, pero la varianza de $\bar{X}$ es $1/n$ veces la varianza de x, así que entre mayor es la n tenemos una mejor estimación de μ.

En el caso del estimador de la media $\bar{X}$, el error estándar quedaría

$$ee(\bar{X}) = [var(\bar{X})]^{1/2} = \sigma/\sqrt{n}.$$

Entonces,

Consideramos los datos de ENLACE edo. de México (ENLACE era una prueba estandarizada que se aplicaba a todos los alumnos de primaria en México), y la columna de calificaciones de español 3º de primaria (`esp_3`).

```
enlace <- read_csv("enlace_15.csv")
```

```
## Rows: 7518 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): turno, tipo
## dbl (6): id, cve_ent, esp_3, esp_6, n_eval_3, n_eval_6
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

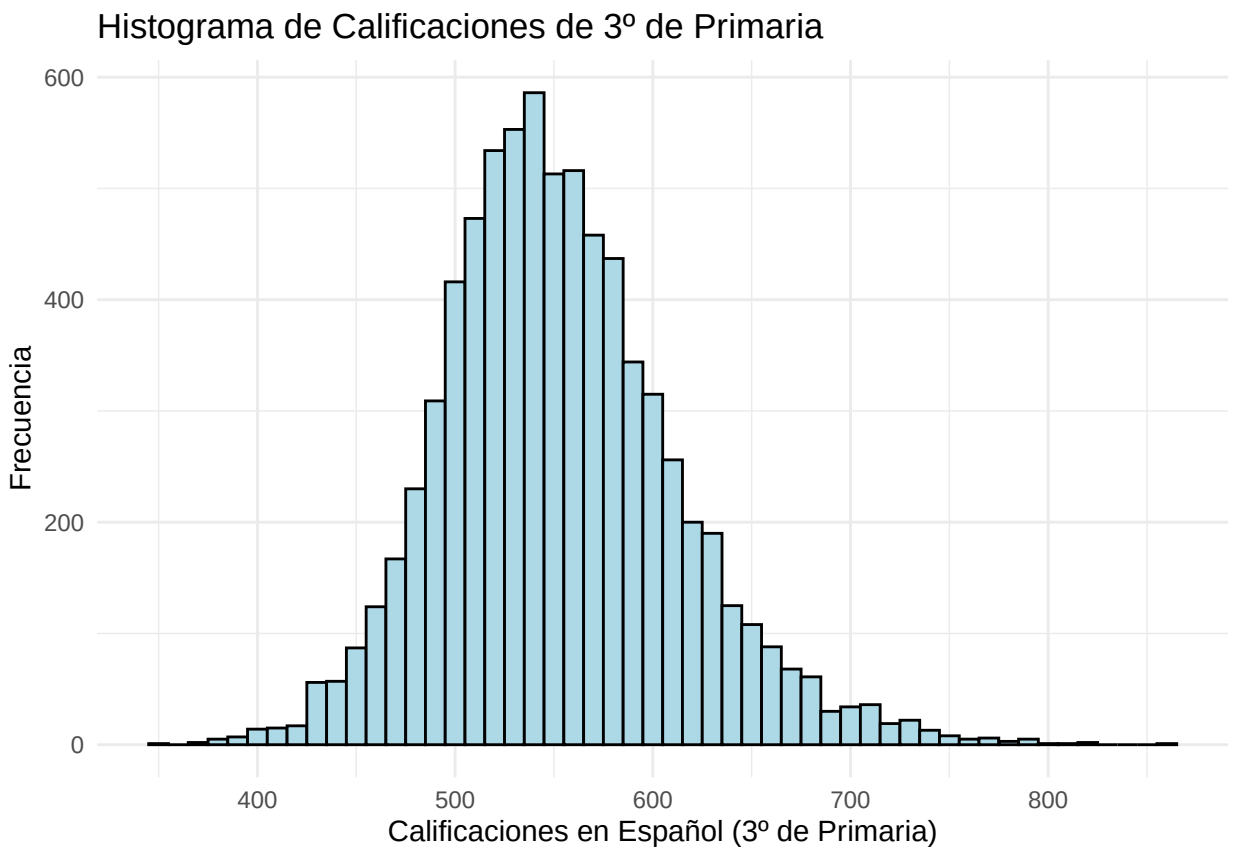
```
glimpse(enlace)
```

```
## Rows: 7,518
## Columns: 8
## $ id       <dbl> 38570, 38571, 38572, 38573, 38574, 38575, 38576, 38577, 38578~
## $ cve_ent  <dbl> 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 1~
## $ turno    <chr> "MATUTINO", "MATUTINO", "MATUTINO", "MATUTINO", "MATUTINO", "~
## $ tipo     <chr> "INDêGENA", "INDêGENA", "INDêGENA", "INDêGENA", "INDêGENA", "~
## $ esp_3    <dbl> 550, 485, 462, 646, 508, 502, 570, 441, 597, 648, 535, 430, 4~
## $ esp_6    <dbl> 483, 490, 385, 613, 452, 500, 454, 427, 582, 614, 443, 562, 4~
## $ n_eval_3 <dbl> 13, 17, 9, 33, 26, 10, 65, 82, 132, 16, 16, 6, 10, 27, 10, 1,~
## $ n_eval_6 <dbl> 19, 18, 9, 26, 35, 13, 49, 78, 110, 18, 9, 2, 12, 34, 9, 6, 7~
```

Genera un histograma de las calificaciones de 3º de primaria. Calcula la media y la desviación estándar.

```
# Cargar la librería ggplot2 para
library(ggplot2)

# Visualizar un histograma de las calificaciones de 3º de primaria
ggplot(enlace, aes(x = esp_3)) +
  geom_histogram(binwidth = 10, fill = "lightblue", color = "black") +
  labs(title = "Histograma de Calificaciones de 3º de Primaria",
       x = "Calificaciones en Español (3º de Primaria)",
```

```
       y = "Frecuencia") +
  theme_minimal()
```




```
media_esp_3 <- mean(enlace$esp_3, na.rm = TRUE)
cat("La media de las calificaciones de 3º de primaria es:", media_esp_3, "\n")
```

```
## La media de las calificaciones de 3º de primaria es: 552.9911
```

```
desviacion_esp_3 <- sd(enlace$esp_3, na.rm = TRUE)
cat("La desviación estándar de las calificaciones de 3º de primaria es:", desviacion_esp_3, "\n")
```

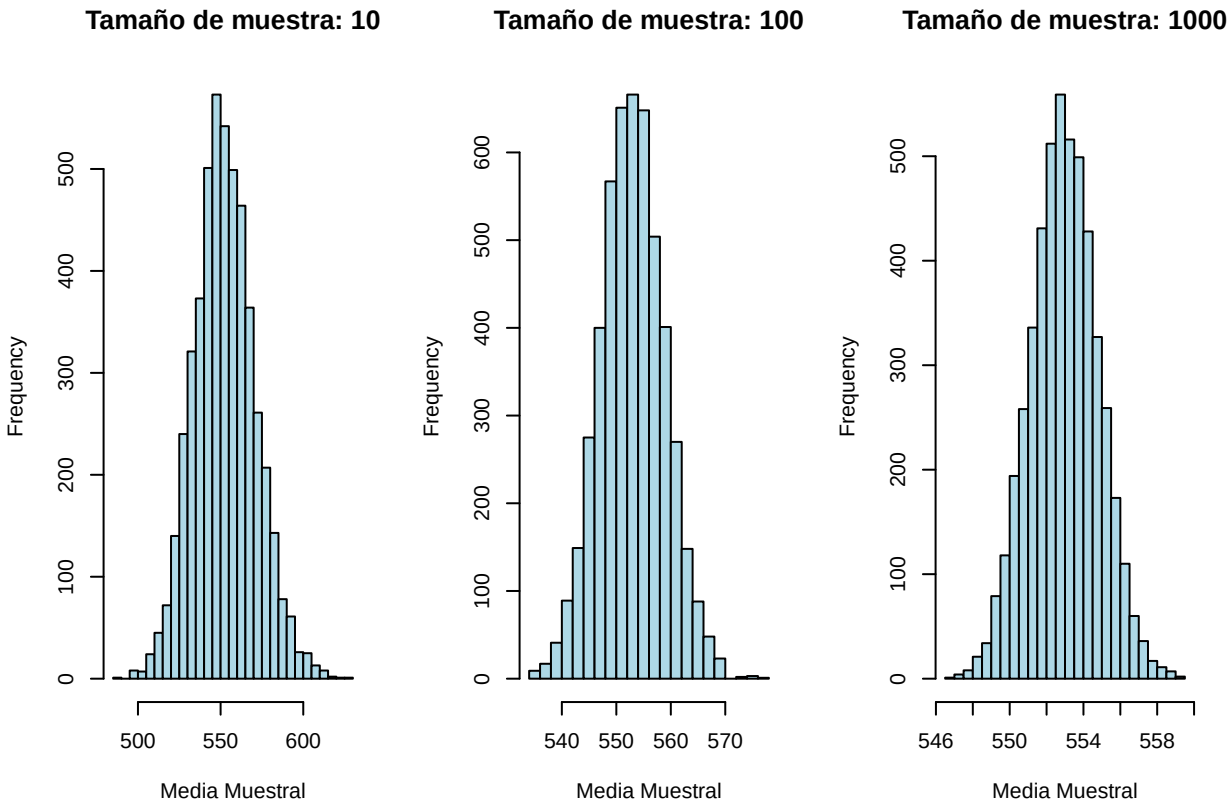
```
## La desviación estándar de las calificaciones de 3º de primaria es: 59.25797
```

Para tamaños de muestra $n = 10, 100, 1000$ aproximaremos la distribución muestral:

i) simula 5,000 muestras aleatorias (con reemplazo)
ii) calcula la media en cada muestra
iii) Realiza un histograma de la distribución muestral de las medias (las medias del paso anterior)

```
simular_medias <- function(n_muestra,n_sim = 5000) {
  medias <- replicate(n_sim, {
    muestra <- sample(enlace$esp_3, size = n_muestra, replace = TRUE)
    mean(muestra, na.rm = TRUE)
  })
  return(medias)
}

# Definir tamaños de muestra
tamaños_muestra <- c(10, 100, 1000)
```

```r
# Calcular las medias muestrales para cada tamaño de muestra
medias_muestrales <- lapply(tamaños_muestra, simular_medias)

# Graficar los histogramas de las medias muestrales
par(mfrow = c(1, 3))
for (i in 1:length(tamaños_muestra)) {
  hist(medias_muestrales[[i]], breaks = 30, main = paste("Tamaño de muestra:", tamaños_muestra[i]),
       xlab = "Media Muestral", col = "lightblue", border = "black")
}
```



iv) aproxima el error estándar calculando la desviación estándar de las medias del paso ii.

- Calcula el error estándar de la media para cada tamaño de muestra usando la fórmula derivada arriba y compara con tus simulaciones.

```r
# Calcular el error estándar empírico para cada tamaño de muestra
error_estandar_empirico <- sapply(medias_muestrales, sd)

# Calcular el error estándar teórico: desviación estándar / sqrt(tamaño de muestra)
error_estandar_teorico <- desviacion_esp_3 / sqrt(tamaños_muestra)

# Mostrar los resultados
for (i in 1:length(tamaños_muestra)) {
  cat("Tamaño de muestra:", tamaños_muestra[i], "\n")
  cat("Error estándar empírico:", error_estandar_empirico[i], "\n")
  cat("Error estándar teórico:", error_estandar_teorico[i], "\n\n")
}
```

```
## Tamaño de muestra: 10
```

```
## Error estándar empírico: 18.46373
## Error estándar teórico: 18.73902
##
## Tamaño de muestra: 100
## Error estándar empírico: 5.877118
## Error estándar teórico: 5.925797
##
## Tamaño de muestra: 1000
## Error estándar empírico: 1.834975
## Error estándar teórico: 1.873902
```

- ¿Cómo se comparan los errores estándar correspondientes a los distintos tamaños de muestra?

A medida que el tamaño de muestra aumenta, tanto el error estándar empírico como el error estándar teórico disminuyen. Esto es consistente con la fórmula teórica del error estándar„ que indica que el error estándar es inversamente proporcional a la raíz cuadrada del tamaño de la muestra. A mayor tamaño de muestra, menor es la variabilidad de la media muestral.