Tarea 06 Bootstrap

Rodrigo Alan Garcia Perez

2024-09-23

```
library(tidyverse)
```

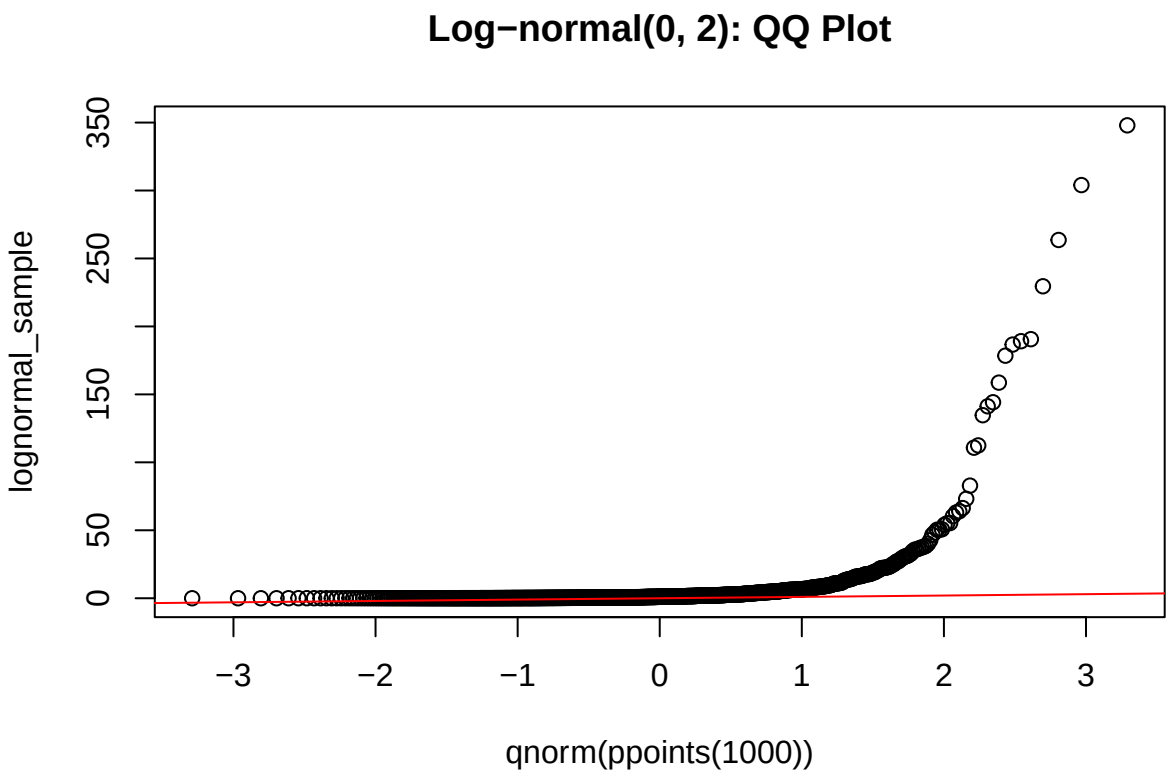
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ 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 error
```

Graficas de cuantiles normales

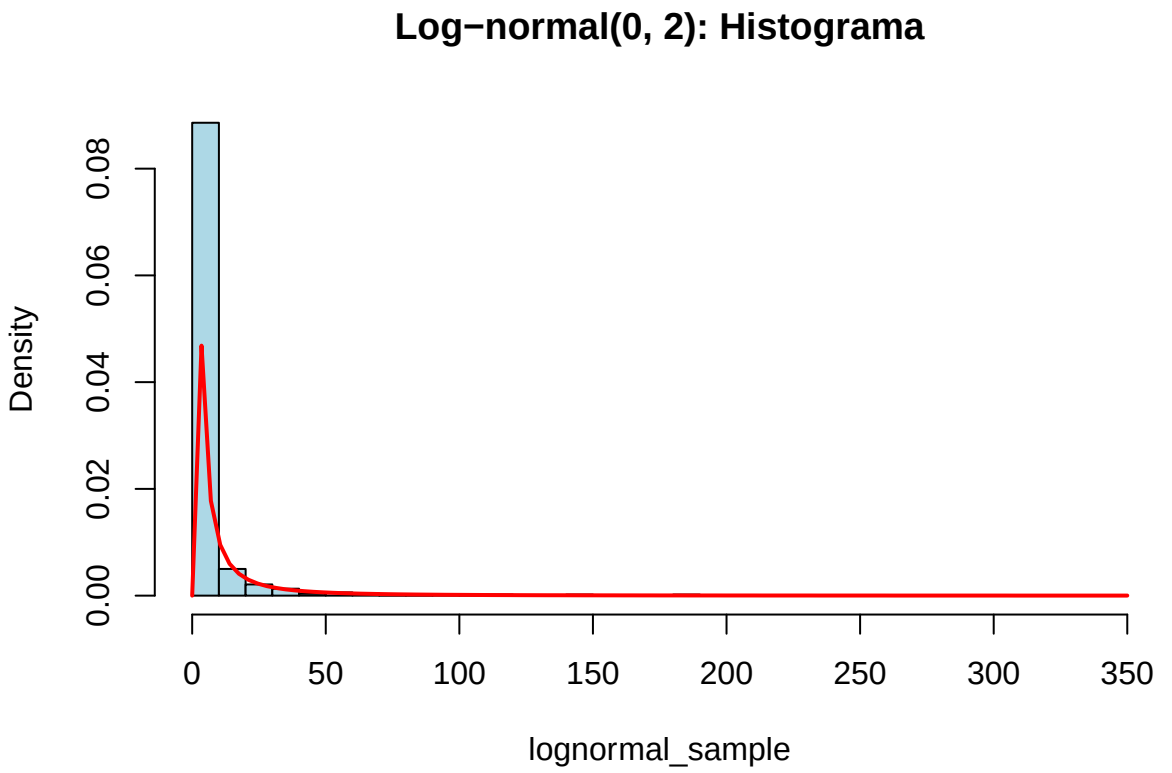
Para las siguientes distribuciones toma una muestra de tamaño 1000 y realiza 1) una gráfica de cuantiles muestrales, 2) un histograma, 3) una gráfica de cuantiles normales:

- Log-normal(0, 2)

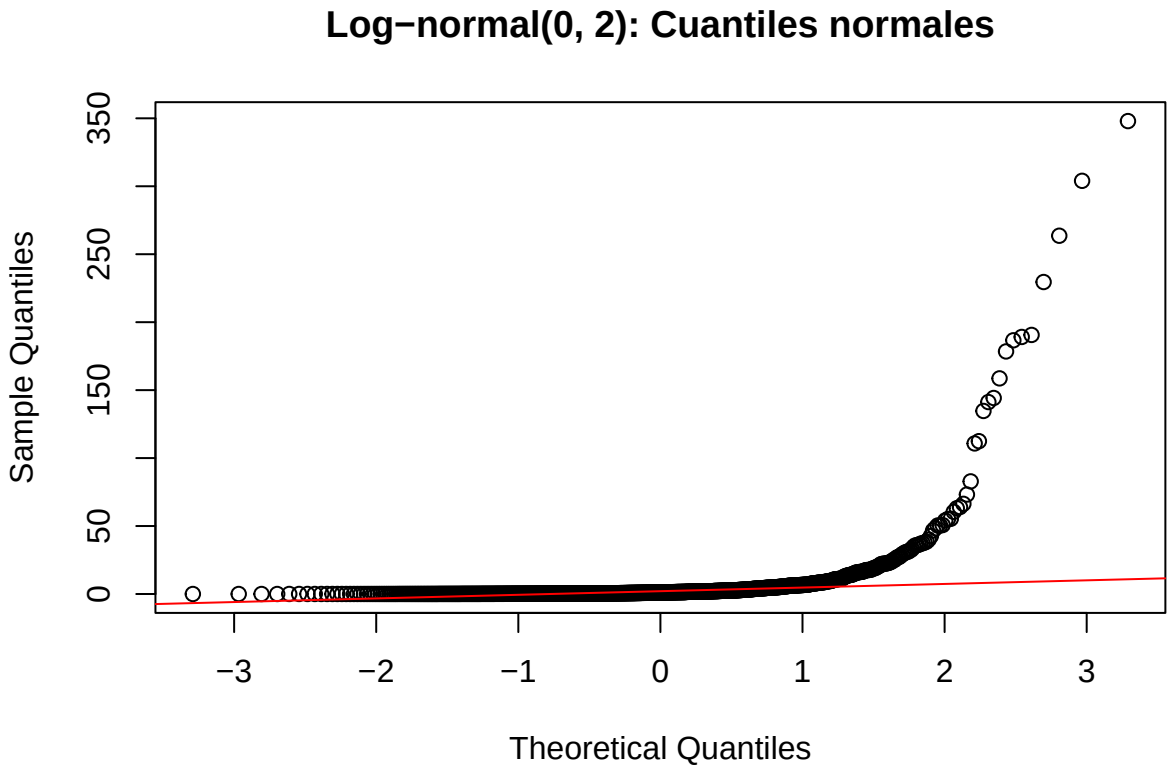
```
set.seed(111)
lognormal_sample <- rlnorm(1000, meanlog = 0, sdlog=2)
qqplot(qnorm(ppoints(1000)), lognormal_sample,  main = "Log-normal(0, 2): QQ Plot")
abline(0, 1, col = "red")
```




```r
# Histograma
hist(lognormal_sample, breaks = 30, main = "Log-normal(0, 2): Histograma", col = "lightblue", freq = FA
curve(dlnorm(x, meanlog = 0, sdlog = 2), add = TRUE, col = "red", lwd = 2)
```

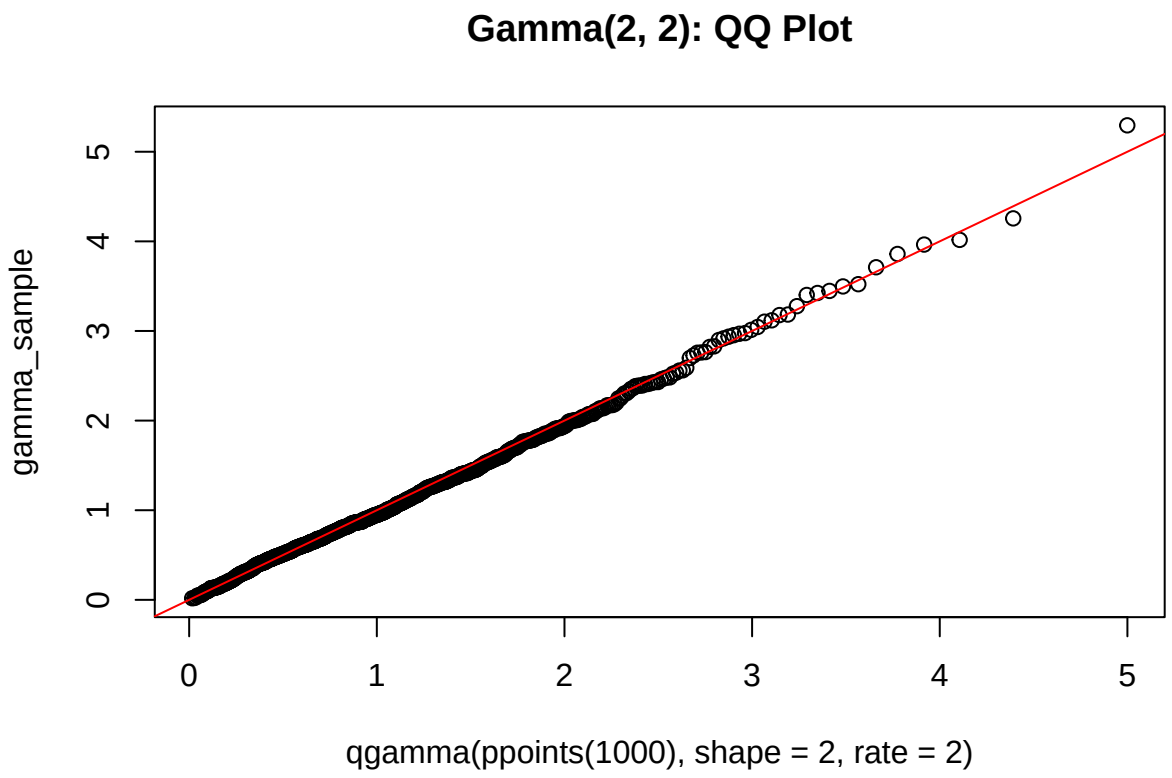


```r
# Gráfica de cuantiles normales
qqnorm(lognormal_sample, main = "Log-normal(0, 2): Cuantiles normales")
qqline(lognormal_sample, col = "red")
```

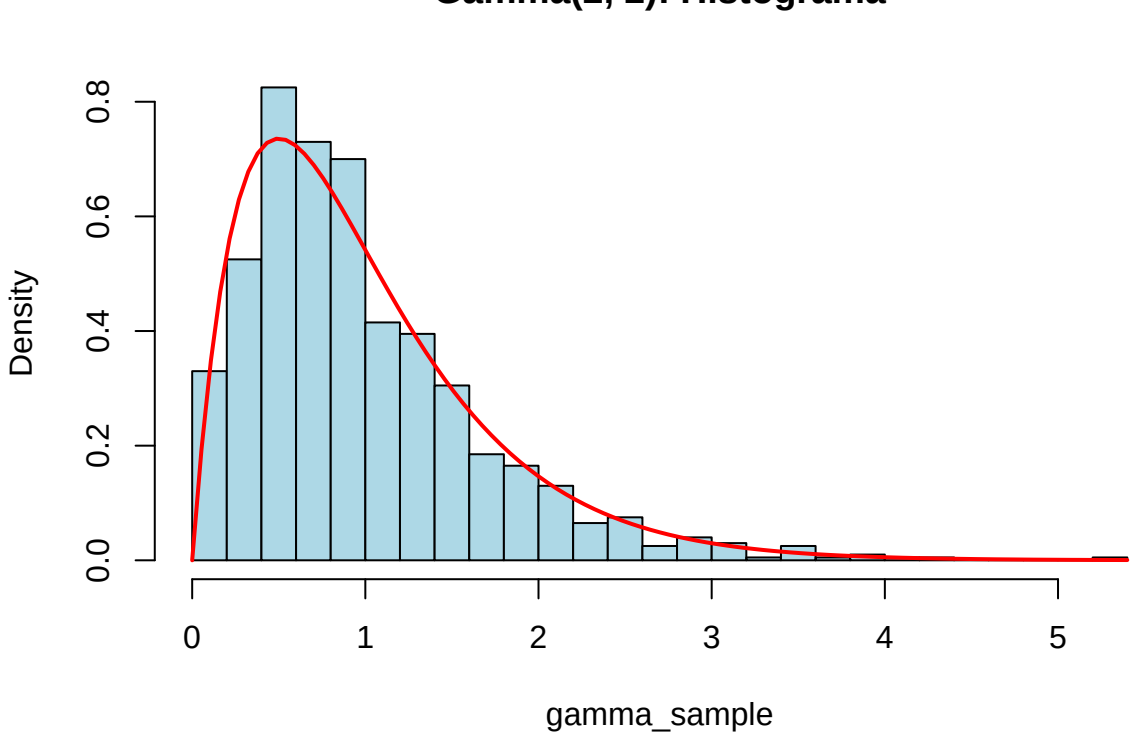


- Gamma(2, 2)

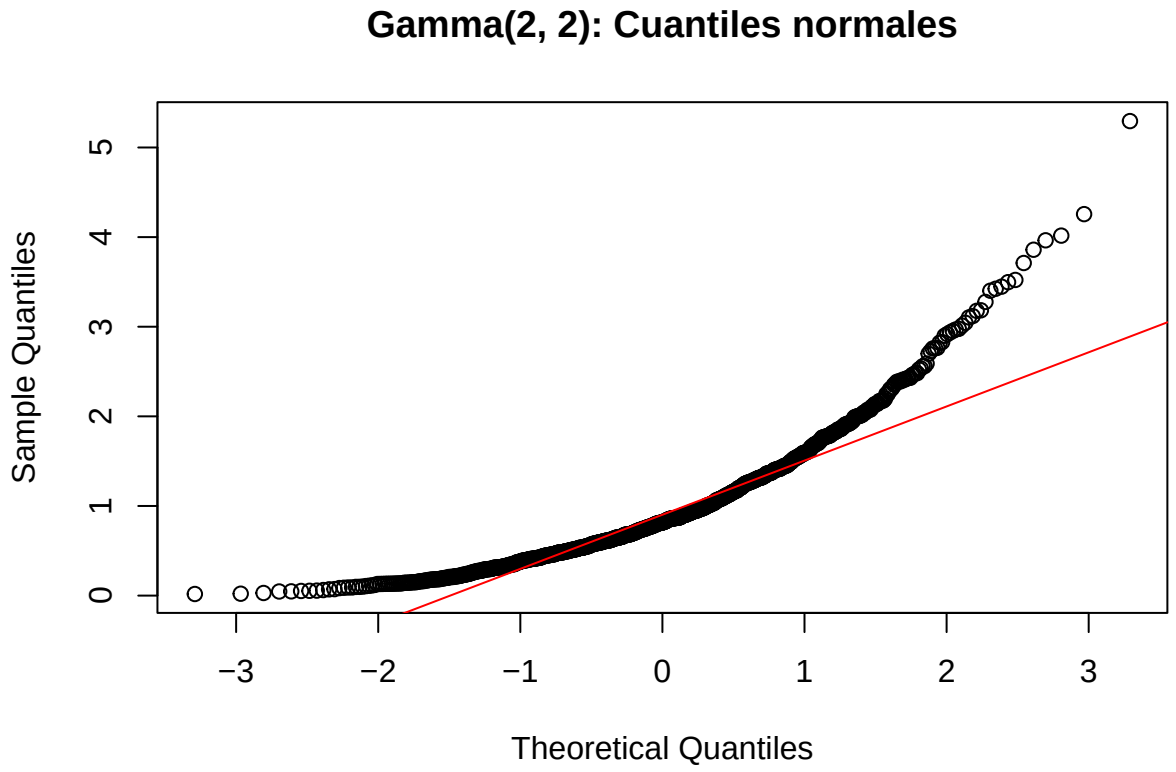
```r
set.seed(111)
gamma_sample <- rgamma(1000, shape = 2, rate = 2)

# Gráfica de cuantiles muestrales
qqplot(qgamma(ppoints(1000), shape = 2, rate = 2), gamma_sample, main = "Gamma(2, 2): QQ Plot")
abline(0, 1, col = "red")
```




```r
# Histograma
hist(gamma_sample, breaks = 30, main = "Gamma(2, 2): Histograma", col = "lightblue", freq = FALSE)
curve(dgamma(x, shape = 2, rate = 2), add = TRUE, col = "red", lwd = 2)
```

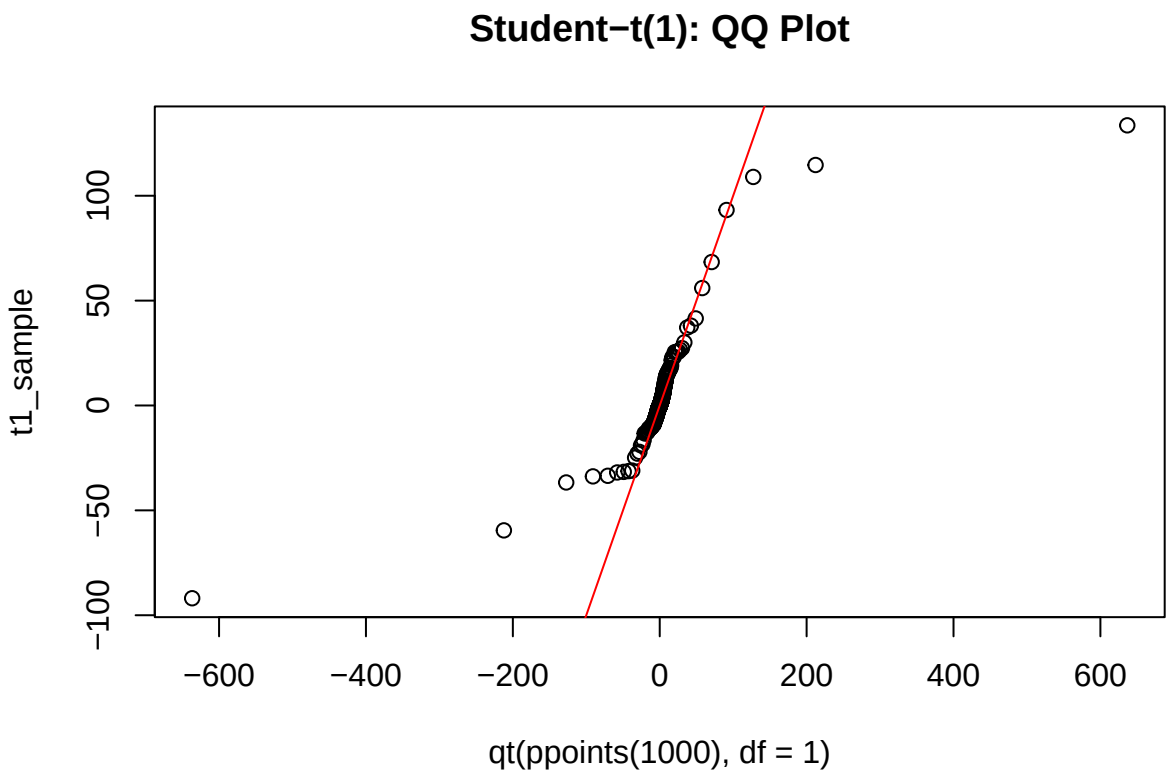


```r
# Gráfica de cuantiles normales
qqnorm(gamma_sample, main = "Gamma(2, 2): Cuantiles normales")
qqline(gamma_sample, col = "red")
```

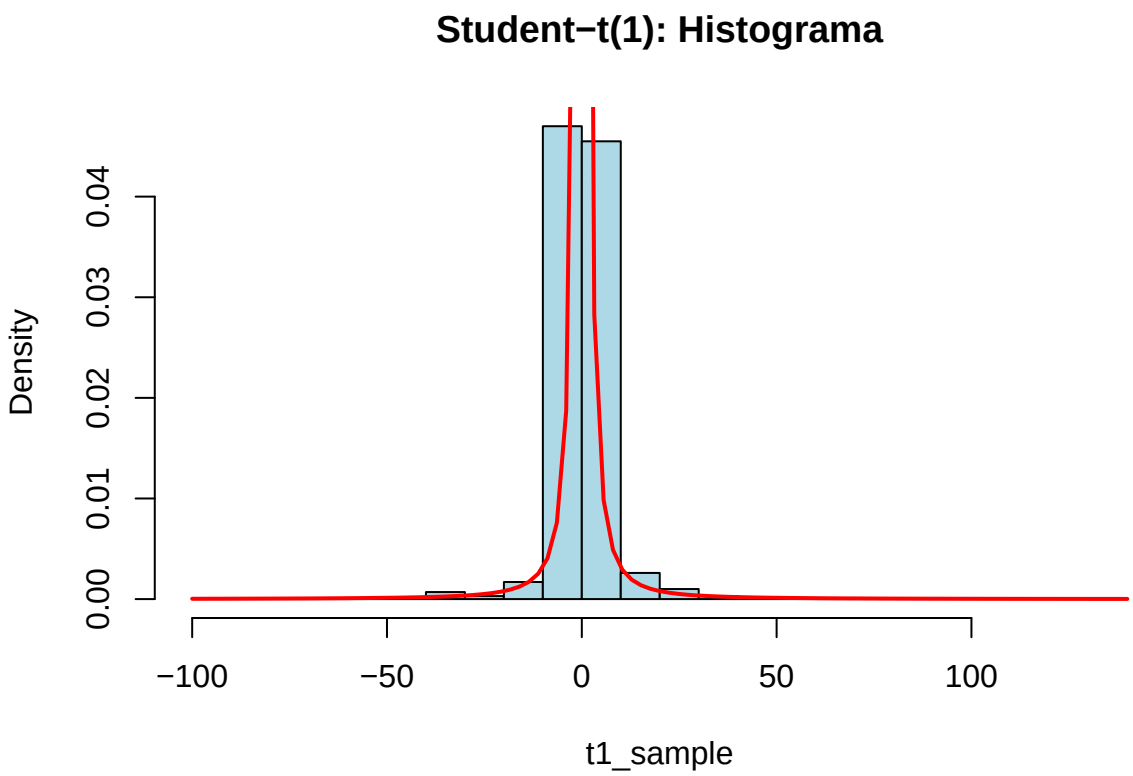


- Student-t con 1 grado de libertad

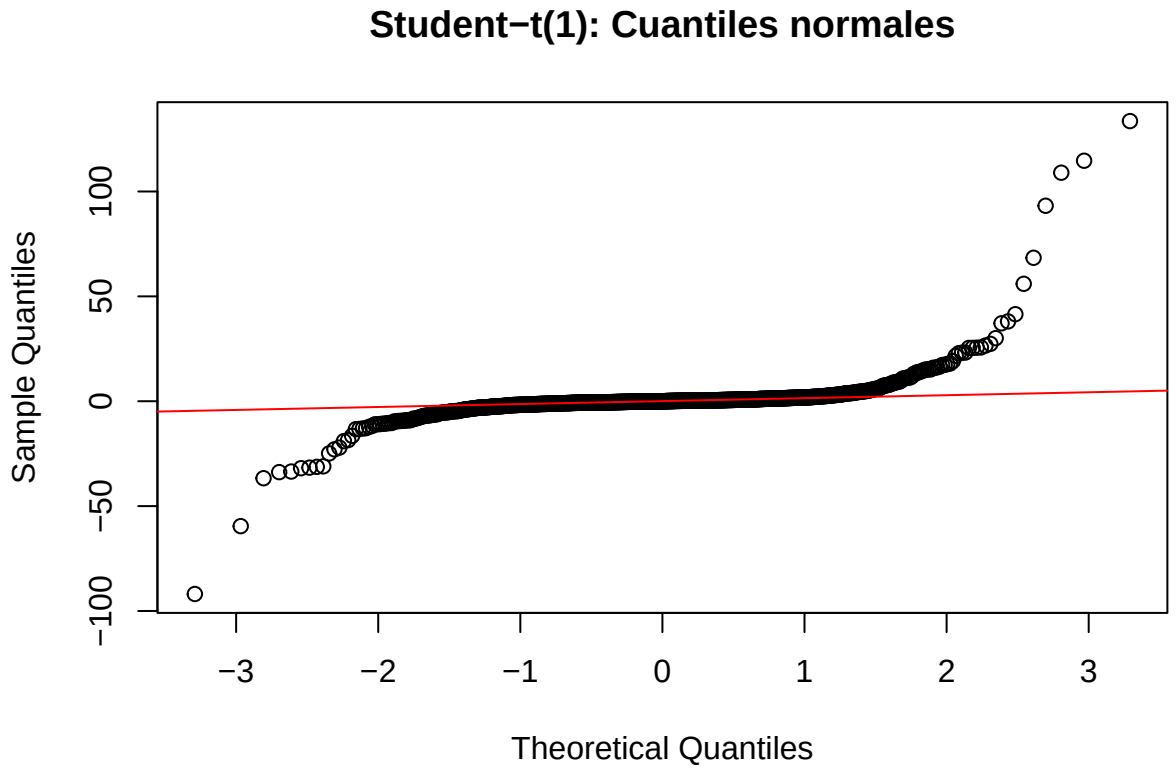
```r
set.seed(111)
t1_sample <- rt(1000, df=1)

# Gráfica de cuantiles muestrales
qqplot(qt(ppoints(1000), df = 1), t1_sample,  main = "Student-t(1): QQ Plot")
abline(0, 1, col = "red")
```




```r
# Histograma
hist(t1_sample, breaks = 30, main = "Student-t(1): Histograma", col = "lightblue", freq = FALSE)
curve(dt(x, df = 1), add = TRUE, col = "red", lwd = 2)
```

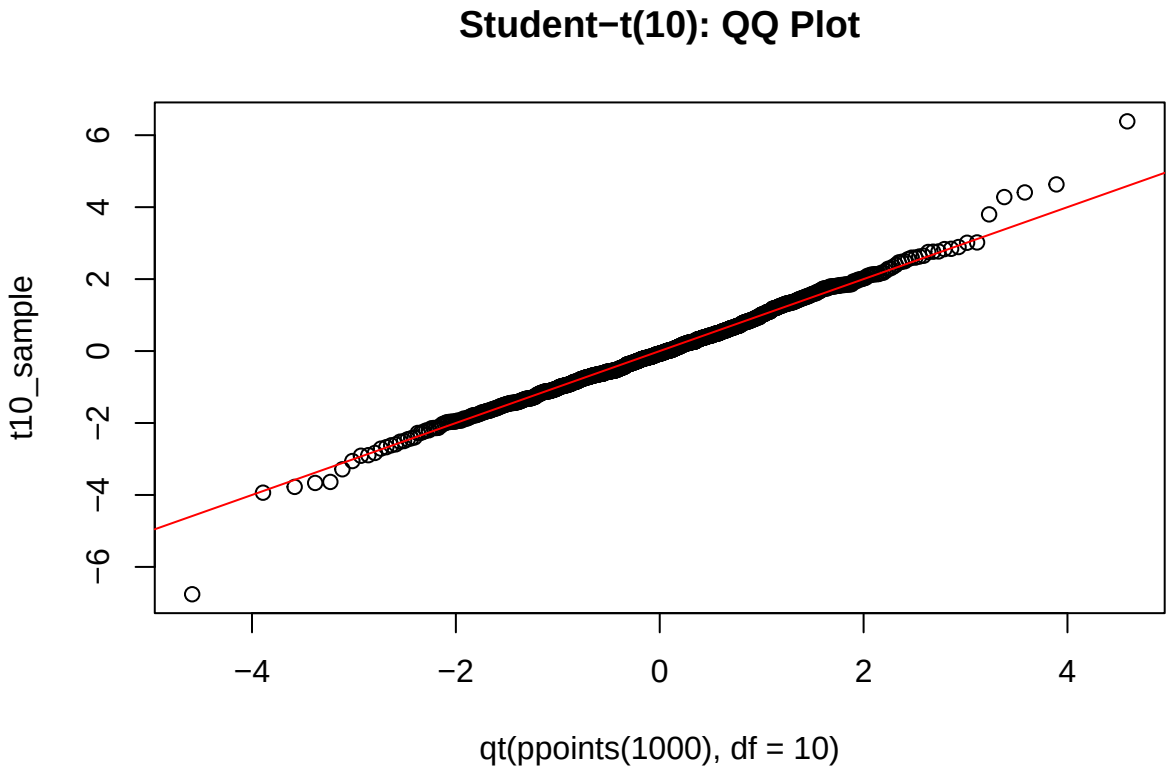


```r
# Gráfica de cuantiles normales
qqnorm(t1_sample, main = "Student-t(1): Cuantiles normales")
qqline(t1_sample, col = "red")
```

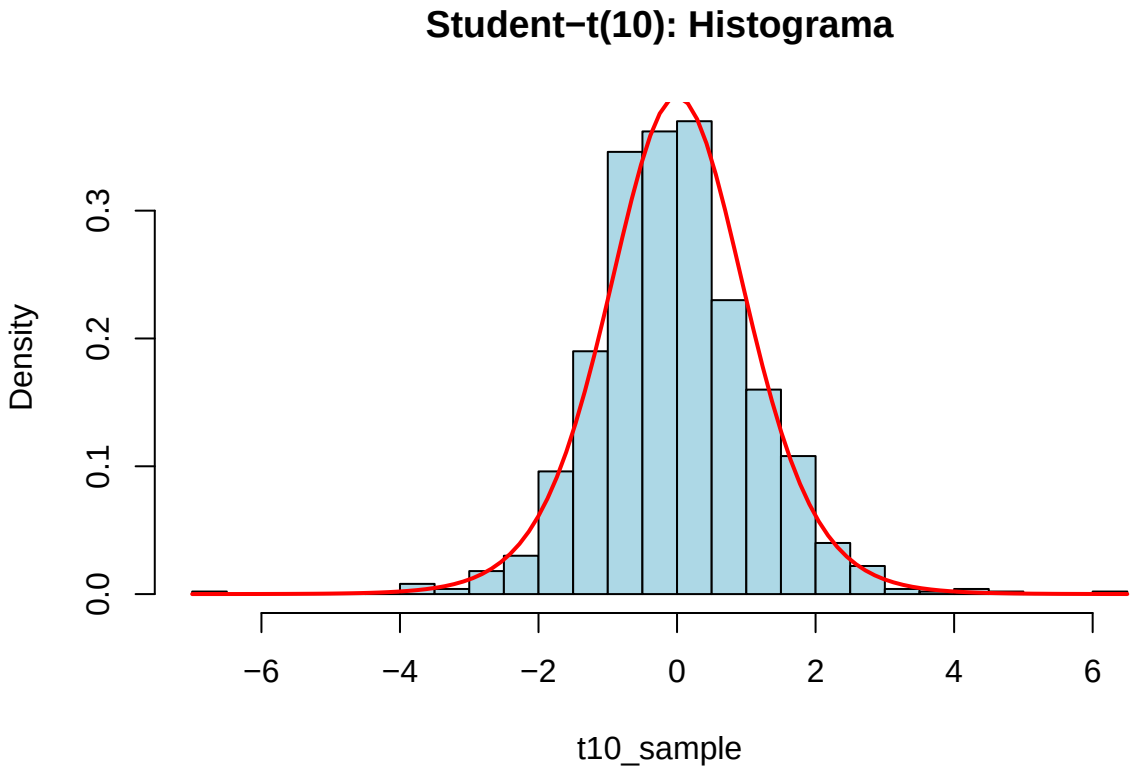


- Student-t con 10 grado de libertad

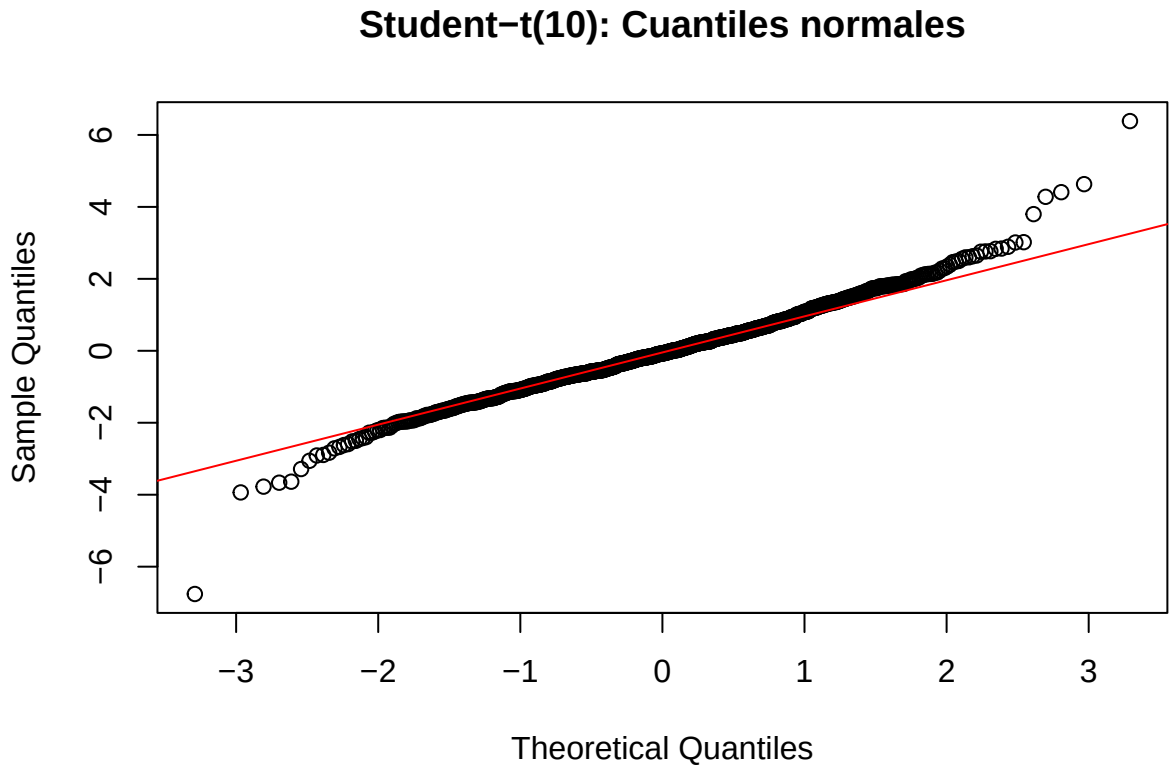
```r
set.seed(111)
t10_sample <- rt(1000, df = 10)

# Gráfica de cuantiles muestrales
qqplot(qt(ppoints(1000), df = 10), t10_sample, main = "Student-t(10): QQ Plot")
abline(0, 1, col = "red")
```




```r
# Histograma
hist(t10_sample, breaks = 30, main = "Student-t(10): Histograma", col = "lightblue", freq = FALSE)
curve(dt(x, df = 10), add = TRUE, col = "red", lwd = 2)
```

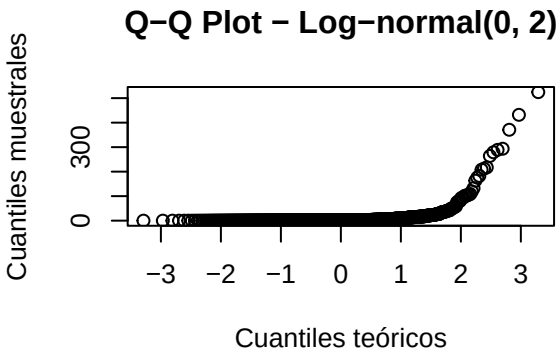


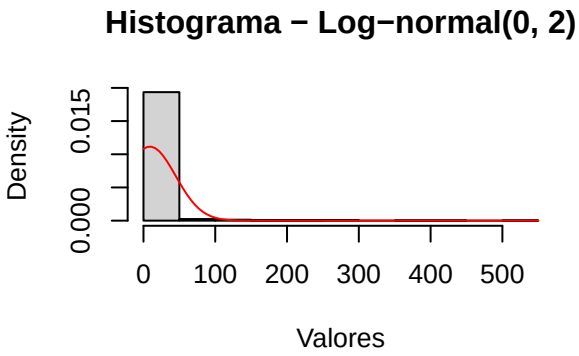
```r
# Gráfica de cuantiles normales
qqnorm(t10_sample, main = "Student-t(10): Cuantiles normales")
qqline(t10_sample, col = "red")
```

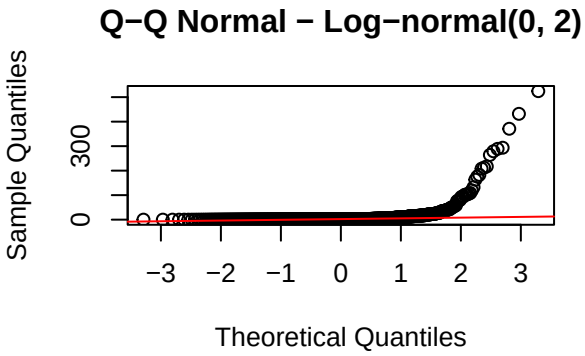


Bootstrap

```r
n <- 1000
muestra_lnorm <- rlnorm(n, meanlog = 0, sdlog = 2)

# Configurar el layout para 2x2 gráficas
par(mfrow = c(2, 2))

# 1. Gráfica de cuantiles muestrales
qqplot(qnorm(ppoints(1000)), muestra_lnorm, main = "Q-Q Plot - Log-normal(0, 2)",
       xlab = "Cuantiles teóricos", ylab = "Cuantiles muestrales")

# 2. Histograma
hist(muestra_lnorm, main = "Histograma - Log-normal(0, 2)", xlab = "Valores", prob = TRUE)
curve(dnorm(x, mean = mean(muestra_lnorm), sd = sd(muestra_lnorm)), add = TRUE, col = "red")

# 3. Gráfica de cuantiles normales
qqnorm(muestra_lnorm, main = "Q-Q Normal - Log-normal(0, 2)")
qqline(muestra_lnorm, col = "red")
```







Muestras independientes en cada grupo

1. Se realiza un experimento en el que se seleccionan 7 ratones de manera aleatoria de un total de 16 ratones. A los siete seleccionados se les suministra un tratamiento mientras que los restantes formarán el grupo de control. El objetivo del tratamiento es prolongar la supervivencia de los ratones. La siguiente tabla muestra el tiempo de supervivencia en días después de suministrar el tratamiento.

Grupo	Datos	Tamaño de muestra
Tratamiento	94, 197, 16, 38, 99, 141, 23	7
Control	52, 104, 146, 10, 51, 30, 40, 27, 46	9

1. Usa las medias de las muestras para determinar si hay diferencias en los grupos, esto es calcula $\bar{x} - \bar{y}$.

```
#Datos
tratamiento <- c(94, 197, 16, 38, 99, 141, 23)
control <- c(52, 104, 146, 10, 51, 30, 40, 27, 46)

media_tratamiento <- mean(tratamiento)
media_control <- mean(control)

diferencia_medias <- media_tratamiento - media_control

# Mostrar los resultados
cat("Media Tratamiento: ", media_tratamiento, "\n")
```

```
## Media Tratamiento:  86.85714
```

```
cat("Media Control: ", media_control, "\n")
```

```
## Media Control:  56.22222
```

```
cat("Diferencia de Medias (Tratamiento - Control): ", diferencia_medias, "\n")
```

```
## Diferencia de Medias (Tratamiento - Control):  30.63492
```

2. Estima el error estándar de la diferencia usando bootsrap.

```
# Función para calcular la diferencia de medias (estadístico de interés)
diff_medias <- function(trat, ctrl) {
  return(mean(trat) - mean(ctrl))
}

# Número de iteraciones bootstrap
# Un número grande (como 10000) asegura una buena estimación de la distribución
n_bootstrap <- 10000

resultados_bootstrap <- numeric(n_bootstrap)

set.seed(123)
for (i in 1:n_bootstrap) {
  # Remuestreo con reemplazo
  # Esto simula la variabilidad de tomar múltiples muestras de la población
  muestra_trat <- sample(tratamiento, replace = TRUE)
  muestra_ctrl <- sample(control, replace = TRUE)

  # Cálculo del estadístico para cada muestra bootstrap
  # Aquí calculamos la diferencia de medias para cada par de muestras
  resultados_bootstrap[i] <- diff_medias(muestra_trat, muestra_ctrl)
}

# La desviación estándar de los estadísticos bootstrap estima el error estándar
error_estandar <- sd(resultados_bootstrap)

# Imprimir el resultado
print(paste("Error estándar estimado:", round(error_estandar, 5)))
```

```
## [1] "Error estándar estimado: 26.89161"
```

3. ¿Dirías que el tratamiento incrementó la supervivencia de los ratones?

Aunque hay una tendencia positiva hacia un incremento en la supervivencia de los ratones tratados (aproximadamente 30 días más en promedio), la gran variabilidad en los datos (reflejada en el alto error estándar) nos impide afirmar con certeza estadística que el tratamiento incrementó significativamente la supervivencia.

4. Supongamos que deseamos comparar los grupos usando las medianas en lugar de las medias, estima la diferencia de las medias y usa bootstrap para estimar el error estándar de la diferencia.

```
diff_medianas <- function(trat, ctrl){
  return(median(trat) - median(ctrl))
}

# Calcular la diferencia de medianas original
diff_original <- diff_medianas(tratamiento, control)

# Número de iteraciones bootstrap
n_bootstrap <- 10000
```

```r
# Vector para almacenar los resultados de cada muestra bootstrap
resultados_bootstrap <- numeric(n_bootstrap)

# Realizar bootstrap
set.seed(123)  # Para reproducibilidad
for (i in 1:n_bootstrap) {
  # Remuestreo con reemplazo
  muestra_trat <- sample(tratamiento, replace = TRUE)
  muestra_ctrl <- sample(control, replace = TRUE)

  # Calcular la diferencia de medianas para esta muestra bootstrap
  resultados_bootstrap[i] <- diff_medianas(muestra_trat, muestra_ctrl)
}

# Calcular el error estándar
error_estandar <- sd(resultados_bootstrap)

# Imprimir los resultados
print(paste("Diferencia de medianas original:", round(diff_original, 2)))
```

```
## [1] "Diferencia de medianas original: 48"
```

```r
print(paste("Error estándar estimado:", round(error_estandar, 5)))
```

```
## [1] "Error estándar estimado: 40.19506"
```

El error estándar para las medianas (40.19506) es mayor que el error estándar para las medias (26.89161), lo que indica una mayor variabilidad en la estimación de la diferencia de medianas.

Datos pareados

2. **Bootstrap correlación.** Nuevamente trabaja con los datos `primaria`, selecciona una muestra aleatoria de tamaño 100 y utiliza el principio del *plug-in* para estimar la correlación entre la calificación de y =español 3 y la de z =español 6: $\hat{corr}(y, z)$. Usa bootstrap para calcular el error estándar de la estimación.

```r
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.
```

```r
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~
```

```r
# Funcion para obtener correlacion
cor_func <- function(data,indices){
  d <- data[indices,]
  return(cor(d$esp_3, d$esp_6))
}

set.seed(123)

# Seleccionar una muestra aleatoria de 100 filas del dataframe 'enlace'
muestra <- enlace[sample(nrow(enlace), 100),]

# Calcular la correlación inicial usando el principio plug-in
# Esta es la estimación puntual de la correlación en nuestra muestra
cor_inicial <- cor(muestra$esp_3, muestra$esp_6)

# Definir el número de repeticiones para el bootstrap
n_bootstrap <- 10000

# Realizar el bootstrap:
# - Repetir 10000 veces
# - En cada repetición, seleccionar 100 índices con reemplazo
# - Aplicar la función cor_func a estos índices
# - Almacenar cada resultado en el vector resultados_boot
resultados_boot <- replicate(n_bootstrap, cor_func(muestra, sample(100, replace = TRUE)))

# Calcular el error estándar como la desviación estándar de los resultados del bootstrap
error_estandar <- sd(resultados_boot)


# Imprimir la correlación estimada inicial
print(paste("Correlación estimada:", round(cor_inicial, 4)))
```

```
## [1] "Correlación estimada: 0.5141"
```

```r
# Imprimir el error estándar estimado
print(paste("Error estándar estimado:", round(error_estandar, 4)))
```

```
## [1] "Error estándar estimado: 0.0875"
```

Existe una correlación positiva moderada entre las calificaciones de español 3 y español 6. Esta relación es estadísticamente significativa y sugiere que el rendimiento en español 3 puede ser un predictor útil, aunque no perfecto, del rendimiento en español 6.