Resampling: Test de permutación, Simulación de Monte Carlo y Bootstrapping

Joaquín Amat Rodrigo *j.amatrodrigo@gmail.com*

Mayo, 2016

Índice

Idea intuitiva de las técnicas de *resampling*

Los métodos de remuestreo o *resampling* se engloban dentro de los test no paramétricos, no requieren de ninguna asunción sobre la distribución de la o las poblaciones estudiadas. Son por lo tanto una alternativa a los test paramétricos cuando no se satisfacen las condiciones del Teorema del Límite Central o cuando se quiere hacer inferencia sobre un parámetro distinto a la media. La información presente en este documento se ha obtenido mayoritariamente de los libros: *Comparing groups Randomization and Bootstrap Methods using R. Andrew S.Zieffler*, *Bootstrap Methods and Permutation Test by Tim Hestemberg* y *The RBook Michael J Crawley*. A continuación se describe la idea general de los diferentes métodos de *resampling*.

Test de permutaciones

Se trata de un test de significancia estadística para la diferencia entre grupos. Fue desarrollado por Ronald Fisher y E. J. G. Pitman en 1930. La distribución del estadístico estudiado (media, mediana...) se obtiene calculando el valor de dicho estadístico para todas las posibles reorganizaciones de las observaciones en los distintos grupos. Dado que implica calcular todas las posibles situaciones se trata de un test exacto. Para ilustrar la idea de los test de permutación, supóngase que un conjunto de sujetos se distribuye en dos grupos, A y B, de tamaños n_A y n_B, cuyas medias muestrales tras el experimento resultan ser $\overline{x}_A$ y $\overline{x}_B$. Se desea determinar si existe una diferencia significativa entre la media de los dos grupos, o lo que es lo mismo, comprobar si hay evidencias en contra de la hipótesis nula de que la diferencia observada es debida únicamente a la asignación al azar de los sujetos a los dos grupos y que ambas muestras proceden de la misma población.

- En primer lugar se calcula la diferencia entre las medias de los dos grupos, a lo que se conoce como diferencia observada ($Dif_{observada}$).
- Todas las observaciones se combinan juntas sin tener en cuenta el grupo al que pertenecían.
- Se calculan todas las posibles permutaciones en las que las observaciones pueden ser agrupadas en dos grupos de tamaño n_A y n_B.
- Para cada permutación se calcula la diferencia entre medias ($Dif_{calculada}$). El conjunto de valores calculados forman la distribución exacta de las posibles diferencias siendo cierta la

hipótesis nula. A esta distribución se le conoce como *permutation distribution of the mean difference*.

- El *p-value* de dos colas se calcula como la proporción de permutaciones muestrales en las que el valor absoluto de la diferencia calculada es mayor o igual al valor absoluto de la diferencia observada.

En el ejemplo ilustrativo se emplea como estadístico la media pero podría ser cualquier otro.

La condición necesaria para un test de permutaciones se conoce como *exchangeability*, según la cual todas las posibles permutaciones tienen la misma probabilidad de ocurrir siendo cierta la hipótesis nula. Las conclusiones de un test de permutación solo son aplicables a diseños de tipo experimental, es decir, en los que tras haber elegido los sujetos del estudio, se realiza una asignación aleatoria de los sujetos a los diferentes grupos.

Los test de permutación son test de significancia y por lo tanto se emplean para calcular *p-values*, no para intervalos de confianza.

Simulación de Monte Carlo o randomization test

En los test de permutación el *p-value* obtenido es exacto ya que se calculan todas las posibles permutaciones de las observaciones. Esto resulta complicado o imposible cuando el tamaño muestral es mediano o grande. La simulación de Monte Carlo consiste en realizar el test empleando únicamente una muestra aleatoria, de ahí que se llame *randomization*, de todas las posibles permutaciones, evitando así tener que calcularlas todas. Al no calcularse todas las posibles permutaciones, no es un test exacto sino aproximado.

El método de Monte Carlo no es insesgado, por lo que se emplea una corrección (Davison and Hinkley) tal que:

$$p_{value} = \frac{r + 1}{k + 1}$$

Siendo r el número de permutaciones igual o más extremas que el estadístico observado y k el número de permutaciones empleadas en la simulación.

A pesar de que la corrección es mínima cuando el número de simulaciones es alto, presenta la ventaja de que si el valor observado es mayor que cualquiera de los calculados, el p-value es muy bajo pero no cero. Por facilidades de cálculo se suele emplear un número de permutaciones terminado en 9 (4999, 9999).

Bootstrapping

El escenario ideal para realizar inferencia estadística sobre una población es disponer de infinitas (o una gran cantidad) de muestras de dicha población, sin embargo, en la práctica raramente es posible. Si solo se dispone de una muestra y se considera que es representativa de la población, los valores de la variable aleatoria en la muestra aparecen aproximadamente con la misma proporción en que lo hacen en la población. El método de *Bootstrapping* se basa en generar nuevas pseudomuestras del mismo tamaño que la muestra real, realizando *sampling with replacement* de las observaciones. Si la muestra original es representativa de la población, la distribución del estadístico calculada a partir de las pseudomuestras (*bootstraping distribution*) se asemejará a la distribución muestral que se obtendría si se pudiera acceder a la población para generar nuevas muestras. Fue desarrollado por Bradley Efron en 1979.

Dado que se podrían generar infinitas nuevas muestras mediante el *sampling with replacement*, el método de bootstrapping emplea únicamente una cantidad determinada por el usuario. Hace uso por lo tanto de la simulación de Monte Carlo.

El Bootstrapping no asume una asignación aleatoria de los grupos, sino que las muestras han sido obtenidas aleatoriamente de la o las poblaciones. Se aplica por lo tanto en diseños muestrales, no experimentales. Esta es la diferencia clave respecto a los test de permutación/*randomization*.

El método bootstrapping se puede emplear para:

Calcular intervalos de confianza para un parámetro poblacional: se emplea el *sampling with replacement* a partir de la muestra original.

Calcular significancia estadística (*p-value*) para la diferencia entre dos poblaciones: Si bien este uso se asemeja al de los test de permutación/*randomization*, no es igual. Los test de permutación se emplean para contrastar la hipótesis nula de que las muestras pertenecen a una misma población (distribución) mediante el estudio de las diferencias debidas a la asignación aleatoria de los grupos. El método de bootstrapping contrasta también la hipótesis de que ambas muestras proceden de la misma población (distribución) pero lo hace mediante el estudio de las diferencias debidas al muestreo aleatorio. Se aplica por lo tanto a estudios en los que no ha habido una asignación aleatoria a los grupos previa realización de los experimentos. Los pasos a seguir son: se mezclan las observaciones de ambas muestras, se emplea el *sampling with replacement* sobre este *pool* para generar una nueva pseudomuestra del mismo tamaño, se separan las observaciones de la pseudomuestra en dos grupos de igual tamaño a los originales y se calcula la diferencia del estadístico entre ambas. El proceso se repite múltiples veces generando así la distribución de las diferencias esperadas debido al muestreo aleatorio. El *p-value* de dos colas se calcula como la proporción de pseudomuestras en las que el valor

absoluto de la diferencia calculada es mayor o igual al valor absoluto de la diferencia observada.

Calcular intervalos de confianza para la diferencia entre dos poblaciones: Para esta finalidad se considera como hipótesis nula que las observaciones proceden de dos poblaciones distintas. Se emplea el *sampling with replacement* con la observaciones de cada muestra (sin mezclarlas) para generar dos nuevas pseudomuestras independientes y se calcula la diferencia del estadístico. Este proceso se repite múltiples veces generando la distribución que se obtendría si se extrajesen cada vez dos muestras, cada una de su respectiva población, y se calculara la diferencia. La distribución resultante estará centrada en la verdadera diferencia entre las poblaciones.

El método de bootstrapping se subdivide en paramétrico o no paramétrico dependiendo si se considera que la distribución generada sigue un determinado modelo teórico.

Comparación de los test de permutación y métodos bootstrapping

Tanto los test de *permutation* como los test de *bootstrap* se pueden emplear para estudiar diferencias entre grupos. Existe una lista muy extensa de referencias en las que se debate cuál de los dos métodos es el más adecuado. En general todas ellas concluyen en que el método más adecuado depende del objetivo de la inferencia, y a su vez, el objetivo de la inferencia determina que diseño del estudio a seguir.

La siguiente tabla contiene los diferentes tipos de diseños que se pueden emplear para comparar dos grupos y el tipo de inferencia (conclusiones) que se puede realizar en cada uno:

Muestreo aleatorio	Asignación de grupos aleatoria	Objetivo de la inferencia	Permite determinar causalidad
Sí	No	Población	No
No	Sí	Muestra	Sí
Sí	Sí	Población y Muestra	Sí

La principal diferencia entre ambos métodos aparece cuando se emplean para calcular *p-values*.

Los test de significancia (cálculo de *p-value*) se basan en la hipótesis nula de que todas las observaciones proceden de la misma población. El objetivo del test es determinar si la diferencia observada entre los grupos se debe a un determinado factor (tratamiento) o solo a la variabilidad esperada por la naturaleza de un proceso aleatorio. Cuando la aleatoriedad se debe a la asignación de los sujetos (supuestamente iguales) a los distintos grupos se emplean los test de permutación o *randomization*. La estructura de un experimento que puede analizarse mediante test de permutación es: selección de sujetos del estudio, asignación aleatoria a diferentes grupos, aplicación de los "tratamientos", comparación de resultados. Los test de permutación/*randomization* responden a la pregunta ¿Cuánta variabilidad se espera en un determinado test estadístico debido únicamente a la aleatoriedad de las asignaciones si todos los sujetos proceden realmente de una misma población? Si se trata de comparar la media entre dos grupos la pregunta anterior equivale a ¿Qué diferencia entre medias cabe esperar dependiendo de cómo se distribuyan los sujetos en los dos grupos si todos proceden de una misma población? Aun siendo todos de una misma población, dado que no serán exactamente idénticos, habrá pequeñas diferencias dependiendo de cómo se agrupen.

El *bootstraping* como test de significancia se emplea cuando la aleatoriedad es debida al proceso de obtención de las muestras y no a la asignación en grupos. Responden a la pregunta ¿Cuánta variabilidad se espera en un determinado estadístico debido únicamente al muestreo aleatorio si todos los sujetos proceden realmente de una misma población? Debido a las pequeñas diferencias entre los individuos de una población, si se extraen dos muestras aleatorias de ella y se comparan, no van a ser exactamente iguales, además esta diferencia será distinta para cada par de muestras aleatorias extraídas. La estructura de un experimento que puede analizarse mediante *bootstrapping* es: Se obtienen dos muestras aleatorias de dos poblaciones y se comparan.

Por lo tanto, aunque ambos test pueden emplearse para calcular *p-values*, sus aplicaciones no se solapan. Los test de permutación/*randomization* se emplean para diseños experimentales y el *bootstraping* para diseños muestrales.

El método de *bootstrapping* puede emplearse además, para generar intervalos de confianza de la verdadera diferencia de un parámetro entre dos poblaciones, mientras que los test de permutación/*randomization* no. El porqué reside en que la simulación bootstrapping, cuando se emplea por separado para cada muestra, genera una aproximación de la verdadera distribución del estadístico que se está estudiando, lo que permite generar intervalos de confianza que acoten su valor con una determinada seguridad. Cuando se quiere calcular un *p-value* se tiene que comparar el valor observado del estadístico con la distribución que cabría esperar de él bajo la hipótesis nula, no respecto a su verdadera distribución. Los test de permutación/*randomization* o los test de bootstrapping (aplicado al conjunto de observaciones mezcladas), generan la distribución esperada si se cumple la H_0.

Es importante tener en cuenta que ninguno de estos métodos está al margen de los problemas que implica tener muestras pequeñas.

Test de permutación exacto

Este tipo de test solo es posible aplicarlo cuando el tamaño de las muestras es limitado, ya que implica generar todas las posibles permutaciones de los datos y calcular para cada una de ellas el estadístico de interés. *No he encontrado en R ningún paquete que lo realice*

Test de permutación con simulación de Monte Carlo

Ejemplo1

Supóngase un estudio que pretende determinar si la participación en actividades extraescolares aumenta la capacidad empática de los estudiantes. Para ello el colegio ofrece un programa voluntario en el que cada participante se designa de forma aleatoria a un grupo "control"" que no recibe clases extraescolares o a un grupo "tratamiento" que sí las recibe. A final del año todos los sujetos del estudio realizan un examen que determina su capacidad empática. En vista de los resultados ¿Se puede considerar que las clases extraescolares tienen un impacto en el promedio de cómo se relacionan socialmente los estudiantes? Ejemplo libro *Comparing Groups Randomization and Bootstrap Methods Using R.*

```
datos <- read.table("http://www.tc.umn.edu/~zief0002/Comparing-
Groups/Data/AfterSchool.csv", header = TRUE, sep = ",", row.names = 1)
datos <- datos[, c("Treatment", "Delinq")]
colnames(datos) <- c("grupo", "puntuacion")
datos$grupo <- as.factor(datos$grupo)
levels(datos$grupo) <- c("control", "tratamiento")
```

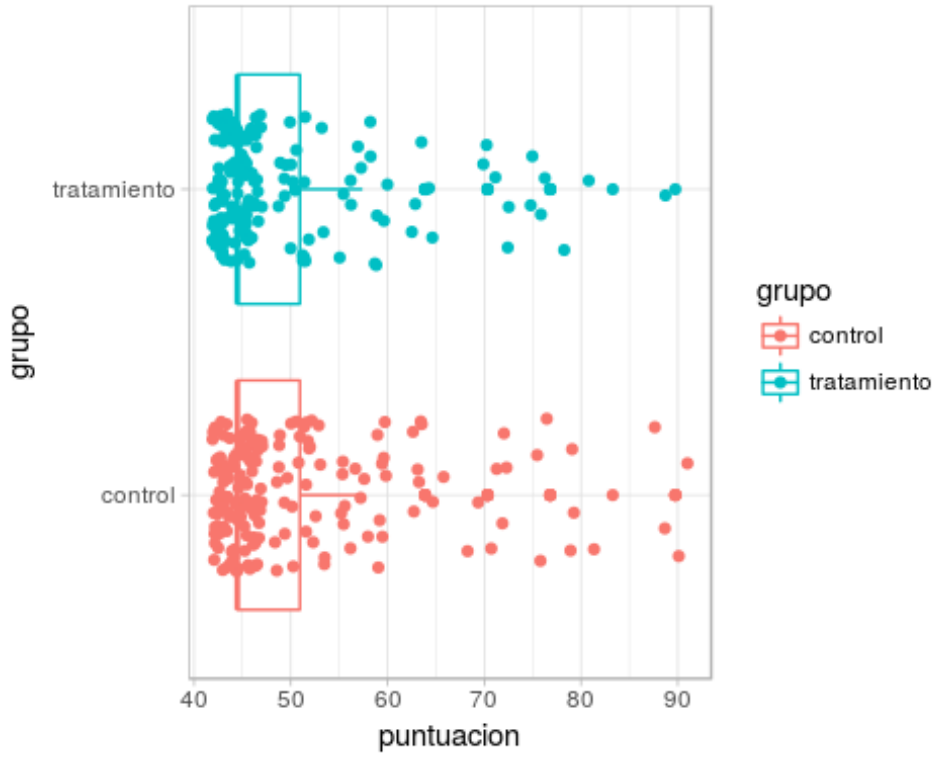
El diseño experimental del estudio emplea una asignación aleatoria de los sujetos a los dos grupos (*tratamiento* y *control*) para posteriormente llevar a cabo el experimento. Esta aleatorización en la asignación implica que, en promedio, los dos grupos son iguales para todas las características, de tal forma que la única diferencia entre ellos será si reciben o no el tratamiento. Determinar si la diferencia observada es significativa equivale a preguntarse cómo de probable es obtener esta diferencia si el tratamiento no tiene efecto y los estudiantes se han

asignado de forma aleatoria en cada grupo. Lo que es lo mismo, determinar si la diferencia observada es mayor de lo que cabría esperar debido únicamente a la variabilidad producida por la formación aleatoria de los grupos.

El diseño experimental de asignación aleatoria a grupos permite obtener conclusiones de tipo causa efecto. Sin embargo, dado que la selección de los sujetos no ha sido aleatoria sino por voluntarios, no son extrapolables a toda la población.

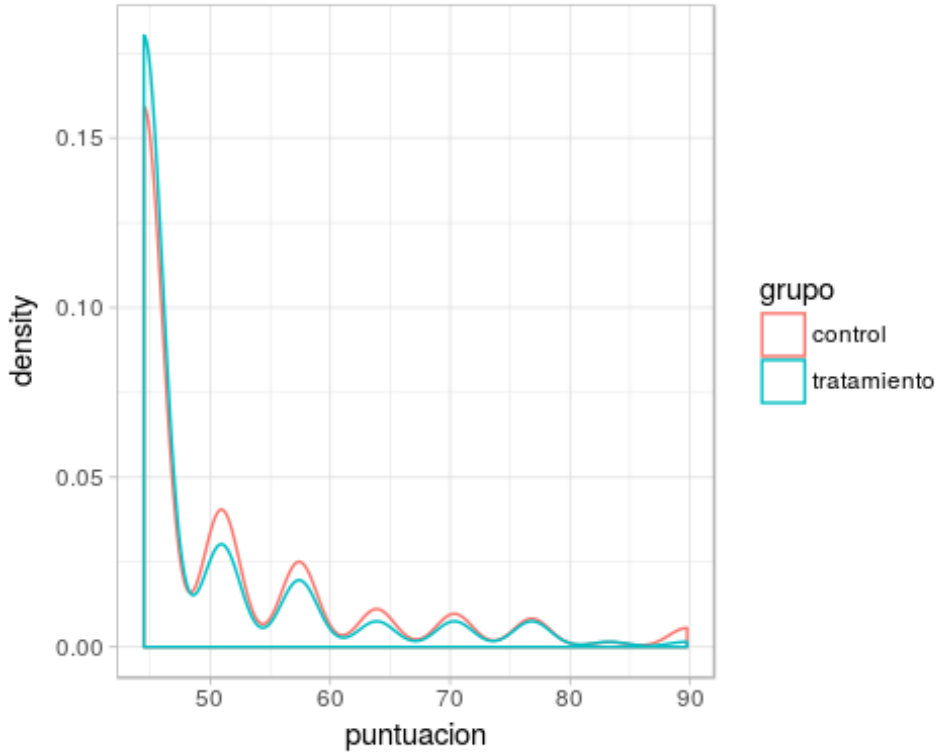
El primer estudio exploratorio (diagrama de cajas y diagrama de densidad) no muestran que exista una diferencia marcada en la distribución de los dos grupos.

```
require(ggplot2)
ggplot(data = datos, aes(x = grupo, y = puntuacion, colour = grupo)) +
  geom_boxplot() +
  geom_jitter(width = 0.25) +
  coord_flip() +
  theme_light()
```




```
ggplot(data = datos, aes(x = puntuacion, colour = grupo)) +
  geom_density() +
```

```
  theme_light()
```



Las medias de ambos grupos y su variabilidad son similares.

```
tapply(X = datos$puntuacion, INDEX = datos$grupo, FUN = mean)
```

```
##     control tratamiento
##    50.72559    49.01896
```

```
tapply(X = datos$puntuacion, INDEX = datos$grupo, FUN = sd)
```

```
##     control tratamiento
##    10.52089     8.97423
```

Dado que las observaciones no se distribuyen de forma normal, los test paramétricos de significancia no son adecuados. *(Dado que el tamaño muestral es grande, el resultado de los test paramétricos sería relativamente bueno.)*

Existen diferentes tipos de test no paramétricos que se pueden aplicar cuando no se cumplen las condiciones del teorema del límite central, por ejemplo el *U test de Mann-Witney* que compara medianas. Otra opción no paramétrica son los métodos de *resampling* que permiten trabajar con cualquier estadístico (media, mediana, varianza...). Un hecho clave a

tener en cuenta a la hora de elegir entre *test de permutación* o *bootstrapping* es que el muestreo no ha sido aleatorio, ya que se ha basado en voluntarios. Una vez obtenidos los sujetos del estudio, sí se han asignado aleatoriamente a los diferentes grupos.

En primer lugar se calcula la diferencia entre las medias de ambos grupos (diferencia observada).

```
dif_obs <- mean(datos[datos$grupo == "control", "puntuacion"]) -
           mean(datos[datos$grupo == "tratamiento", "puntuacion"])
dif_obs
```

```
## [1] 1.706636
```

Determinar si la diferencia observada es significativa equivale a preguntarse cómo de probable es obtener esta diferencia si el tratamiento no tiene efecto y los estudiantes se han asignado de forma aleatoria en cada grupo. Para poder obtener la probabilidad exacta, se requiere calcular todas las posibles permutaciones en las que 356 sujetos pueden repartirse en dos grupos y calcular la diferencia de medias para cada una. El número de permutaciones posibles es muy elevado, $(3.93x10^{105})$, por lo que se recurre a una simulación de Monte Carlo.

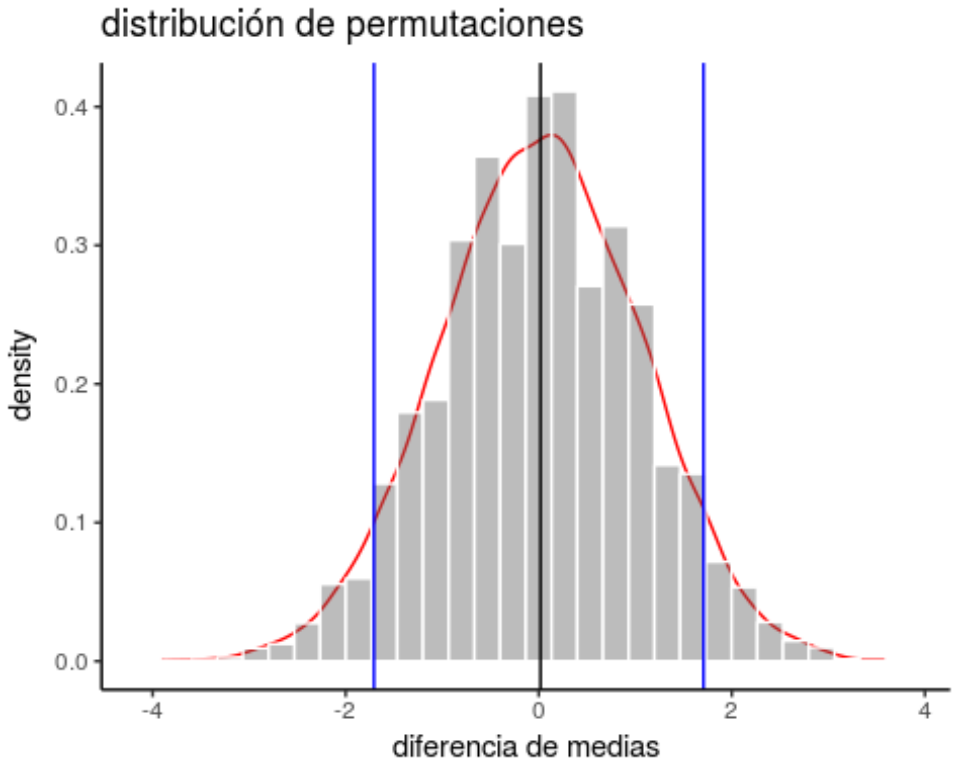
En cada iteración: se asignan aleatoriamente los sujetos a cada uno de los grupos manteniendo el tamaño original de cada uno, se calcula la diferencia de medias y se almacena el valor. Existen múltiples formas de realizar las permutaciones, lo importante es que mimeticen una asignación aleatória de los grupos manteniendo el tamaño original

```
distribucion_permutaciones <- rep(NA, 9999)
n_control <- length(datos$grupo[datos$grupo == "control"])
n_tratamiento <- length(datos$grupo[datos$grupo == "tratamiento"])

for (i in 1:9999) {
    datos_aleatorizados <- sample(datos$puntuacion)  #mezclado aleatorio de las
observaciones
    distribucion_permutaciones[i] <- mean(datos_aleatorizados[1:n_control]) -
                              mean(datos_aleatorizados[n_control + 1:n_tratamiento])
}
```

Los datos simulados forman lo que se conoce como *distribución de randomization o de Monte Carlo* y representa la variación esperada en la diferencia de medias debida únicamente a la asignación aleatoria de grupos.

```
require(ggplot2)
qplot(distribucion_permutaciones, geom = "blank") +
 geom_line(aes(y = ..density..), stat = "density", colour = "red") +
 geom_histogram(aes(y = ..density..), alpha = 0.4, colour = "white") +
 geom_vline(xintercept = mean(distribucion_permutaciones)) +
 geom_vline(xintercept = dif_obs, colour = "blue") +
 geom_vline(xintercept = -dif_obs, colour = "blue") +
 labs(title = "distribución de permutaciones", x = "diferencia de medias") +
 theme_classic()
```




```
summary(distribucion_permutaciones)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -3.90500 -0.69850  0.03030  0.01786  0.75910  3.82000
```

```
sd(distribucion_permutaciones)
```

```
## [1] 1.043322
```

Como era de esperar, la diferencia media entre grupos si el tratamiento no es efectivo es muy próxima a cero (línea vertical negra). La desviación típica (1.0433219) de la distribución de permutaciones indica que la variabilidad debida a la asignación aleatoria de los sujetos a los diferentes grupos es muy pequeña.

Finalmente, se calcula la probabilidad (*p-value*) de obtener diferencias igual o más extremas que la observada (líneas verticales azules) con y sin corrección de continuidad:

```
p_value = (sum(abs(distribucion_permutaciones) > abs(dif_obs)))/9999
p_value
```

```
## [1] 0.1058106
```

```
p_value_corregido = ((sum(abs(distribucion_permutaciones) > abs(dif_obs))) +
    1)/(9999 + 1)
p_value_corregido
```

```
## [1] 0.1059
```

Conclusión

Los 356 sujetos del estudio fueron asignados de forma aleatoria a un grupo control (n=187) o a un grupo tratamiento (n=169) que asistió a clases extra escolares. Un test de permutación se empleó para determinar si existía una diferencia significativa en la capacidad empática promedio entre ambos grupos. El *p-value* fue calculado mediante una simulación de Monte Carlo con 9999 permutaciones usando la corrección de continuidad sugerida por Davison and Hikley(1997). El *p-value* obtenido muestra una evidencia muy débil en contra de la hipótesis nula de que el tratamiento no tiene efecto, sugiriendo que asistir a clases extra escolares no mejora la capacidad empática para los estudiantes que formaron parte del experimento. *Siendo estadísticamente estrictos, no se puede extrapolar a la población de estudiantes ya que la selección de sujetos no fue aleatoria*

A modo de comprobación, siendo los tamaños muestrales de más de 30 observaciones, el t-test debería dar un resultado similar.

```
t.test(puntuacion ~ grupo, data = datos, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  puntuacion by grupo
## t = 1.6379, df = 354, p-value = 0.1023
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.3425852  3.7558562
## sample estimates:
##     mean in group control mean in group tratamiento
##                   50.72559                   49.01896
```

Los *p-values* son similares: Bootstrapping = 0.1059, t-test = 0.1023

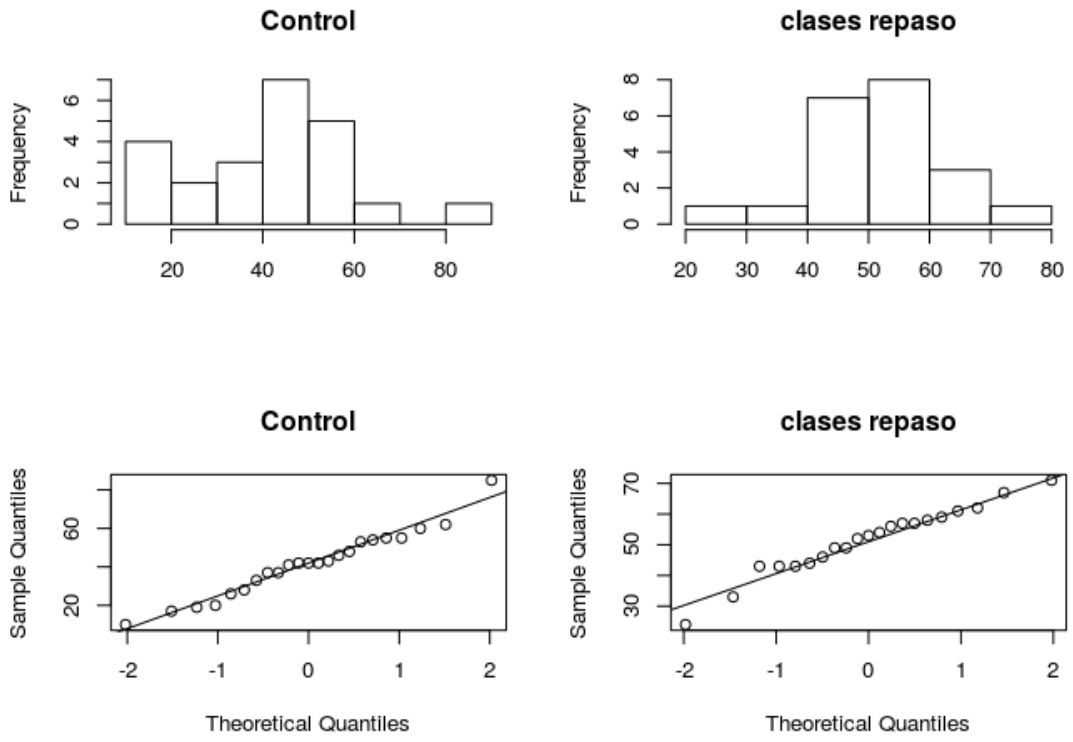
Ejemplo2

Una investigación quiere determinar si las clases de repaso tienen un efecto significativo en el resultado de los exámenes de los estudiantes. Para ello un conjunto de estudiantes se reparte al azar en dos grupos (uno que asiste a clases de repaso y otro que no) y se evalúa su conocimiento con un examen. ¿Existe una diferencia significativa en el promedio entre ambos grupos? Solucionarlo con t-test y test de permutación si se cumplen sus respectivas condiciones. Ejemplo del libro *Bootstrap Methods ans Permutation Test by Tim Hestemberg*.

```
datos <- data.frame(grupo = c("clases repaso", "clases repaso", "clases repaso",
    "clases repaso", "clases repaso", "clases repaso", "clases repaso", "clases
repaso", "clases repaso", "clases repaso", "clases repaso", "clases repaso",
"clases repaso", "clases repaso", "clases repaso", "clases repaso", "clases
repaso", "clases repaso", "clases repaso", "clases repaso", "clases repaso",
"control", "control", "control", "control", "control", "control", "control",
"control", "control", "control", "control", "control", "control", "control",
"control", "control", "control", "control", "control", "control", "control",
"control", "control"), resultado = c(24, 43, 58, 71, 43, 49, 61, 44, 67, 49, 53,
56, 59, 52, 62, 54, 57, 33, 46, 43, 57, 26, 62, 37, 42, 43, 55, 54, 20, 85, 33, 41,
19, 60, 53, 42, 46, 10, 17, 28, 48, 37, 42, 55))
```

Los *t-test* se caracterizan por ser válidos calculando *p-values* para la diferencia de medias poblacionales siempre y cuando se cumpla que los datos son independientes y que las observaciones proceden de distribuciones normales. Son robustos frente a cierta asimetría si el tamaño de ambas muestras es >= 30.

```
par(mfrow = c(2, 2))
hist(datos[datos$grupo == "control", "resultado"], main = "Control", xlab = "")
hist(datos[datos$grupo == "clases repaso", "resultado"], main = "clases repaso",
    xlab = "")
qqnorm(datos[datos$grupo == "control", "resultado", ], main = "Control")
qqline(datos[datos$grupo == "control", "resultado"])
qqnorm(datos[datos$grupo == "clases repaso", "resultado"], main = "clases repaso")
qqline(datos[datos$grupo == "clases repaso", "resultado"])
```



```
par(mfrow = c(1, 1))
tapply(X = datos$resultado, INDEX = datos$grupo, FUN = function(x) {
    shapiro.test(x)
})
```

```
## $`clases repaso`
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.96635, p-value = 0.6517
##
##
## $control
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.97181, p-value = 0.7322
```

```
tapply(X = datos$resultado, INDEX = datos$grupo, FUN = mean)
```

```
## clases repaso       control
##      51.47619      41.52174
```

```
tapply(X = datos$resultado, INDEX = datos$grupo, FUN = sd)
```

```
## clases repaso       control
##      11.00736      17.14873
```

Ambas muestras se distribuyen aproximadamente de forma normal por lo que se puede aplicar un t-test con corrección de Welch dado que las varianzas no son iguales.

```
t.test(resultado ~ grupo, data = datos, altermative = "two.sided", mu = 0,
var.equal = FALSE,
    paired = FALSE, conf.level = 0.95)
```

```
##  Welch Two Sample t-test
##
## data:  resultado by grupo
## t = 2.3109, df = 37.855, p-value = 0.02638
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   1.23302 18.67588
## sample estimates:
## mean in group clases repaso       mean in group control
##                    51.47619                    41.52174
```

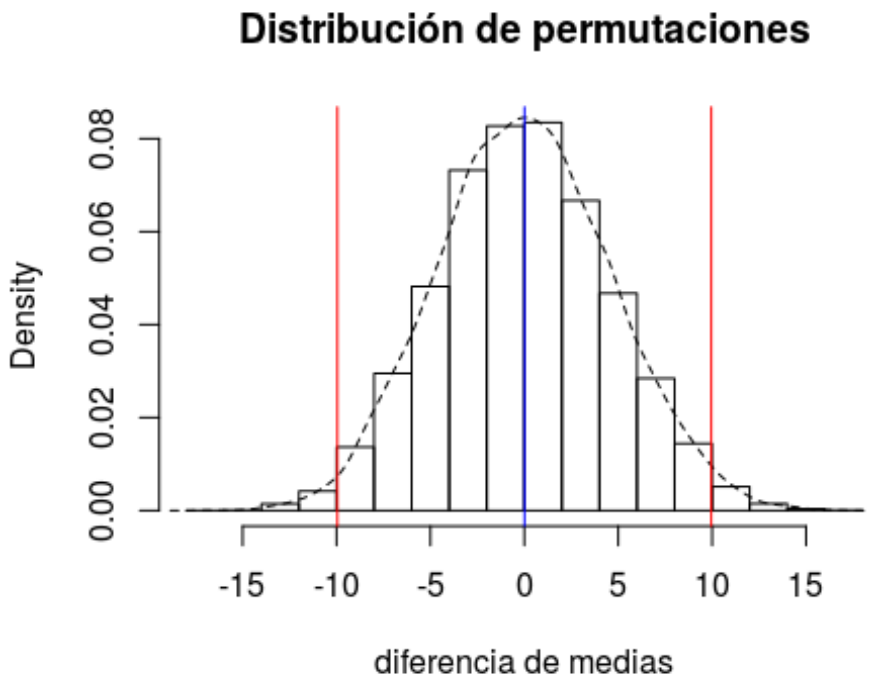
Solución mediante test de permutación

```
distribucion_permutaciones <- rep(NA, 9999)
n_control <- length(datos$grupo[datos$grupo == "control"])
n_tratamiento <- length(datos$grupo[datos$grupo == "clases repaso"])

for (i in 1:9999) {
    datos_aleatorizados <- sample(datos$resultado)
    distribucion_permutaciones[i] <- mean(datos_aleatorizados[1:n_control]) -
                        mean(datos_aleatorizados[n_control + 1:n_tratamiento])
}

diferencia_observada <- mean(datos[datos$grupo == "control", "resultado"]) -
                        mean(datos[datos$grupo == "clases repaso", "resultado"])


hist(distribucion_permutaciones, freq = FALSE, main = "Distribución de
permutaciones", xlab = "diferencia de medias")
lines(density(distribucion_permutaciones), lty = "dashed")
abline(v = diferencia_observada, col = 2)
abline(v = -diferencia_observada, col = 2)
abline(v = mean(distribucion_permutaciones), col = 4)
```




```
p_value = (sum(abs(distribucion_permutaciones) > abs(diferencia_observada)))/9999
p_value
```

```
## [1] 0.02720272
```

```
p_value_corregido = ((sum(abs(distribucion_permutaciones) >
abs(diferencia_observada))) +
    1)/(9999 + 1)
p_value_corregido
```

```
## [1] 0.0273
```

Ambos *p-values* (t-test y permutation test) se asemejan mucho. Esto es debido a que, como muestra la *permutation distribution*, la diferencia de medias se distribuye de forma normal por lo que los t-test son válidos. En caso de que no fuese así el test de permutación es una mejor opción.

En caso de querer saber si el t-test es adecuado, una forma de comprobarlo es generar la *permutation distribution* y verificar si se distribuye de forma normal.

Test de permutación para comparar varianzas

En la mayoría de casos, la comparación entre grupos se centra en estudiar diferencias en la posición de las distribuciones (media, mediana...), sin embargo también puede ser de interés comparar si las varianzas de dos grupos son iguales. Para este tipo de estudios se puede emplear un test de permutación en el que el estadístico es la varianza.

Ejemplo

Empleando los datos del ejemplo de los estudiantes y la capacidad empática, se va a suponer que los datos pertenecen a un estudio que está realizando un departamento médico para evaluar si un fármaco aumenta el razonamiento matemático de los estudiantes. Se sabe que no existe diferencia en el promedio de las puntuaciones obtenidas en el examen, pero se quiere evaluar si hay diferencia en la variabilidad. Esto es interesante porque aunque el fármaco no sea capaz de incrementar el promedio de capacidad de razonamiento, podría disminuir la diferencia entre sujetos

La hipótesis nula considera que ambos grupos son iguales por lo que:

$$\sigma^2_{control} = \sigma^2_{tratamiento}$$

$$\sigma^2_{control} - \sigma^2_{tratamiento} = 0$$

```
datos <- read.table("http://www.tc.umn.edu/~zief0002/Comparing-
Groups/Data/AfterSchool.csv",  header = TRUE, sep = ",", row.names = 1)
datos <- datos[, c("Treatment", "Delinq")]
colnames(datos) <- c("grupo", "puntuacion")
datos$grupo <- as.factor(datos$grupo)
levels(datos$grupo) <- c("control", "tratamiento")
```

La diferencia observada en las varianzas es:

```
aggregate(formula = puntuacion ~ grupo, data = datos, FUN = var)
```

```
##         grupo puntuacion
## 1     control   110.6891
## 2 tratamiento    80.5368
```

```
dif_obs <- var(datos[datos$grupo == "control", "puntuacion"]) -
var(datos[datos$grupo ==
    "tratamiento", "puntuacion"])
dif_obs
```

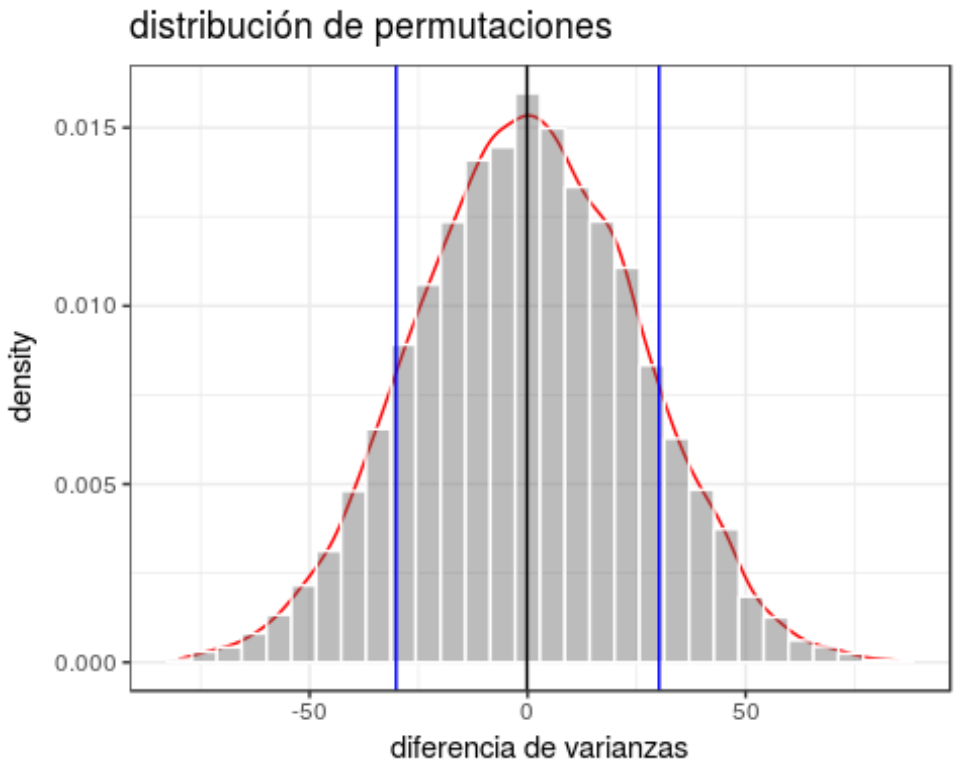
```
## [1] 30.15232
```

Mediante permutaciones se obtiene la distribución de la diferencia de varianza esperada únicamente debido a la asignación aleatoria de las observaciones a cada grupo, siendo cierta la hipótesis nula.

```
distribucion_permutaciones <- rep(NA, 9999)
n_control <- length(datos$grupo[datos$grupo == "control"])
n_tratamiento <- length(datos$grupo[datos$grupo == "tratamiento"])

for (i in 1:9999) {
    datos_aleatorizados <- sample(datos$puntuacion)
    distribucion_permutaciones[i] <- var(datos_aleatorizados[1:n_control]) -
                        var(datos_aleatorizados[n_control + 1:n_tratamiento])
}
```

```
qplot(distribucion_permutaciones, geom = "blank") +
  geom_line(aes(y = ..density..), stat = "density", colour = "red") +
  geom_histogram(aes(y = ..density..), alpha = 0.4, colour = "white") +
  geom_vline(xintercept = mean(distribucion_permutaciones)) +
  geom_vline(xintercept = dif_obs, colour = "blue") +
  geom_vline(xintercept = -dif_obs, colour = "blue") +
  labs(title = "distribución de permutaciones", x = "diferencia de varianzas") +
. theme_bw()
```




```
summary(distribucion_permutaciones)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -80.0700 -17.5000  -0.1347  -0.1533  17.5800  85.2500
```

```
sd(distribucion_permutaciones)
```

```
## [1] 25.45499
```

La distribución de las permutaciones muestra que la diferencia media entre varianzas si el fármaco no tiene efecto es muy próxima a cero (-0.1533, línea vertical negra). La desviación típica de la distribución de permutaciones indica la variabilidad de la diferencia de varianzas debida a la asignación aleatoria de los sujetos a los diferentes grupos.

Finalmente, se calcula la probabilidad (*p-value*) de obtener diferencias igual o más extremas que la observada (líneas verticales azules) con y sin corrección de continuidad.

```
p_value = (sum(abs(distribucion_permutaciones) > dif_obs))/9999
p_value
```

```
## [1] 0.2391239
```

```
p_value_corregido = ((sum(abs(distribucion_permutaciones) > dif_obs)) + 1)/(9999 +
    1)
p_value_corregido
```

```
## [1] 0.2392
```

Conclusión

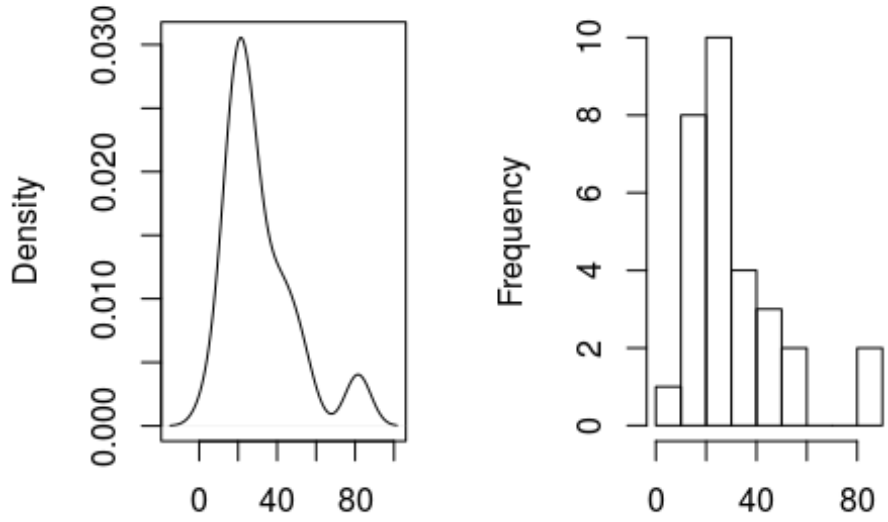
Los 356 sujetos del estudio fueron asignados de forma aleatoria a un grupo control (n=187) o a un grupo tratamiento (n=169) que asistió a clases extra escolares. Un test de permutación se empleó para determinar si existía una diferencia significativa en la variabilidad de la capacidad de razonamiento matemático entre ambos grupos. El *p-value* fue calculado mediante una simulación de Monte Carlo con 9999 permutaciones usando la corrección de continuidad sugerida por Davison and Hikley(1997). El *p-value* obtenido muestra una evidencia muy débil en contra de la hipótesis nula de que el tratamiento no tiene efecto, sugiriendo que tomar el medicamento no reduce la varianza en la capacidad de razonamiento matemático de los estudiantes que formaron parte del experimento.

Boostraping para estimar un parámetro poblacional mediante intervalos de confianza (media, mediana...)

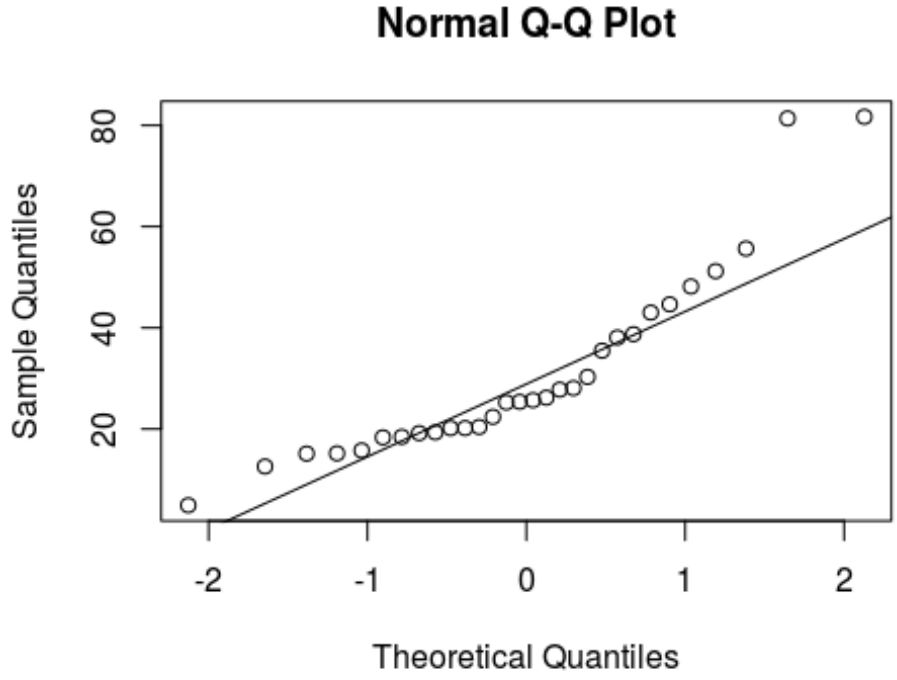
Ejemplo1

Supóngase que se dispone de una muestra formada por 30 observaciones de una variable aleatoria continua. Se quiere inferir en el valor medio de dicha variable en la población generando un intervalo de confianza del 95%. Ejemplo obtenido de *TheRbook*.

```
datos <- c(81.372918, 25.700971, 4.942646, 43.020853, 81.690589, 51.195236,
    55.659909, 15.153155, 38.74578, 12.610385, 22.415094, 18.355721, 38.081501,
    48.171135, 18.462725, 44.642251, 25.391082, 20.410874, 15.778187, 19.351485,
    20.189991, 27.795406, 25.2686, 20.177459, 15.196887, 26.206537, 19.190966,
    35.481161, 28.094252, 30.305922)
par(mfrow = c(1, 2))
plot(density(datos), main = "", xlab = "")
hist(datos, xlab = "", main = "")
```




```
par(mfrow = c(1, 1))
qqnorm(datos)
qqline(datos)
```



```
shapiro.test(datos)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos
## W = 0.85765, p-value = 0.0009003
```

La representación gráfica de los datos y el test de normalidad Shapiro-Wilk muestran que no se distribuyen de forma normal, lo que implica que la aproximación basada en el teorema del límite central para estimar el error estándar $SE = \frac{sd}{\sqrt{n}}$ deja de ser buena y por lo tanto tampoco los intervalos paramétricos basados en la estructura $[parámetro\ estimado \pm t_{\alpha} SE]$ que se generen con esa estimación del SE. Una alternativa es emplear el Bootstrapping.

El método de *Bootstrapping* se basa en generar nuevas pseudomuestras del mismo tamaño que la muestra real, realizando *sampling with replacement* de las observaciones. Si la muestra original es representativa de la población, la distribución del estadístico calculada a partir de las pseudomuestras (*bootstraping distribution*) se asemejará a la distribución muestral que se obtendría si se pudiera acceder a la población para generar nuevas muestras. De tal forma que:

- La *sd* de la *bootstraping distribution* es un estimador del SE.
- La media de la *bootstraping distribution* es un estimador del verdadero parámetro poblacional.

Así pues, a partir de la *bootstraping distribution* se pueden obtener los valores necesarios para crear el intervalo de confianza sin recurrir al teorema del límite central.

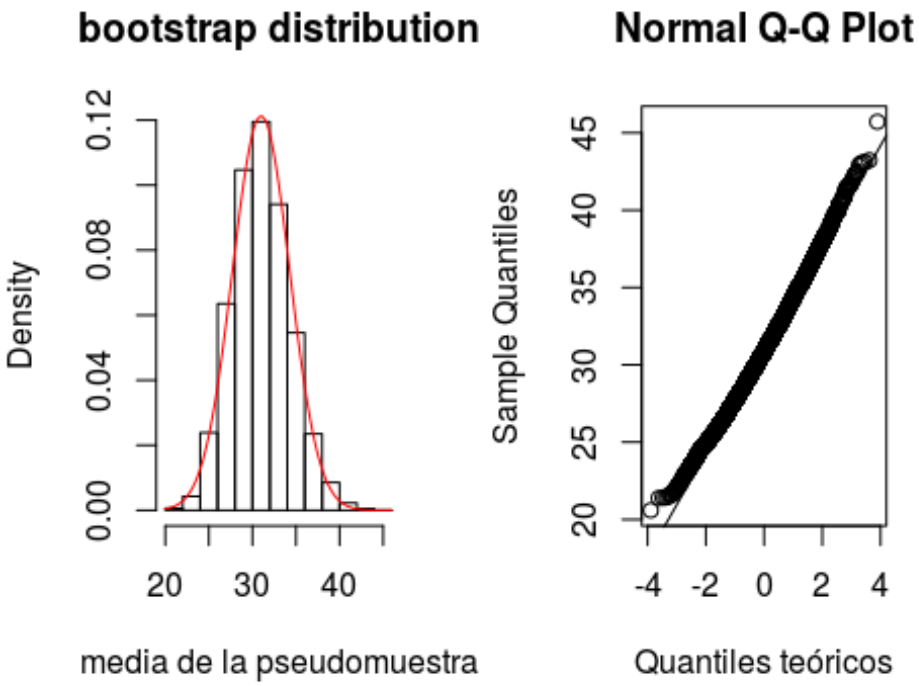
1.Generar la bootstrapping distribution

```
boot_distribution <- rep(NA, 9999)
for (i in 1:9999) {
    boot_distribution[i] <- mean(sample(x = datos, size = 30, replace = TRUE))
}
```

2.Estudio de la bootstrapping distribution

Distribución:

```
par(mfrow = c(1, 2))
hist(boot_distribution, prob = TRUE, main = "bootstrap distribution", xlab = "media
de la pseudomuestra")
curve(dnorm(x, mean = mean(boot_distribution), sd = sd(boot_distribution)),
    add = TRUE, col = "red")
qqnorm(boot_distribution, xlab = "Quantiles teóricos")
qqline(boot_distribution)
```



Si la distribución obtenida por *bootstraping* no es de tipo normal, no se pueden emplear intervalos de confianza basados en la t-distribution ni en percentiles.

Centro:

La media de la *bootstraping distribution* debe de ser cercana a la media de la muestra inicial a partir de la cual se está generando el bootstrapping. A esta diferencia se le llama *bias*.

```
mean(datos) - mean(boot_distribution)
```

```
## [1] -0.02631589
```

En este caso el *bias* es muy pequeño comparado al valor de la media calculada.

3.Intervalo de confianza

Existen varios métodos para generar intervalos de confianza a partir de una *bootstrapping distribution*. Los intervalos *t* y los basados en percentiles emplean la desviación estándar de la bootstrapping distribution como estimación del SE.

Intervalo basado en *t-distribution*:

$$[parámetro\ estimado \pm t_{\alpha,df} SE]$$

- media = mean(datos) = 30.9686559
- $t_{\alpha=0.05,df=29}$ = *qt(p=1-0.05/2, df = 29)* = 2.045
- SE = *sd(boot_distribution)* = 3.2916364
- IC 95% = [24.24, 37.7]

Intervalo de confianza basado en percentiles: un IC del 95% debe abarcar desde el percentil 0.025 al 0.975.

```
quantile(x = boot_distribution,probs = c(0.025,0.975))
```

```
##     2.5%    97.5%
## 25.00097 37.80903
```

Tanto el intervalo basado en el valor *t* como el basado en *percentiles* son únicamente válidos cuando la *bootstrapping distribution* se asemeja a una normal y el *bias* es pequeño. El intervalo basado en percentiles no ignora la asimetría de los datos por lo que se suele considerar más adecuado. En caso de no cumplirse estas condiciones, o de que ambos intervalos no se asemejen entre sí, ninguno es fiable.

El intervalo de confianza más adecuado para *bootstrapping* se conoce como *Intervalo Bootstrapping Bias Corrected Acelerated BCa*. En R existen varios paquetes que permiten obtener la distribución de bootstrapping así como el intervalo *BCa*.

Solución mediante la función boot()

La función `boot()` recibe como argumentos un vector con las observaciones, una función que calcule el estadístico de interés, que tiene que ser definida previamente por el usuario, y el número de simulaciones.

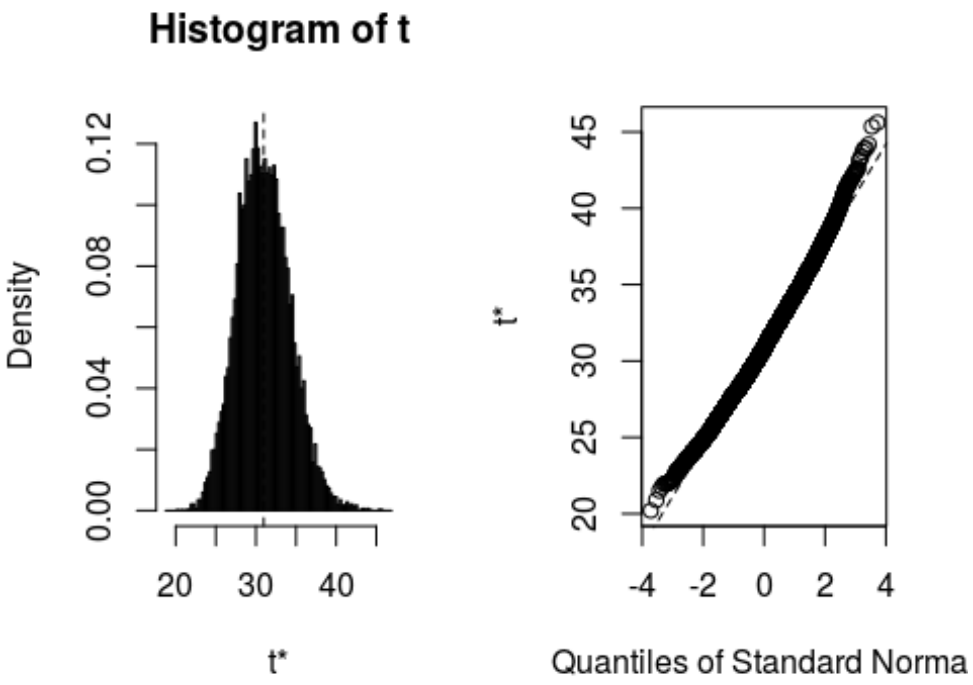
```
require(boot)

media <- function(valores, i) {
    mean(valores[i])
}
boot_distribution <- boot(datos, media, R = 9999)
boot_distribution
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = datos, statistic = media, R = 9999)
##
##
## Bootstrap Statistics :
##     original     bias    std. error
## t1* 30.96866 0.04246029     3.29509
```

Una vez generado el objeto *boot*, se puede tanto mostrar gráficamente la distribución como calcular los diferentes tipos de intervalos.

```
plot(boot_distribution)
```




```
boot.ci(boot_distribution, conf = 0.95)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 9999 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_distribution, conf = 0.95)
##
## Intervals :
## Level      Normal              Basic
## 95%   (24.47, 37.38 )   (24.07, 37.01 )
##
## Level     Percentile            BCa
## 95%   (24.93, 37.87 )   (25.47, 38.81 )
## Calculations and Intervals on Original Scale
```

Otra opción es la función `bootES()` del paquete *bootES* que internamente hace una llamada a la función `boot()` pero sin necesidad de definir la función del estadístico.

```
bootES(data = datos, R = 9999, mean)
```

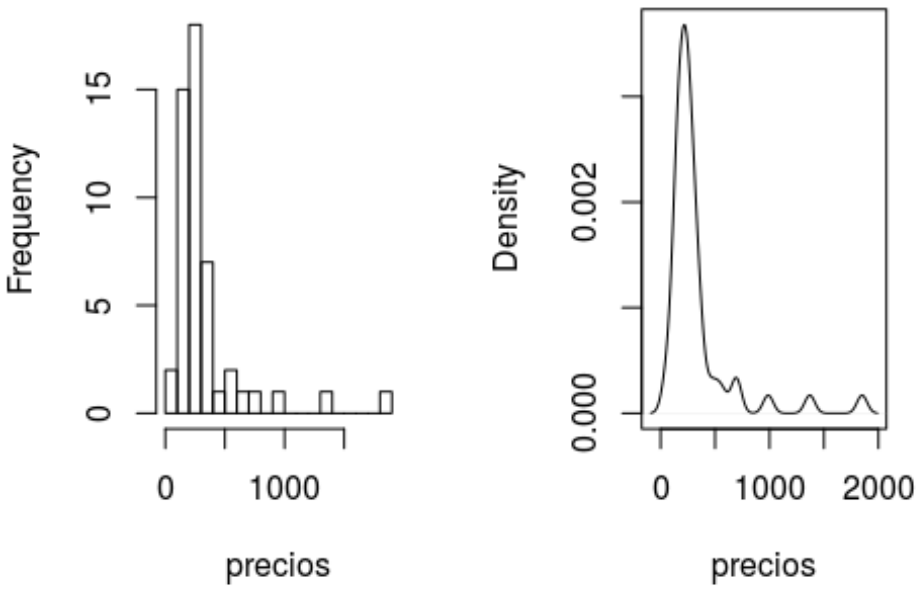
Ejemplo 2

Supóngase que se quiere estimar el precio de las viviendas en una ciudad a partir de una muestra de 50 propiedades. ¿Cuál es el intervalo de confianza del 95% del precio de las viviendas de la ciudad? Ejemplo del libro *Bootstrap Methods ans Permutation Test by Tim Hestemberg*.

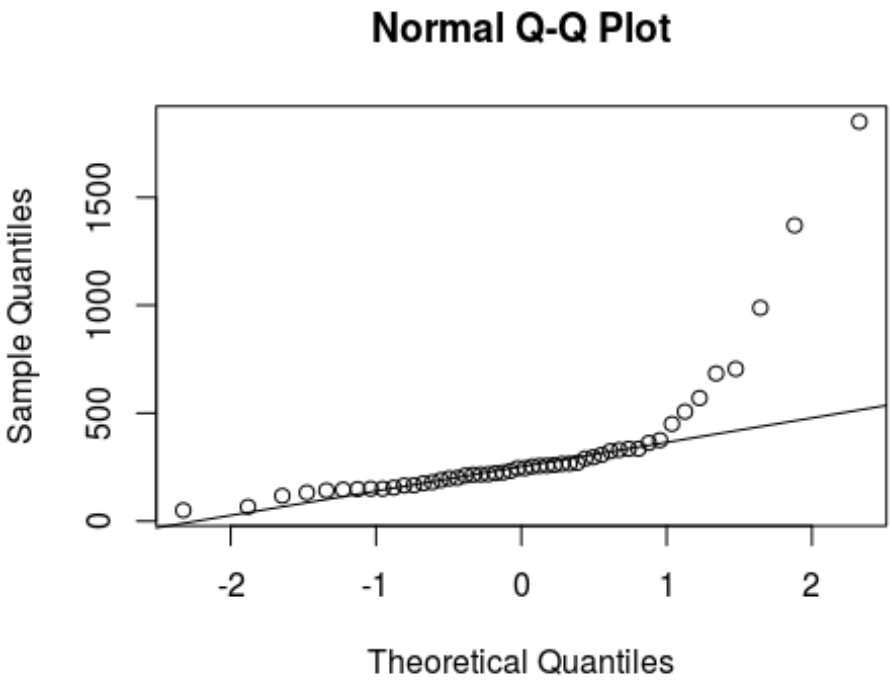
```
precios <- c(142, 175, 198, 150, 705, 232, 50, 146, 155, 1850, 132, 215, 117,
    245, 290, 200, 260, 450, 66, 165, 362, 307, 266, 166, 375, 245, 211, 265,
    296, 335, 335, 1370, 256, 148, 988, 324, 216, 684, 270, 330, 222, 180, 257,
    253, 150, 225, 217, 570, 507, 190)
```

La representación gráfica de los datos refleja que la muestra es muy asimétrica:

```
par(mfrow = c(1, 2))
hist(precios, breaks = 20, main = "")
plot(density(precios), main = "", xlab = "precios")
```



```
par(mfrow = c(1, 1))
qqnorm(precios)
qqline(precios)
```



Si los datos no se distribuyen de forma normal, no se pueden aplicar métodos paramétricos basados en el teorema del límite central. Además, dado que hay asimetría, es recomendable emplear estadísticos más robustos que la media como la mediana o la media truncada (*trimmed mean*). Para métodos de bootstrapping se recomienda la media truncada antes que la mediana.

25% media truncada: Se define como la media del 50% de los valores centrales. Es decir, la media de los valores que caen entre el primer y el tercer percentil (lo que forma la caja en los diagramas box-plot). En R se calcula mediante `mean(x=, trim= 0.25)`.

La idea de Bootstrapping es que la *bootstrapping distribution*, se debe asemejar a la *sampling distribution* del parámetro estimado (en este caso la trimmed mean). Y que la *sd* de la *bootstrapping distribution* se aproxima al *SE* de la *sampling distribution*. Es decir, mediante bootstrapping podemos obtener los valores necesarios para calcular el intervalo de confianza sin recurrir a los métodos teóricos basados en teorema del límite central.

En este caso se emplean las función `boot()` de R.

1.Cálculo de la bootstraping distribution

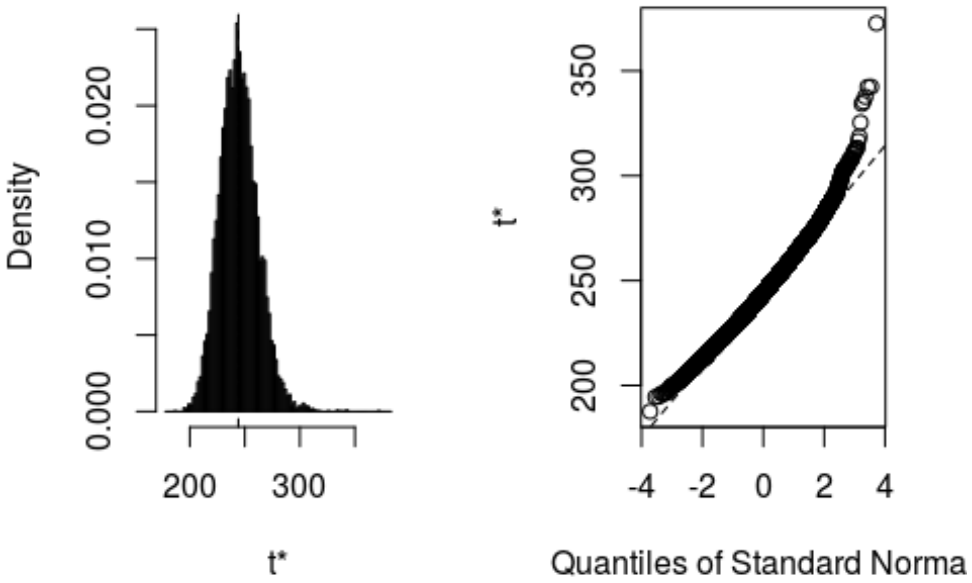
```
require(boot)
media_truncada <- function(valores, i) {
    mean(valores[i], trim = 0.25)
}
boot_distribution <- boot(precios, media_truncada, R = 9999)
boot_distribution
```

```
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = precios, statistic = media_truncada, R = 9999)
##
##
## Bootstrap Statistics :
##     original    bias    std. error
## t1* 244.0385 0.7124366    17.39653
```

2.Estudio de la bootstrapping distribution

Distribución:

```
plot(boot_distribution)
```



La distribución generada no cumple las condiciones de normalidad. Esto implica que los intervalos de confianza obtenidos por percentiles o basados en la t-distribution no son fiables.

Centro:

La media de la *bootstrapping distribution* debe de ser cercana al valor del estadístico estudiado en la muestra inicial a partir de la cual se está generando el bootstrapping. A esta diferencia se le llama *bias*.

```
mean(precios, trim = 0.25) - mean(boot_distribution$t)
```

```
## [1] -0.7124366
```

3.Intervalo de confianza

```
boot.ci(boot_distribution)
```

```
## Warning in boot.ci(boot_distribution): bootstrap variances needed for
## studentized intervals

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 9999 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_distribution)
##
## Intervals :
## Level      Normal              Basic
## 95%   (209.2, 277.4 )   (207.2, 274.8 )
##
## Level     Percentile            BCa
## 95%   (213.3, 280.8 )   (213.8, 281.9 )
## Calculations and Intervals on Original Scale
```

Bootstrapping para cálculo de *p-value*.

El Bootstrapping como test de significancia para la diferencia entre grupos se emplea cuando se quiere estudiar si la diferencia entre dos poblaciones es significativa, empleando muestras aleatorias y separadas de cada una de las poblaciones.

Ejemplo1

Un estudio pretende determinar si existe una diferencia entre el promedio de los logros académicos de los estudiantes inmigrantes procedentes de México y los procedentes de otros países latinoamericanos. Para ello se obtiene una muestra aleatoria de 2000 estudiantes que llegaron al país entre 2005 y 2006 entre los que hay mexicanos y no mexicanos. Ejemplo libro *Comparing Groups Randomization and Bootstrap Methods Using R*.

Dado que se trata de un diseño muestral en el que se ha obtenido una muestra aleatoria con individuos de cada uno de los grupos que se van a comparar, el método adecuado es bootstrapping. *Para este tipo de estudios no se me ocurre como podría hacerse un diseño experimental con asignación aleatoria a los grupos.*

```
datos <- read.table("http://www.tc.umn.edu/~zief0002/Comparing-
Groups/Data/LatinoEd.csv", header = TRUE, sep = ",", row.names = 1)
datos <- datos[, c("Achieve", "Mex")]
colnames(datos) <- c("nota", "nacionalidad")
datos$nacionalidad <- as.factor(datos$nacionalidad)
levels(datos$nacionalidad) <- c("no mexicana", "mexicana")
head(datos)
```

```
##   nota nacionalidad
## 1 59.2     mexicana
## 2 63.7     mexicana
## 3 62.4     mexicana
## 4 46.8     mexicana
## 5 67.6     mexicana
## 6 63.1  no mexicana
```

El promedio del rendimiento académico es de 5.92 puntos menor en estudiantes de origen mexicano que el de otras nacionalidades latinas.

```
aggregate(formula = nota ~ nacionalidad, data = datos, FUN = mean)
```

```
##   nacionalidad     nota
## 1  no mexicana 64.51471
## 2     mexicana 58.59310
```

```
aggregate(formula = nota ~ nacionalidad, data = datos, FUN = sd)
```

```
##   nacionalidad     nota
## 1  no mexicana 13.03141
## 2     mexicana 15.62688
```

```
aggregate(formula = nota ~ nacionalidad, data = datos, FUN = length)
```

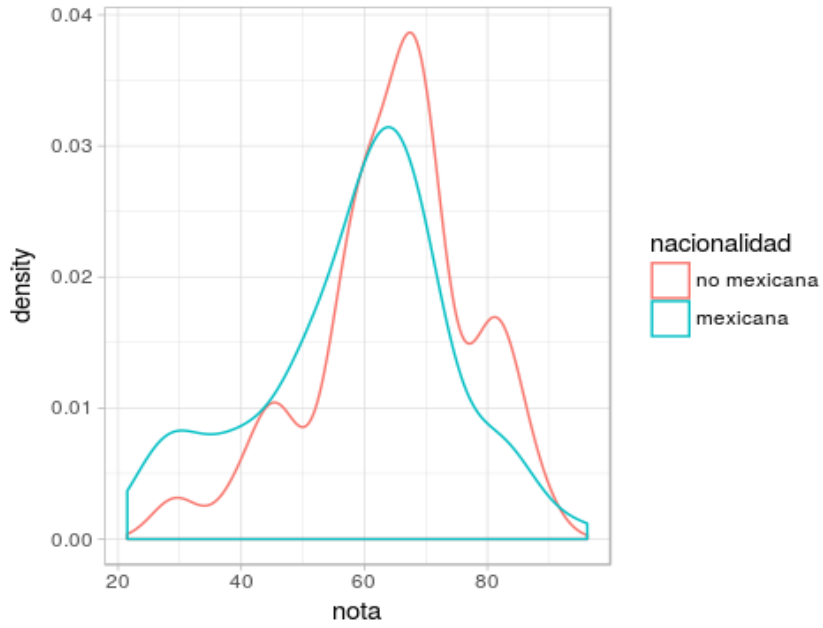
```
##   nacionalidad nota
## 1  no mexicana   34
## 2     mexicana  116
```

```
dif_obs <- mean(datos[datos$nacionalidad == "mexicana", "nota"]) -
           mean(datos[datos$nacionalidad == "no mexicana", "nota"])
dif_obs
```

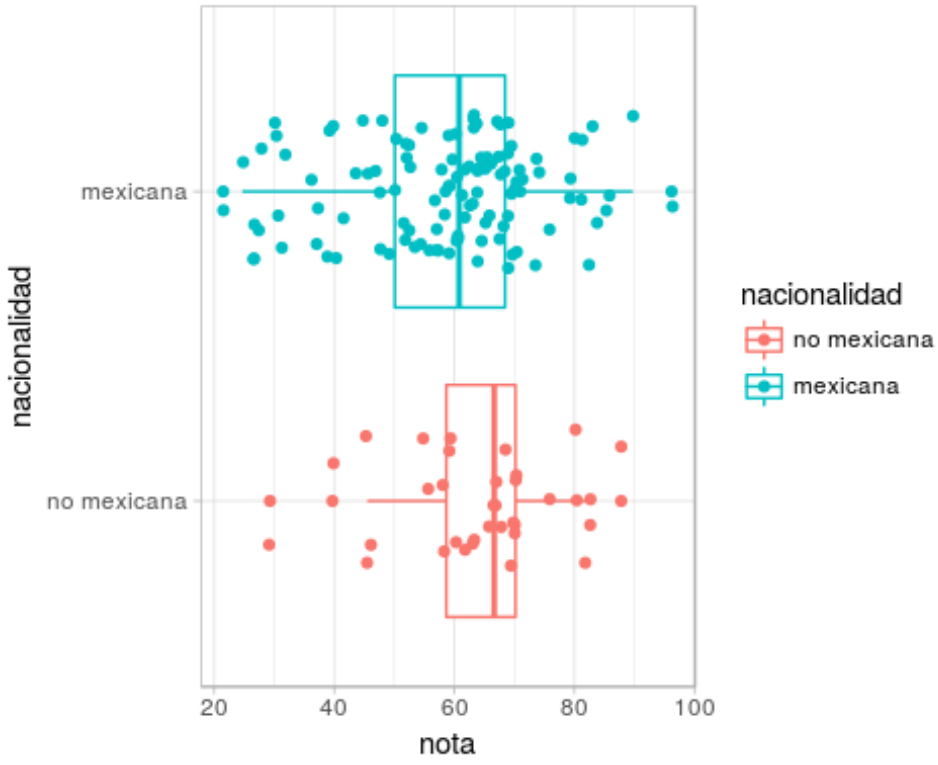
```
## [1] -5.921602
```

La exploración gráfica de los datos indica posibles diferencias en la posición de la distribución.

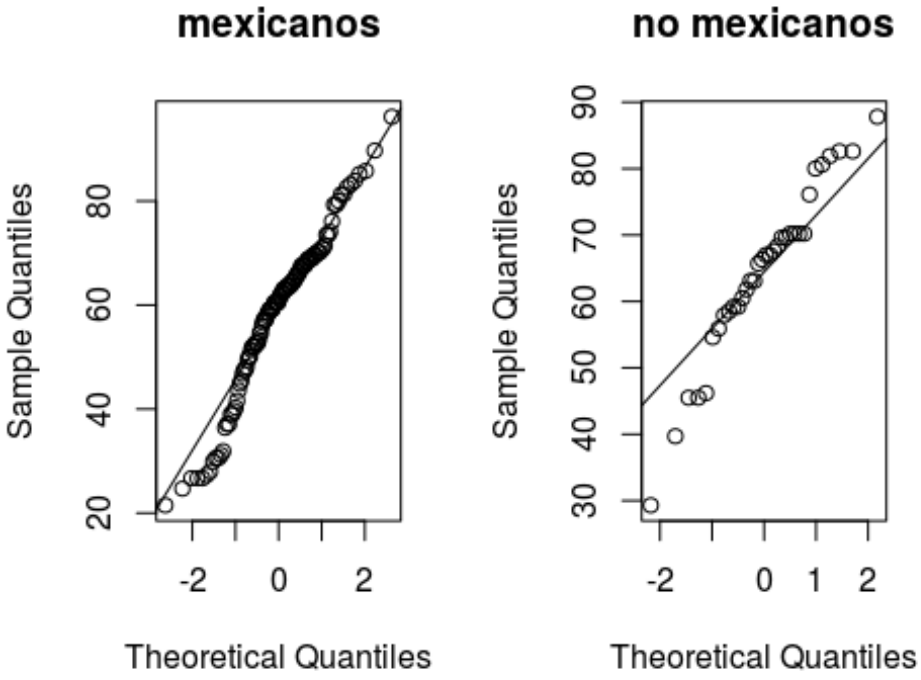
```
ggplot(data = datos, aes(nota, colour = nacionalidad)) +
 geom_density() +
 theme_light()
```




```
ggplot(data = datos, aes(x = nacionalidad, y = nota, colour = nacionalidad)) +
  geom_boxplot() +
  coord_flip() +
  geom_jitter(width = 0.25) +
  theme_light()
```



```
par(mfrow = c(1, 2))
qqnorm(datos[datos$nacionalidad == "mexicana", "nota"], main = "mexicanos")
qqline(datos[datos$nacionalidad == "mexicana", "nota"])
qqnorm(datos[datos$nacionalidad == "no mexicana", "nota"], main = "no mexicanos")
qqline(datos[datos$nacionalidad == "no mexicana", "nota"])
```



Los datos no se distribuyen de forma normal, siendo más marcada la falta de normalidad en las observaciones de nacionalidad no mexicana que es la muestra de menor tamaño.

El objetivo del estudio es determinar si la diferencia observada de 5.9 unidades está dentro de lo que cabría esperar por puro azar debido al muestreo aleatorio si no existiera diferencia entre las poblaciones (H_0: todas las observaciones proceden de la misma población/distribución). Si se obtuviera una nueva muestra de 2000 estudiantes la diferencia promedio entre estudiantes mexicanos y no mexicanos sería ligeramente distinta aunque no existiera una diferencia real entre nacionalidades. Por lo tanto, dar respuesta al problema pasa por determinar cuanta diferencia se espera por el simple hecho de repetir el muestreo.

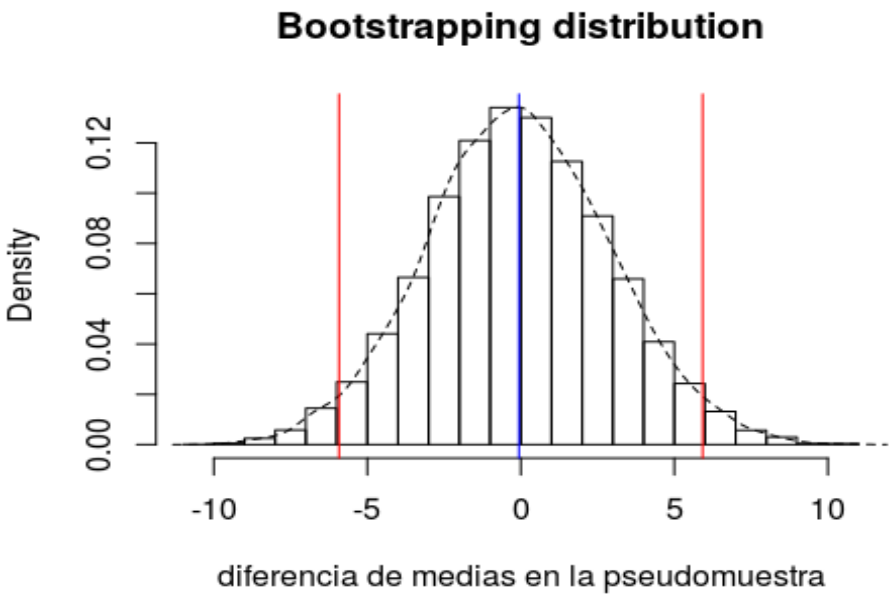
Mediante bootstrapping se generan nuevas pseudomuestras de 2000 individuos empleando las dos muestras originales (mexicanos y no mexicanos) combinadas. Con cada una de las pseudomuestras se generan dos grupos de tamaños iguales a los grupos de las muestras originales (34, 116) y se calcula la diferencia del estadístico, en este caso la media.

```
boot_distribution <- rep(NA, 9999)
n_mexicana <- length(datos$nacionalidad[datos$nacionalidad == "mexicana"])
n_nomexicana <- length(datos$nacionalidad[datos$nacionalidad == "no mexicana"])
for (i in 1:9999) {
    pseudomuestra <- sample(datos$nota, size = length(datos$nota), replace = TRUE)
    boot_distribution[i] <- mean(pseudomuestra[1:n_mexicana]) -
                            mean(pseudomuestra[n_mexicana + 1:n_nomexicana])
}
```

El proceso se asemeja mucho al empleado en los test de permutación. La diferencia radica en que en los test de permutación se emplean siempre las observaciones de la muestra original ordenadas de forma distinta en cada iteración. En este caso, en cada iteración y antes del reparto en grupos, se genera una nueva pseudomuestra que tiene el mismo tamaño que la muestra original pero formada por distintas observaciones.

Los datos simulados forman lo que se conoce como *bootstrapping distribution* y representa la variación esperada en la diferencia de medias debida únicamente al muestreo aleatorio.

```
hist(boot_distribution, freq = FALSE, main = "Bootstrapping distribution", xlab =
"diferencia de medias en la pseudomuestra")
lines(density(boot_distribution), lty = "dashed")
abline(v = dif_obs, col = 2)
abline(v = -dif_obs, col = 2)
abline(v = mean(boot_distribution), col = 4)
```



```
summary(boot_distribution)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -10.05000  -2.08400  -0.09823  -0.06154   1.95300  10.93000
```

```
sd(boot_distribution)
```

```
## [1] 2.985099
```

Como era de esperar, la diferencia entre la media de los grupos considerando que todos proceden de una misma población está centrada en cero. La desviación típica de la *bootsrapping distribution* indica que la variabilidad en la diferencia de medias debida al muestreo es de 2.9851 unidades.

Finalmente, se calcula la probabilidad (*p-value*) de obtener diferencias igual o más extremas que la observada (líneas verticales rojas) con y sin corrección de continuidad.

```
p_value = (sum(abs(boot_distribution) > abs(dif_obs)))/9999
p_value
```

```
## [1] 0.04970497
```

```
p_value_corregido = ((sum(abs(boot_distribution) > abs(dif_obs))) + 1)/(9999 +
    1)
p_value_corregido
```

```
## [1] 0.0498
```

Conclusión

Un test de bootstrapping no paramétrico se empleó para determinar si existe diferencia entre el promedio de logros académicos entre inmigrantes mexicanos y no mexicanos. Los datos se obtuvieron mediante muestreo aleatorio asegurando así la independencia de los datos. El *p-value* se calculó mediante 9999 simulaciones de Monte Carlo, empleando la corrección de continuidad sugerida por Davison y Hinkley (1997) resultando en 0.049. Esta es una evidencia moderada en contra de la hipótesis nula de que no hay diferencia entre poblaciones y sugiere que los logros académicos entre inmigrantes de nacionalidad mexicana y no mexicana son distintos.

Boostraping para intervalos de confianza y tamaño del efecto (*effect size*) de la diferencia entre dos poblaciones

Los test de hipótesis que evalúan la diferencia entre dos poblaciones generan un p-value basado en la hipótesis nula de que no existe diferencia del estadístico (media, mediana...) entre las poblaciones. Por esta razón, cuando el *bootstrapping* se emplea como test de hipótesis, las observaciones se mezclan totas juntas y posteriormente se extraen las pseudomuestras. Cuando la finalidad del estudio es generar intervalos de confianza para la verdadera diferencia de un estadístico entre dos poblaciones, la hipótesis nula considera que las muestras sí proceden de dos poblaciones diferentes cada una con un valor distinto para el estadístico estudiado. Dada esta hipótesis nula, las pseudomuestras obtenidas por *sampling with replacement* se tienen que generar de forma separada para cada grupo. Esta es la diferencia entre el empleo de *bootstrapping* para calcular *p-values* y para calcular intervalos de confianza.

Pasos del Bootstrapping no paramétrico para intervalos de confianza (*hipótesis alternativa*)

1. Generar una nueva pseudomuestra del grupo A del mismo tamaño que la muestra original n_A y empleando únicamente las observaciones pertenecientes a dicho grupo.
2. Generar una nueva pseudomuestra del grupo B del mismo tamaño que la muestra original n_B y empleando únicamente las observaciones pertenecientes a dicho grupo.
3. Calcular la diferencia del estadístico entre las dos nuevas pseudomuestras.
4. Repetir el proceso múltiples veces, almacenando la diferencia calculada en cada iteración.
5. El conjunto de valores generado forma lo que se conoce como la *bootstap distribution* de la diferencia del estadístico dada la hipótesis nula de que ambas muestras proceden de dos poblaciones distintas. Esta distribución tiende a estar centrada en el verdadero valor de la diferencia entre ambas poblaciones.
6. A partir de la *bootstap distribution* generar un intervalo de confianza para el parámetro poblacional. Las mismas consideraciones explicadas para intervalos de confianza de una única población se aplican también a la diferencia entre poblaciones.

Tamaño del efecto (effect size)

El cálculo de *p-value* y los intervalos de confianza permiten responder a dos de las preguntas fundamentales cuando se comparan poblaciones:

- ¿Existen evidencias que apunten a la existencia de una diferencia real entre las dos poblaciones? o ¿Es la diferencia observada la esperada por simple variabilidad?
- ¿Si la diferencia es real, cómo de grande es esta diferencia?

A pesar de la importancia de estas dos preguntas, existe una tercera que es fundamental a la hora de tomar decisiones en base a los resultados. ¿Es la diferencia suficientemente grande como para ser práctica? Cuando las unidades en la que se expresa la diferencia son frecuentes en el día a día, se puede interpretar fácilmente. Sin embargo, en muchas ocasiones la escala en la que se mide la diferencia no es intuitiva, lo que complica su interpretación así como su divulgación. Por esta razón siempre se debe reportar la diferencia obtenida en unidades estandarizadas, lo que se conoce como tamaño de efecto o *effect size*.

Uno de los tamaños de efecto más empleados es la *Cohen's d*, que consiste en expresar la diferencia entre las medias en términos de desviación típica (*pooled sd*).

Ejemplo 1

Empleando los datos del estudio de logros académicos entre inmigrantes de nacionalidad mexicana y no mexicana, generar un intervalo de confianza para la diferencia en la media de rendimiento académico entre ambas poblaciones.

```
datos <- read.table("http://www.tc.umn.edu/~zief0002/Comparing-
Groups/Data/LatinoEd.csv",
    header = TRUE, sep = ",", row.names = 1)
datos <- datos[, c("Achieve", "Mex")]
colnames(datos) <- c("nota", "nacionalidad")
datos$nacionalidad <- as.factor(datos$nacionalidad)
levels(datos$nacionalidad) <- c("no mexicana", "mexicana")
head(datos,4)
```

```
##   nota nacionalidad
## 1 59.2     mexicana
## 2 63.7     mexicana
## 3 62.4     mexicana
## 4 46.8     mexicana
```

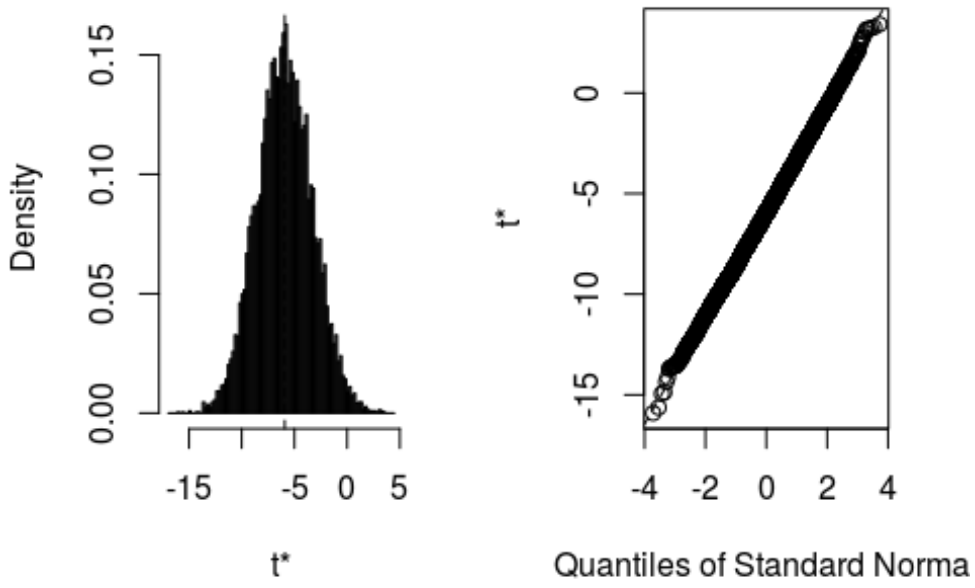
Se recurre a dos funciones de R que automatizan el proceso de *resampling* teniendo en cuenta cada grupo:

Función boot()

Esta función requiere que se defina previamente la función que calcula el estadístico de interés, en este caso la media.

```
require(boot)
# Función que calcula la diferencia entre medias de los dos grupos
diferencia_medias <- function(data, i) {
    pseudomuestra <- data[i, ]
    mean(pseudomuestra$nota[pseudomuestra$nacionalidad == "mexicana"]) -
    mean(pseudomuestra$nota[pseudomuestra$nacionalidad == "no mexicana"])
}

# El argumento strata de la funcion boot() permite identificar los grupos
set.seed(5290)
boot_distribution <- boot(data = datos, statistic = diferencia_medias, R = 9999,
                      strata = datos$nacionalidad)
plot(boot_distribution)
```



```
boot_distribution
```

```
##
## STRATIFIED BOOTSTRAP
##
##
## Call:
## boot(data = datos, statistic = diferencia_medias, R = 9999, strata =
datos$nacionalidad)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* -5.921602 -0.009100393    2.629031
```

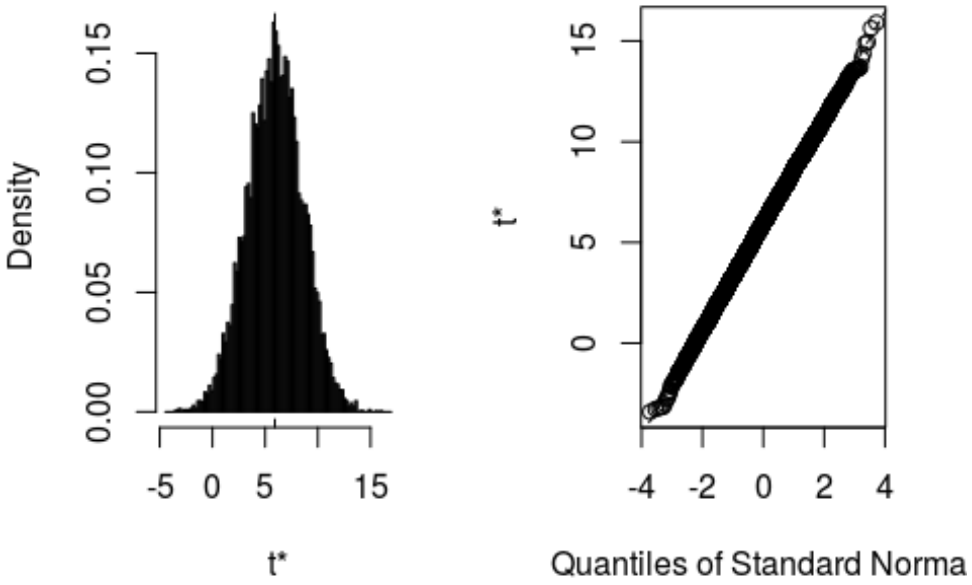
```
boot.ci(boot_distribution)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 9999 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_distribution)
##
## Intervals :
## Level      Normal              Basic
## 95%   (-11.065,  -0.760 )   (-11.126,  -0.860 )
##
## Level     Percentile            BCa
## 95%   (-10.983,  -0.717 )   (-10.958,  -0.670 )
## Calculations and Intervals on Original Scale
```

Funcion bootES()

La función `bootES()` hace una llamada interna a `boot()` pero facilita la sintaxis al no necesitar definir la función del estadístico cando ya existen en R (mean, median...).

```
require(bootES)
set.seed(5290)
bootES(data = datos, data.col = "nota", group.col = "nacionalidad", contrast =
c("mexicana",
    "no mexicana"), R = 9999, plot = TRUE)
```




```
##
## User-specified lambdas: (mexicana, nomexicana)
## Scaled lambdas: (-1, 1)
## 95.00% bca Confidence Interval, 9999 replicates
## Stat        CI (Low)    CI (High)   bias        SE
## 5.922       0.670       10.958      0.009       2.629
```

Como es de esperar, la distribución generada está centrada en el valor de la diferencia observada ya que el resampling se está haciendo acorde a la hipótesis nula de que las muestras proceden de poblaciones distintas.

Dado que la distribución obtenida cumple la normalidad, todos los diferentes tipos de intervalos (percentiles, t, BCa) se consideran válidos.

Tamaño del efecto

La función `bootES()` permite, además de generar los intervalos de confianza para la diferencia de dos medias en las unidades en las que se está midiendo la variable, calcular intervalos para múltiples tipos de tamaño de efecto. *Se puede encontrar una descripción detallada de todos ellos en BootES: An R package for bootstrap confidence intervals on effect sizes (Kris N.Kirby & Daniel Gerlanc).*

```
require(bootES)
bootES(data = datos, data.col = "nota", group.col = "nacionalidad", contrast =
c("mexicana",
    "no mexicana"), R = 9999, effect.type = "cohens.d")
```

```
##
## User-specified lambdas: (mexicana, nomexicana)
## Scaled lambdas: (-1, 1)
## 95.00% bca Confidence Interval, 9999 replicates
## Stat        CI (Low)    CI (High)   bias        SE
## 0.392       0.031       0.725       0.004       0.178
```

Como muestra el resultad de la función, el tamaño de efecto es de 0.392, y su CI del 95% (0.03, 0.725). Este valor sugiere que la diferencia observada en las notas aunque significativa, es pequeña.

Ejemplo 2

Se dispone de un set de datos con información sobre el tiempo que una empresa de telefonía tarda en reparar los problemas que tienen dos grupos de consumidores (CLEC y ILEC). Se desea conocer cuál es la diferencia en la media de los tiempos de reparación de ambos grupos empleando un intervalo de confianza del 95%. Ejemplo del libro *Bootstrap Methods ans Permutation Test by Tim Hestemberg*.

```
require("resample")
data("Verizon")
head(Verizon)
```

```
##    Time Group
## 1 17.50  ILEC
## 2  2.40  ILEC
## 3  0.00  ILEC
## 4  0.65  ILEC
## 5 22.23  ILEC
## 6  1.20  ILEC
```

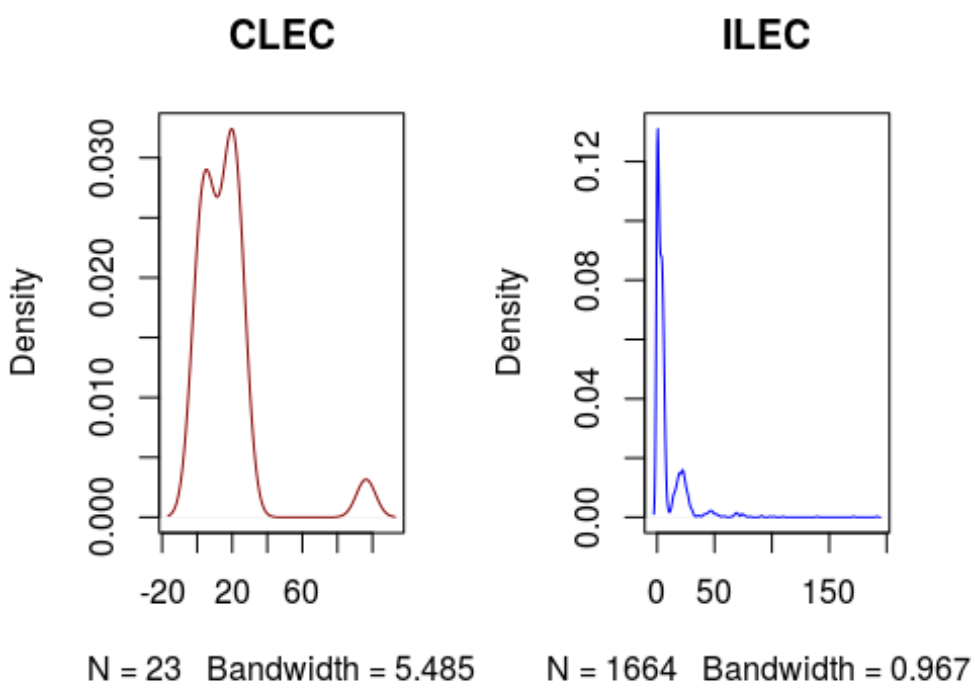
```
CLEC <- Verizon[Verizon$Group == "CLEC", "Time"]
ILEC <- Verizon[Verizon$Group == "ILEC", "Time"]
summary(CLEC)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   5.425  14.330  16.510  20.720  96.320
```

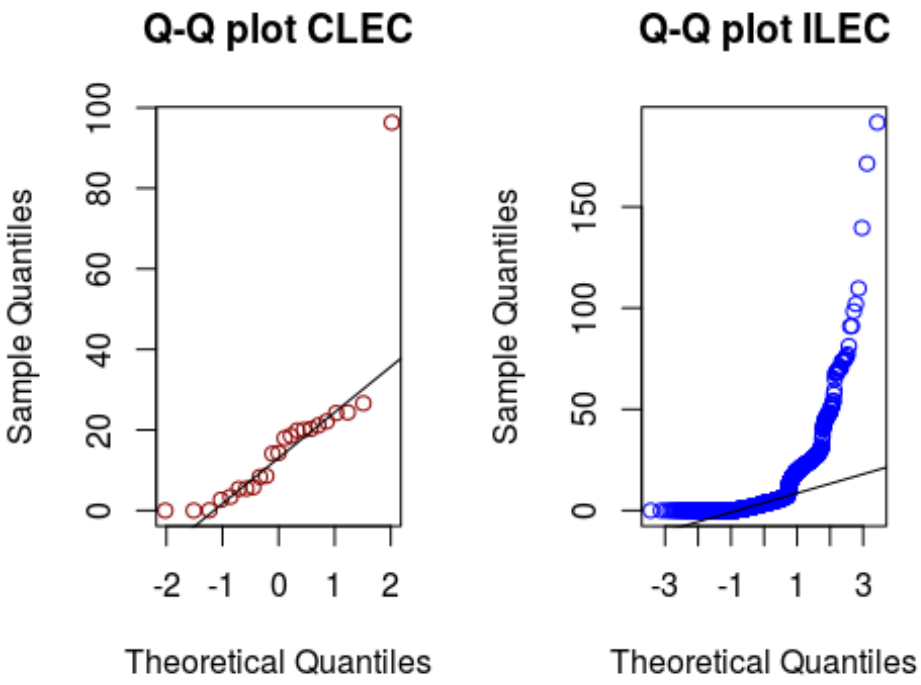
```
summary(ILEC)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.730   3.590   8.412   7.080 191.600
```

```
par(mfrow = c(1, 2))
plot(density(CLEC), main = "CLEC", col = "darkred")
plot(density(ILEC), main = "ILEC", col = "blue")
```




```
qqnorm(CLEC, col = "darkred", main = "Q-Q plot CLEC")
qqline(CLEC)
qqnorm(ILEC, col = "blue", main = "Q-Q plot ILEC")
qqline(ILEC)
```




```
par(mfrow = c(1, 2))
```

Los métodos paramétricos basados en el teorema del límite central, tales como los *t-test*, emplean como Error Estándar la ecuación $SE = \sqrt{\frac{sd_1^2}{n_1} + \frac{sd_2^2}{n_2}}$. Cuando las condiciones de normalidad no se cumplen, está aproximación no es valida. El *bootstrapping* permite obtener el valor de SE sin tener que recurrir al teorema del límite central, ya que la desviación estándar de la *bootstrapping distribution* se aproxima al SE de la sampling distribution. Es decir, mediante *bootstrapping* se pueden obtener los valores necesarios para calcular un intervalo de confianza.

La representación gráfica muestra una marcada asimetría de los datos, siendo además el tamaño de uno de los grupos de solo 23 observaciones. En estas condiciones, los métodos paramétricos no son recomendables.

```
aggregate(formula = Time ~ Group, data = Verizon, FUN = mean)
```

```
##   Group      Time
## 1  CLEC 16.509130
## 2  ILEC  8.411611
```

```
aggregate(formula = Time ~ Group, data = Verizon, FUN = sd)
```

```
##   Group     Time
## 1  CLEC 19.50358
## 2  ILEC 14.69004
```

La diferencia observada entre los tiempos de reparación de las dos muestras es de 8.0975199. El siguiente paso es generar un intervalo de confianza para la diferencia entre ambas poblaciones.

1.Cálculo de la bootstrapping distribution

Se obtienen nuevas pseudomuestras por separado de cada una de las muestras iniciales manteniendo el tamaño original y se calcula la diferencia.

```
boot_distribution <- rep(NA, 9999)
for (i in 1:9999) {
    boot_distribution[i] <- (mean(sample(x = ILEC, size = length(ILEC), replace =
                             TRUE))) - (mean(sample(x = CLEC, size = length(CLEC),
                             replace = TRUE)))
}
```

R contiene funciones que realizan el proceso de forma más eficiente:

```
require(bootES)
set.seed(5290)
boot_distribution <- bootES(data = Verizon, data.col = "Time", group.col = "Group",
                     contrast = c("CLEC", "ILEC"), R = 9999)
```

2.Estudio la bootstrapping distribution

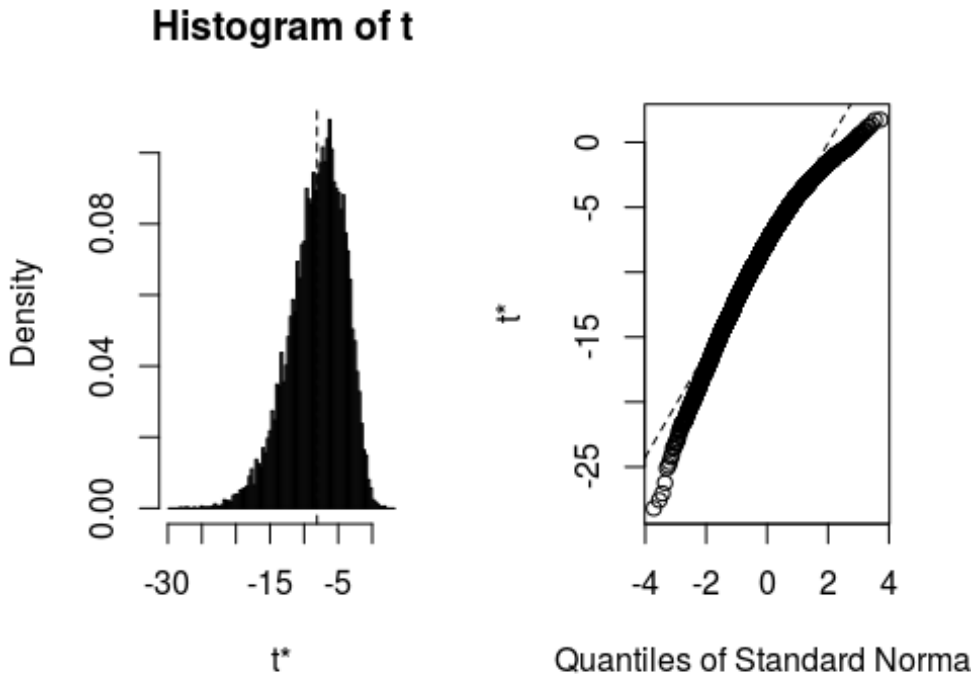
```
mean(boot_distribution$t)
```

```
## [1] -8.142566
```

```
sd(boot_distribution$t)
```

```
## [1] 4.029708
```

```
plot(boot_distribution)
```



Dado que la distribución no es normal, los intervalos basados en *t-distribution* o percentiles no son adecuados.

3.Intervalo de confianza

```
boot_distribution
```

```
##
## User-specified lambdas: (CLEC, ILEC)
## Scaled lambdas: (-1, 1)
## 95.00% bca Confidence Interval, 9999 replicates
## Stat        CI (Low)     CI (High)   bias        SE
## -8.098      -17.923      -1.996      -0.045      4.030
```

4.Tamaño del efecto

```
set.seed(5290)
bootES(data = Verizon, data.col = "Time", group.col = "Group", contrast = c("CLEC",
    "ILEC"), effect.type = "cohens.d", R = 9999)
```

```
##
## User-specified lambdas: (CLEC, ILEC)
## Scaled lambdas: (-1, 1)
## 95.00% bca Confidence Interval, 9999 replicates
## Stat        CI (Low)    CI (High)   bias        SE
## -0.549      -1.168      -0.110      -0.007      0.276
```

Bootstrapping para datos pareados

Un determinado estudio pretende determinar si existe relación entre la luna llena y la agresividad de las personas dementes. Para ello se mide con un test de comportamiento el nivel de agresividad de 15 pacientes los días de luna llena y los días sin luna llena. Obtener el intervalo de confianza del 95% para la verdadera diferencia entre las dos condiciones

Dado que se trata de un diseño muestral, el método adecuado es el bootstrapping.

```
datos <- data.frame(paciente = c(1:15), luna_llena = c(3.33, 3.67, 2.67, 3.33,
    3.33, 3.67, 4.67, 2.67, 6, 4.33, 3.33, 0.67, 1.33, 0.33, 2), sin_luna = c(0.27,
    0.59, 0.32, 0.19, 1.26, 0.11, 0.3, 0.4, 1.59, 0.6, 0.65, 0.69, 1.26, 0.23,
    0.38))
head(datos)
```

```
##   paciente luna_llena sin_luna
## 1        1       3.33     0.27
## 2        2       3.67     0.59
## 3        3       2.67     0.32
## 4        4       3.33     0.19
## 5        5       3.33     1.26
## 6        6       3.67     0.11
```

```
mean(datos$luna_llena)
```

```
## [1] 3.022
```

```
mean(datos$sin_luna)
```

```
## [1] 0.5893333
```

```
dif_obs <- mean(datos$luna_llena) - mean(datos$sin_luna)
dif_obs
```

```
## [1] 2.432667
```

Cuando se quiere emplear *bootstrapping* para estudiar la diferencia en el promedio de una variable bajo dos condiciones y los datos son pareados, al igual que ocurre con los t-test, se obtiene una nueva variable a partir de la diferencia entre las dos condiciones para cada sujeto.

```
datos$diff <- datos$luna_llena - datos$sin_luna
head(datos,4)
```

```
##   paciente luna_llena sin_luna diff
## 1        1       3.33     0.27 3.06
## 2        2       3.67     0.59 3.08
## 3        3       2.67     0.32 2.35
## 4        4       3.33     0.19 3.14
```

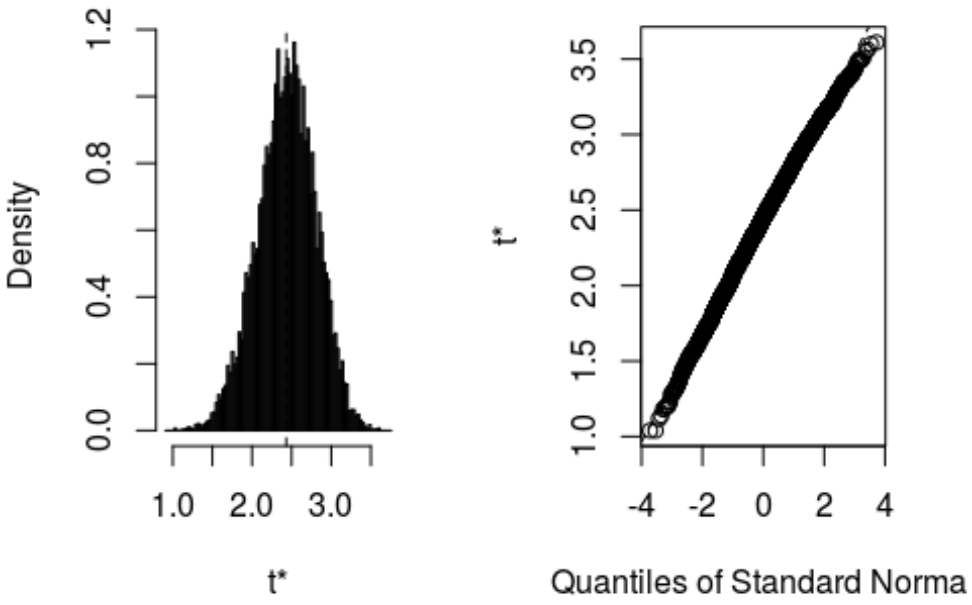
Una vez se dispone de esta nueva variable, el proceso es igual al seguido cuando se quiere realizar inferencia sobre el parámetro de una población (descrito previamente).

Mediante boot()

```
require(boot)
media <- function(valores, i) {
    mean(valores[i])
}
set.seed(5290)
boot_distribution <- boot(datos$diff, media, R = 9999)
boot_distribution
```

```
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
## Call:
## boot(data = datos$diff, statistic = media, R = 9999)
## Bootstrap Statistics :
##     original      bias    std. error
## t1* 2.432667 0.004540854   0.3674775
```

```
plot(boot_distribution)
```

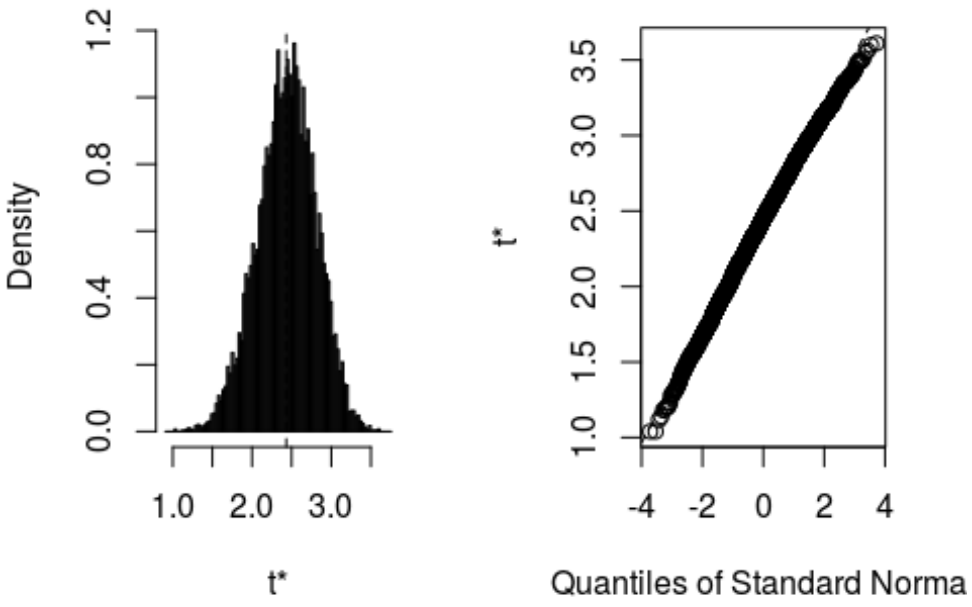



```
boot.ci(boot_distribution)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 9999 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_distribution)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.708,  3.148 )   ( 1.738,  3.177 )
##
## Level     Percentile            BCa
## 95%   ( 1.688,  3.127 )   ( 1.619,  3.073 )
## Calculations and Intervals on Original Scale
```

Mediante bootES()

```
set.seed(5290)
bootES(datos$diff, R = 9999, plot = TRUE)
```




```
## 95.00% bca Confidence Interval, 9999 replicates
## Stat        CI (Low)    CI (High)   bias        SE
## 2.433       1.619       3.073       0.005       0.367
```

Dado que el intervalo de confianza no contiene el valor 0, sí existe evidencia para un alpha de 0.05 de que el promedio de agresividad no es igual con y sin luna llena.

La función `bootES()` permite calcular también el tamaño del efecto:

```
set.seed(5290)
bootES(datos$diff, R = 9999, effect.type = "cohens.d")
```

```
##
## 95.00% bca Confidence Interval, 9999 replicates
## Stat        CI (Low)    CI (High)   bias        SE
## 1.666       0.947       2.707       0.164       0.571
```

Bootstrapping para coeficientes de correlación

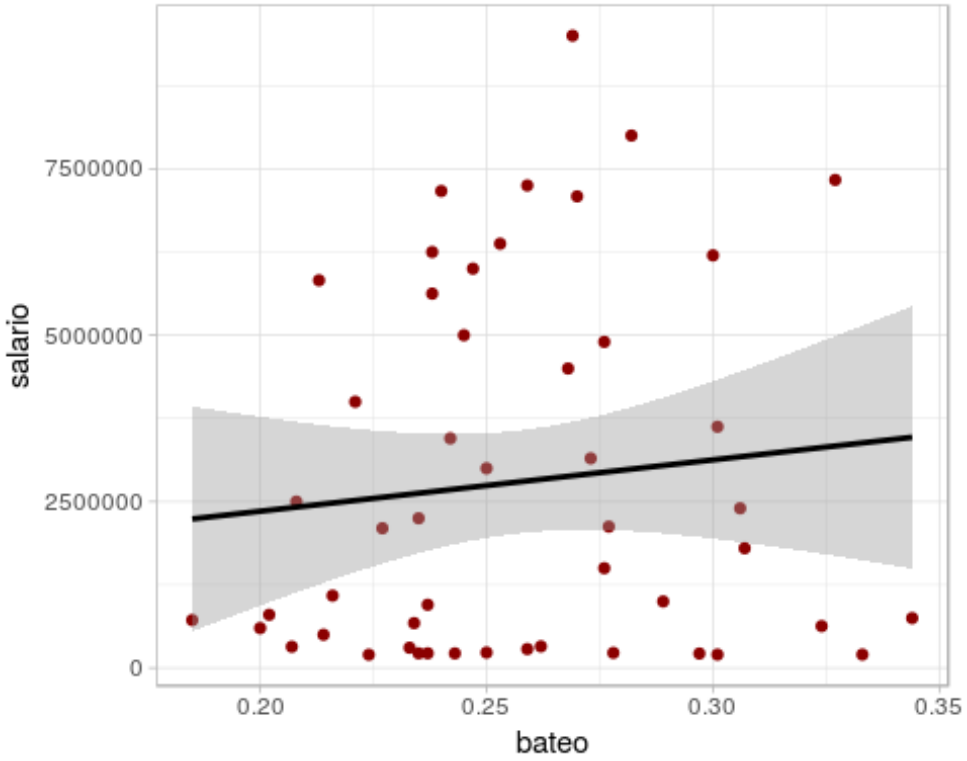
Cuando se estudia la correlación entre dos variables continuas, junto con el valor *R* de la correlación hay que calcular su intervalo de confianza o su significancia. Por muy alto que sea un coeficiente de correlación, si no es significativo, se tiene que considerar como inexistente la correlación puesto que podría deberse simplemente al azar.

Supóngase que se quiere saber si existe una relación entre el salario de los jugadores y su calidad (éxito al batear la pelota). Se dispone de una muestra de 50 jugadores seleccionados al azar de la liga. ¿Se puede considerar real la relación, en caso de que exista?

```
datos <- data.frame(jugador = c("Matt_Williams", "Jim_Thome", "Jim_Edmonds",
"Fred_McGriff", "Jermaine_Dye", "Edgar_Martinez", "Jeff_Cirillo", Rey_Ordonez",
"Edgardo_AlfonzoMoises_Alou", "Travis_Fryman", "Kevin_Young",
"M._GrudzielanekTony_Batista", "Fernando_Tatis", "Doug_Glanville", "Miguel_Tejada",
"Bill_Mueller", "Mark_McLemore", "Vinny_Castilla", "Brook_Fordyce", "Torii_Hunter",
"Michael_Tucker", "Eric_Chavez", "Aaron_Boone", "Greg_Colbrunn", "Dave_Martinez",
"Einar_Diaz", "Brian_L._HunterDavid_Ortiz", "Luis_Alicea", "Ron_Coomer",
"Enrique_Wilson", "Dave_Hansen", "Alfonso_SorianoKeith_Lockhart", "Mike_Mordecai",
"Julio_Lugo", "Mark_L._JohnsonJason_LaRue", "Doug_MientkiewiczJay_Gibbons",
"Corey_PattersonFelipe_Lopez", "Nick_Johnson", "Thomas_Wilson", "Dave_Roberts",
"Pablo_Ozuna", "Alexis_Sanchez", "Abraham_Nunez", "Corey_PattersonFelipe_Lopez",
"Nick_Johnson", "Thomas_Wilson", "Dave_Roberts", "Pablo_Ozuna", "Alexis_Sanchez",
"Abraham_Nunez"), salario = c(9500000, 8e+06, 7333333, 7250000, 7166667,
 7086668, 6375000, 6250000, 6200000, 6e+06, 5825000, 5625000, 5e+06, 4900000,
 4500000, 4e+06, 3625000, 3450000, 3150000, 3e+06, 2500000, 2400000, 2250000,
 2125000, 2100000, 1800000, 1500000, 1087500, 1e+06, 950000, 8e+05, 750000,
 720000, 675000, 630000, 6e+05, 5e+05, 325000, 320000, 305000, 285000, 232500,
 227500, 221000, 220650, 220000, 217500, 202000, 202000, 2e+05), bateo = c(0.269,
 0.282, 0.327, 0.259, 0.24, 0.27, 0.253, 0.238, 0.3, 0.247, 0.213, 0.238,
 0.245, 0.276, 0.268, 0.221, 0.301, 0.242, 0.273, 0.25, 0.208, 0.306, 0.235,
 0.277, 0.227, 0.307, 0.276, 0.216, 0.289, 0.237, 0.202, 0.344, 0.185, 0.234,
 0.324, 0.2, 0.214, 0.262, 0.207, 0.233, 0.259, 0.25, 0.278, 0.237, 0.235,
 0.243, 0.297, 0.333, 0.301, 0.224))
head(datos)
```

```
##          jugador salario bateo
## 1  Matt_Williams 9500000 0.269
## 2      Jim_Thome 8000000 0.282
## 3    Jim_Edmonds 7333333 0.327
## 4   Fred_McGriff 7250000 0.259
## 5   Jermaine_Dye 7166667 0.240
## 6 Edgar_Martinez 7086668 0.270
```

```
ggplot(data = datos, mapping = aes(x = bateo, y = salario)) +
 geom_point(colour = "darkred") +
 geom_smooth(method = "lm", colour = "black") +
 theme_light()
```

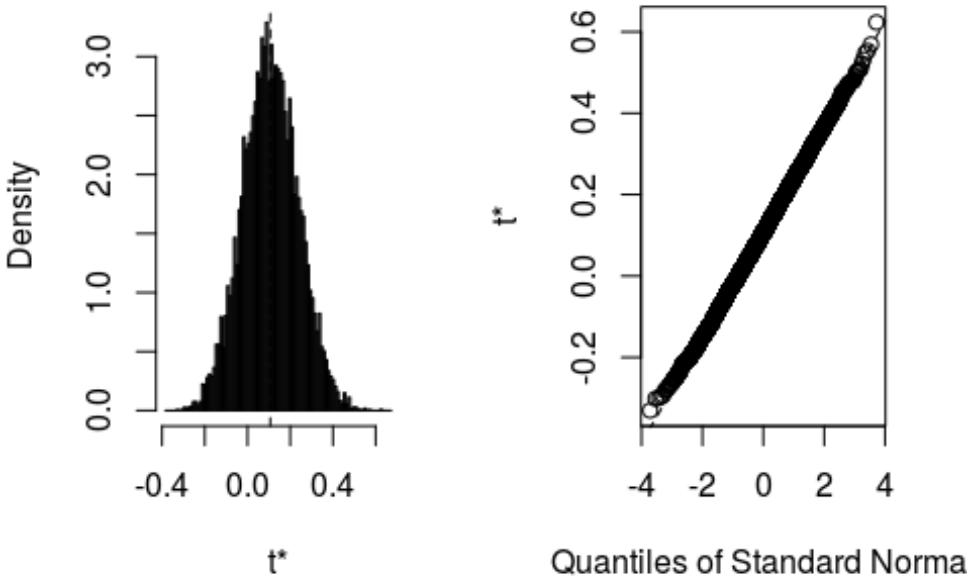



```
cor(x = datos$bateo, y = datos$salario)
```

```
## [1] 0.1067575
```

La representación de los datos y el cálculo de la correlación lineal muestran un r = 0.107.

```
bootES(data = datos[c("bateo", "salario")], R = 9999, ci.type = "bca", plot = TRUE)
```




```
##
## 95.00% bca Confidence Interval, 9999 replicates
## Stat        CI (Low)    CI (High)   bias        SE
## 0.107       -0.144      0.363       0.001       0.129
```

Dado que el intervalo de confianza contiene el valor 0, no se puede considerar significativa la correlación para un α de 0.05.

Bootstrapping para variables cualitativas (proporciones)

A la hora de comparar variables cualitativas (proporciones), si se cumplen una serie de condiciones, se pueden emplear métodos basados en el teorema del límite central.

- Independencia: El muestreo es aleatorio y el tamaño de la muestra es menor que el 10% de la población.
- Tamaño y distribución: Debe haber al menos 10 eventos verdaderos y 10 eventos falsos esperados dada la hipótesis nula. Esto implica que en el cálculo hay que emplear el *null value* de p establecido en H_0. $np_0 >= 10$, $n(1 - p_0) >= 10$.

Si estas condiciones no se satisfacen, la aproximación mediante el teorema del límite central no es válida. En estos casos es posible calcular el *p-value* mediante *bootstrapping*.

Ejemplo

El porcentaje de complicaciones que ocurren en una prueba clínica con el instrumental que tiene un hospital es del 10%. Una compañía considera que su instrumental consigue minimizar el riesgo de complicaciones argumentando que solo 3 intervenciones de las 62 realizadas han tenido complicaciones. ¿Hay evidencias de que sea cierto que su instrumental es mejor, utilizando un límite de significancia del 5%?

Hipótesis

$\boldsymbol{H_0}$: No existe una asociación entre el nuevo instrumental y el indice de complicaciones, p del nuevo instrumental es igual al del instrumento actual, $p = 0.1$.

$\boldsymbol{H_a}$: Con el nuevo instrumental se consigue reducir el porcentaje de complicaciones $p < 0.1$.

Estadístico

Se calcula el valor de la proporción en la muestra $\hat{p} = 3/62 = 0.0483871$

Condiciones para solucionarlo mediante el teorema del límite central

Independencia: El muestreo es aleatorio y el tamaño de la muestra es menor que el 10% de la población.

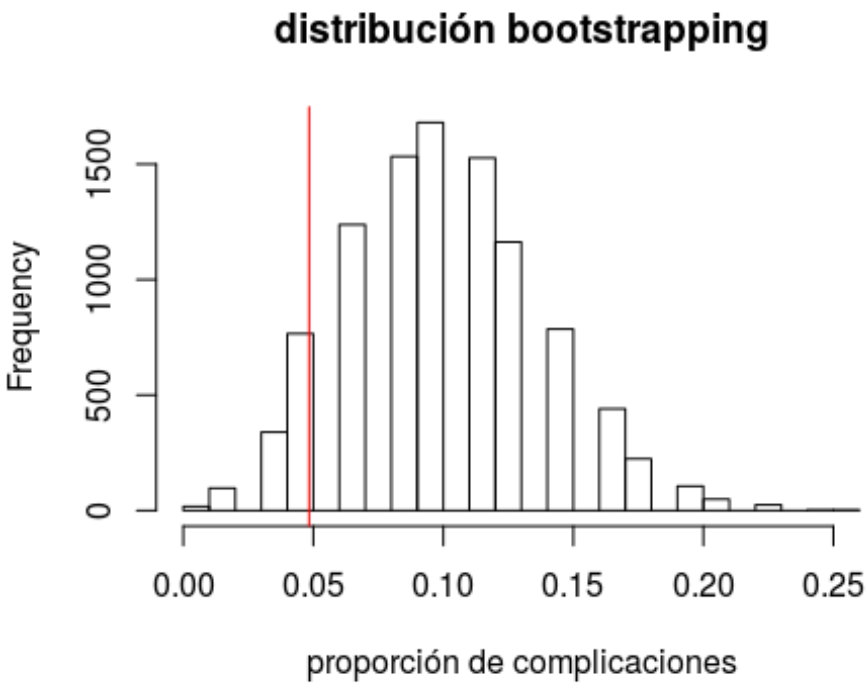
Tamaño y distribución: Debe haber en la muestra al menos 10 eventos verdaderos y 10 eventos falsos dada la hipótesis nula.

- np_0 = 62 x 0.1 = 6.2 *no se cumple la condición*
- $n(1 - p_0)$ = 62 x 0.9 = 55.8

Cálculo de *p-value*

Al no cumplirse las condiciones, no se puede considerar que $\hat{p}$ se distribuye de forma normal. Aun así, para poder saber cual es la probabilidad de que siendo Ho cierta ($p = 0.1$) de 62 eventos solo 3 o menos sean positivos, es decir, de que $\hat{p}$ = 0.0483871 se pueden simular muestreos con estas condiciones generando la distribución bootstrapping.

```
boot_distribution <- rep(NA, 9999)
# La muestra tiene un 10% de true. Esto equivale a la Ho
muestra_original <- c(rep(TRUE, 10), rep(FALSE, 90))
for (i in 1:9999) {
    boot_muestra <- sample(muestra_original, 62, replace = TRUE)
    # hacer la media de un vector lógico es como calcular la proporción de True
    boot_distribution[i] <- mean(boot_muestra)
}
hist(boot_distribution, breaks = 30, main = "distribución bootstrapping", xlab =
"proporción de complicaciones")
abline(v = 0.0483871, col = "red")
```



p-value = (número de observaciones de $\hat{p}$ <= 0.0483871) / (número total observaciones) = 0.1221122.

Conclusión

Dado que *p-value* es mayor que alpha, no hay evidencias suficientes para considerar que los resultados que muestra la compañía no se puedan obtener por fluctuaciones aleatorias siendo la verdadera probabilidad de complicaciones del 10%.

Test de permutaciones con simulación de montecarlo para variables cualitativas (proporciones)

Un estudio quiere determinar si un fármaco reduce el riesgo de muerte tras una intervención del corazón. Se diseña un experimento en el que pacientes que tienen que ser operados se distribuyen aleatoriamente en dos grupos (control y tratamiento). En vista de los resultados, ¿Se puede decir que el fármaco es efectivo? Nivel de significancia del 5%.

grupo	vivo	muerto	total
control	11	39	50
tratamiento	14	26	40
----------	----	------	-----
total	25	65	90

Hipótesis

$\boldsymbol{H_0}$: El porcentaje de supervivencia es independiente del tratamiento, la proporción de vivos es la misma en ambos grupos, *p(control) - p(tratamiento) = 0*.

$\boldsymbol{H_a}$: El porcentaje de supervivencia es dependiente del tratamiento, *p(control) - p(tratamiento) != 0*.

Estadístico

$\hat{p}\,(control)$ - $\hat{p}\,(tratamiento)$ = 11/50 - 14/40 = -0.13

Cálculo de p_{pool}

Siendo $p_{pool} = \frac{total\ positivos}{total\ eventos} = \frac{positivos1+positivos2}{n1+n2}$

p_{pool} = (11 + 14)/(50+40) = 0.278

Condiciones para solucionarlo mediante el teorema del límite central

Independencia:

- Dentro de cada muestra: El muestreo debe ser aleatorio y el tamaño de la muestra menor que el 10% de la población.
- Entre grupos/muestras: los datos no pueden ser pareados.

Tamaño y distribución

$n_1 p_{pool}$ & $n_1(1 - p_{pool})$ = 13.9, 36.1 >= 10 Al menos 10 eventos positivos y negativos esperados en la muestra 1.

$n_2 p_{pool}$ & $n_{2(}1 - p_{pool})$= 11.12, 28.88 >= 10 Al menos 10 eventos positivos y negativos esperados en la muestra 2.

A pesar de que las condiciones se cumplen, están muy cerca del límite. La aproximación por teorema del límite central no es del todo precisa.

Cálculo de *p-value*

Mediante *bootstrapping* se puede obtener la distribución muestral de *p(control) - p(tratamiento)*. Acorde a la H_0, la probabilidad de supervivencia es la misma en ambos grupos, para simular esto, se redistribuyen aleatoriamente (manteniendo el tamaño de cada grupo) los sujetos y se calcula la diferencia de proporciones. De los 90 individuos hay 50 control y 40 tratados.

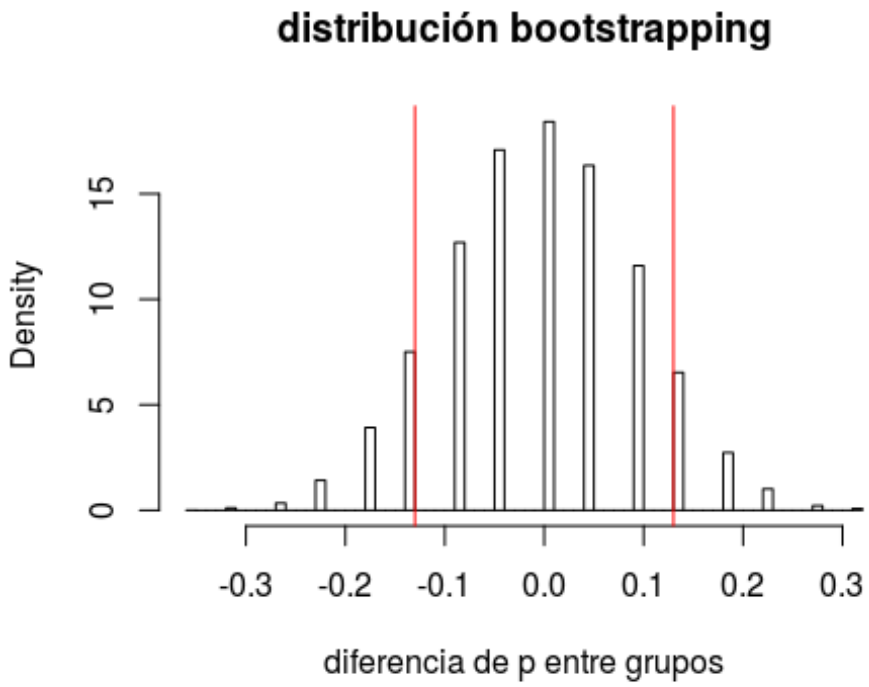
```
total_individuos <- c(rep(TRUE, 25), rep(FALSE, 65))
posicion_individuos <- c(1:90)
permutation_distribution <- rep(NA, 9999)

for (i in 1:9999) {
    a <- sample(posicion_individuos, 50)
    control <- total_individuos[a]
    # El resto de individuos va al grupo tratamiento
    b <- posicion_individuos[!(posicion_individuos %in% a)]
    tratamiento <- total_individuos[b]

    # La media de un vector lógico es la proporcion de TRUEs
    permutation_distribution[i] <- mean(control) - mean(tratamiento)
```

```
}

hist(permutation_distribution, breaks = 50, freq = FALSE, main = "distribución
bootstrapping",
    xlab = "diferencia de p entre grupos")
abline(v = -0.13, col = "red")
abline(v = 0.13, col = "red")
```



p-value = número de observaciones de -0.13 <= p => 0.13 dividido por el número total observaciones = 0.1637164

Conclusión

Dado que *p-value* es mayor que alpha. No hay evidencias suficientes para considerar que los resultados muestren una relación entre el tratamiento y la proporción de supervivientes.

Bibliografía

Comparing groups Randomization and Bootstrap Methods using R. Andrew S.Zieffler

Bootstrap Methods and Permutation Test by Tim Hestemberg

Points of Significance: Sampling distributions and bootstrap, Nature Methods

The RBook Michael J Crawley