Ejemplosbootstrap_I

Pedro Luque

```
##############################################
##ESTADISTICA COMPUTACIONAL I.               #
##GRADO EN ESTADISTICA                       #
##DOBLE GRADO EN MATEMATICAS Y ESTADISTICA   #
##EJEMPLOS BOOTSTRAP  (I)                    #
##############################################
source("ananor.r")   #Para analizar la normalidad


#####################################################
#PREVIO: cómo generar muestras bootstrap
#####################################################
#i) Muestras univariantes
#-----------------------
set.seed(123)
x=rnorm(5)
x
```

```
## [1] -0.56047565 -0.23017749  1.55870831  0.07050839  0.12928774
```

```
#Se trata de generar una m.a.s. de los elementos de x,
#con reemplazmiento
#o sea x*=(x1*....x5*) donde las xi* son iid según x
#tres posibles muestras
sample(x)
```

```
## [1]  0.07050839 -0.23017749  0.12928774 -0.56047565  1.55870831
```

```
sample(x,rep=TRUE)
```

```
## [1]  1.5587083  0.1292877  1.5587083  1.5587083 -0.5604756
```

```
sample(x,rep=TRUE)
```

Valores repetidos

```
## [1]  0.07050839 -0.56047565 -0.56047565  0.12928774  1.55870831
```

```
sample(x,rep=TRUE)
```

```
## [1] -0.23017749 -0.23017749 -0.56047565  1.55870831  0.07050839
```

```
#En este caso habría 5^5=3125 muestras bootstrap posibles
#Se observa que en una muestra bootstrap usualmente habrá
#casos repetidos y casos que no aparecen
```

```
#ii) Muestras multivariantes
#---------------------------
#hay que seleccionar casos completos, por ejemplo seleccionando posiciones
#supongamos el siguiente conjunto de datos
```

```r
data(attitude)
datasub=attitude[1:10,1:4]
n=nrow(datasub)
datasub
```

```
##    rating complaints privileges learning
## 1      43         51         30       39
## 2      63         64         51       54
## 3      71         70         68       69
## 4      61         63         45       47
## 5      81         78         56       66
## 6      43         55         49       44
## 7      58         67         42       56
## 8      71         75         50       55
## 9      72         82         72       67
## 10     67         61         45       47
```

```r
datasub[sample(n,replace=TRUE),]
```

```
##      rating complaints privileges learning
## 1        43         51         30       39
## 7        58         67         42       56
## 5        81         78         56       66
## 10       67         61         45       47
## 7.1      58         67         42       56
## 9        72         82         72       67
## 9.1      72         82         72       67
## 10.1     67         61         45       47
## 7.2      58         67         42       56
## 5.1      81         78         56       66
```

Podemos generar tantas muestras como queramos

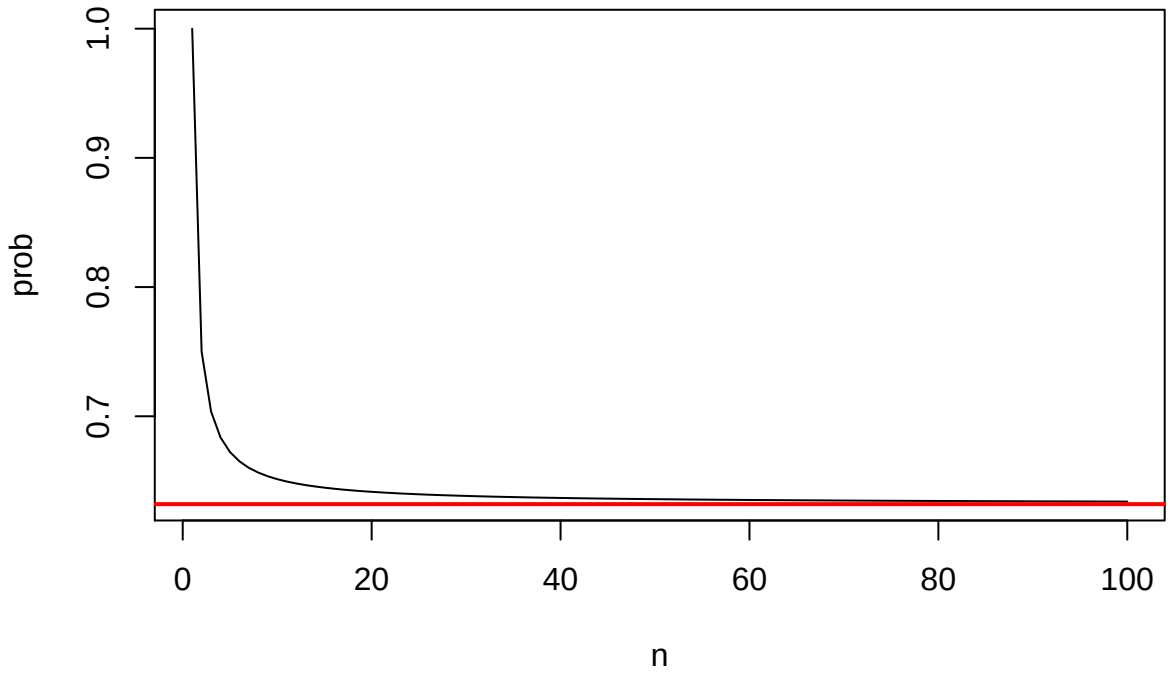
```r
#Las filas que aparecen con .1, .2, significa que aparecen repetidas
#en la muestra bootstrap

#iii) Probabilidad de que un caso pertenezca a una muestra
#     bootstrap= 1-(1-1/n)^n, tiende a 0.632
```

caso = fila

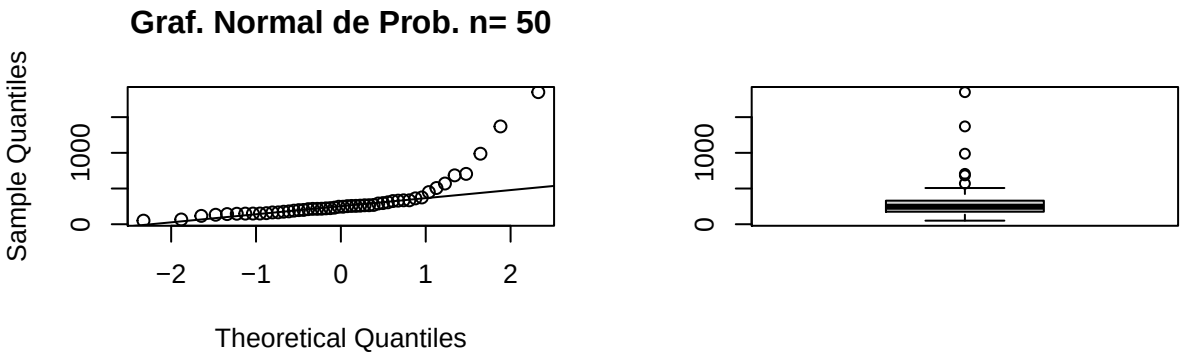
Para ver esto: Es el complementaario

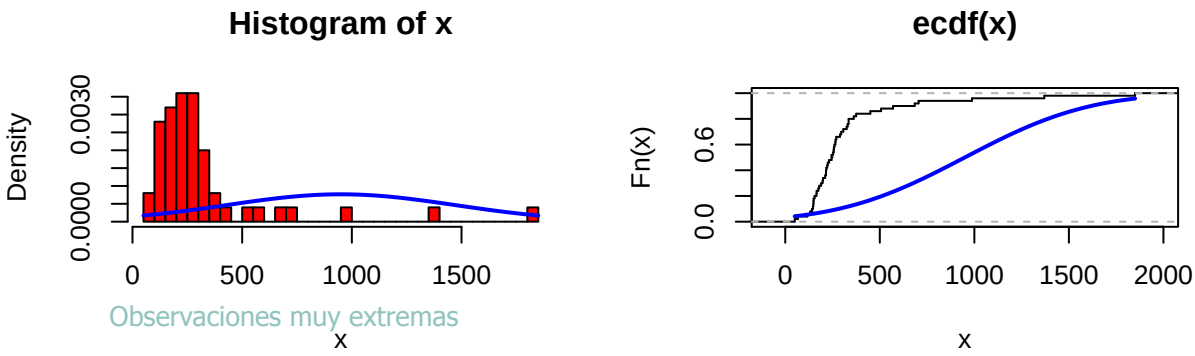
```r
n=seq(1,100,1)
prob=1-(1-1/n)^n

plot(n,prob,type="l")
abline(h=0.632,col="red",lwd=2)
```

A medida de que n crece, la probabilidad baja, pero nunca será menor que 0,632




```
#####################################################
#Ejemplo 1.
#Precios de venta (miles de euros) de viviendas
#en cierto lugar
#Estimador: media recortada
#Parámetro: Media poblacional
#Objetivo:  Estimar el sesgo y la varianza de este
#           estimador
#####################################################
x=c(142,175,197.5,149.4,705,232,50,146.5,155,1850,
    132.5,215,116.7,244.9,290,200,260,449.9,66.407,
    164.95,362,307,266,166,375,244.95,210.95,265,296,335,
    335,1370,256,148.5,987.5,324.5,215.5,684.5,270,330,
    222,179.8,257,252.95,149.95,225,217,570,507,190)

ananor(x)
```





Observaciones muy extremas

No parece que tenga comportamiento normal

```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.60636, p-value = 2.424e-10
```

```
#La presencia de outliers afecta sensiblemente a
#la media muestral
#Alternativa: media recortada
```

Para evitar valores extremos

```
mean(x)
```

```
## [1] 329.2571
```

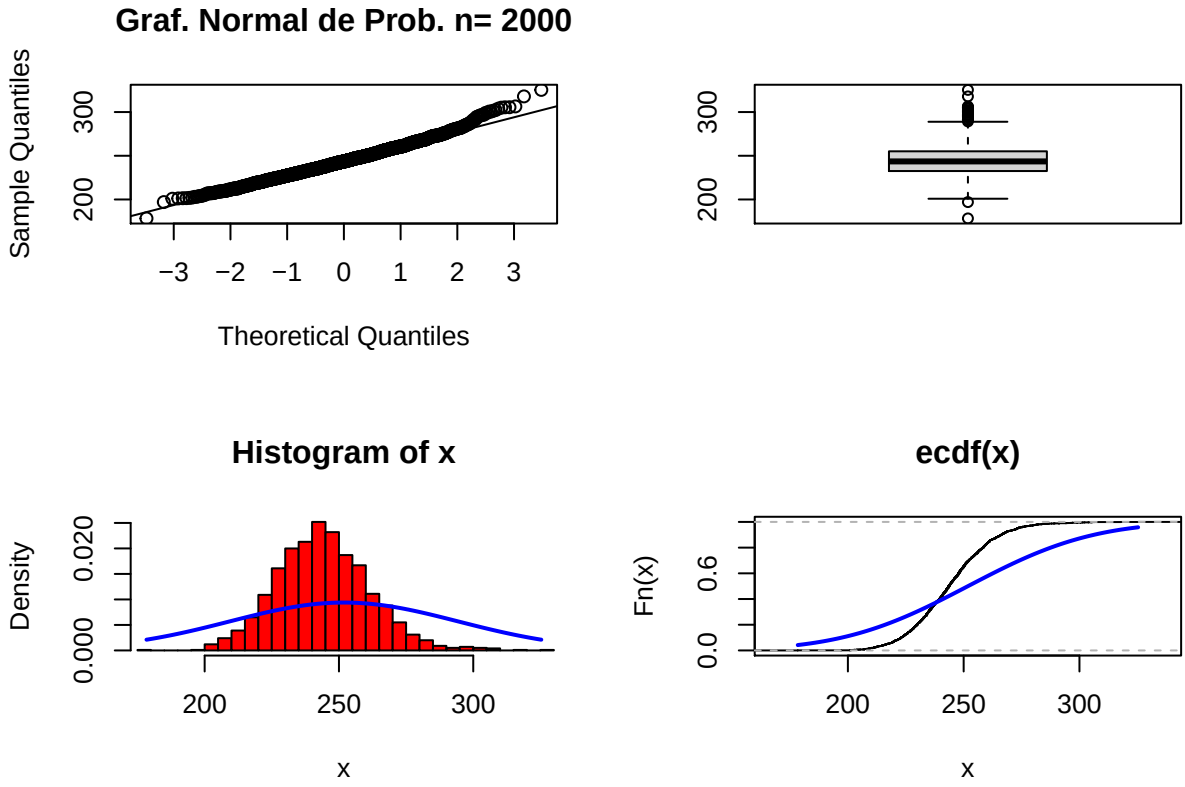
```
(medrec0= mean(x,trim=0.25))
```

Descarta el 25% de los valores por arriba y por abajo

```
## [1] 244.0019
```

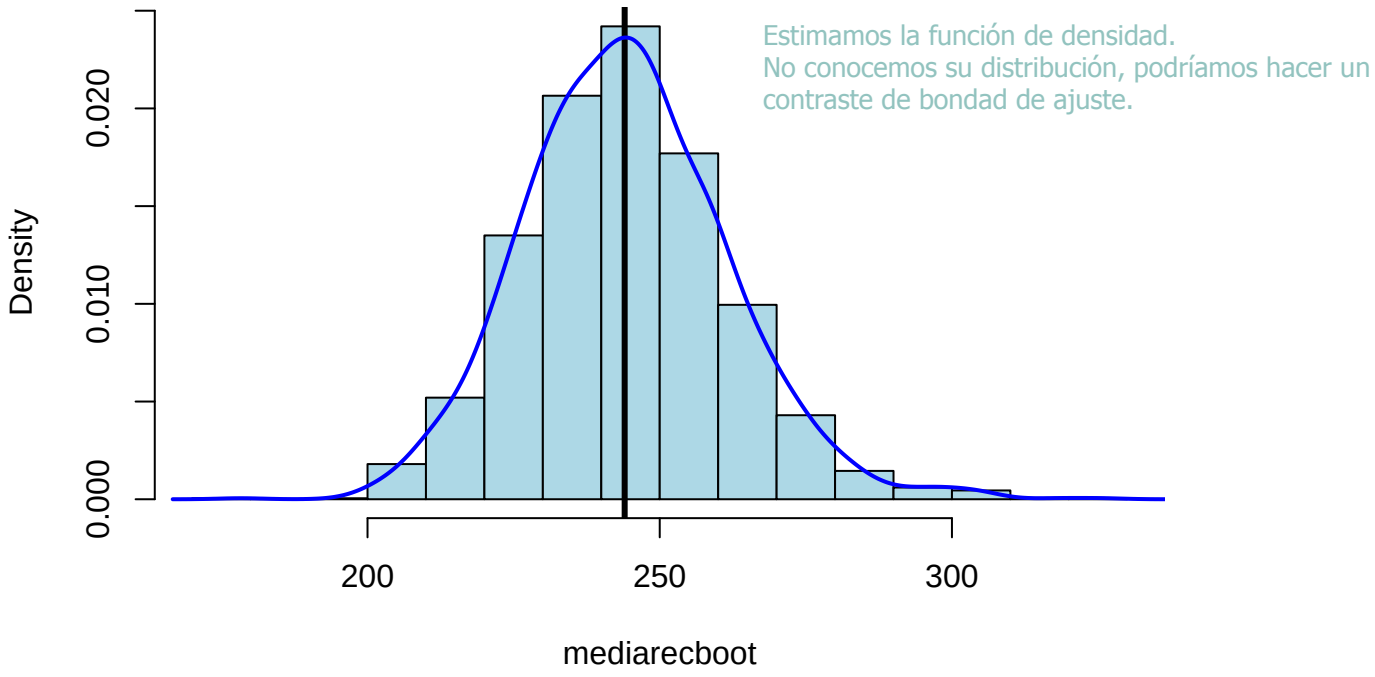
```
#se descartan el 25% de valores mayores
#y el 25% de valores menores

B=2000
#En el siguiente vector se guardarán los B valores
#del estadístico bootstrap
mediarecboot= numeric(B)
for (b in 1:B)
{   xboot= sample(x,replace=TRUE)
    mediarecboot[b]= mean(xboot,trim=0.25)
}  #siguiente b
```

```
#Suele ser de interés estudiar la forma de la distribución bootstrap
#sobre todo cuando vayamos a calcular IC
ananor(mediarecboot) #no parece que siga una distribución normal
```




```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.99146, p-value = 1.937e-09
```

```
hist(mediarecboot,br=20,prob=T,
     main="Distribuc. bootstrap de la media recortada 0.25",
     col="lightblue")
abline(v=medrec0,lwd=3)
lines(density(mediarecboot,bw="SJ"),col="blue",lwd=2)
```



Estimamos la función de densidad.
No conocemos su distribución, podríamos hacer un contraste de bondad de ajuste.

```
#Estimación del Sesgo:
sesgob=mean(mediarecboot)-  medrec0
sesgob
```

```
## [1] 0.187225
```

Estimación del sesgo pequeña

```
#Sesgo bajo comparado con medrec0
#Al igual que en el Jackknife se puede corregir la
#estimación
medrec0  - sesgob
```

```
## [1] 243.8147
```

```
#ES:
varBoot= var(mediarecboot); varBoot
```

```
## [1] 300.7658
```

```
ESBoot= sd(mediarecboot); ESBoot
```

```
## [1] 17.3426
```

```
##########################################
##Ejemplo 2. Estudiar mediante el bootstrap
##el Recorrido intercuartílico
##########################################

#Una función que devuelva el RI a partir
#de una muestra x
RI= function(x)
{
  Cuartiles= summary(x)[c(2,5)]
  res= Cuartiles[2]-Cuartiles[1]
```

Rango intercuartílico de la muestra

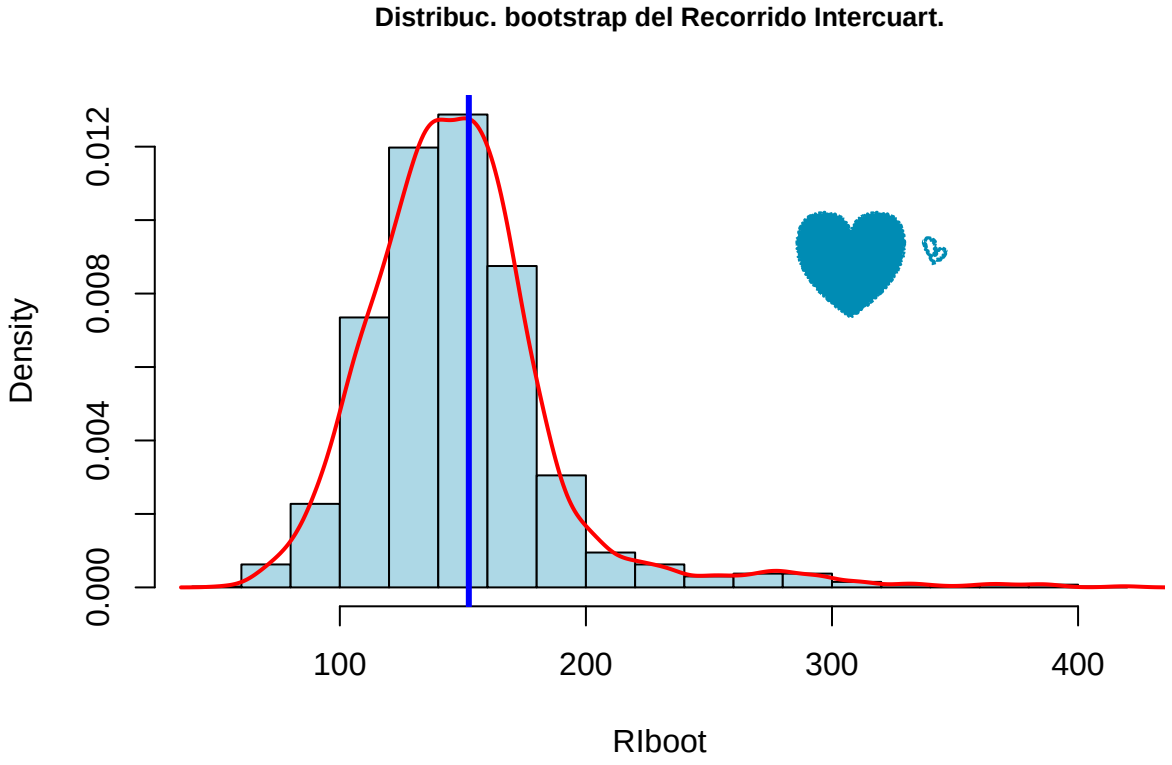
```
  names(res)= "R.I."
  res
}

RI0=RI(x); RI0
```

Rango Intercuartílico de la muestra Original

```
##    R.I.
## 152.425
```

```
B=2000
RIboot= numeric(B)
for (b in 1:B)
{   xboot= sample(x,replace=TRUE)
    RIboot[b]= RI(xboot)
}  #siguiente b
hist(RIboot,br=20,prob=TRUE,
     main="Distribuc. bootstrap del Recorrido Intercuart.",
     col="lightblue",cex.main=0.8)
lines(density(RIboot,bw="SJ"),col="red",lwd=2)
abline(v=RI0,lwd=3,col="blue")
```

Remuestreamos sobre esa muestra original que habíamos seleccionado de los datos del ejercicio 1




```
# sesgo
sesgob=mean(RIboot)-  RI0
#Estimador corregido:
RI0-sesgob
```

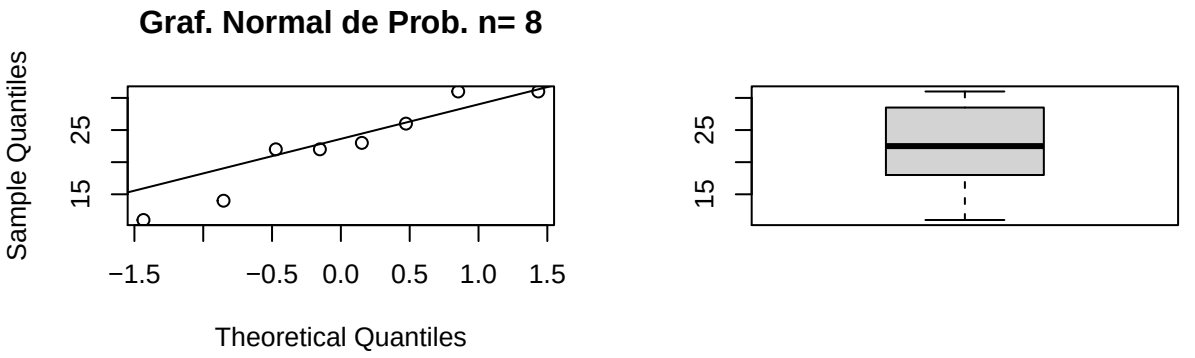
```
##     R.I.
## 156.5924
```

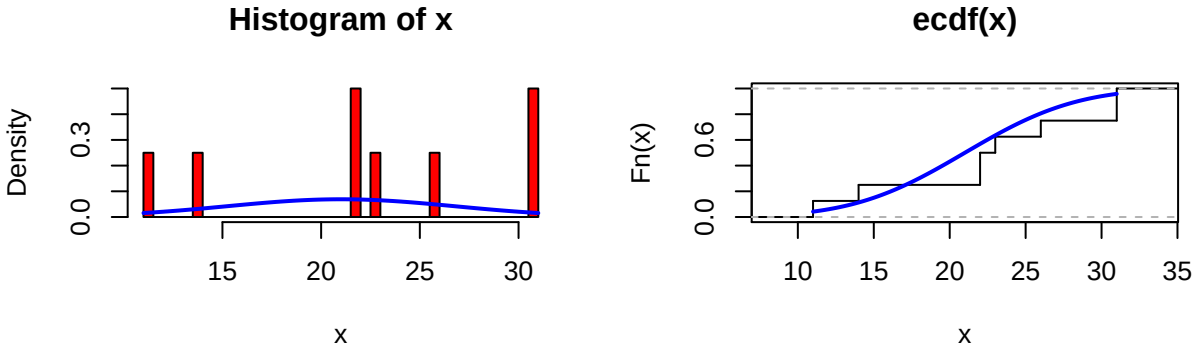
```
#ES
ESBoot= sd(RIboot); ESBoot
```

Error estandar ó cuasidesviación típica de las muestras obtenidas

```
## [1] 39.28767
```

```
#####################################
#Ejemplo 3. Cantidad de vitaminas en 8
#lotes de una mezcla de maíz y soja
#Un contraste de hipótesis mediante el
#bootstrap
#####################################
x=c(26, 31, 23, 22, 11, 22, 14, 31)
ananor(x)
```






```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.91858, p-value = 0.4184
```

```
#según Shapiro-Wilk no se rechaza la normalidad
#pero lo vamos a hacer con el bootstrap

#interesa que la media sea superior a 18, por
#lo que se plantea el siguiente contraste
#H0: E[X]<=18, H1:E[X]>18
#Vamos a usar el estadístico del test-t
t.test(x,mu=18)$statistic
```

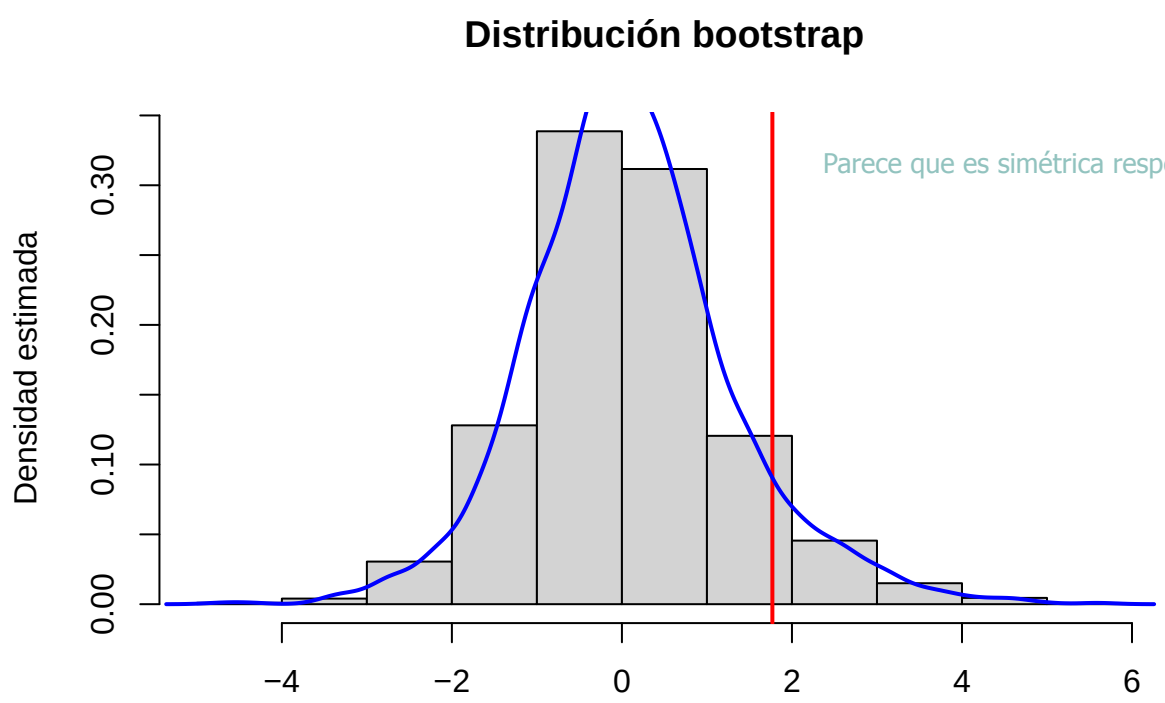
```
##        t
## 1.769914
```

```
#o sea:
(mean(x)-18)*sqrt(length(x))/sd(x)
```

```
## [1] 1.769914
```

```
t0= t.test(x,mu=18)$statistic
B=1999
#Aquí van los B valores del estadístico bootstrap:
taste= numeric(B)
for (b in 1:B)
{
  xboot= sample(x,rep=T)   media de la muestra original
  taste[b]= t.test(xboot,mu=mean(x))$statistic  Estadístico no usando mu=18 de H0
}
#Importante: el estadístico bootstrap compara la media
#de la muestra bootstrap con la media de la muestra disponible
#t.test(xboot,mu=mean(x))  no se pone mu=18
#en las transparencias y el script Bootstrap_ilustrarConsejos.r
#se ilustra este hecho  Mejora el resultado

hist(taste,prob=T,main="Distribución bootstrap",
     xlab="",ylab="Densidad estimada",col="lightgrey")
abline(v=t0,col="red",lwd=2)
lines(density(taste,bw="SJ"),col="blue",lwd=2) Estimación de la función de densidad con el método del núcleo SJ
```

Si la distribución es normal, el estadístico se tiene que distribuir según una t-student



Parece que es simétrica respecto a 0

```
#Calcular la estimación del p valor
cat("P[T*>=T0]~",(sum(taste>t0)+1)/(B+1),"\n")  En cuantas muestras el estadístico bootstrap es mayor que t0
```

```
## P[T*>=T0]~ 0.083
```

→ 8 3 %

```
#Contraste paramétrico
t.test(x,mu=18,alternative="greater")  No tenemos pruebas para concluir que la media es mayor que 18
```

```
##
##  One Sample t-test
##
## data:  x
```

```
## t = 1.7699, df = 7, p-value = 0.06003
## alternative hypothesis: true mean is greater than 18
## 95 percent confidence interval:
##  17.68304      Inf
## sample estimates:
## mean of x
##      22.5
```

```
#Para el contraste bilateral, el p-valor aproximado
#se calcularía como sigue
```

Calor absoluto de las obs a la derecha del rojo y a la izquierda del valor rojo negativo

```
cat("P[|T*|>=|T0|]~",(sum(abs(taste)>=abs(t0))+1)/(B+1),"\n")
```

```
## P[|T*|>=|T0|]~ 0.1265
```

No rechazo el contraste

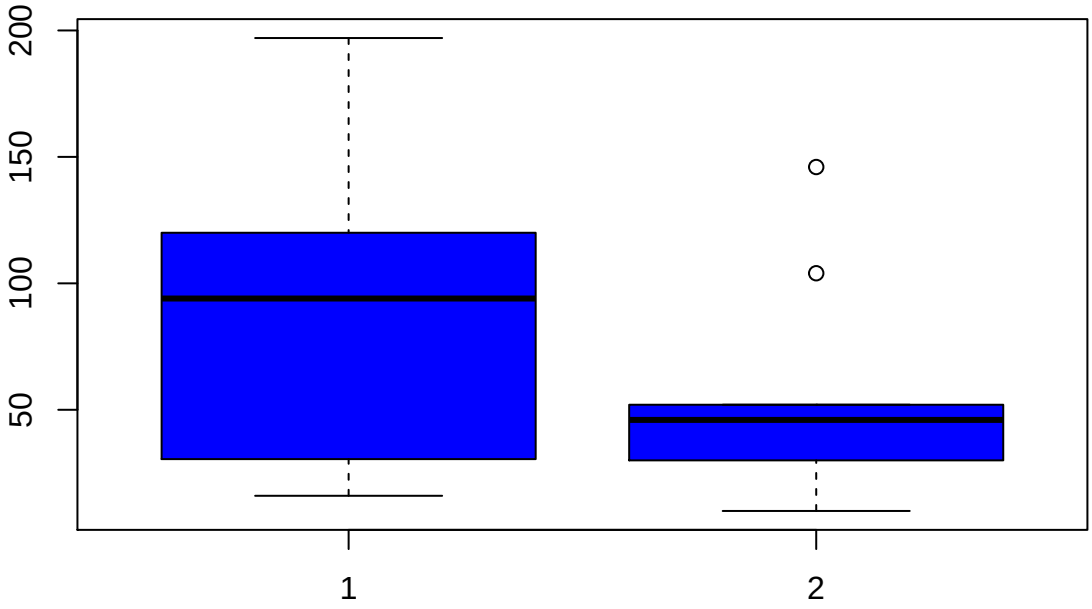
```
##########################################
##Ejemplo 4.Contraste bootstrap bilateral
##comparando las medias de dos poblaciones
##########################################
x=c(94,38,23,197,99,16,141)
y=c(52,10,40,104,51,27,146,30,46)
nx=length(x)
ny=length(y)
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.00   30.50   94.00   86.86  120.00  197.00
```

```
summary(y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.00   30.00   46.00   56.22   52.00  146.00
```

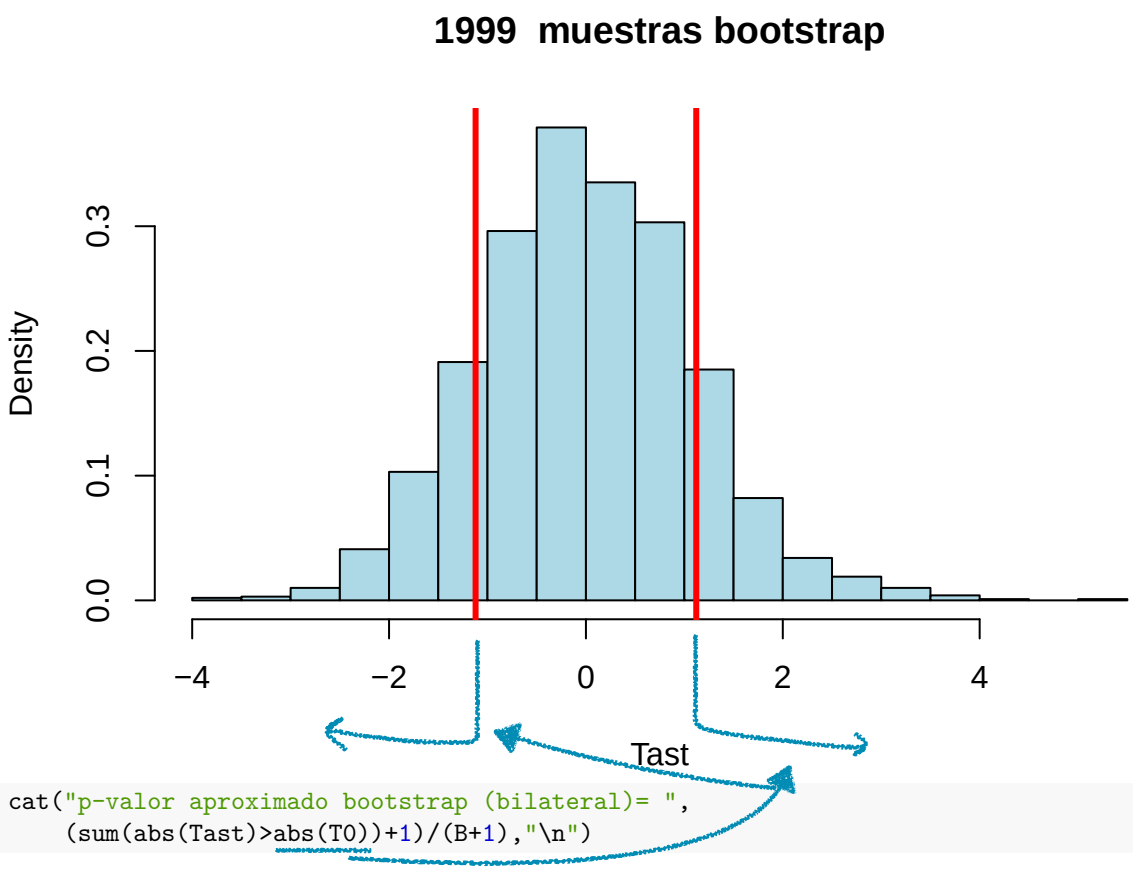
```
boxplot(x,y,col="blue")
```




```
B=1999
T0=t.test(x,y,var.equal=TRUE)$statistic
Tast=numeric(B)
```

Supongo hipótesis de homocedasteceidad, y ambas distribuciones prodecen de la misma distribución por lo que ambas varianzas han de ser iguales. Hago el t-test, por lo que obtengo el estadístico t-student

```
#las muestras bootstrap se forman con elementos
#extraídos de forma indistinta de cualquiera de
#las submuestras originales
xy=c(x,y)
for (b in 1:B)
{xast=sample(xy,nx,replace=TRUE) #Muestreo bajo HO
 yast=sample(xy,ny,replace=TRUE) #Muestreo bajo HO
 Tast[b]=t.test(xast,yast,var.equal=TRUE)$statistic
}

hist(Tast,br=30,prob=TRUE,
     main=paste(B," muestras bootstrap"),
     col="lightblue")
abline(v=T0, lwd=3,col="red")
abline(v=-T0, lwd=3,col="red")
```




```
cat("p-valor aproximado bootstrap (bilateral)= ",
    (sum(abs(Tast)>abs(T0))+1)/(B+1),"\n")
```

```
## p-valor aproximado bootstrap (bilateral)=  0.285
```

No rechazo

```
##########################################
##Ejemplo 5. Intervalo de confianza
##para la diferencia de medias de dos poblaciones
#mismos datos del ejemplo anterior
##########################################
x=c(94,38,23,197,99,16,141)
y=c(52,10,40,104,51,27,146,30,46)
nx=length(x)
ny=length(y)
```

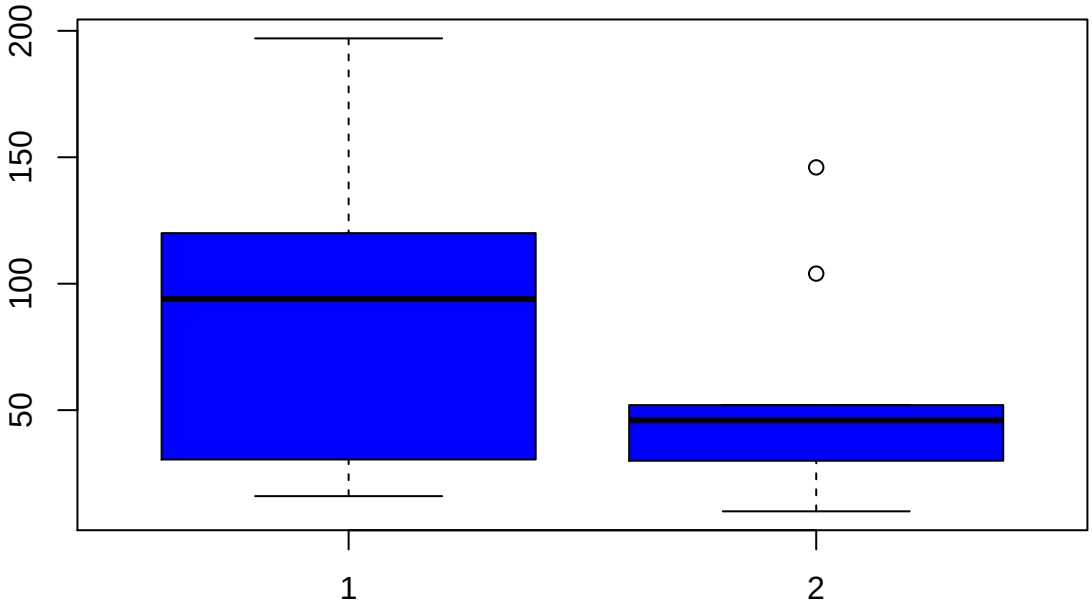
```
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.00   30.50   94.00   86.86  120.00  197.00
```

```
summary(y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.00   30.00   46.00   56.22   52.00  146.00
```

```
boxplot(x,y,col="blue")
```




```
B=2000
dife0=mean(x)-mean(y)
dife0
```

```
## [1] 30.63492
```

La de x es bastante mayor que la de y (30ud)

```
difeast=numeric(B)

#Para calcular IC, como no hay una hipótesis nula
#H0, se muestrea por separado
for (b in 1:B)
{xast=sample(x,replace=TRUE) #Muestra boot de x
 yast=sample(y,replace=TRUE) #Muestra boot de y
 difeast[b]=mean(xast)-mean(yast)
}

hist(difeast,br=30,prob=TRUE,
     main=paste(B," muestras bootstrap"),
     xlab="diferencia de medias",
     col="lightblue")
#Si la forma de la distribución boostrap es razonablemente
#normal, se pueden usar el método normal y/o el método percentil
#para calcular IC

#Método percentil, simplemente los cuantiles de difeast
```

No tengo hipótesis, sólo quiero IC

Calculo el estadístico para cada una de las muestras

```
alfa= 0.05
ICperc=quantile(difeast,prob=c(alfa/2,1-alfa/2))
cat("ICBootstrap (Percentil) para la diferencia de medias al 95% :\n(",
   ICperc[1],",",
  ICperc[2],")\n")
```

Método percentil: calculo cuantiles

```
## ICBootstrap (Percentil) para la diferencia de medias al 95% :
## ( -18.38611 , 83.63611 )
```

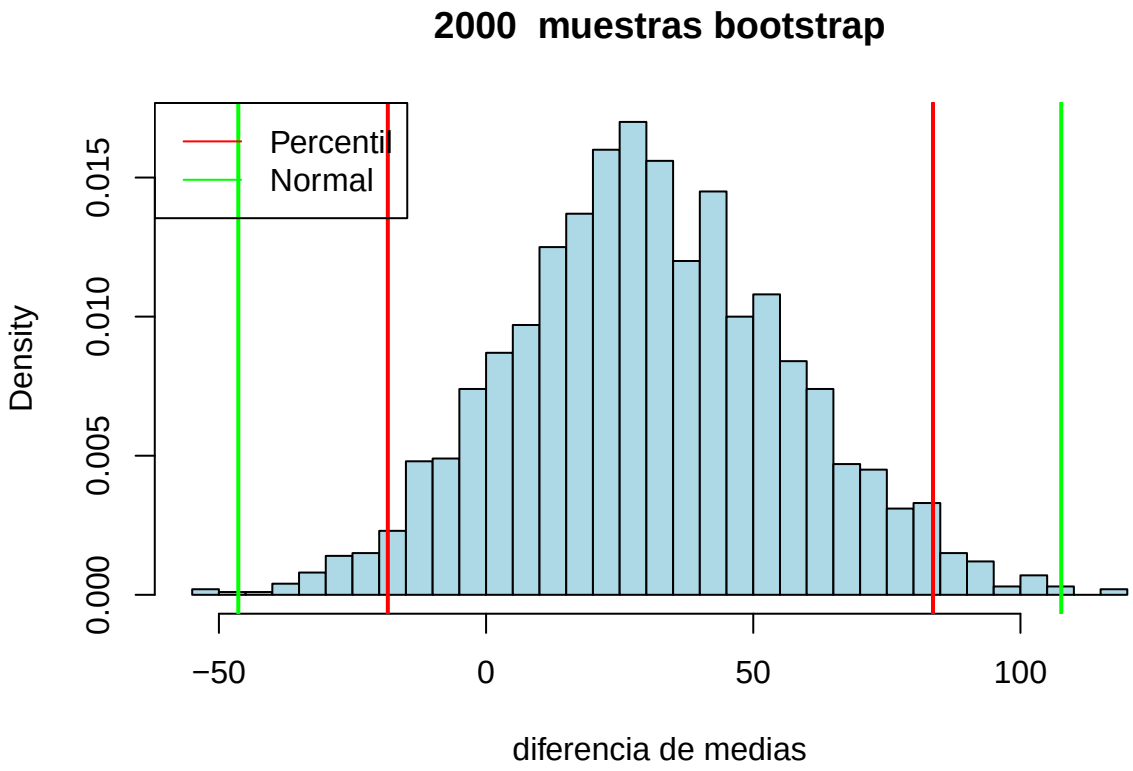
Ojo, es una errata y coge de otro ejercicio
calcular ESBoot = sd(dife0)= 27.03379

```
#Método normal
cuantil= qnorm(1-alfa/2)
ICnormal=c(dife0- cuantil*ESBoot,dife0+ cuantil*ESBoot)
cat("ICBootstrap (Normal) para la diferencia de medias al 95% :\n(",
    ICnormal[1],",",
    ICnormal[2],")\n")
```

```
## ICBootstrap (Normal) para la diferencia de medias al 95% :
## ( -46.3675 , 107.6373 )
```

(-22,83)

```
abline(v=ICperc,col="red",lwd=2)
abline(v=ICnormal,col="green",lwd=2)

legend("topleft",col=c("red","green"),lty=1,
       legend=c("Percentil","Normal"))
```




```
#####################################################################
##Ejemplo 6. Una tabla de contingencia, contraste de independencia
##y cálculo de IC para la diferencia de probabilidades
#####################################################################
#    Infarto:    SI     NO    Total
# Fuman          12    12   | 24
```

```
# No Fuman        3    16   | 19
#H0:P[Infarto=SI/Fumar]=P[Infarto=SI/No Fuman]
#H1:P[Infarto=SI/Fumar]!=P[Infarto=SI/No Fuman]
fuman=24
fumaneinf=12
nofuman=19
nofumaneinf=3

RR=(fumaneinf/fuman)/(nofumaneinf/nofuman)
cat(" Frec. relativa de infartos en fumadores=",fumaneinf/fuman,"\n",
    "Frec. relativa de infartos en no fumadores=",nofumaneinf/nofuman,"\n",
    "Riesgo relativo de sufrir infarto Fumador/No fumador= ",RR,"\n")
```

```
##  Frec. relativa de infartos en fumadores= 0.5
##  Frec. relativa de infartos en no fumadores= 0.1578947
##  Riesgo relativo de sufrir infarto Fumador/No fumador=  3.166667
```

Casi un tercio la segunda probabilidad

```
#Construir una tabla con los datos
tabla= matrix(c(fumaneinf,fuman-fumaneinf,
                nofumaneinf,nofuman-nofumaneinf),byrow=T,2,2)
rownames(tabla)= c("Fuman","No Fuman")
colnames(tabla)=c("Infarto","No Infarto")
tabla
```

```
##          Infarto No Infarto
## Fuman         12         12
## No Fuman       3         16
```

La muestra de partida bootstrap es la tabla

```
100*prop.table(tabla,1)
```

```
##           Infarto No Infarto
## Fuman    50.00000   50.00000
## No Fuman 15.78947   84.21053
```

```
100*prop.table(tabla,2)
```

```
##          Infarto No Infarto
## Fuman         80   42.85714
## No Fuman      20   57.14286
```

```
#Se puede hacer el contraste de independencia con el test-chicuadrado
chisq.test(tabla)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tabla
## X-squared = 4.0616, df = 1, p-value = 0.04387
```

Con un nivel de sig del 5%, esto nos lleva a rechazar la independencia y la igualdad de probabilidades

```
#Si fallaran las reglas de Cochran
#en tablas 2x2:frecuencias esperadas >5
#podemos recurrir al bootstrap

#5.1 Contraste de independencia bootstrap
#----------------------------------------
#Para aplicar el bootstrap se puede elegir el estadistico del test
#chi-cuadrado; como depende de  los valores observados, que
```

```r
#son fijos, y de los esperados, que se calculan bajo H0,
#se puede muestrear sobre los valores esperados,
#que al dividir por n dan las probabilidades bajo H0
#Recordemos que el estadístico es la suma de las (obs-esp)^2/esp
#donde las frecuencias esperadas (bajo H0) están disponibles en
#el elemento expected

resul = chisq.test(tabla)
resul
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tabla
## X-squared = 4.0616, df = 1, p-value = 0.04387
```

```r
(p.hat = resul$expected / sum(tabla))
```

```
##            Infarto No Infarto
## Fuman    0.1946998  0.3634397
## No Fuman 0.1541374  0.2877231
```

Divido por n, el total de observaciones
y hallo las estimaciones de las probs conjuntas
de cada una de las celdas

```r
#Probabilidades conjuntas estimadas bajo H0
```

Son indepe

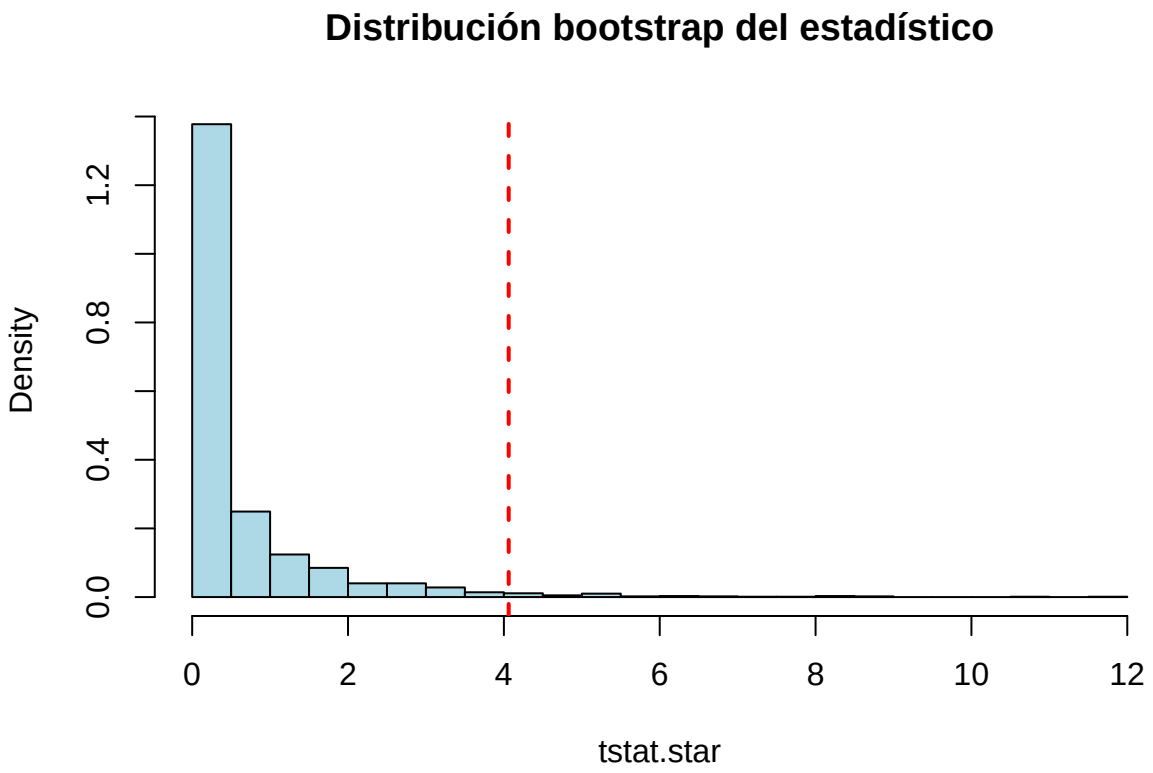
```r
#Estadístico calculado en la tabla original:
(tstat.hat = resul$statistic)
```

```
## X-squared
##  4.061616
```

```r
#Cada muestra bootstrap se puede obtener muestreando con
#reemplazamiento, con tamaño n, el conjunto de las cuatro posiciones
#de la tabla, cada posición con probabilidad la que aparece en p.hat
n = sum(tabla)
k = length(tabla)   #Número de celdas=nrow(x)*ncol(x)  2*2=4
B = 1999
tstat.star = numeric(B)   #Estadístico bootstrap
for (i in 1:B) {
  #Muestra bootstrap de la tabla conjunta, p.hat (bajo H0):
  y.star = sample(1:k, n, replace = TRUE,
                  prob = as.numeric(p.hat))
  #as.numeric(p.hat) pasa la matriz a un vector
  #recorriendo las columnas
  #Disponer como tabla la muestra y.star
  #el siguiente tabulate guarda un 0 para aquellas posiciones
  #que correspondan a elementos no observados en y.star
  #tabulate va a contar cuántas veces aparece el 1,2,3,4 (k)
  tabla.star = matrix(tabulate(y.star, k), 2, 2)
  suppressWarnings({resul.boot = chisq.test(tabla.star)})
  tstat.star[i]=resul.boot$statistic
  #podría dar errores si una fila o columna está vacía
}

cat("P-valor bootstrap =",
    (sum(tstat.star >= tstat.hat) + 1) / (B + 1),"\n")
```

```
## P-valor bootstrap = 0.0215
```

```
hist(tstat.star,br=20,prob=TRUE,
     col="lightblue",main="Distribución bootstrap del estadístico")
abline(v = tstat.hat, lty = 2,lwd=2,col="red")
```



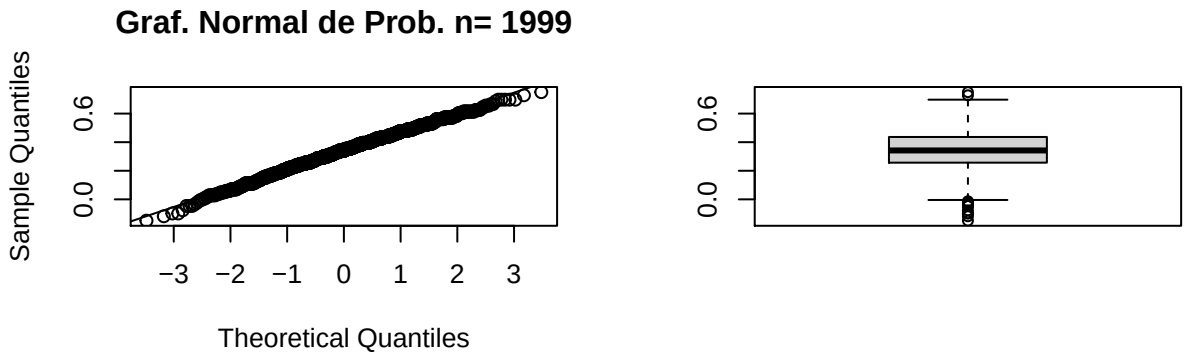

```
#5.2 Intervalos de Confianza boostrap
#--------------------------------------
#IC-bootstrap (normal y percentil) para la diferencia de probabilidades
#P[Infarto=SI/Fumar]-P[Infarto=SI/No Fuman]
#En este caso se muestrea por separado !!!
alfa=0.05
difeboot=numeric(B)
dife0= (fumaneinf/fuman)-(nofumaneinf/nofuman)
cat("Diferencia de proporciones observadas en la tabla=",dife0,"\n")
```

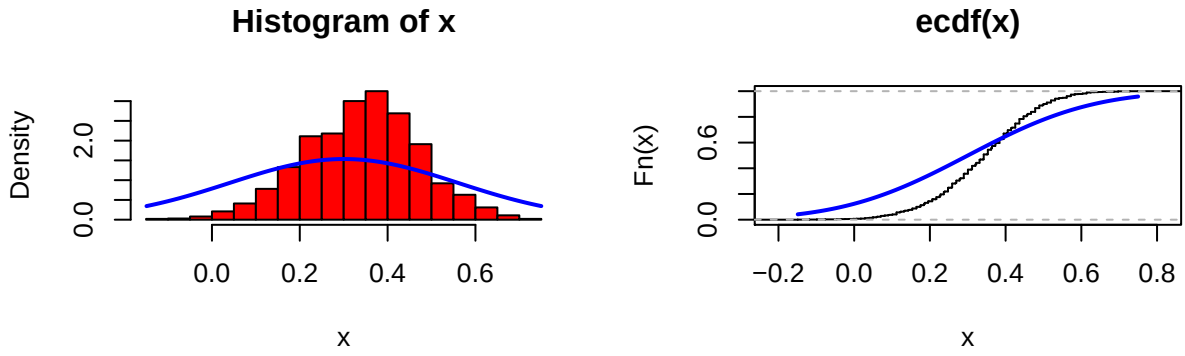
Método de muestreo es por separado, porque tengo por un lado los fumadores y por otro los no fumadores

```
## Diferencia de proporciones observadas en la tabla= 0.3421053
```

```
for (i in 1:B)
{
  #Se muestrea por separado, por ejemplo en los fumadores
  #se le ha asignado un número de orden entre 1 fuman
  #comparando con el total de fumaneinf obtenemos la proporción a
  a=sum(sample(1:fuman,rep=TRUE)<=fumaneinf)/fuman
  b=sum(sample(1:nofuman,rep=TRUE)<=nofumaneinf)/nofuman
  difeboot[i]= a-b
}
ananor(difeboot)
```

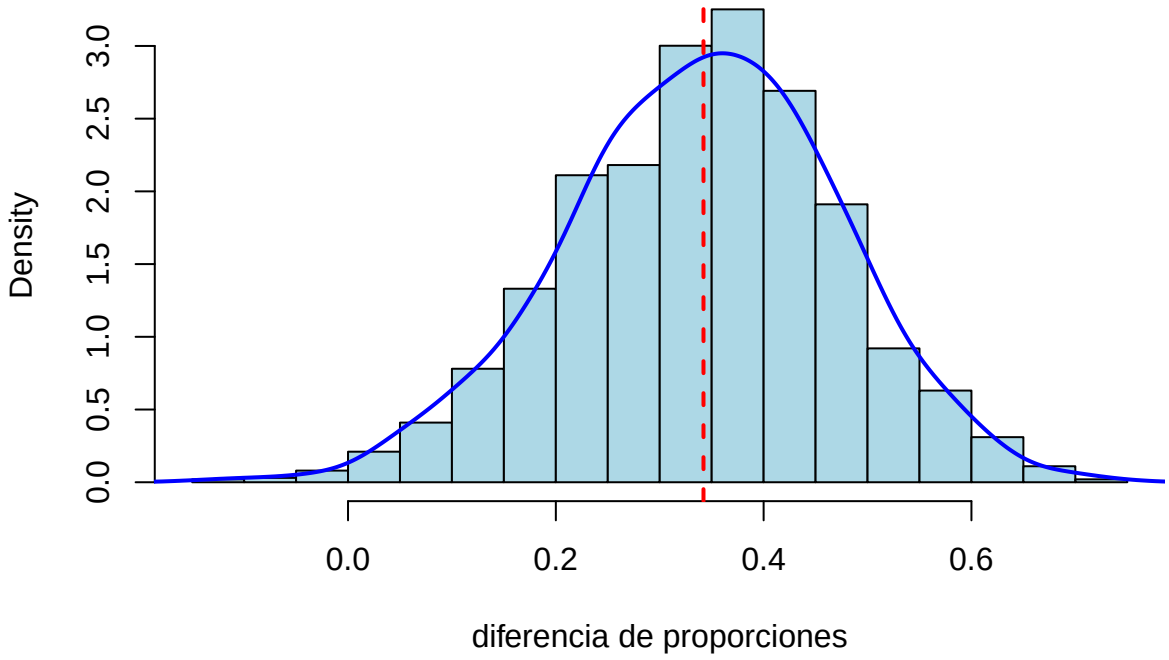
Genero un valor entre 1 y 24 con reemp y miro si es menor o igual que 12.
Cuento de la muestra cuantos han sido menor o igual q 12 (han tenido infarto) y divido por los que fuman: obtengo la frecuencia relativa.
12 es el valor frontera.






```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.99716, p-value = 0.001022
```

```
hist(difeboot,br=20,prob=TRUE,
     xlab="diferencia de proporciones",
     col="lightblue",main="Distribuc. bootstrap")
abline(v = dife0, lty = 2,lwd=2,col="red")
lines(density(difeboot,bw="SJ"),col="blue",lwd=2)
```




```
#Parece que hay ligera asimetría a la izquierda, se podría
#utilizar el método BCa que se verá en el script siguiente
cat(" Intervalo de confianza normal 95%= (",
    dife0-qnorm(1-alfa/2)*sd(difeboot), ", ",
    dife0+qnorm(1-alfa/2)*sd(difeboot),") \n",
    "Intervalo de confianza Percentil 95%= (",
    quantile(difeboot,probs=c(alfa/2,1-alfa/2)), ") \n")
```

Metodo percentil: quantiles de nivel alfa/2 y 1-alfa/2

```
##  Intervalo de confianza normal 95%= ( 0.08429555 ,  0.599915 )
##  Intervalo de confianza Percentil 95%= ( 0.07017544 0.5833333 )
```