

MÉTODOS DE REMUESTREO

Tema 3. Estimación de errores estándar mediante remuestreo

basado en

- B. Efron, R. Tibshirani (1993). An Introduction to the bootstrap.
- O. Kirchkamp (2017). Resampling methods.

Curso 2018/19

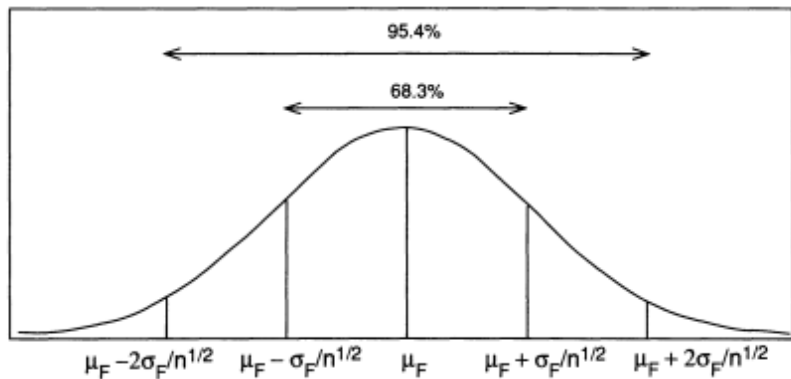
Errores estándar

- ▶ Los estadísticos muestrales se usan frecuentemente en Estadística, por ello es necesario conocer su precisión.
- ▶ El bootstrap permite encontrar el error estándar de los estadísticos basándose en el principio de *plug-in*.
- ▶ Supongamos una v.a. X con media μ_F y varianza σ_F^2
- ▶ Sea $\mathbf{x} = (x_1, x_2, \dots, x_n)$ una m.a.s. procedente de la distribución F .
- ▶ La media muestral

$$\bar{X} \sim \left(\mu_F, \frac{\sigma_F^2}{n} \right)$$

Cálculo de errores estándar mediante el TCL

- ▶ De este modo la esperanza de \bar{x} es la misma que la de la v.a. X original, pero la varianza es igual a $1/n$ veces la varianza de X .
- ▶ Así el error estándar de \bar{x} es simplemente $\frac{\sigma_F}{\sqrt{n}}$.
- ▶ En una distribución normal se espera que X sea menor que una vez la desviación estándar de μ_F aproximadamente el 68 % de las ocasiones y menor que dos desviaciones estándar alrededor del 95 % de las veces, aplicando el *TCL* (teorema central del límite).



Limitaciones del TCL

- ▶ La aproximación del TCL funciona bien cuando el tamaño muestral n es grande pero la aproximación tiene limitaciones.
- ▶ Supongamos que X sigue una distribución de Bernoulli:

$$P_F \{X = 1\} = p$$

$$P_F \{X = 0\} = 1 - p$$

- ▶ El parámetro p es la probabilidad de éxito que está entre 0 y 1

Limitaciones del TCL

- Una m.a.s. es una sucesión de unos y ceros de modo que la suma

$$s = \sum_{i=1}^n x_i \sim \text{Bin}(n, p)$$

- La media $\bar{x} = \frac{s}{n}$ es igual a \hat{p} que es el estimador *plug-in* de p , de modo que

$$\hat{p} \sim N\left(p, \frac{p(1-p)}{n}\right)$$

Limitaciones del TCL

- ▶ Se toma el ejemplo de una distribución binomial con $n = 25$ en los casos de $p = 0,25$ y $p = 0,9$.
- ▶ Para el caso de $p = 0,9$ la aproximación a la normal por el TCL **no** es muy buena.

```
n = 25  
  
p09 = rbinom(20000, n, 0.9)/n  
p025 = rbinom(20000, n, 0.25)/n
```

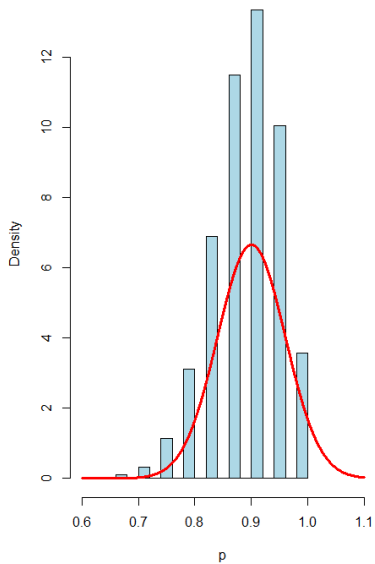
Limitaciones del TCL

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

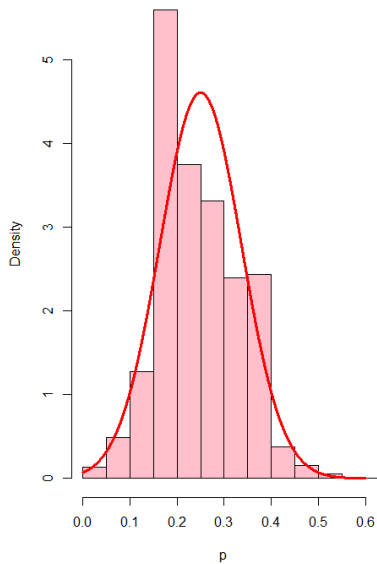
hist(p09, prob=T, xlim=c(0.6,1.1), col="lightblue",
     xlab="p", main="p = 0.9")
xs1 = seq(0.6,1.1,0.0001)
ys1 = dnorm(xs1, 0.9, sqrt((0.9*0.1)/n))
lines(xs1,ys1,lwd=3,col="red")

hist(p025, prob=T, xlim=c(0,0.6), col="pink",
     xlab="p", main="p = 0.25")
xs2 = seq(0,0.6,0.0001)
ys2 = dnorm(xs2, 0.25, sqrt((0.25*0.75)/n))
lines(xs2, ys2, lwd=3, col="red")
```


$p = 0.9$



$p = 0.25$



Bootstrap y errores estándar

- ▶ El bootstrap permite calcular errores estándar sin que tenga importancia lo complicado que sea el estimador que se considere.
- ▶ Los métodos bootstrap dependen del concepto de *muestra bootstrap*.
- ▶ Partimos de la función de distribución empírica \hat{F} que asigna probabilidad $1/n$ a cada uno de los elementos de la muestra observada.
- ▶ Una muestra bootstrap se define como una muestra aleatoria de tamaño n extraída de \hat{F}

$$\hat{F} \rightarrow \mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$$

- ▶ La notación *estrella* $*$ indica que \mathbf{x}^* no es el conjunto de datos original sino una versión *remuestreada* de la muestra original \mathbf{x} .

Bootstrap y errores estándar

- ▶ Alternativamente, se puede decir que una muestra bootstrap $x_1^*, x_2^*, \dots, x_n^*$ es una muestra aleatoria de tamaño n tomada **con reemplazamiento** de la muestra original (que hace el papel de *población*).
- ▶ El algoritmo se denomina **bootstrap no paramétrico** porque depende solo de la función de distribución empírica.
- ▶ Por ejemplo podríamos tener una muestra bootstrap como

$$x_1^* = x_7$$

$$x_2^* = x_3$$

$$x_3^* = x_3$$

$$\dots \quad \dots \quad \dots$$

$$x_n^* = x_2$$

Algoritmo Bootstrap y errores estándar

1. Seleccionar B muestras bootstrap $\mathbf{x}^{*1}, \mathbf{x}^{*2}, \dots, \mathbf{x}^{*B}$ cada una obtenida a partir de la muestra original \mathbf{x} con **reemplazamiento**.
2. Evaluar la réplica bootstrap en el estimador correspondiente

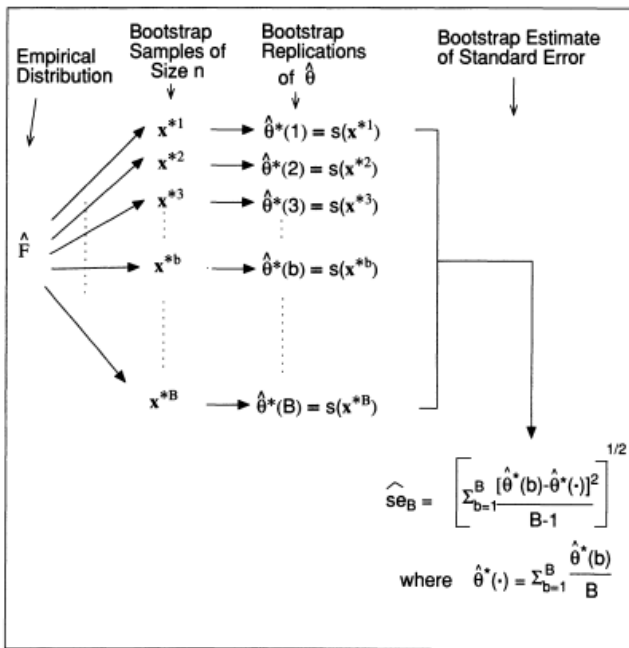
$$\hat{\theta}^*(b) = s(\mathbf{x}^{*b})$$

para $b = 1, 2, \dots, B$.

3. Estimar el error estándar $se_F(\hat{\theta})$ mediante

$$\hat{se}_B = \sqrt{\frac{1}{B-1} \sum_{b=1}^B \left(\hat{\theta}^*(b) - \hat{\theta}^*(\cdot) \right)^2}$$

$$\text{donde } \hat{\theta}^*(\cdot) = \frac{1}{B} \sum_{b=1}^B \hat{\theta}^*(b)$$



Ejemplo de los institutos de máster en leyes

- ▶ La correlación entre GPA y LSAT es

```
library(bootstrap)
(lawCor = with(law, cor(GPA, LSAT)))
```

```
[1] 0.7763745
```

- ▶ Pero ¿cómo es de preciso el estimador del coeficiente de correlación lineal?
- ▶ Si la distribución conjunta de ambas variables F es normal bivalente, entonces $\hat{\rho}$ (siguiendo a Efron&Tibshirani) tiene un error estándar igual a

$$\hat{\sigma}_{\hat{\rho}} = \frac{1 - \hat{\rho}^2}{\sqrt{n - 3}} \approx 0,115$$

Ejemplo de los institutos de máster en leyes

► Así,

```
(se = (1-lawCor^2)/sqrt(dim(law)[1]-3))
```

```
[1] 0.1146741
```

```
c((lawCor - 1.96*se), min(1, (lawCor + 1.96*se)))
```

```
[1] 0.5516133 1.00000
```

► Alternativamente se puede usar la librería psychometric:

```
psychometric::CIr(lawCor, dim(law)[1])
```

```
[1] 0.4385108 0.9219648
```

Ejemplo de los institutos de máster en leyes

- ▶ Usando bootstrap se puede **evitar** asumir que F se distribuye como una normal bivalente.

```
ssamplesize = dim(law)[1]
ind = 1:samplesize

law.boot =
replicate(1000, {indB = sample(ind,replace=TRUE);
with(law[indB,], cor(GPA,LSAT))})

sd(law.boot)
```

```
[1] 0.1336493
```


Error estándar del coeficiente de correlación

- ▶ Desde el punto de vista clásico, el error estándar del estimador de la correlación, se puede estimar así:

```
# Asumes SE_r = sqrt((1-r^2)/(n-2))

cor.test.plus = function(x) {
  list(x,
    Standard.Error =
      unname(sqrt((1 - x$estimate^2)/x$parameter)))
}
```

Error estándar del coeficiente de correlación

```
library(bootstrap)  
cor.test.plus(cor.test(law$GPA, law$LSAT))
```

```
Pearson's product-moment correlation  
  
data: law$GPA and law$LSAT  
t = 4.4413, df = 13, p-value = 0.0006651  
alternative hypothesis: true correlation is not equal to 0  
  
95 percent confidence interval:  
 0.4385108 0.9219648  
  
sample estimates:  
      cor  
0.7763745  
  
Standard.Error  
[1] 0.174806
```

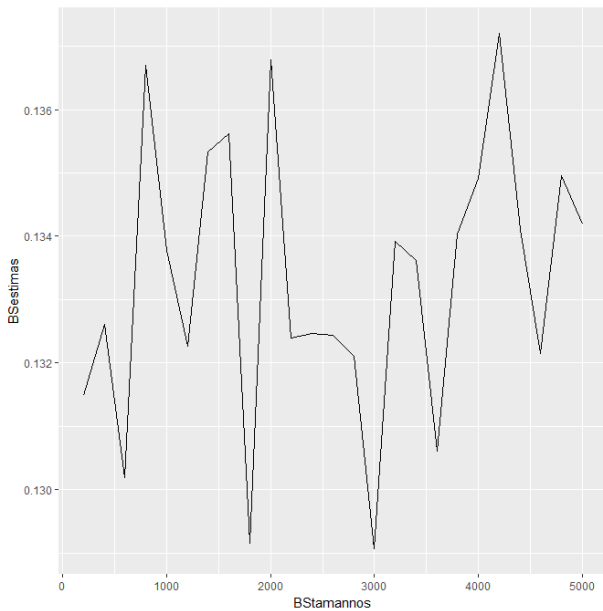
Ejemplo de los institutos de máster en leyes

- ¿Cómo converge de rápido el estimador bootstrap?

```
samplesize = dim(law)[1]
ind = 1:samplesize
lawBS = function(B) sd(replicate(B,
{indB = sample(ind,replace=TRUE);
with(law[indB,],cor(GPA,LSAT))}))

BStamannos = seq(200,5000,200)
BSestimas = sapply(BStamannos,lawBS)

library(ggplot2)
qplot(BStamannos, BSestimas, geom="path")
```



Ejemplo de los institutos de máster en leyes

- Se compara la distribución empírica muestral de la población $\hat{F}(\hat{\theta}^*)$ con la distribución poblacional $F(\hat{\theta}^*)$

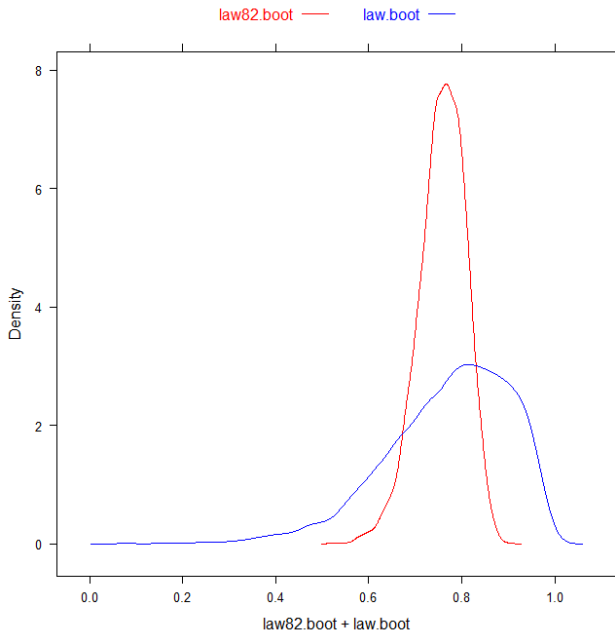
```
ind1 = 1:dim(law)[1]
law.boot = replicate(5000, {indB = sample(ind1,
size=length(ind1), replace=TRUE);
with(law[indB,], cor(GPA,LSAT))})

ind2 = 1:dim(law82)[1]
law82.boot = replicate(5000,{indB = sample(ind2,
size=length(ind2), replace=TRUE);
with(law82[indB,], cor(GPA,LSAT))})
```

Ejemplo de los institutos de máster en leyes

```
library(latticeExtra)

densityplot(~law82.boot + law.boot, plot.points=FALSE,
auto.key=list(columns=2,size=3,between=1,
col=c("red","blue")),
par.settings=list(superpose.line=
list(col=c("red","blue"))), col=c("red","blue"))
```



Ejemplo de los institutos de máster en leyes

- ▶ Se compara el estimador bootstrap del error estándar a partir de la muestra $se_{\hat{F}}(\hat{\rho})$ con respecto al error estándar a partir de la población $se_F(\hat{\rho})$

```
sd(law.boot)
```

```
[1] 0.1327579
```

```
sd(law82.boot)
```

```
[1] 0.1309014
```


Bootstrap Paramétrico

- ▶ En muchas ocasiones se tienen fórmulas analíticas para calcular los errores. En este caso se puede aplicar el bootstrap aprovechando que éstas se conocen.
- ▶ Se denomina a este tipo de remuestreo como **bootstrap paramétrico** y se define el estimador bootstrap del error estándar como

$$se_{\hat{F}_{par}}(\hat{\theta}^*)$$

- ▶ donde \hat{F}_{par} es un estimador de F que se obtiene a partir de un modelo paramétrico aplicado a los datos.

Bootstrap Paramétrico

- ▶ En el ejemplo [law82](#), en lugar de estimar la función de distribución F mediante la función de distribución empírica, se puede asumir que la población se distribuye como una normal bivalente.
- ▶ Para la media y la matriz de covarianzas de esta distribución, los estimadores razonables serían respectivamente

$$(\bar{x}, \bar{y})$$

$$\frac{1}{14} \begin{pmatrix} \sum_i (x_i - \bar{x})^2 & \sum_i (x_i - \bar{x})(y_i - \bar{y}) \\ \sum_i (x_i - \bar{x})(y_i - \bar{y}) & \sum_i (y_i - \bar{y})^2 \end{pmatrix}$$

- ▶ Se denota a la población normal bivalente que se obtiene con esta media y matriz de covarianzas como \hat{F}_{par} .

Bootstrap Paramétrico

- ▶ Se denomina al estimador bootstrap paramétrico del error estándar del parámetro como $se_{\hat{F}_{par}}(\hat{\theta}^*)$.
- ▶ En lugar de muestrear con reemplazamiento a partir de los datos originales, se sacan B muestras de tamaño n del estimador paramétrico de la población \hat{F}_{par}

$$\hat{F}_{par} \rightarrow (x_1^*, x_2^*, \dots, x_n^*)$$

- ▶ Posteriormente se siguen los mismos pasos 2 y 3 del algoritmo general del bootstrap *no paramétrico*: se calcula el correspondiente estadístico en cada muestra bootstrap y luego se calcula la desviación estándar de las B réplicas.

Ejemplo de los centros de estudios de máster

- En el ejemplo de los datos de los centros de estudios de máster en leyes se tiene que si (x, y) se distribuyen como una normal bivalente entonces se pueden generar observaciones de este vector, definiendo

$$\begin{aligned}x &= \mu_x + \sigma_x z_1 \\ y &= \mu_y + \sigma_y \frac{z_1 + c \cdot z_2}{\sqrt{1 + c^2}}\end{aligned}$$

donde $z_1, z_2 \sim N(0, 1)$

$$c = \sqrt{\frac{\sigma_x^2 \sigma_y^2}{\sigma_{xy}^2} - 1}$$

Ejemplo de los institutos de máster en leyes

```
library(bootstrap)

paraBoot = function(datos) {
  ndatos = dim(datos)[1]
  sigma = cov(datos)
  mu = rapply(datos, mean)

  c = sqrt(prod(diag(sigma))/sigma[1,2]^2-1)
  z1 = rnorm(ndatos)
  z2 = rnorm(ndatos)
  x = mu[1] + sqrt(sigma[1,1])*z1
  y = mu[2] + sqrt(sigma[2,2])*(z1+c*z2)/sqrt(1+c^2)
  cbind(x,y)
}
```

Ejemplo de los institutos de máster en leyes

Alternativamente, se puede programar con una librería específica que trata la normal multivariante: `mvrnorm`.

```
paraBoot = function(datos) {  
  ndatos = dim(datos)[1]  
  sigma = cov(datos)  
  mu = rapply(datos, mean)  
  
  x = MASS::mvrnorm(ndatos, mu=mu, Sigma=sigma)  
  return(x)  
}
```

Ejemplo de los institutos de máster en leyes

```
# Bootstrap parametrico
pBoot = replicate(5000, cor(paraBoot(law))[2,1])

# Aproximacion bootstrap
sd(pBoot)
```

```
[1] 0.1214904
```

```
samplesize = dim(law)[1]
# Aproximacion asintotica
(1-cor(law)[2,1]^2)/sqrt(samplesize-3)
```

```
[1] 0.1146741
```

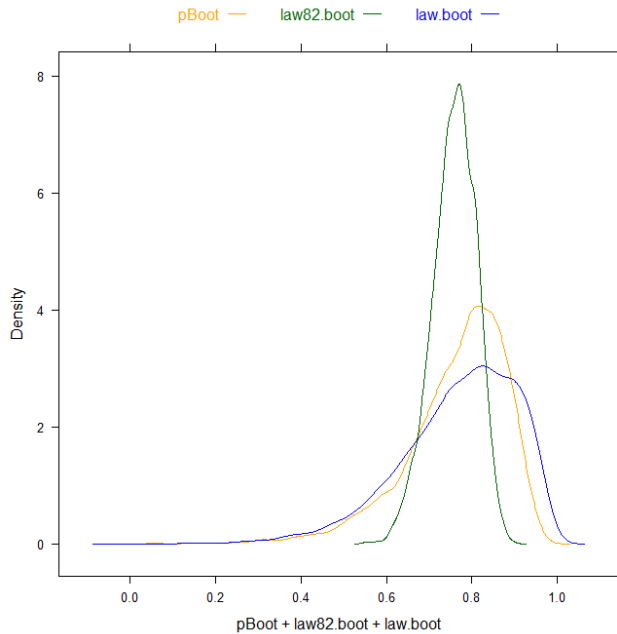
Ejemplo de los institutos de máster en leyes

```
library(latticeExtra)

ind1 = 1:dim(law)[1]
law.boot = replicate(5000, {indB = sample(ind1,
size=dim(law)[1], replace=TRUE);
with(law[indB,], cor(GPA,LSAT))})

ind2 = 1:dim(law82)[1]
law82.boot = replicate(5000,{indB = sample(ind2,
size=dim(law82)[1], replace=TRUE);
with(law82[indB,], cor(GPA,LSAT))})

densityplot(~pBoot+law82.boot+law.boot,plot.points=FALSE,
auto.key=list(columns=3,size=2,between=1,
col=c("orange","darkgreen","blue")),
par.settings=list(superpose.line=
list(col=c("orange","darkgreen","blue"))),
col=c("orange","darkgreen","blue"))
```

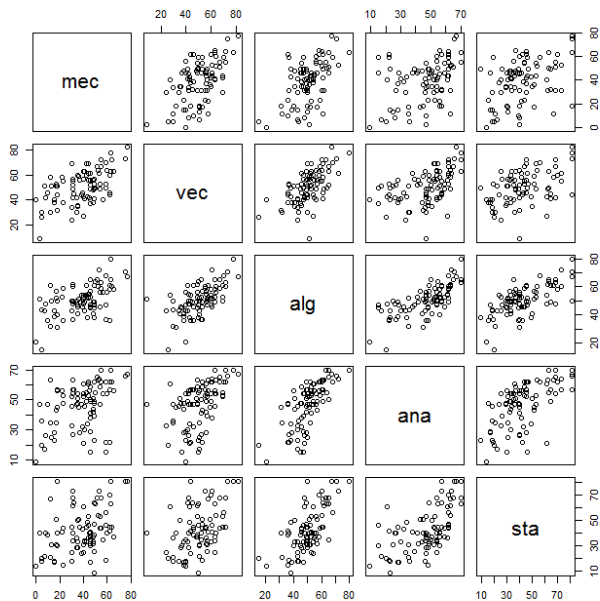
Bootstrap Paramétrico

- ▶ La mayor parte de los errores estándar son aproximaciones basadas en la distribución normal.
- ▶ Estas aproximaciones se parecen a los resultados que se obtienen con el bootstrap paramétrico cuando se hace remuestreo de la distribución normal.
- ▶ Cuando se usa bootstrap paramétrico se obtienen resultados más precisos que en las aproximaciones asintóticas cuando éstas existen.

Aplicación a datos multivariantes

- ▶ **Ejemplo:** Se tienen unos datos sobre calificaciones en 5 asignaturas de 88 alumnos (ver el libro sobre Análisis Multivariante, de Mardia, Kent and Bibby, (1979)):
 - ▶ **mec:** mechanics
 - ▶ **vec:** vectors
 - ▶ **alg:** algebra
 - ▶ **ana:** analysis
 - ▶ **sta:** statistics

```
library(bootstrap)
data(scor)
plot(scor)
```



Aplicación a datos multivariantes

- El vector de medias y la correspondiente matriz de covarianzas son:

```
colMeans(scor)
```

	mec	vec	alg	ana	sta
	38.95455	50.59091	50.60227	46.68182	42.30682

```
cov(scor)
```

	mec	vec	alg	ana	sta
mec	305.7680	127.22257	101.57941	106.27273	117.40491
vec	127.2226	172.84222	85.15726	94.67294	99.01202
alg	101.5794	85.15726	112.88597	112.11338	121.87056
ana	106.2727	94.67294	112.11338	220.38036	155.53553
sta	117.4049	99.01202	121.87056	155.53553	297.75536

Aplicación a datos multivariantes

- ▶ Se calculan los autovalores y autovectores de la matriz de covarianzas.
- ▶ La matriz 5×5 de covarianzas tiene 5 autovalores positivos en orden decreciente: $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \hat{\lambda}_3 \geq \hat{\lambda}_4 \geq \hat{\lambda}_5$ y a cada uno de ellos le corresponde un autovector diferente.

```
round(eigen(cov(scor))$values,3)      # Autovalores
```

```
[1] 686.990 202.111 103.747  84.630  32.153
```

```
round(eigen(cov(scor))$vectors,3)     # Autovectores
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.505	0.749	-0.300	0.296	-0.079
[2,]	-0.368	0.207	0.416	-0.783	-0.189
[3,]	-0.346	-0.076	0.145	-0.003	0.924
[4,]	-0.451	-0.301	0.597	0.518	-0.286
[5,]	-0.535	-0.548	-0.600	-0.176	-0.151

Aplicación a datos multivariantes

- ▶ Los autovalores y autovectores de la matriz de covarianzas son importantes para explicar la estructura multivariante de los datos.
- ▶ Se observa que las calificaciones en los exámenes están altamente correlacionados entre sí: un estudiante con calificaciones altas en mecánica suele tenerlas altas también en cálculo vectorial.
- ▶ Un modelo posible para las medidas correlacionadas sería

$$x_i = Q_i \mathbf{v}$$

para $i = 1, \dots, 88$

Aplicación a datos multivariantes

- ▶ Donde Q_i es un número que representa la capacidad del estudiante i mientras que \mathbf{v} es un vector de valores fijos para todos los estudiantes
- ▶ Q_i se puede interpretar como el *coeficiente intelectual* (IQ) del estudiante i -ésimo.
- ▶ Si el modelo anterior fuese cierto, entonces solo el primer autovalor $\hat{\lambda}_1$ sería positivo y el resto de autovalores serían igual a 0.
- ▶ También, \mathbf{v} sería igual al primer autovector $\hat{\mathbf{v}}_1$.

Aplicación a datos multivariantes

- ▶ Se define el ratio del mayor autovalor con respecto al total $\hat{\theta}$,

$$\hat{\theta} = \frac{\hat{\lambda}_1}{\sum_{i=1}^5 \hat{\lambda}_i}$$

- ▶ Así el modelo anterior es equivalente a $\hat{\theta} = 1$.
- ▶ Aunque, en la práctica, no se espera que sea *exactamente* igual a 1
- ▶ En el caso de las calificaciones, la estimación de $\hat{\theta}$ es

$$\hat{\theta} = \frac{686,990}{686,990 + 202,111 + 103,747 + 84,630 + 32,153} = 0,619$$

Aplicación a datos multivariantes

- ▶ En muchas circunstancias es interesante tener un valor alto de $\hat{\theta}$ porque eso indica un alto poder explicativo del modelo.
- ▶ El valor de $\hat{\theta}$ mide el porcentaje de varianza explicada por el primer componente.
- ▶ Cuanto más cerca estén los puntos respecto al eje del componente principal, mayor será el valor de $\hat{\theta}$.
- ▶ ¿Qué precisión tiene $\hat{\theta}$? ¿Cuál es el error estándar de $\hat{\theta}$?
- ▶ Esta sería una aplicación directa del bootstrap en este caso.

Aplicación a datos multivariantes

- ▶ La complejidad del cálculo de $\hat{\theta}$ no resulta relevante, en tanto que se pueda calcular $\hat{\theta}^*$ para cualquier muestra bootstrap.
- ▶ En este caso, una muestra bootstrap es una matriz remuestreada \mathbf{X}^* de tamaño 88×5 .
- ▶ Las filas \mathbf{x}_i^* de \mathbf{X}^* proceden de una m.a.s de tamaño 88 de las filas de la matriz de datos original.

$$x_1^* = x_{i_1}, x_2^* = x_{i_2}, \dots, x_{88}^* = x_{i_{88}}$$

- ▶ De este modo, algunas filas de \mathbf{X} aparecerán varias veces y otras ninguna en la matriz remuestreada \mathbf{X}^* .

Aplicación a datos multivariantes

- ▶ Una vez generada \mathbf{X}^* se calcula la matriz de covarianzas \mathbf{G}^* de la manera habitual y luego se calculan los autovalores correspondientes.
- ▶ Se calcula la réplica bootstrap de $\hat{\theta}$

$$\hat{\theta}^* = \frac{\hat{\lambda}_1^*}{\sum_{j=1}^5 \hat{\lambda}_j^*}$$

- ▶ Y se aplica el algoritmo general bootstrap para calcular el error estándar.

Aplicación a datos multivariantes

```
library(bootstrap)

# En componente principales svd es numericamente mas
# estable que la descomposicion por autovectores y
# autovalores, pero para aplicar bootstrap
# esta ultima es mas rapida

autovals = eigen(var(scor), symmetric=TRUE,
only.values=TRUE)$values
(teta = autovals[1] / sum(autovals))
```

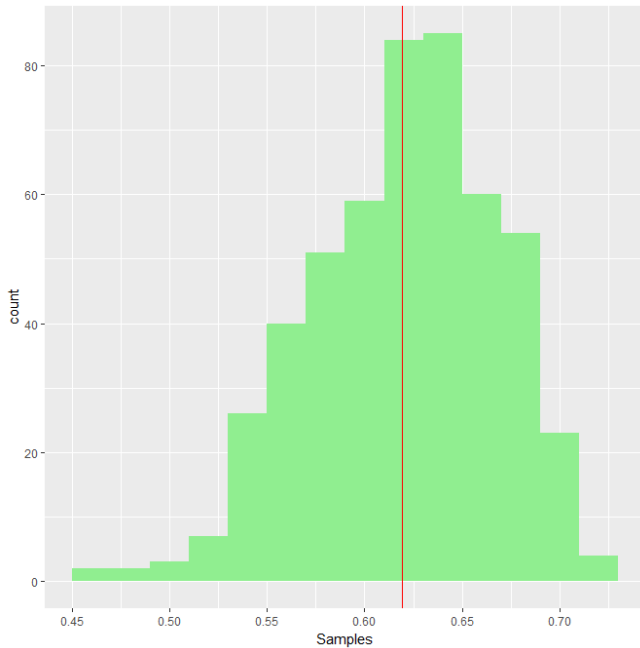
```
[1] 0.619115
```

Aplicación a datos multivariantes

```
theta = function(ind){  
  vals = eigen(var(scor[ind,]), symmetric=TRUE,  
    only.values=TRUE)$values  
  vals[1] / sum(vals)  
}  
  
scor.boot = bootstrap(1:dim(scor)[1], 500, theta)  
sd(scor.boot$thetastar) # error estandar del bootstrap
```

```
[1] 0.04570752
```

```
library(ggplot2)  
qplot(scor.boot$thetastar, geom="histogram", binwidth=0.02,  
  fill=I("lightgreen"), xlab="Samples") +  
  geom_vline(xintercept=teta, col="red")
```



Aplicación a datos multivariantes

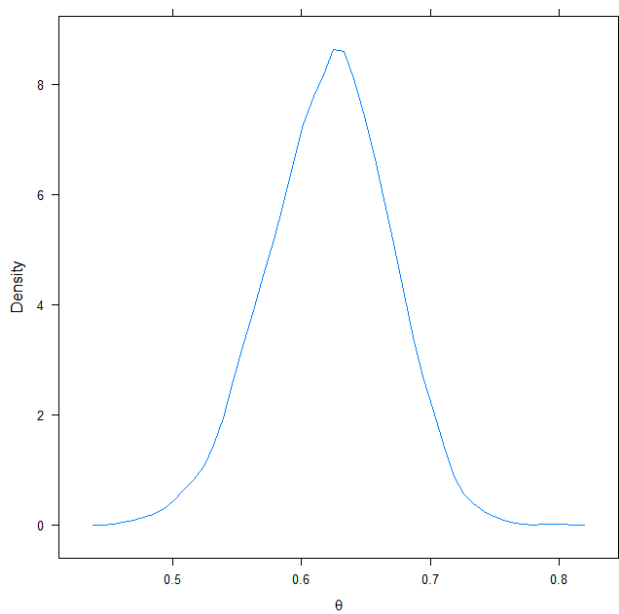
```
library(bootstrap)
data(scor)
X = scor

eigenTeta = function(X) {
  ee = eigen(cov(X))["values"]
  ee[1]/sum(ee)
}

ind = 1:dim(X)[[1]]

eigendist = replicate(5000,
  eigenTeta(X[sample(ind, replace=TRUE),]))

library(latticeExtra)
densityplot(eigendist, plot.points=FALSE,
  xlab=expression(theta))
```

Aplicación a datos multivariantes

- ▶ El autovector $\hat{\mathbf{v}}_1$ que corresponde al mayor autovalor se le denomina primer componente principal de \mathbf{G}
- ▶ Supongamos que se trata de resumir el rendimiento de los estudiantes mediante un solo número, en lugar de con 5 notas.
- ▶ Se puede demostrar que la mejor combinación lineal de las 5 notas es

$$y_i = \sum_{k=1}^5 \hat{v}_{1k} x_{ik}$$

es decir, una combinación lineal donde los componentes $\hat{\mathbf{v}}_1$ equivalen a los pesos de las notas originales.

- ▶ Esta combinación lineal es óptima en el sentido de que captura la mayor parte de la variabilidad de las 5 puntuaciones originales de entre todos los posibles \mathbf{v} .

Aplicación a datos multivariantes

- ▶ La segunda combinación lineal

$$z_i = \sum_{k=1}^5 \hat{v}_{2k} x_{ik}$$

es el segundo componente principal $\hat{\mathbf{v}}_2$ es decir, el segundo autovector de \mathbf{G} .

- ▶ El primer componente se puede asociar a la *media de puntuaciones* de un estudiante, mientras que el segundo parece asociarse más bien a la relación que hay entre exámenes con libro *abierto* frente a *cerrado*.

Aplicación a datos multivariantes

- Tanto $\hat{\mathbf{v}}_1$ como $\hat{\mathbf{v}}_2$ son estadísticos del mismo modo que lo es $\hat{\theta}$, y de este modo se puede aplicar el bootstrap para calcular su variabilidad.

```
library(bootstrap)
data(scor)
X = scor

eigenVec = function(X) {
  ee = eigen(cov(X))["vectors"]
  return(cbind(ee[,1], ee[,2]))
}
```

Aplicación a datos multivariantes

```
ind = 1:dim(X)[[1]]  
eigendist = replicate(500,  
  eigenVec(X[sample(ind, replace=TRUE),]))  
  
apply(eigendist[1:5,1,], 1, sd)
```

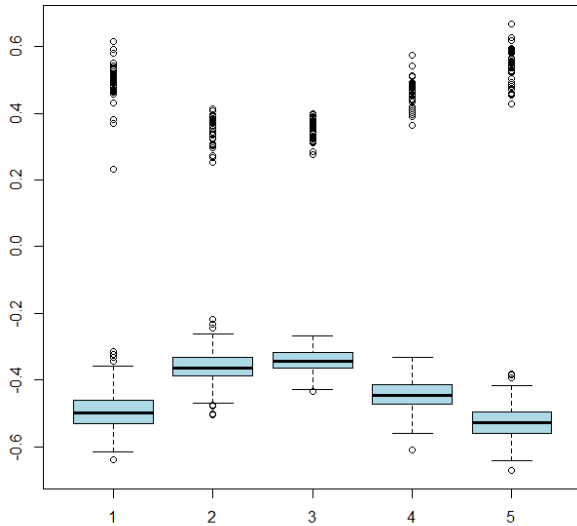
```
[1] 0.2525616 0.1828719 0.1704597 0.2257041 0.2645642
```

```
apply(eigendist[1:5,2,], 1, sd)
```

```
[1] 0.50333116 0.20130403 0.07855624 0.23908044 0.41575549
```

```
boxplot(cbind(eigendist[1,1,], eigendist[2,1,],  
  eigendist[3,1,], eigendist[4,1,], eigendist[5,1,]),  
  main="Componente 1", col="lightblue")
```

Componente 1



Cuando puede fallar el bootstrap

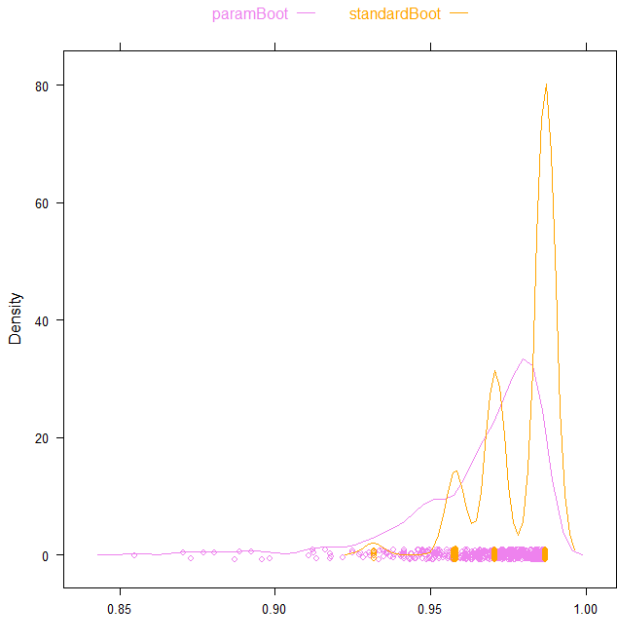
- ▶ Consideramos el siguiente problema:
 X se distribuye como una distribución uniforme en $(0, \theta)$.
El estimador de máxima verosimilitud para θ es el $\max(X_i)$
- ▶ Tenemos una muestra de 50 observaciones.
- ▶ Comparamos el estimador bootstrap no paramétrico de θ con respecto al estimador paramétrico del mismo.

```
N = 50  
X = runif(N)  
(thetaHat = max(X))
```

```
[1] 0.990335
```

Cuando puede fallar el bootstrap

```
standardBoot = replicate(500,  
max(sample(X, N, replace=TRUE)))  
  
paramBoot = replicate(500,  
max(runif(N, min=0, max=thetaHat)))  
  
library(latticeExtra)  
  
densityplot(~paramBoot + standardBoot, xlab="",  
auto.key=list(columns=2, size=2, between=1,  
col=c("violet", "orange")),  
par.settings=list(superpose.line=  
list(col=c("violet", "orange"))),  
col=c("violet", "orange"))
```

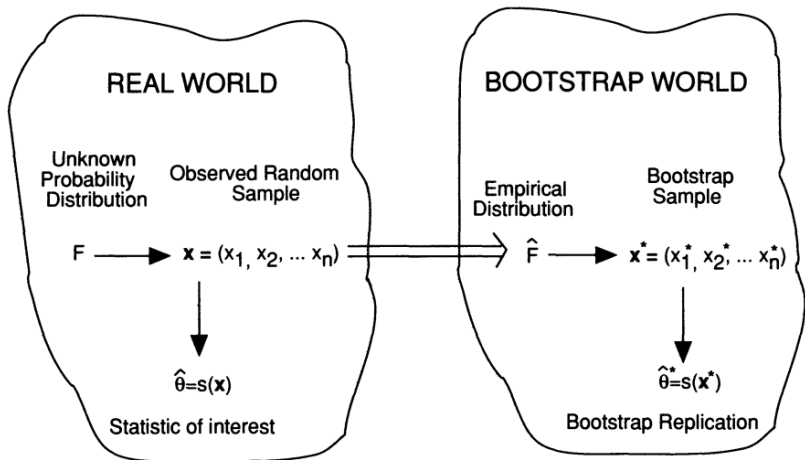



Estructuras de datos generales

- ▶ Hasta ahora se ha considerado una estructura simple de los datos: el modelo unimuestral donde una distribución de probabilidad desconocida F genera los datos X mediante muestreo aleatorio.
- ▶ Pero algunos datos x_i pueden ser bastante complejos, como vectores, mapas o imágenes.
- ▶ Estructuras complejas de datos aparecen en modelos como series temporales, análisis de varianza, modelos de regresión, datos censurados o muestreo estratificado.
- ▶ Pero el método bootstrap se puede adaptar a estructuras de datos generales.

Problemas unimuestrales

- ▶ El esquema del método bootstrap para problemas unimuestrales se basa en la existencia de dos *mundos paralelos*.
- ▶ Por un lado está el mundo real con una distribución desconocida F de la que se toma una muestra aleatoria y se calcula un estadístico a partir de \mathbf{x} digamos $\hat{\theta} = s(\mathbf{x})$. Después se trata de estudiar su comportamiento: errores, intervalos de confianza, etc.
- ▶ Por otro lado está el *mundo bootstrap* de modo que la población se reduce a la muestra original y a partir de la distribución empírica \hat{F} se obtienen las muestras bootstrap \mathbf{x}^* .
- ▶ A partir de ella se calcula el estadístico de interés $\hat{\theta}^* = s(\mathbf{x}^*)$ y se estudia su comportamiento.



Problemas unimuestrales

- ▶ La doble flecha del esquema indica el cálculo de \hat{F} a partir de F .
- ▶ Conceptualmente este es el paso fundamental del bootstrap y el resto de pasos se definen por analogía.
- ▶ El procedimiento bootstrap para estructuras más complejas es inmediato una vez que se sabe como realizar el proceso de la *doble flecha*, es decir cómo estimar el mecanismo probabilístico a partir de los datos.
- ▶ Se usa la notación $P \rightarrow \mathbf{x}$ para indicar que un modelo de probabilidad desconocido P ha generado el conjunto de datos \mathbf{x} .

Problemas de dos muestras

- ▶ En el caso del problema de inferencia de dos muestras, el modelo de probabilidad se puede considerar como $P = (F, G)$ donde F es la distribución de probabilidad del primer grupo y G la del segundo grupo.
- ▶ Se obtienen dos muestras aleatorias independientes $\mathbf{x} = (\mathbf{z}, \mathbf{y})$ de modo que la aplicación $P \rightarrow \mathbf{x}$ se describe como $F \rightarrow \mathbf{z}$ e independientemente $G \rightarrow \mathbf{y}$
- ▶ En este caso se toman las respectivas funciones de distribución empíricas y el estimador natural de P se construye como $\hat{P} = (\hat{F}, \hat{G})$ y se obtiene una muestra bootstrap $\mathbf{x}^* = (\mathbf{z}^*, \mathbf{y}^*)$ como $\hat{F} \rightarrow \mathbf{z}^*$ e independientemente $\hat{G} \rightarrow \mathbf{y}^*$

Ejemplo de los ratones

- ▶ Se toma el ejemplo de las diferencias de medias entre ratones según son tratamiento o control

```
Trata = c(94,197,16,38,99,141,23)
Cont = c(52,104,146,10,51,30,40,27,46)

mean(Trata) - mean(Cont)
```

```
[1] 30.63492
```

```
B = 1000

sd(replicate(B, mean(sample(Trata,replace=TRUE)) -
mean(sample(Cont,replace=TRUE))))
```

```
[1] 26.19047
```

Ejemplo de los ratones

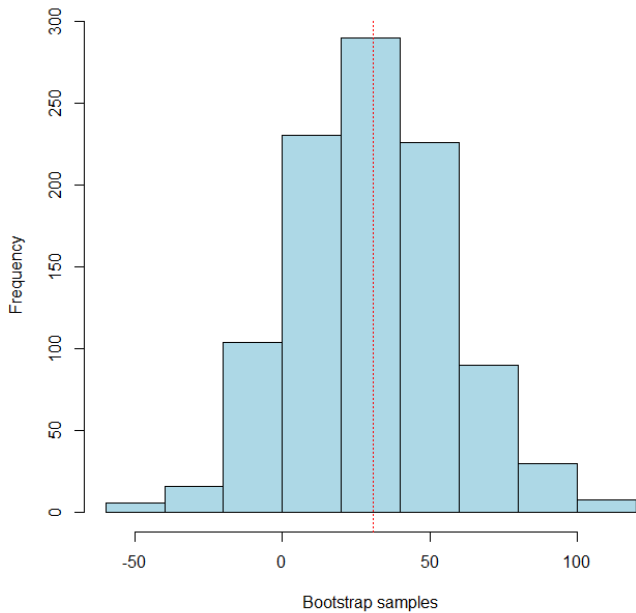
```
library(simpleboot)
```

```
b = two.boot(Trata, Cont, mean, R=B)
```

```
sd(b$t)
```

```
[1] 27.27626
```

```
hist(b, col="lightblue")
```

Ejemplo de los ratones

```
# Programa original de la libreria bootstrap

B = 1000
library(bootstrap)
mouse.boot.c = bootstrap(mouse.c, B, mean)
mouse.boot.t = bootstrap(mouse.t, B, mean)

mouse.boot.diff =
mouse.boot.t$thetastar - mouse.boot.c$thetastar

sd(mouse.boot.diff)
```

```
[1] 26.58504
```

Ejemplo de los ratones

```
Trata = c(94,197,16,38,99,141,23)
Cont = c(52,104,146,10,51,30,40,27,46)
B = 1000
n = length(Trata)
Losratones = c(Trata,Cont)

(t.obs = mean(Trata)-mean(Cont))
```

```
[1] 30.63492
```

Ejemplo de los ratones

```
library(boot)
t.fun = function(data,i,n){
  bobo = data[i]
  mean(bobo[1:n]) - mean(bobo[-c(1:n)])
}

(mouse.boot = boot(Losratones, t.fun, R=1000, n=n))
```

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

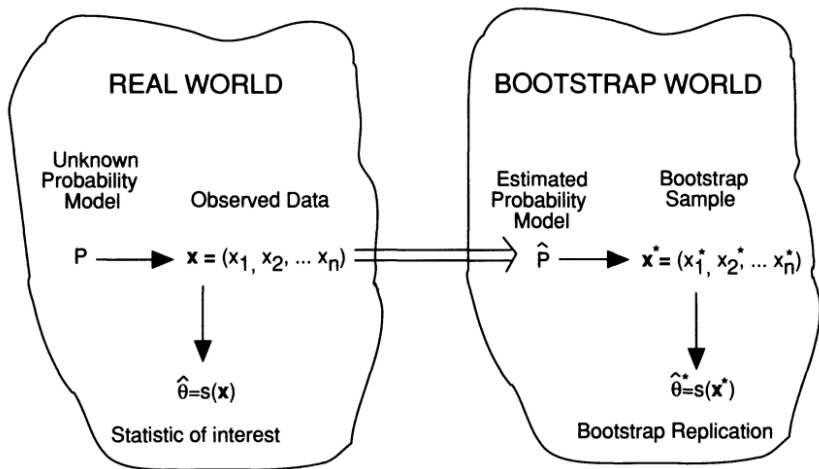
```
boot(data = Losratones, statistic = t.fun, R = 1000, n = n)
```

Bootstrap Statistics :

	original	bias	std. error
t1*	30.63492	-29.97352	27.83245

Estructuras de datos generales

- ▶ El esquema siguiente se aplica a estructuras generales $P \rightarrow \mathbf{x}$
- ▶ En el mundo real se tiene una distribución desconocida P que da lugar al conjunto de datos \mathbf{x}
- ▶ El paso principal es el indicado por \Rightarrow que da lugar a un estimador \hat{P} de la distribución original P .
De este modo $\hat{P} \rightarrow \mathbf{x}^*$ es equivalente a $P \rightarrow \mathbf{x}$
- ▶ Y así $\mathbf{x}^* \rightarrow \hat{\theta}^* = s(\mathbf{x}^*)$ es la misma función que $\mathbf{x} \rightarrow \hat{\theta} = s(\mathbf{x})$.
- ▶ Generalmente la generación de muestras bootstrap $\hat{P} \rightarrow \mathbf{x}^*$ requiere menos tiempo de computación que el cálculo de $\hat{\theta}^* = s(\mathbf{x}^*)$.



Otro ejemplo

- ▶ Se toma el ejemplo de los datos correspondientes a 20 observaciones donde se mide el efecto de unas pastillas para dormir (el incremento de horas de sueño en relación a mediciones control).

```
data(sleep)

# test t Student
with(sleep, t.test(extra~group)$statistic)
```

```
      t
-1.860813
```

Otro ejemplo

```
scores = sleep$extra
R = 1000
t.valores = numeric(R)

scoresG1 = subset(scores, sleep$group==1)
scoresG2 = subset(scores, sleep$group==2)

for (i in 1:R) {
  grupo1 = sample(scoresG1, size=10, replace=T)
  grupo2 = sample(scoresG2, size=10, replace=T)
  t.valores[i] = t.test(grupo1, grupo2)$statistic
}

sd(t.valores)
```

```
[1] 1.103293
```

```
ggplot2::qplot(t.valores, geom="histogram", binwidth=1,
fill=I("lightgreen"), col=I("red"))
```