Demo R: Remuestreo y estimación de densidades

Rocío Maribel Ávila Ayala

Conceptos importantes

Inferencia no paramétrica mediante remuestreo

Estimación del error estándar de un estimador

Supóngase que se cuenta con una muestra observada $\mathbf{x} = (x_1, \ldots, x_n)$ de un fenómeno aleatorio de interés asociado a una variable aleatoria con distribución desconocida F, y que se desea estimar un parámetro de interés $\theta = T(F)$.

Una forma de medir la precisión del estimador $\hat{\theta} = g(\mathbf{x})$ de θ es mediante su error estándar o su varianza. El error estándar puede estimarse por alguna técnica de remuestreo. A continuación se presentan dos de ellas:

Estimación del error estándar por Bootstrap

1. Simular de $\hat{F}_n$ B muestras bootstrap de tamaño n, $\mathbf{x}^{*1}, \mathbf{x}^{*2}, \ldots, \mathbf{x}^{*B}$. Donde $\mathbf{x}^{*b} = (x_1^{*b}, \ldots, x_n^{*b})$, para $b = 1, \ldots, n$.
2. Calcular las correspondientes réplicas bootstrap $\hat{\theta}^*(b) = g(\mathbf{x}^{*b})$, para $b = 1, \ldots, n$.
3. Estimar el error estándar $se_F(\hat{\theta})$ como la desviación estándar de las B réplicas

$$\hat{se}_B = \left\{ \sum_{b=1}^{B} \left[\hat{\theta}^*(b) - \hat{\theta}^*(\cdot) \right]^2 / (B-1) \right\}^{1/2},$$

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

Estimación del error estándar por Jackknife

1. Calcular las n muestras jackknife

$$\mathbf{x}_{(i)} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$$

para $i = 1, \ldots, n$.

2. Calcular las correspondientes réplicas jackknife para $i = 1, \ldots, n$

$$\hat{\theta}_{(-i)} = g(\mathbf{x}_{(i)})$$

3. Calcular el promedio de las réplicas jackknife

$$\hat{\theta}_{(\cdot)} = \frac{\sum_{i=1}^{n} \hat{\theta}_{(-i)}}{n}$$

4. El estimador del error estándar por jackknife está dado por

$$\hat{se}_{jack} = \left[\frac{n-1}{n} \sum_{i=1}^{n} \left(\hat{\theta}_{(-i)} - \hat{\theta}_{(\cdot)} \right)^2 \right]^{1/2}$$

Estimación del sesgo de un estimador

El sesgo de un estimador $\hat{\theta} = g(\mathbf{x})$ es otra medida de la precisión del mismo como estimador de $\theta = T(F)$. El sesgo se define como la diferencia entre la esperanza del estimados $\hat{\theta}$ y el parámetro θ que se está estimando.

Estimación del sesgo por Bootstrap

1. Simular de $\hat{F}_n$ B muestras bootstrap de tamaño n, $\mathbf{x}^{*1}, \mathbf{x}^{*2}, \ldots, \mathbf{x}^{*B}$. Donde $\mathbf{x}^{*b} = (x_1^{*b}, \ldots, x_n^{*b})$, para $b = 1, \ldots, n$.
2. Calcular las correspondientes réplicas bootstrap $\hat{\theta}^*(b) = s(\mathbf{x}^{*b})$, para $b = 1, \ldots, n$.
3. Estimar el sesgo de $\hat{\theta}$ como
$$\hat{bias}_B(\hat{\theta}) = \hat{\theta}^*(\cdot) - T(\hat{F}),$$
donde $\hat{\theta}^*(\cdot) = \frac{\sum_{b=1}^{B} \hat{\theta}(b)}{B} = \frac{\sum_{b=1}^{B} g(\mathbf{x}^{*b})}{B}$

Estimación del sesgo por Jackknife

1. Calcular las n muestras jackknife
$$\mathbf{x}_{(i)} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$$
para $i = 1, \ldots, n$.
2. Calcular las correspondientes réplicas jackknife para $i = 1, \ldots, n$
$$\hat{\theta}_{(i)} = g(\mathbf{x}_{(i)})$$
3. Calcular el promedio de las réplicas jackknife
$$\hat{\theta}_{(\cdot)} = \frac{\sum_{i=1}^{n} \hat{\theta}_{(i)}}{n}$$
4. El estimador del sesgo por jackknife está dado por
$$\hat{bias}_{jack} = (n-1)\left(\hat{\theta}_{(\cdot)} - \hat{\theta}\right).$$

Intervalos de confianza por remuestreo

A partir de las técnicas Bootstrap y Jackknife también pueden obtenerse intervalos de $100(1-\alpha)\%$ confianza para el estadístico observado $\hat{\theta}$.

1. Intervalo normal con error estándar bootstrap
$$\hat{\theta} \pm z_{\alpha/2} \hat{se}_B$$
2. Intervalo normal con error estándar jackknife
$$\hat{\theta} \pm z_{\alpha/2} \hat{se}_{jack}$$
3. Intervalo pivotal
Sea θ^*_β el cuantil β de la muestra de réplicas bootstrap $(\hat{\theta}^*(1), \ldots, \hat{\theta}^*(B))$,
$$\left(2\hat{\theta} - \theta^*_{1-\alpha/2}, 2\hat{\theta} - \theta^*_{\alpha/2}\right)$$

4. Intervalo percentil

$$\left(\theta^*_{\alpha/2}, \theta^*_{1-\alpha/2}\right)$$

Estimación de densidades por Kernel

Se dice que una función $K : \mathbb{R} \longrightarrow \mathbb{R}$ es un núcleo si $\int_{-\infty}^{\infty} K(u)du = 1$ y K es una función simétrica, i.e. $K(-x) = K(x)$.

Sea $(x_1, \ldots, x_n)$ una muestra i.i.d. proveniente de una variable aleatoria con densidad f, y sean K un Kernel sobre $\mathbb{R}$ y h un ancho de banda fijo. El estimador por núcleo K de la densidad f es

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right).$$

Ejemplos de implementación computacional

Bootstrap

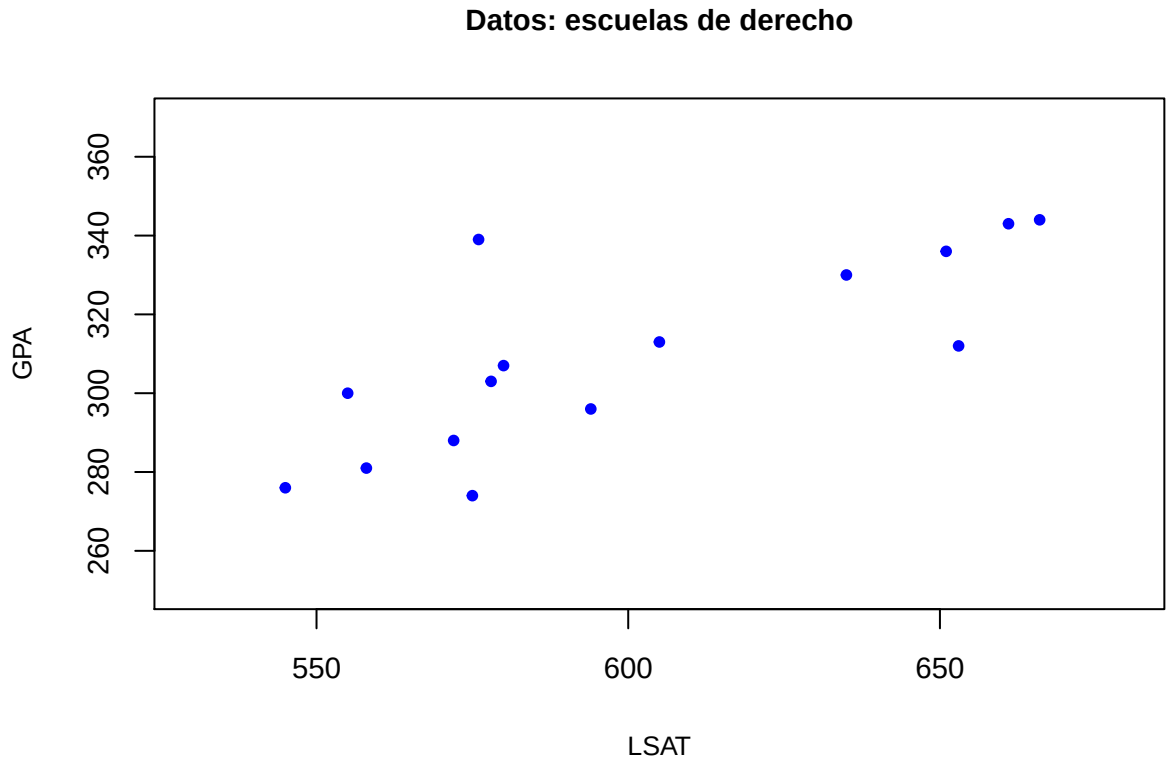
Se cuenta con datos de 15 alumnos de la escuela de leyes correspondientes a calificaciones en la prueba de admisión LSAT (Law School Admission Test), así como calificaciones GPA (Grade Point Average), las cuales miden su desempeño académico. A partir de dichos datos se estima un coeficiente de correlación de 0.776.

```
# Calificación en el examen de admisión
lsat <- c(576, 635, 558, 578, 666, 580, 555, 661, 651, 605,
          653, 575, 545, 572, 594)

# Desempeño en los cursos
gpa <- c(339, 330, 281, 303, 344, 307, 300, 343, 336, 313,
         312, 274, 276, 288, 296)

# Correlación observada
(rho_obs <- cor(lsat, gpa))
```

```
## [1] 0.7763745
```

A continuación se muestra la dispersión de los puntos observados. El comportamiento valida una correlación positiva.

```
# Dispersión de los puntos
plot(lsat, gpa, xlab = "LSAT", ylab = "GPA",
     col = "blue", cex.axis = 0.9, pch = 20,
     xlim = c(530, 680), ylim = c(250, 370), cex.lab = 0.8,
     main = "Datos: escuelas de derecho", cex.main = 0.9)
```



Se desea conocer la precisión del estimador observado de la correlación. A continuación se presenta una función para estimar el error estándar para cualquier número de remuestreos (B)

```
se_boot <- function(B){
  n <- length(gpa)
  b <- c()
  for(i in 1:B){
    sel <- sample(1:n, size = n, replace  = T)
    b[i] <- cor(lsat[sel], gpa[sel])
  }
  # Estimación del error estándar del estimador de la correlación
  s <- sd(b)
  out <- list(replicas = b, error_est = s)
  return(out)
}
```

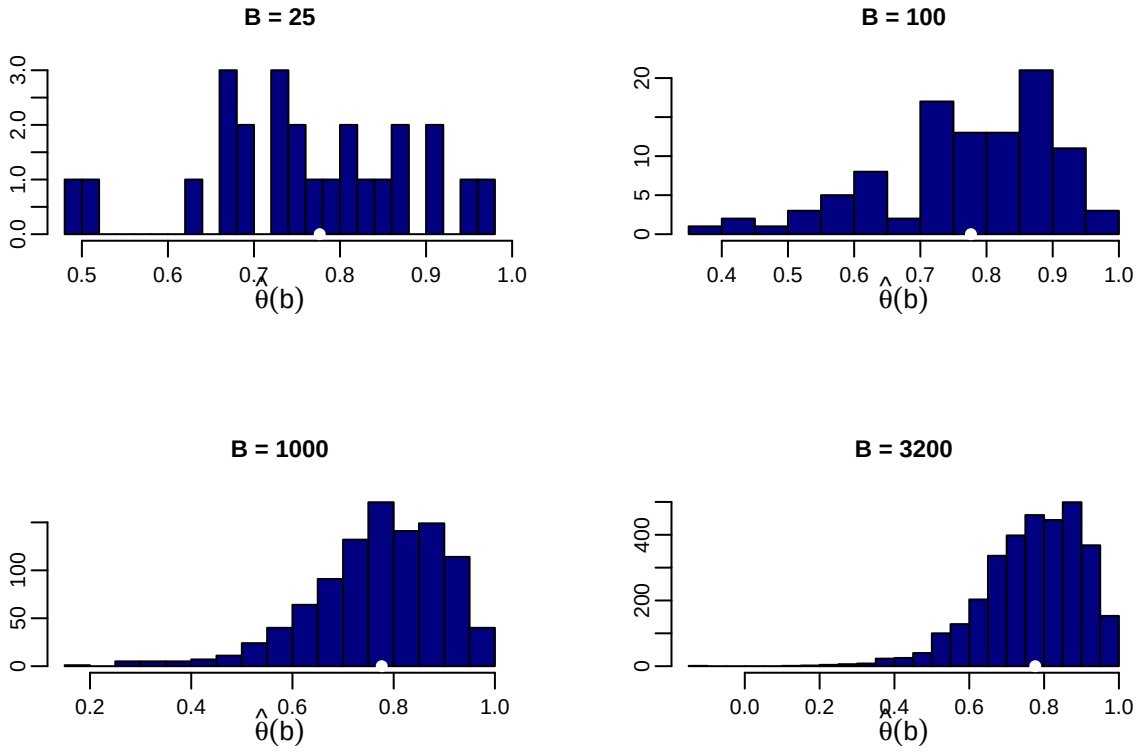
Ahora se calcula por remuestreo el error estándar del estimador para $B = 25, 100, 400, 3200$

```
set.seed(381)
estim_sd_boot <- lapply(c(25, 100, 1000, 3200), se_boot)
error_est <- c(estim_sd_boot[[1]]$error_est, estim_sd_boot[[2]]$error_est,
               estim_sd_boot[[3]]$error_est, estim_sd_boot[[4]]$error_est)
names(error_est) <- paste("B =", c(25, 100, 1000, 3200))
error_est
```

```
##    B = 25   B = 100  B = 1000  B = 3200
## 0.1228328 0.1319545 0.1303034 0.1328138
```

Se observa que el número de remuestreos varía ligeramente la estimación. Por último, se grafica la distribución de las réplicas bootstrap correspondientes para cada número de remuestreos.

```
# Histograma del estimador de la correlación
par(mfrow = c(2, 2), mar = c(5, 4, 3, 1))
for(i in 1:4){
  b <- estim_sd_boot[[i]]$replicas
  hist(b, main = paste("B =", length(b)),
       cex.main = 0.9, xlab = expression(hat(theta)*(b)), ylab = "",
       cex.axis = 0.8, mgp = c(1.5, 0.5, 0), col = "navyblue", nclass = 20)
  points(rho_obs, 0, pch = 16, col = "white")
}
```



Se observa que al considerar pocos remuestreos, a pesar de que el estimador del error estándar no difiere mucho del que toma 3200 (ver punto blanco bajo el histograma), el histograma resulta muy deficiente, por lo que se deduce que se necesitan más réplicas bootstrap para estimar la forma de la densidad que para calcular el error estándar.

Jackknife

A partir de los siguientes datos

```
dat <- rcauchy(100)
```

se desea calcular el coeficiente de variación $CV = \frac{sd(X)}{\mathbb{E}(X)}$, el cual es una medida de dispersión que describe la cantidad de variabilidad en relación con la media. El CV observado resulta

```r
CV <- function(x) sd(x)/mean(x)
(CV_obs <- CV(dat))
```

```
## [1] 6.345344
```

Ahora interesa ver qué tan preciso es este estimador. A continuación se calculan las n réplicas Jackknife posibles del CV:

```r
CV_jk <- c()
for(i in 1:length(dat)){
  CV_jk[i] <- CV(dat[-i])
}
```

A partir de las réplicas anteriores, estimamos el error estándar del CV observado:

```r
med_jack <- mean(CV_jk)
n <- length(dat)
(sd_CV_jack <- ((n-1)/n))*sum((CV_jk-med_jack)^2)
```

```
## [1] 9.082111
```

Y el sesgo del CV observado se obtiene como:

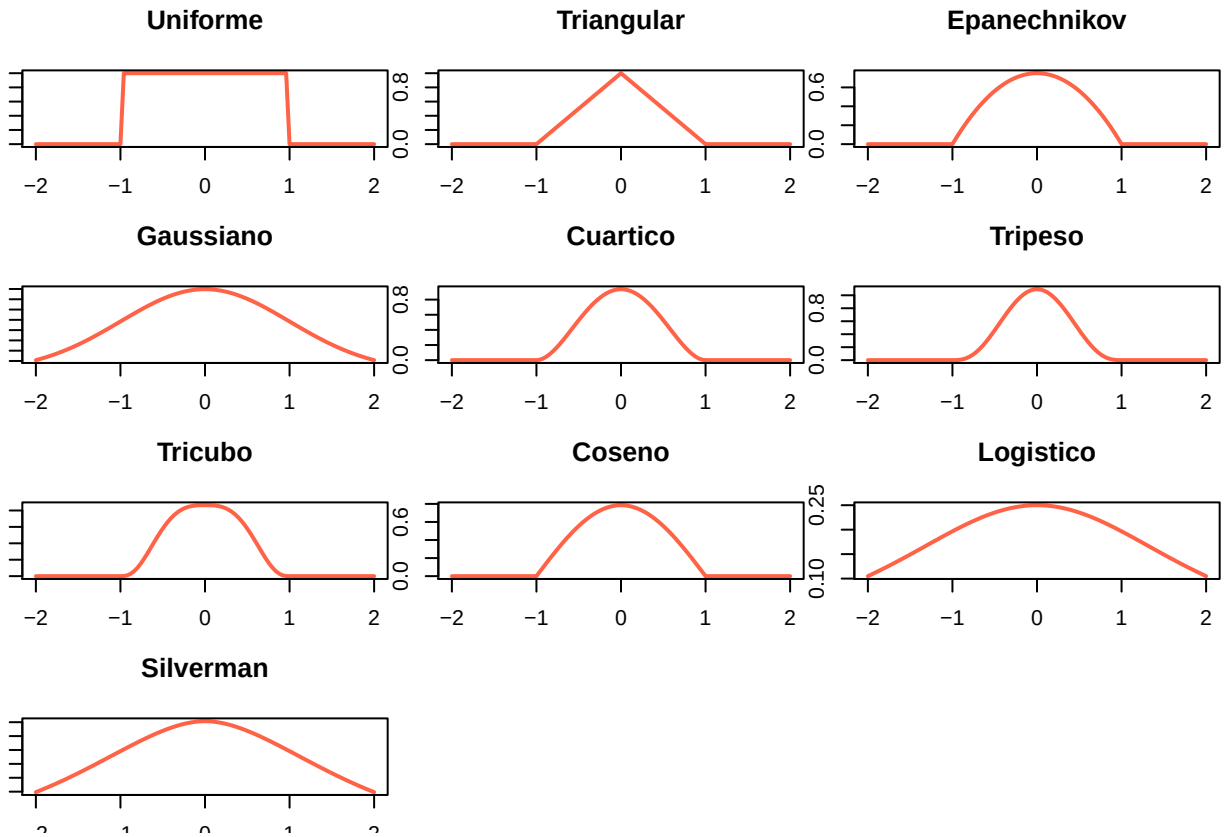
```r
(sesgo_CV_jack <- (n-1)*(med_jack-CV_obs))
```

```
## [1] 1.43623
```

Estimación de densidades por Kernel

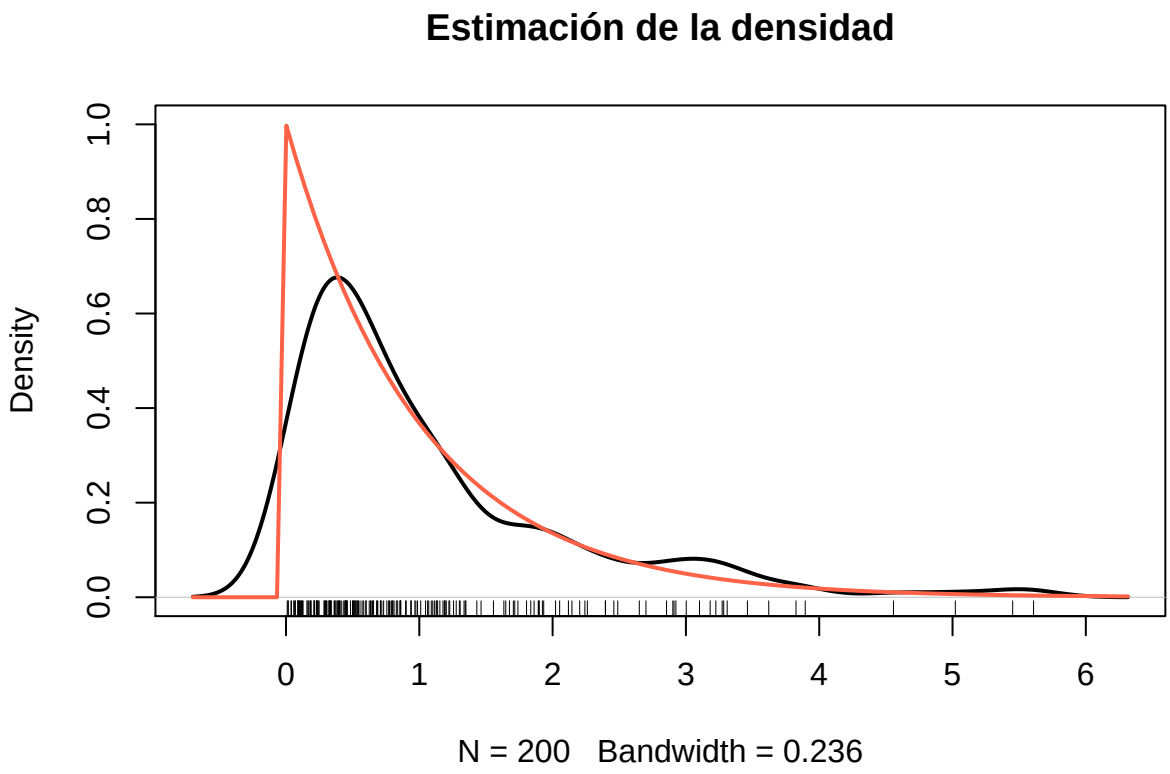
A continuación se muestran las gráficas de algunas funciones Kernel que pueden usarse en la estimación de una densidad:

```r
K<-function(u,nombre){
  if (nombre=="Uniforme") return(0.5*(u>-1&u<1))
  if(nombre=="Triangular") return((1-abs(u))*(u>-1&u<1))
  if(nombre=="Epanechnikov") return(0.75*(1-u^2)*(u>-1&u<1))
  if(nombre=="Gaussiano") return(1/sqrt(2*pi)*exp(-0.5*u^2))
  if(nombre=="Cuartico") return(15/16*(1-u^2)^2*(u>-1&u<1))
  if(nombre=="Tripeso") return(35/32*(1-u^2)^3*(u>-1&u<1))
  if(nombre=="Tricubo") return(70/81*(1-abs(u)^3)^3*(u>-1&u<1))
  if(nombre=="Coseno") return(pi/4*cos(pi/2*u)*(u>-1&u<1))
  if(nombre=="Logistico") return((exp(u)+2+exp(-u))^(-1))
  if(nombre=="Silverman") return(1/2*exp(-abs(u)/sqrt(2))*sin(abs(u)/sqrt(2)+pi/4))
}

par(mfrow = c(4, 3), mar = c(1.5, 1, 4, 1))
nomK<-c("Uniforme","Triangular","Epanechnikov","Gaussiano","Cuartico","Tripeso",
           "Tricubo","Coseno","Logistico","Silverman")

for(i in nomK)
curve(K(x, i), xlim = c(-2, 2), main = i, col = "tomato", lwd = 2)
```



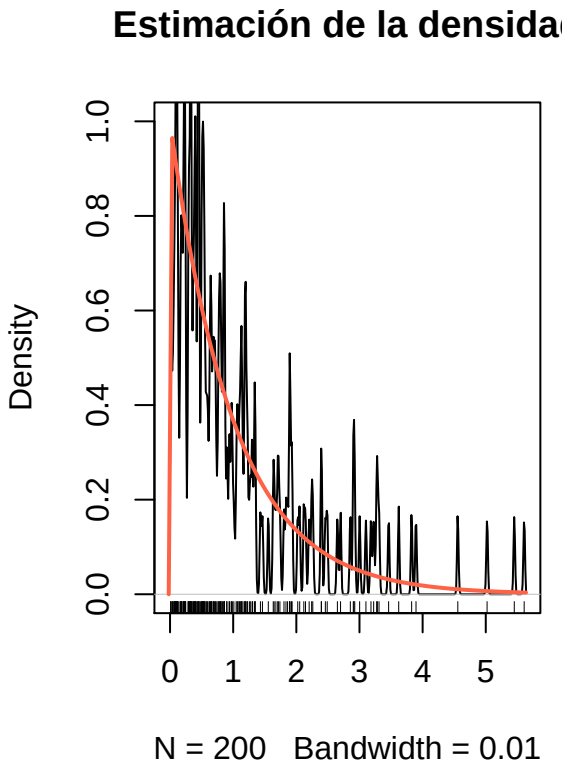
La función *density* estima la densidad de los datos y calcula un ancho de banda óptimo. Veamos un ejemplo con datos exponenciales. La curva roja corresponde a la densidad real a partir de la cual fueron simulados. La fución *rug* agrega sobre el eje x los datos que se observaron.

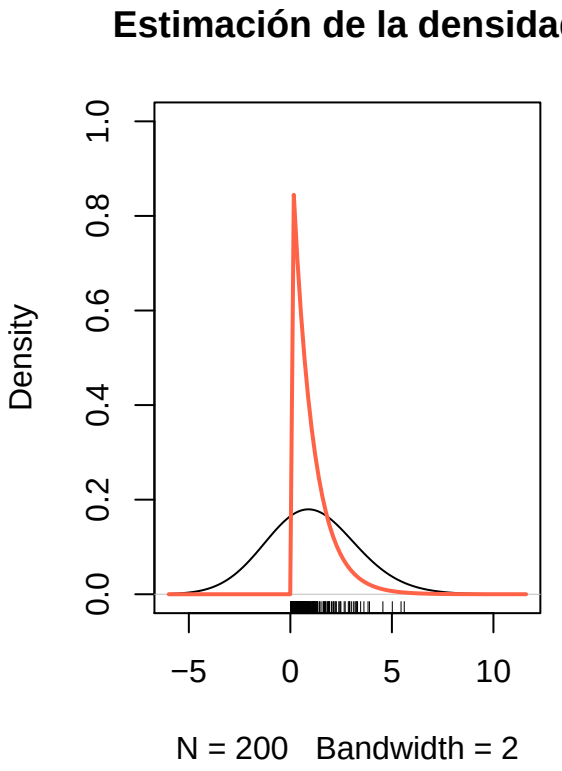
```
set.seed(207)
x <- rexp(200, 1)
plot(density(x), ylim = c(0, 1), lwd = 2, main = "Estimación de la densidad")
curve(dexp(x, rate = 1), add = T, col = "tomato", lwd = 2)
rug(x)
```



En la parte inferior de la gráfica observamos que el ancho de banda óptimo calculado fue se 0.2799. Ahora veremos un ejemplo de que h muy pequeño produce una curva muy ruidosa y h grande una curva demasiado burda.

```
par(mfrow = c(1,2))
# h pequeño
set.seed(207)
x <- rexp(200, 1)
plot(density(x, bw = 0.01), ylim = c(0, 1), main = "Estimación de la densidad")
curve(dexp(x, rate = 1), add = T, col = "tomato", lwd = 2)
rug(x)
# h grande
set.seed(207)
x <- rexp(200, 1)
plot(density(x, bw = 2), ylim = c(0, 1), main = "Estimación de la densidad")
curve(dexp(x, rate = 1), add = T, col = "tomato", lwd = 2)
rug(x)
```


Estimación de la densidad
Density
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
N = 200 Bandwidth = 0.01


Estimación de la densidad
Density
0.0
0.2
0.4
0.6
0.8
1.0
−5
0
5
10
N = 200 Bandwidth = 2