Sesión 10: Aproximación de densidad

Aplicaciones en Computación Estadística

Natalie Julian - www.nataliejulian.com

Estadística UC y Data Scientist en Zippedi Inc.

Presiones atmosféricas

El archivo `presion` contiene mediciones semanales de fluctuaciones de presiones atmosféricas. Se realizaron varias mediciones por semana.

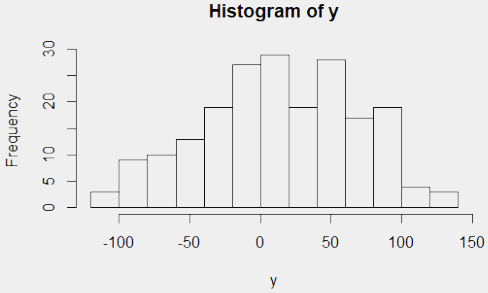
Interesa estimar la densidad de la variable fluctuación, explorando distintas variantes.

Vía Histograma

Histograma

Si realizamos un histograma de la variable obtenemos lo siguiente:

```
hist(y)

summary(y)
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-117.41  -22.73   13.18   13.82   54.22  127.73
```

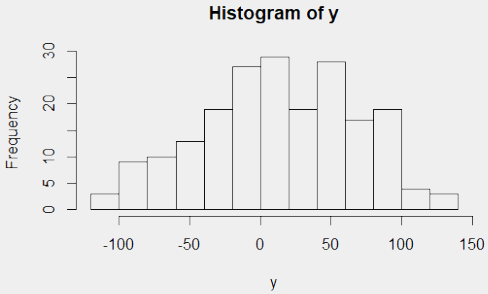


Variantes de la función `hist`

Una variante muy relevante es el argumento `breaks` de la función `hist()`.

Por default, se utiliza la Regla de Sturges:

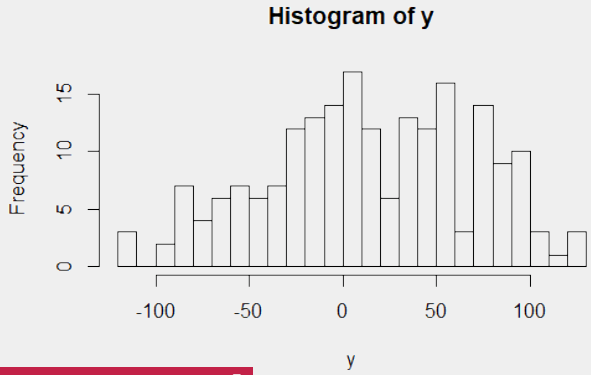
```
#Sturges

n<-length(y)
ceiling(log(n,2))+1
[1] 9

hist(y, breaks=9) #por default
```



Variantes de la función `hist`

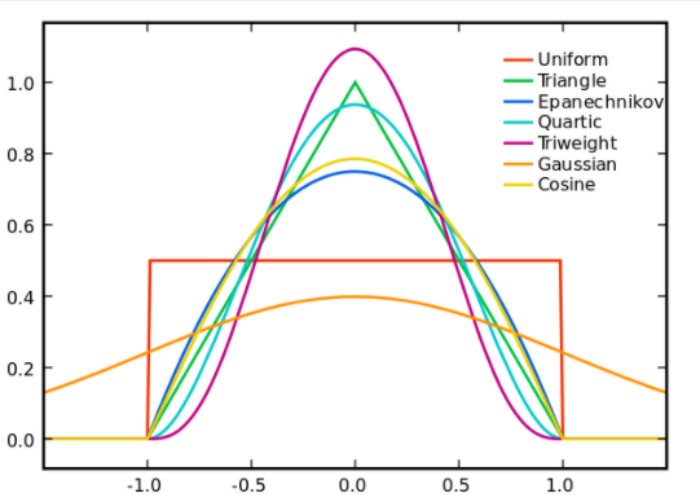
Si utilizamos la Regla de Freedman-Diaconis:

```
#Freedman-Diaconis

r<-IQR(y)
2*r/n^(1/3)
[1] 26.31567

hist(y, breaks=26)
```

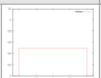
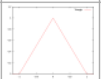
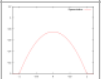
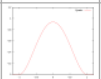
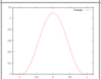
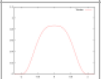
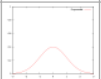


Vía Kernel

Distintos kernels, distintos pesos

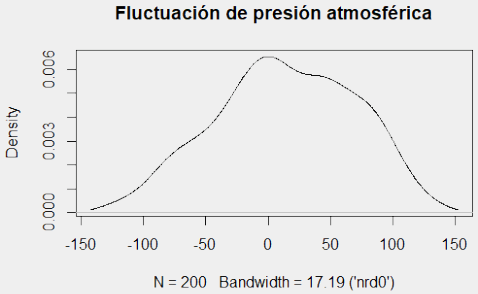


Ver fuente `https://www.wikiwand.com/en/Kernel_(statistics)`

Kernel Functions, $K(u)$			$\int u^2 K(u)\,du$	$\int K(u)^2\,du$	Efficiency[n] relative to the Epanechnikov kernel
Uniform ("rectangular window")	$K(u) = \frac{1}{2}$ Support: $\lvert u\rvert \le 1$	 "Boxcar function"	$\frac{1}{3}$	$\frac{1}{2}$	92.9%
Triangular	$K(u) \quad (1 - \lvert u\rvert)$ Support: $\lvert u\rvert \le 1$		$\frac{1}{6}$	$\frac{2}{3}$	98.6%
Epanechnikov (parabolic)	$K(u) = \frac{3}{4}(1 \quad u^2)$ Support: $\lvert u\rvert \le 1$		$\frac{1}{5}$	$\frac{3}{5}$	100%
Quartic (biweight)	$K(u) \quad \frac{15}{16}(1 \quad u^2)^2$ Support: $\lvert u\rvert \le 1$		$\frac{1}{7}$	$\frac{5}{7}$	99.4%
Triweight	$K(u) \quad \frac{35}{32}(1 \quad u^2)^3$ Support: $\lvert u\rvert \le 1$		$\frac{1}{9}$	$\frac{350}{429}$	98.7%
Tricube	$K(u) \quad \frac{70}{81}(1 \quad \lvert u\rvert^3)^3$ Support: $\lvert u\rvert \le 1$		$\frac{35}{243}$	$\frac{175}{247}$	99.8%
Gaussian	$K(u) = \frac{1}{\sqrt{2\pi}} e^{\quad \frac{1}{2}u^2}$		1	$\frac{1}{2\sqrt{\pi}}$	95.1%

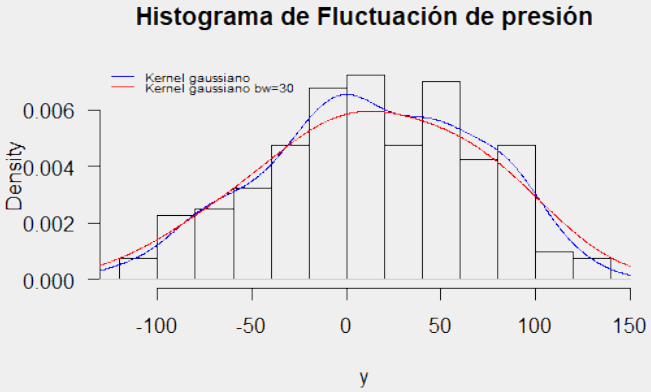
Kernel gaussiano

```
kernelg <- kdensity(y, kernel = "gaussian") #Calcula un óptimo de bandwith
summary(kernelg)
Call:
kdensity(x = y, kernel = "gaussian")

Data:         y (200 obs.)
Bandwidth:    17.19 ('nrd0')
Support:      (-Inf, Inf)
Kernel:       gaussian
Start:        uniform
Range:        (-117.4, 127.7)
NAs in data: FALSE
Adjustment:   1

plot(kernelg, main = "Diferencias de presión")
```



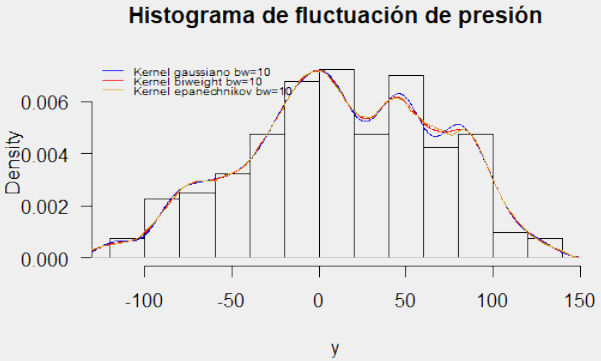
Kernel gaussiano cambiando el bandwith

```
hist(y, freq=FALSE, main="Histograma de diferencias de presión", las=1)
lines(kernelg, col="blue")
lines(kernelg30, col="red")

legend("topleft", c("Kernel gaussiano",
                    "Kernel gaussiano bw=30"), lty = "solid", lwd=1,
       col=c("blue", "red"), bty="n", cex=0.6)
```



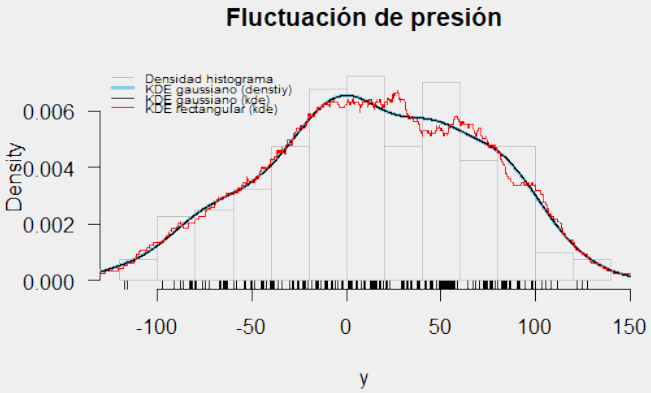
Otros kernel

```
kernelg10 <- kdensity(y, kernel = "gaussian", bw=10)
kernelb10 <- kdensity(y, kernel = "biweight", bw=10)
kernele10 <- kdensity(y, kernel = "epanechnikov", bw=10)

hist(y, freq=FALSE, main="Histograma de fluctuación de presión", las=1)
lines(kernelg10, col="blue")
lines(kernelb10, col="red")
lines(kernele10, col="goldenrod3")
legend("topleft", c("Kernel gaussiano bw=10",
                    "Kernel biweight bw=10", "Kernel epanechnikov bw=10"), lty = "solid", lwd=1,
       col=c("blue", "red", "goldenrod3"), bty="n", cex=0.6)
```



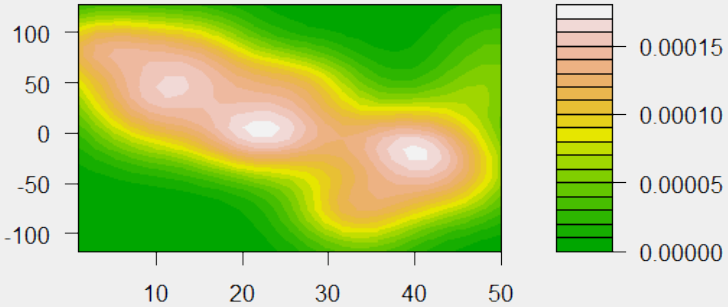
Librería demoKde

```
hist(y, freq=FALSE, border="grey", las=1, main="Diferencia de presiones")
lines(stats::density(y), col="skyblue", lwd=3)
lines(kde(y))
lines(kde(y, kernel = kernelUniform), col="red")
rug(jitter(y), col="black")
legend("topleft", c("density histogram",
                    "KDE gaussiano (denstiy)", "KDE gaussiano (kde)",
                    "KDE rectangular (kde)"), lty = "solid", lwd=c(1,8,1,1),
       col=c("grey", "skyblue", "black", "red"), bty="n", cex=0.6)
```



kde2d

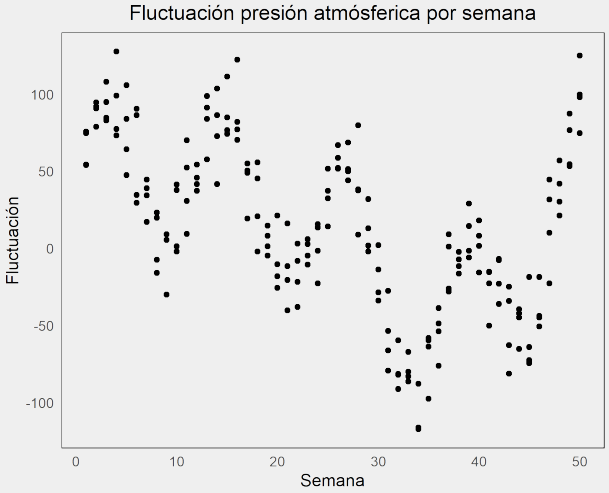
```
library(MASS)

x<-presion$x

f2hat <- kde2d(x, y)
contour(f2hat)

filled.contour(f2hat,color.palette=terrain.colors)
```



Introducción a regresión no parámetrica

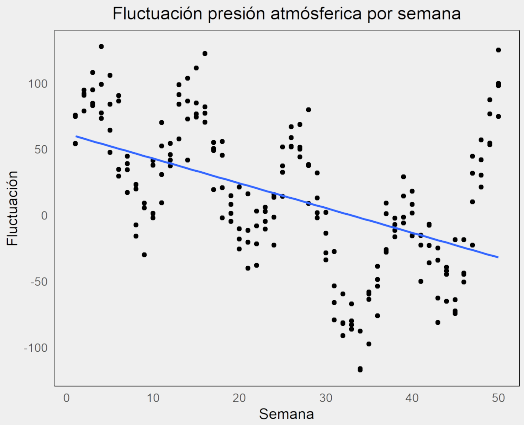
Fluctuación de presión atmosférica por semana

Si graficamos las fluctuaciones de presión atmosférica por semana, obtenemos el siguiente gráfico:



Extensiones

Evidentemente, ajustar una recta no sería lo más adhoc:



Tampoco calcular la correlación:

```
cor(x,y)
[1] -0.4920225
```

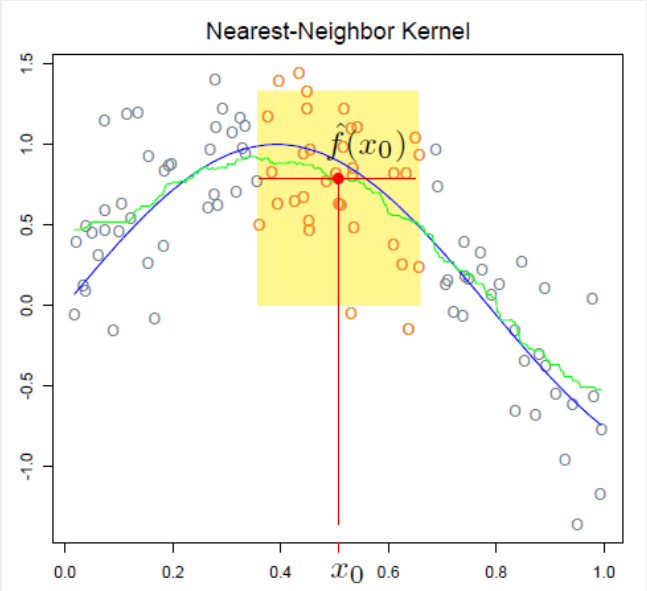
Promedios locales

Podemos plantear la misma idea del kernel pero para este caso, considerando la variable x e y.

Idea:

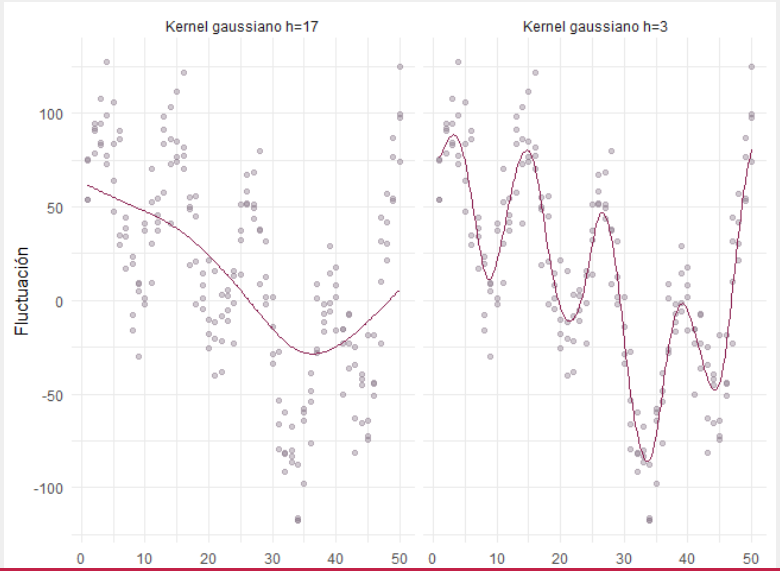
- Sea $h > 0$ un valor constante y para cada x en el eje de las abscisas considere una vecindad en torno a x de ancho $2h$, es decir, el intervalo es $(x - h, x + h)$.
- Un approach bastante simple es estimar el valor para cada x calculando el promedio de las observaciones en la vecindad en torno a x.
- **Problema de la idea anterior:** si tomo un h grande no estamos representando bien la información cerca de x (sobretodo si hay datos atípicos extremos), por lo tanto, una manera de solucionar esto es dando **pesos** a las observaciones. Estos pesos se determinan por la función $K(u)$ (el kernel).

Idea visualmente



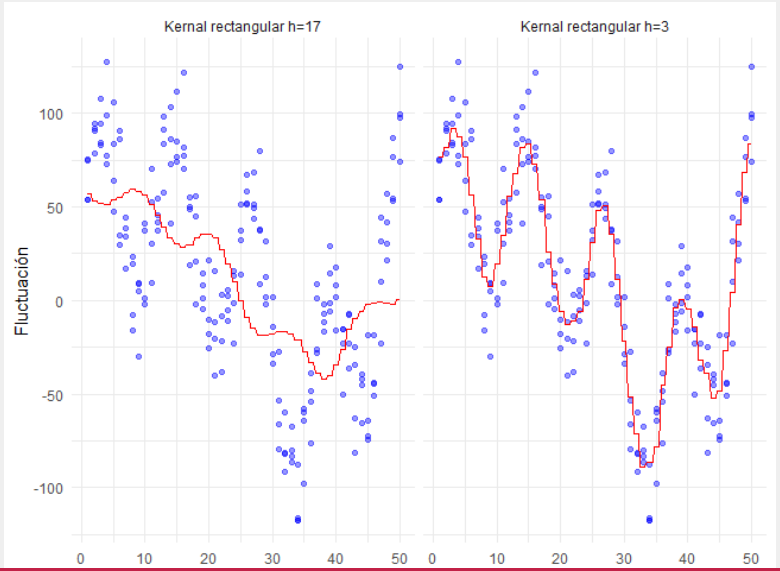
Suavizamiento gaussiano efecto del bandwith

```
suvnor<-stats::ksmooth(x,y, bandwidth = 17, kernel="normal")
suvnor2<-stats::ksmooth(x,y, bandwidth = 3, kernel="normal")
```



Suavizamiento boxcar o rectangular efecto del bandwith

```
suavbox1<-stats::ksmooth(x, y, bandwidth=17 , kernel="box")
suavbox2<-stats::ksmooth(x, y, bandwidth=3 , kernel="box")
```



Suavizamiento efecto del kernel