Reconocimiento Estadístico de Patrones

Johan Van Horebeek

horebeek@cimat.mx

1. Temario del curso

1. Métodos exploratorios para datos multivariados (y minería de datos)
2. Predicción
3. Métodos de agrupamiento
4. Métodos de clasificación
5. Métodos lineales con regularización

Referencias metodológicas:

• Johnson, R.A. y Wichern, D.W. (1992), **Applied multivariate statistical analysis**

• Bishop, C. (2006), **Pattern Recognition and Machine Learning**, Springer-verlag (en pdf)

• Duda, Hart & Stork (2001), **Pattern Classification**, Wiley (en pdf)

• Hastie, T., Tibshirani, R. y Friedman, J. (2001), **The elements of statistical learning: data mining, inference and prediction**. Springer - Verlag (en pdf).

• Ripley, **Data Mining**, Unpublished manuscript.

• Izenman (2008), **Modern multivariate statistical techniques: regression, classification and manifold learning**.

Referencias de programación:

- Venables, W. y Ripley, B. (2002), **Modern applied statistics with S**. (4a ed) Springer - Verlag
- y muchos manuales en http://www.r-project.org/

2. Algunos Ejemplos

Visualización de datos

Situación clásica:

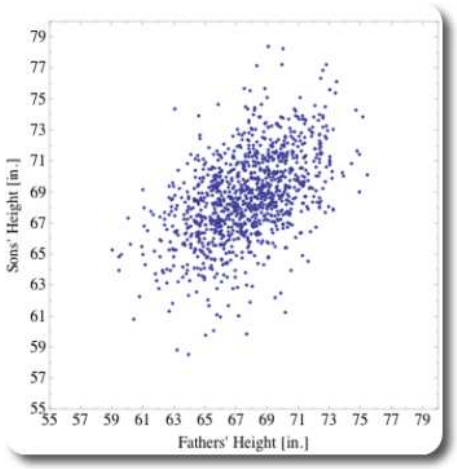
	A	B
1	120	137
2	189	187
3	160	163
4	179	189
5	156	158
6	190	195
7	164	173
8	186	176
9	...	...
10		
11		
12		
13		
14		
15		

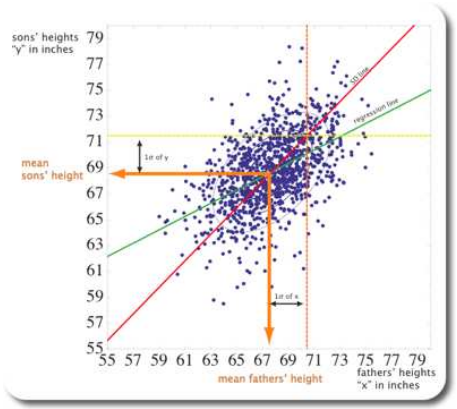
2. Algunos Ejemplos

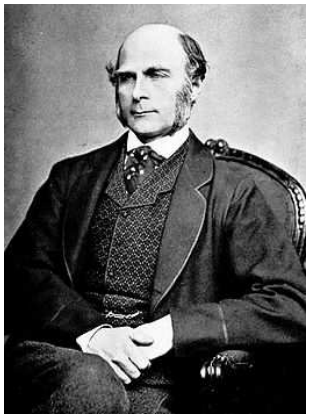
Visualización de datos

Situación clásica:

	A	B
1	120	137
2	189	187
3	160	163
4	179	189
5	156	158
6	190	195
7	164	173
8	186	176
9	...	...
10		
11		
12		
13		
14		
15		





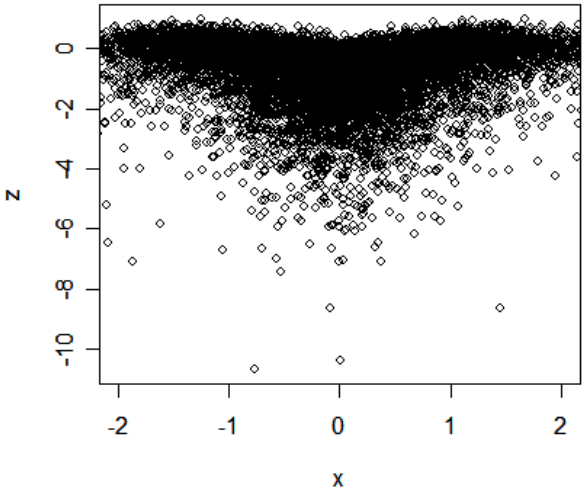


Francis Galton 1822 - 1911

2. Algunos Ejemplos

Visualización de datos

¿ Cómo manejar situaciones donde hay muchísimas observaciones?



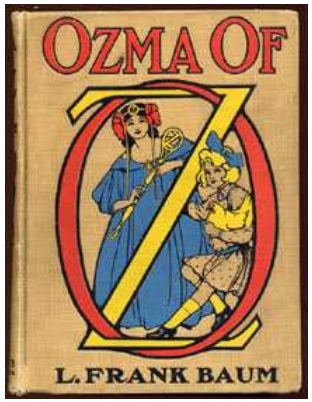
2. Algunos Ejemplos

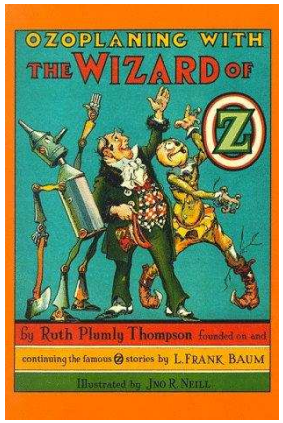
Visualización de datos

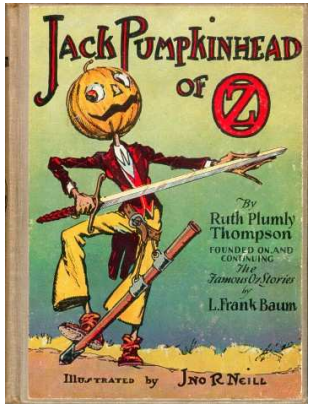
¿ Cómo manejar situaciones donde hay muchísimas observaciones?

¿ Cómo manejar situaciones donde hay muchísimas variables?

	word_1	word_2	. . . . .	word_d
doc_1	.	.	. . . . .	.
.	.	.	. . . . .	.
doc_n	.	.	. . . . .	.







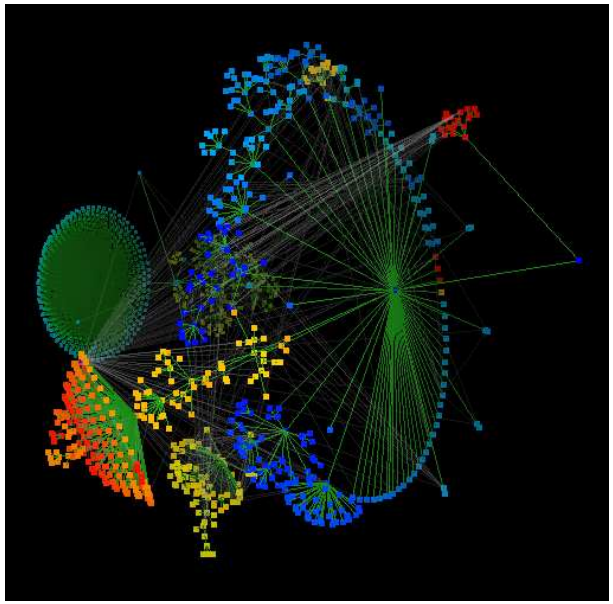
2. Algunos Ejemplos

Visualización de datos

¿ Cómo manejar situaciones donde hay muchísimas observaciones?

¿ Cómo manejar situaciones donde hay muchísimas variables?

¿ Cómo manejar situaciones cuando hay estructuras especiales?



2. Algunos Ejemplos

Visualización de datos

¿ Cómo manejar situaciones donde hay muchísimas observaciones?

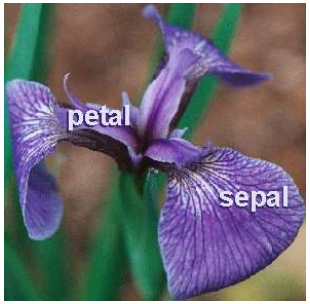
¿ Cómo manejar situaciones donde hay muchísimas variables?

¿ Cómo manejar situaciones cuando hay estructuras especiales?

¿ Cómo manejar situaciones cuando los datos no tienen una representación vectorial?

2. Algunos Ejemplos

Clasificación

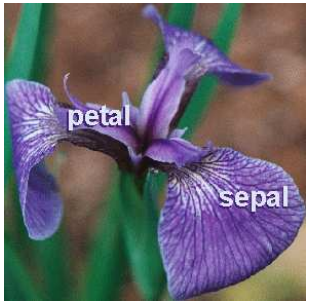


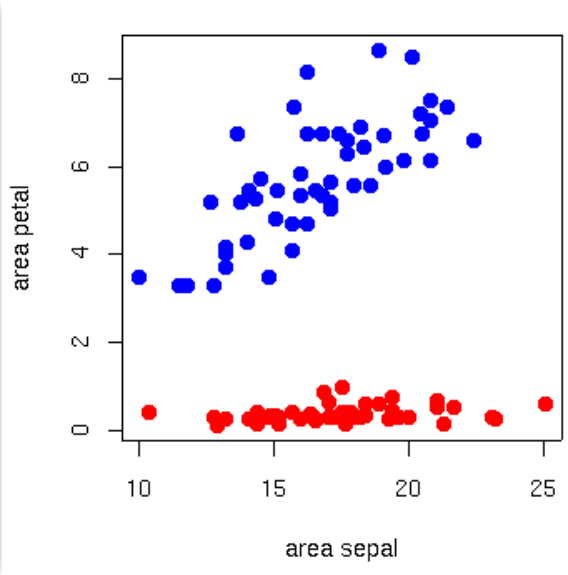
Familia Setosa

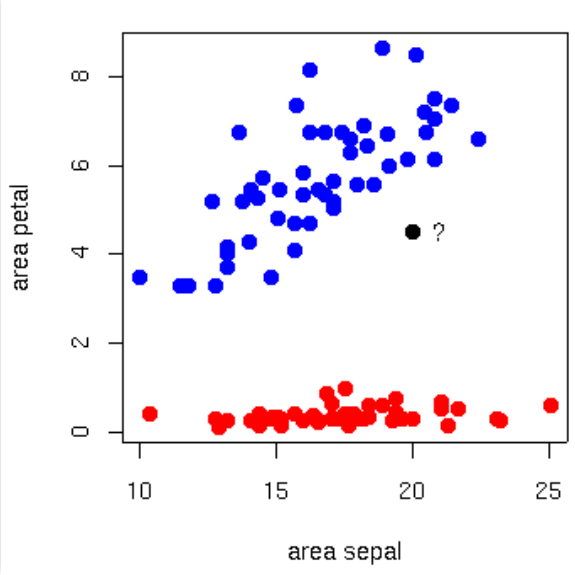
Familia Versicolor

2. Algunos Ejemplos

Clasificación





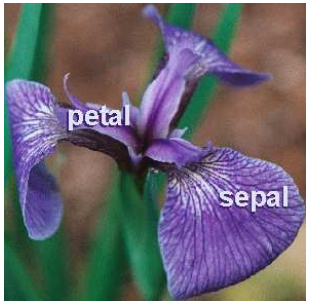


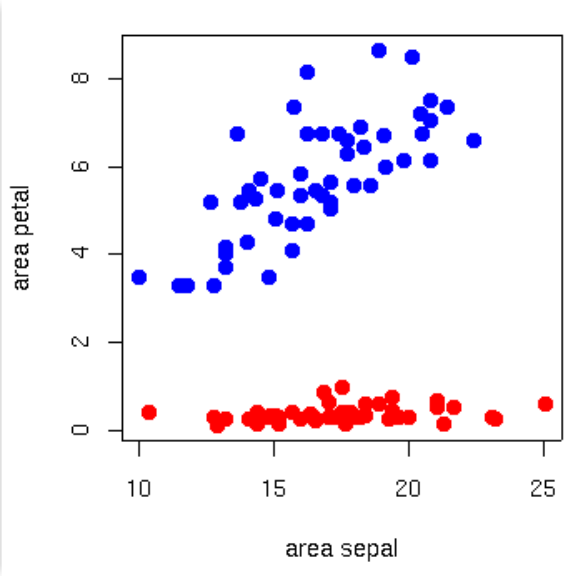
Familia Setosa

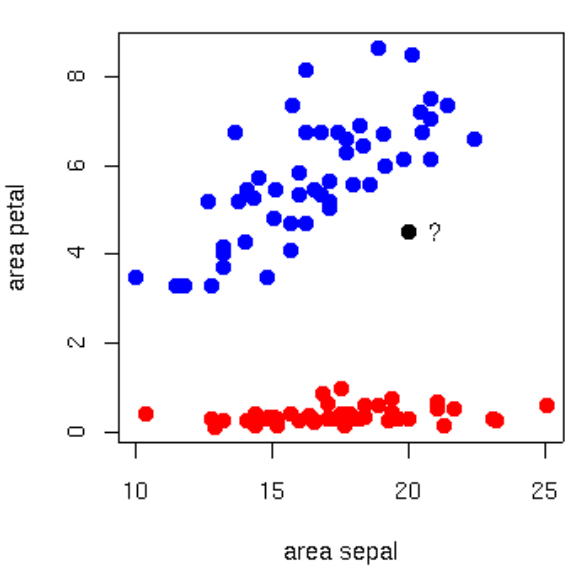
Familia Versicolor

2. Algunos Ejemplos

Clasificación







Es un problema de predicción/clasificación (reconocimiento de patrones):

$$x \to y = f(x), \quad y \in \{0, 1\} \quad x \text{ resume características de un iris}$$

Buscar, estimar o aprender $f()$ + definir x

→ Aprendizaje (Aprendizaje máquina): a partir de informacion externa al sistema (=ambiente), resolver una tarea de forma eficiente. El sistema se conoce como un sistema experto.

3. Trabajar y analizar datos grandes en R

Toma en cuenta la estructura del lenguaje

$\mathcal{R}$ no está hecho en primer lugar para procesar millones de datos pero ... se puede ganar muchisimo tomando en cuenta la estructura del lenguaje.

Hay que aprovechar la vectorización.

```
a<-1:10000; for ( i in 2 :10000) a[i]<-a[i-1];
```

Con `system.time()` se mide el tiempo de ejecución.

```
>system.time(a<-1:10000; for ( i in 2 :10000) a[i]<-a[i-1];)
[1] 5.220 0.032 5.323 0.000 0.000
```

Compara lo anterior con lo que es 50(!) veces más rápido:

```
>system.time({
 a<-1:1000000; a[2:1000000]<-a[1:(1000000-1)]
 })
[1] 0.132 0.040 0.171 0.000 0.000
```

Evita ciclos anidados; usa comandos como `apply`

```
m<-matrix(rnorm(50),ncol=10); apply(m,2,mean); apply(m,1,mean);
```

Cuida el uso de la memoria.

Problema: no puedes liberar explícitamente la memoria.

```
a<- 1:1000000; rm(a); gc() # garbage collector: libera memoria
apartada que ya no se usa
```

Cuidado con duplicar data.

```
a<- 1:1000000; b<- a # en este momento no se copia el objeto a
 b[1] <- 0 # en este momento  se copia el objeto a
```

Usa los comando adecuados.

Para datos grandes: en lugar de `read.table()` usa `scan()` .

Hay algunos trucos; especifiar parámetros explicitamente también puede ayudar.

```
A <- matrix(scan("matrix.dat", n = 200*2000), 200, 2000, byrow =T)
A<- as.matrix(read.table("matrix.dat")) #mas lento
```

Toma en cuenta la limitación de tu máquina

La cantididad de memoria depende del RAM de tu máquina y del sistema operativa.

Regla: puedes cargar datos hasta 50% del RAM sin mayor problema

Puedes extender la memoria que R tiene apartado (cf. `memory.size()`)

No carges más en la memoria de lo que necesitas.

Usa por ejemplo el paqueta `filehash` (veremos un demo después).

Conéctate directamente con la base de datos (veremos un demo después).

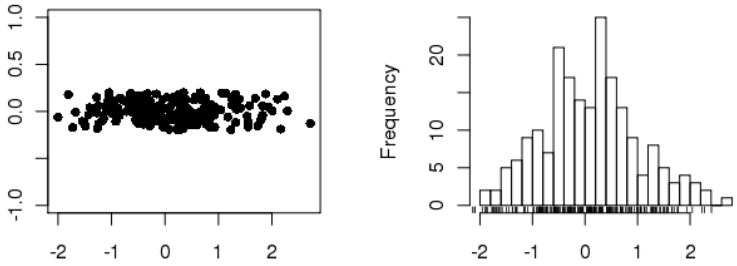
No hagas todo en R

Cada tipo de procesamiento de información tiene su lenguaje óptimo.

Usa R como plataforma de comunicación.

4. Visualizar datos unidimensionales (caso continuo)

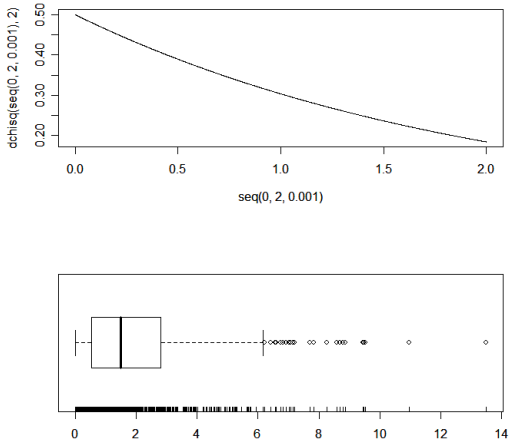
4.1 Histograma

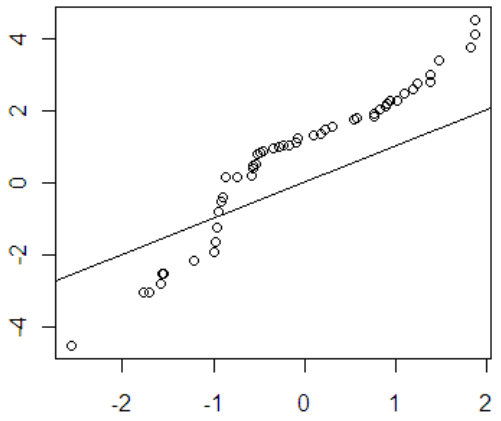



```
par(mfrow=c(1,2));
plot(data,jitter(rep(0,length(data)),amount=0.2),ylim=c(-1,1),cex=1,pch=16)
hist(data,breaks=20); rug(d);
```

Histograma: breaks=? / ancho = ? / lugar primer bin = ?

Boxplots: ciudado con multimodalidad y con distribuciones sesgadas.

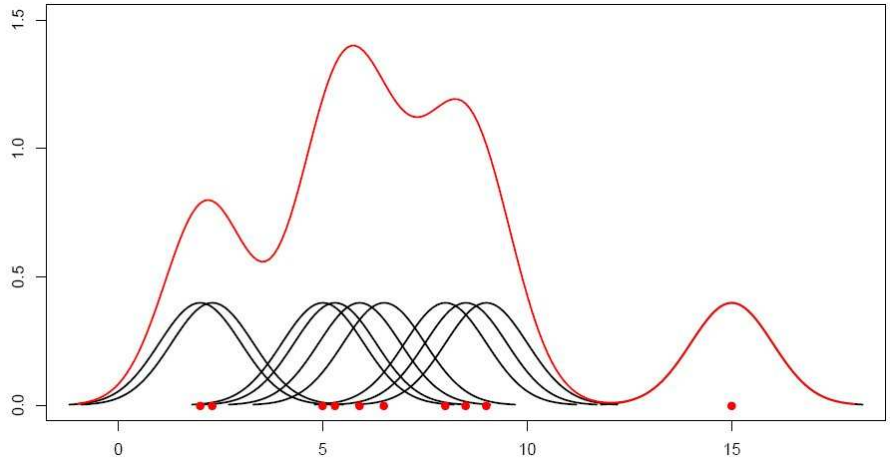




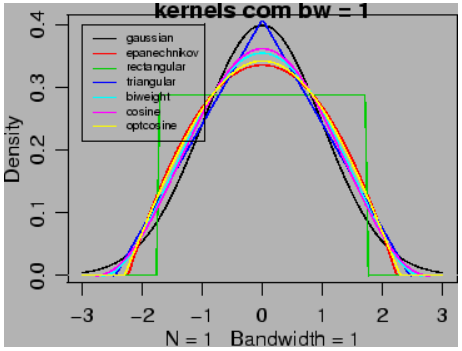
Para comparaciones: usa QQplots

4.2 Kerneles

Idea:



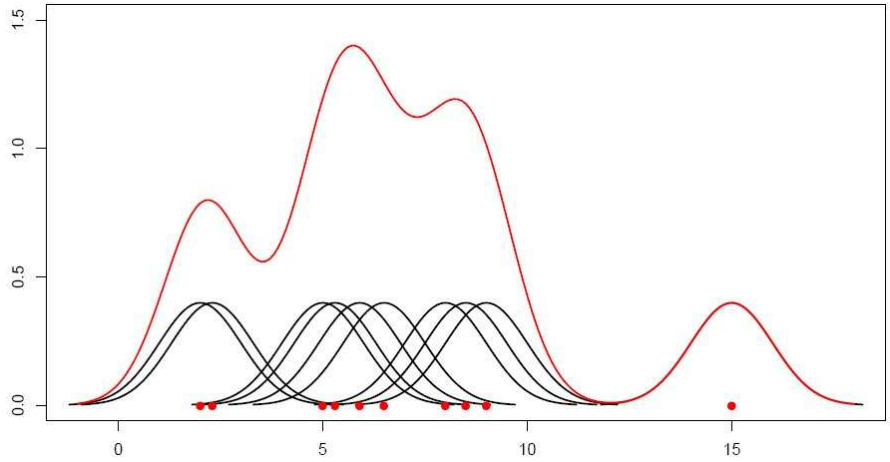
Punto de partida: una función kernel $K(x)$.



En general se elige una K simétrica (en 0) y que forma una densidad.

3.2 Kerneles

Idea:



Define $K_h(x) = \frac{1}{h}K(\frac{x}{h})$

Por ejemplo si $K \sim \mathcal{N}(0,1)$ (=es densidad de un estandard normal) y $K_h \sim \mathcal{N}(0,h^2)$

Define

$$\hat{f}_n(x) = \frac{1}{n}\sum_i K_h(x - x_i).$$

Observa: para x fija, $\hat{f}_n(x)$ es una v.a.! Por ejemplo:

$$E_{X_1,\cdots,X_n}\hat{f}_n(x) = E\frac{1}{n}\sum_i K_h(x - X_i) = EK_h(x - X_1),$$

$$= \int K_h(x - x_i)f(x_i)dx_i = (K_h * f)(x),$$

Si $X_i \sim \mathcal{N}(0,\sigma^2)$ y $K_h \sim \mathcal{N}(0,h^2)$, $(K_h * f)(x)$ es la densidad de $\mathcal{N}(0,h^2+\sigma^2)$

Dos decisiones: ¿ Cómo elegir h (=bandwidth)? y ?Cómo elegir kernel K?

Mientras el kernel es continua y derivable, la forma no influye tanto: el parámetro más importante es h.

Dos aspectos: **variabilidad** y **sesgo**

cf.: variabilidad de un estimador $Var(\widehat{\Theta}_n)$ y el sesgo $(E(\widehat{\Theta}_n) - \theta)^2$

Vemos en el demo `kernel.r` que mientras más grande h:

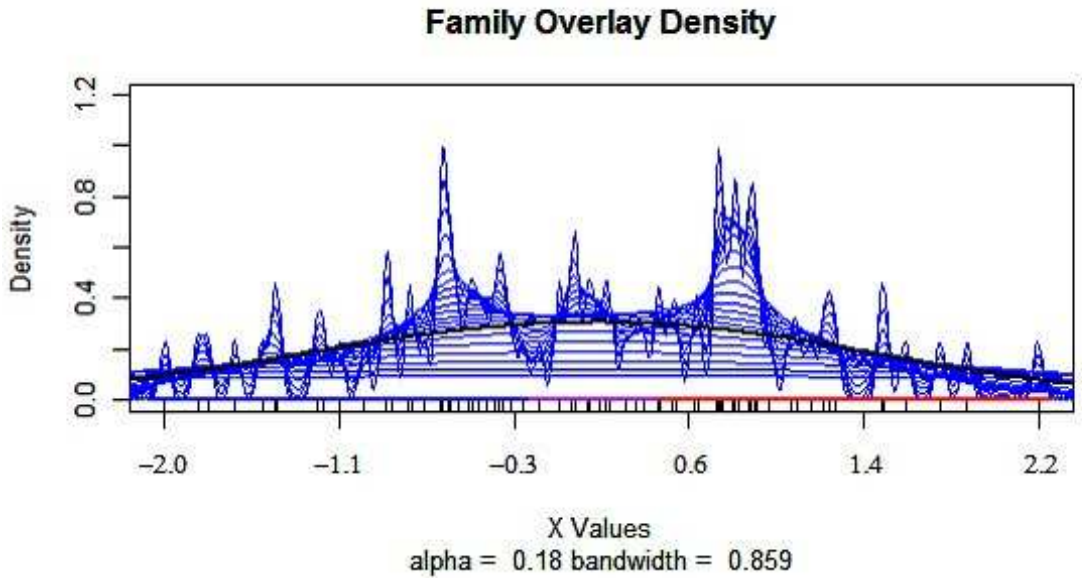
- más grande el sesgo
- más chica la variabilidad

Para n fija, este comportamento **contrario** veremos en muchas aplicaciones más adelante.

Muchas maneras para definir una h óptima; **no existe la h óptima**

INSERTAR DECOMPOSICION

Un enfoque muy particular es la de SIZER: considera h como un parámetro de escala como en imágenes.



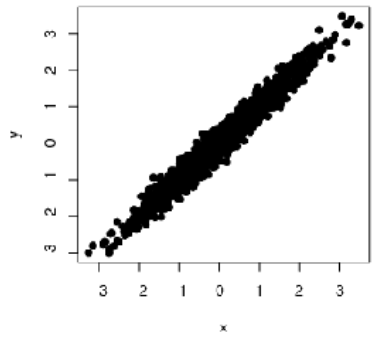

```
plot(density(faithful$eruptions, bw = 0.01, kernel="gaussian"))
```

5. Visualizar datos bidimensionales (caso continuo)

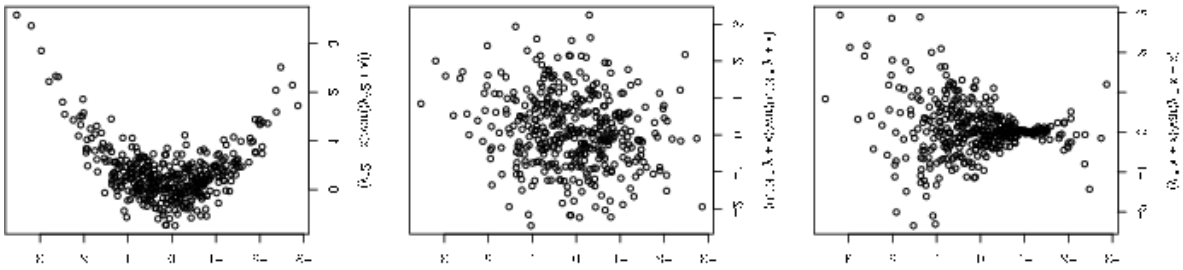
5.1 XY-Plot y la correlación

Situación idonea:

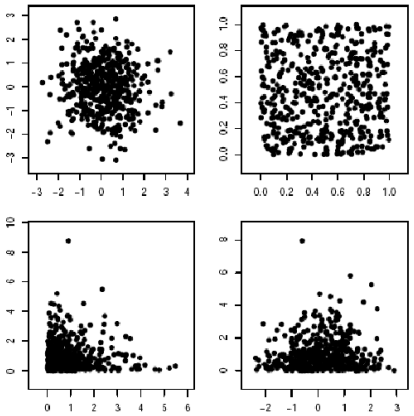
```
x<- rnorm(2000); y<- x+0.2*rnorm(2000);
plot(x,y, cex=1,pch=16)
```



Podemos cuantificar la depencia a través de la **Correlación** La correlación no funciona si las dependencias no son (casi) lineales.



Tres veces la misma correlación!!!



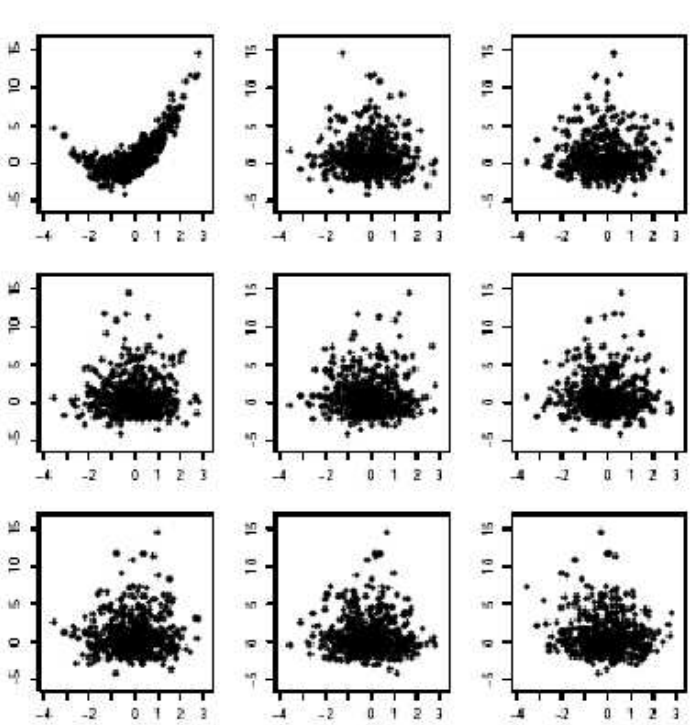
Cuatro veces hay independencia!!!

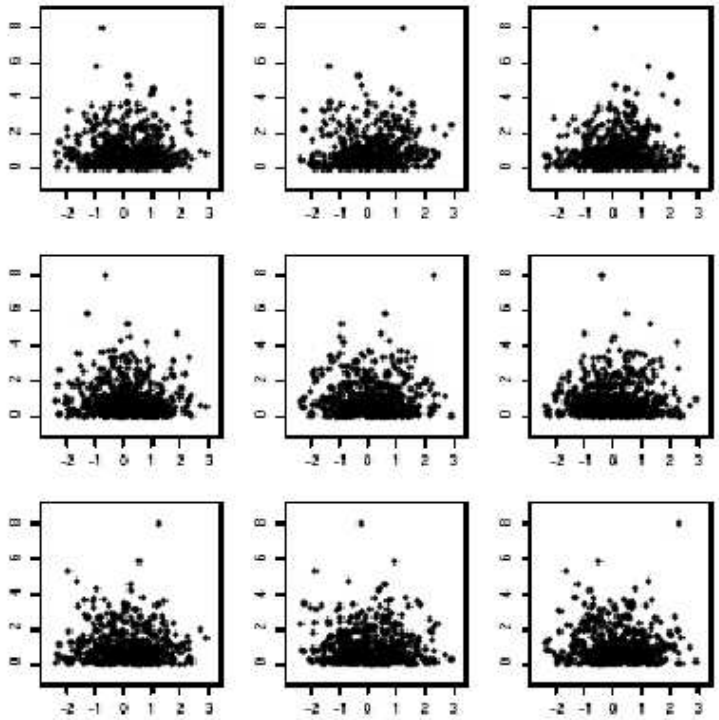
lo que cambia aquí es la distribución marginal.

Di Cook et al. sugieren usar permutaciones aleatorias: su propuesta :

permutar varias veces al azar los valores de cada variable y hacer un xy plot de cada permutación;

si no se logra distinguir visualmente los datos orginales de los demás: eso será un indicador **en favor** de la hipótesis de independencia.





5.2 Visualización de la densidad

Antes usamos para el caso 1D:

$$\hat{f}_n(x) = \frac{1}{n}\sum_i K_h(x - x_i).$$

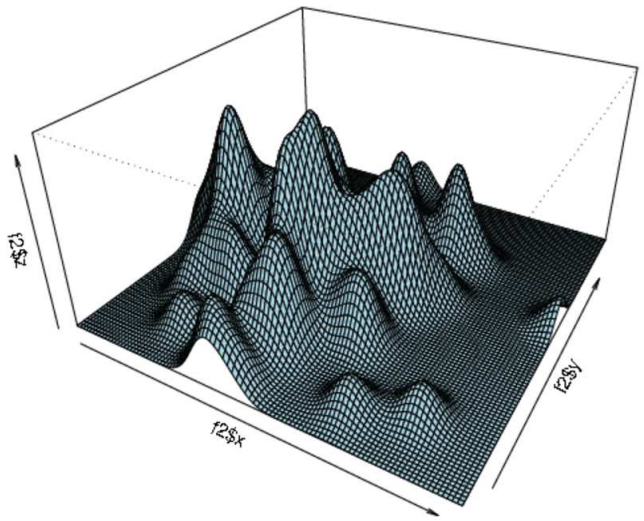
Una extensión directa es:

$$\hat{f}_n(x, y) = \frac{1}{n}\sum_i K_h(x - x_i, y - y_i).$$

¿ Cómo construir K_h en 2D? Por ejemplo, como producto de dos kerneles 1D:

$$K_h(x - x_i, y - y_i) = K_h^1(x - x_i)K_h^1(y - y_i),$$

con K^1 un kernel 1D.



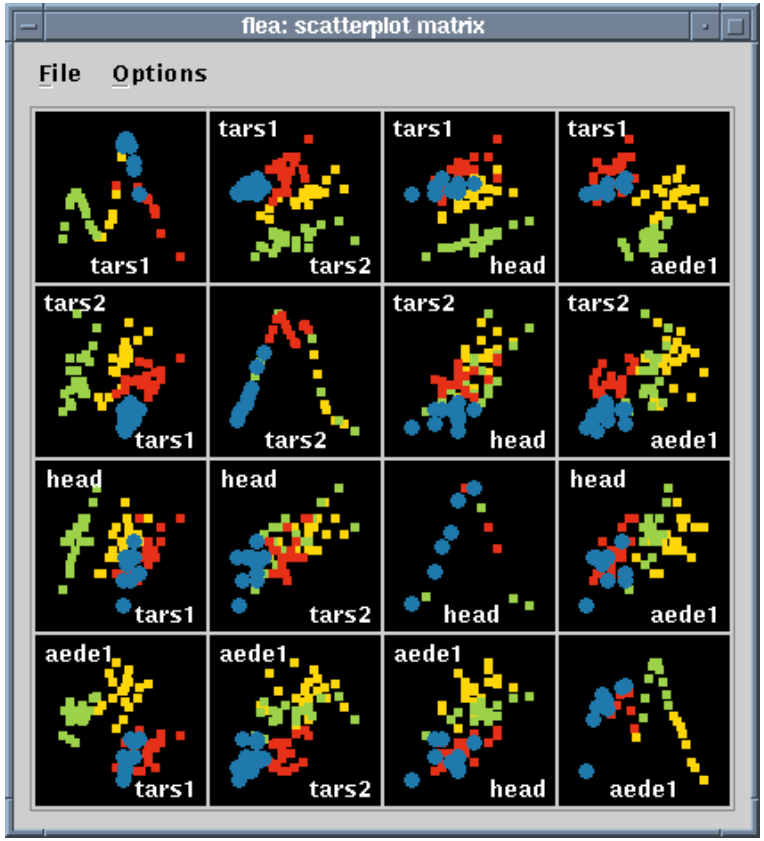
Insertar demo `kernel2.r`

6. Visualizar datos multidimensionales (caso continuo)

Dos caminos grandes: marginalizar (abstraer) y condicionar (especificar)

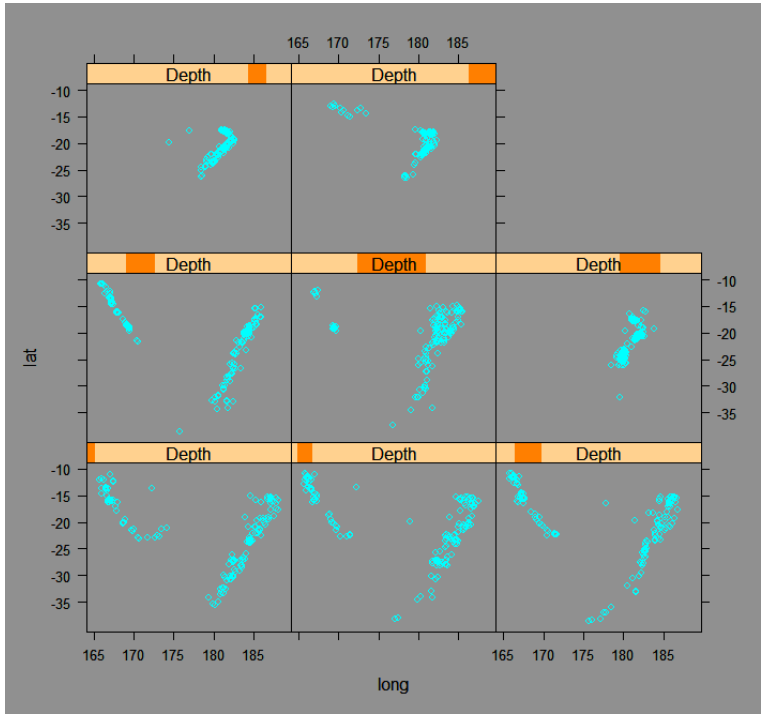
Además podemos usar color, movimiento y linking.

6.1 Pairsplot




```
pairs(data)
```

6.2 Trelisplot




```
library(lattice)
x<-rnorm(50); y<-rnorm(50); z<-rbinom(50,1,0.5)
xyplot(x ~ y  | z)

grupo<- rbinom(50,3,0.5)
xyplot(x ~ y  | z, group= grupo)
z2<-rbinom(50,1,0.5)
xyplot(x ~ y  | z*z2)
```

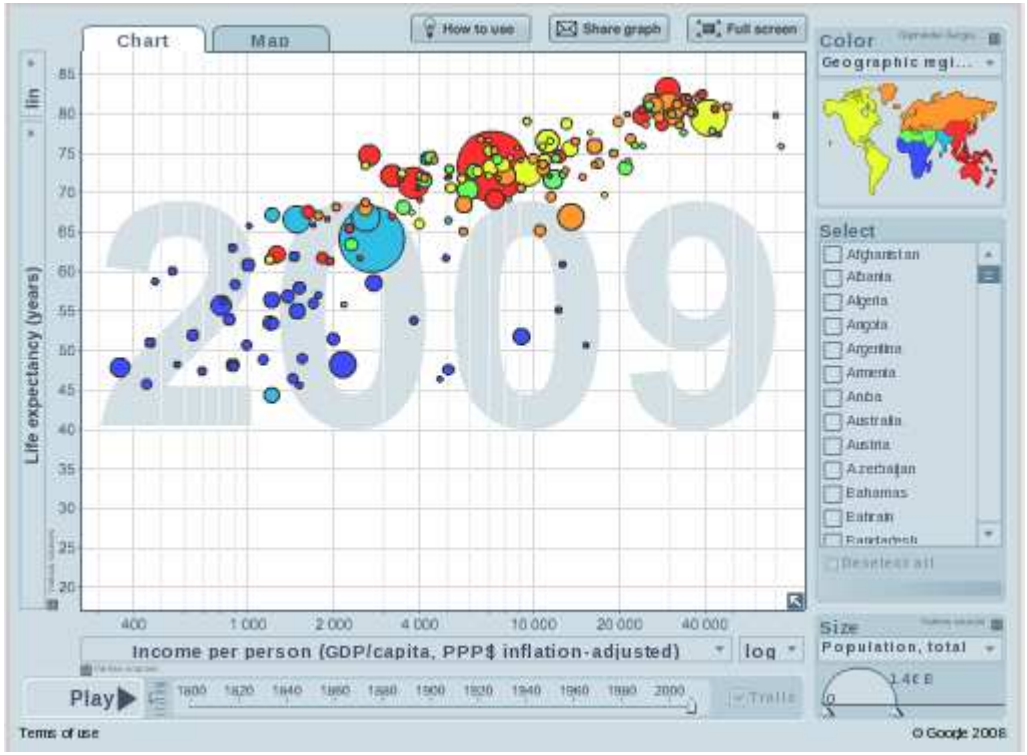
Para discretizar una variable se puede usar:

```
equal.count(rnorm(50), number=5,overlap=0.1)
zc<-cut(rnorm(50),br=-5:5)
```

Ejemplo:

```
require(stats)
Depth <- equal.count(quakes$depth, number=8, overlap=.1)
xyplot(lat ~ long | Depth, data = quakes)
```

Ejemplo bonito: bubbleplots (Gapminder)



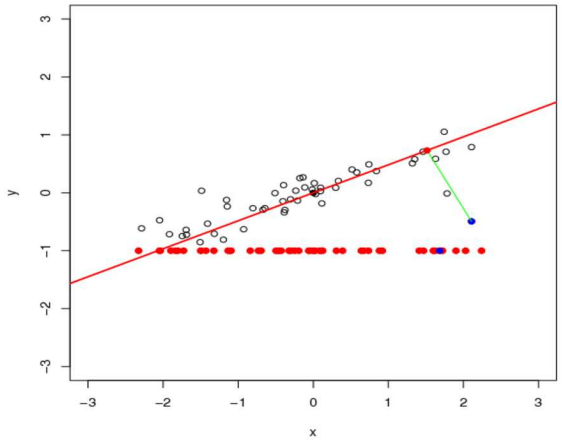
6.3. Plots basados en proyecciones

Idea: buscamos proyecciones interesantes de los datos para reducir la dimensionalidad.

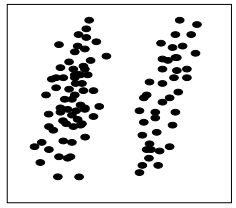
Muchas maneras para definir cuando una proyección es interesante.

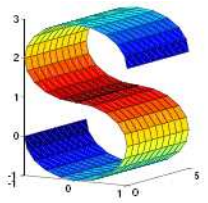
Elección más popular: **maximizar la varianza**.

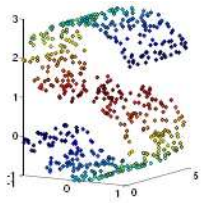
$\Rightarrow$ es el punto de partida de **análisis de componentes principales** (PCA)



no siempre buena idea:







Insertar `demo1.r`.

6.3.1 PCA

Se puede introducir como método estadístico o geométrico.

• Si consideramos $X = (X_1, \cdots, X_d)$ como v.a.:

$$\max_{||l||=1} Var(l'X)$$

.

$$\max_{||l||=1} Var(l'X) \leftrightarrow \max_{l} \frac{Var(l'X)}{||l||^2} = \max_{l} \frac{l'Cov(X)l}{||l||^2}$$

Solución: l es el primer vector propio de $Cov(X)$.

En general, buscamos direcciones l_i tal que

$$\max_{||l_i||=1} Var(l_i'X), \qquad l_i \perp l_1, \cdots, l_{i-1}$$

Solución: $\{l_i\}$ son los vectores propios de $Cov(X)$.

Llamamos $Y_i = l_i'X$; son v.a. decorrelacionadas.

PCA para predicción óptima

Busca transformaciones lineales $X \to Y \to \hat{X}$ con Y de dim. $d_2 < d$ tal que $E||X - \hat{X}||^2$ es mínima.

Si $EX = 0$ y $d_2 = 1$:

$$X\ (\in \mathcal{R}^d) \implies Y = \langle l, X \rangle\ (\in \mathcal{R}^1) \implies \hat{X} = Yl\ (\in \mathcal{R}^d)$$

$$\text{tal que } E||X - \hat{X}||^2 \text{ sea mínima.}$$

Solución: Toma la proyección sobre los primeros d_2 componentes principales.

¿ Cómo sabemos que una proyeccion es informativa?

No existe una respuesta única.

Lo más usado es a través de la varianza; si los datos son centrados:

$$E||X - \hat{X}||^2 = \sum_{i=d_2+1}^{d} var(Y_i) = \sum_{i=d_2+1}^{d} \lambda_i$$

Con $\{\lambda_i\}$ los valores propios correspondientes.

$\Rightarrow$ se trabaja con los valores relativos $\dfrac{\sum_{i=d_2+1}^{d} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$