

Proyecciones

a.k.a.: PCA – Principal
components analysis y
otros

Outline

- Motivación
- Derivación
- Ejemplos
- Otras proyecciones

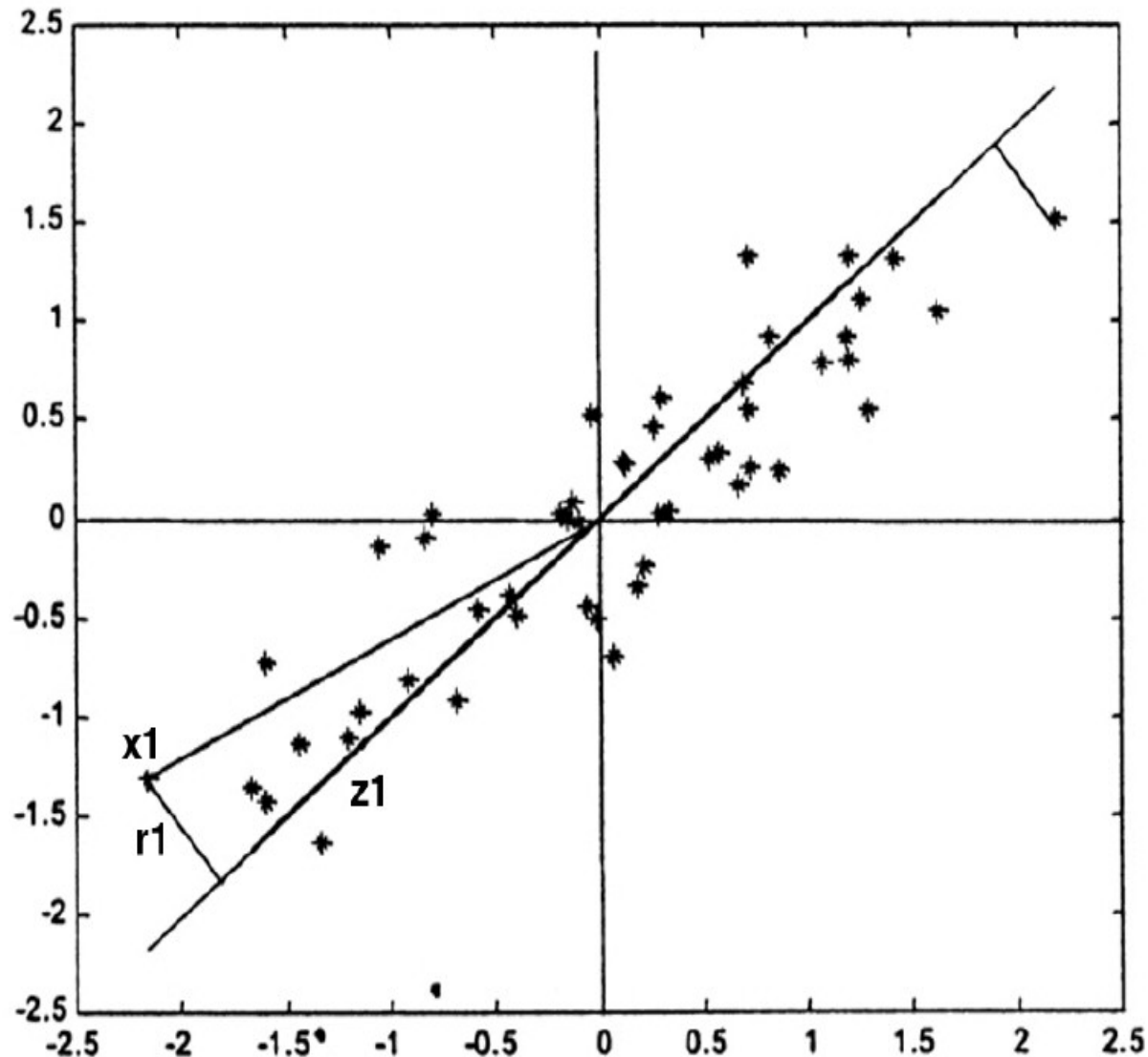
Motivación general

- Tenemos un dataset \mathbf{X} con n datos y p dimensiones, centrado (medias 0).
- Queremos encontrar una representación de esos mismos datos en un espacio de menor dimensión
 - Para visualizarlos (entenderlos!)
 - Para modelarlos mejor
- Cuál es la proyección óptima?

Motivación

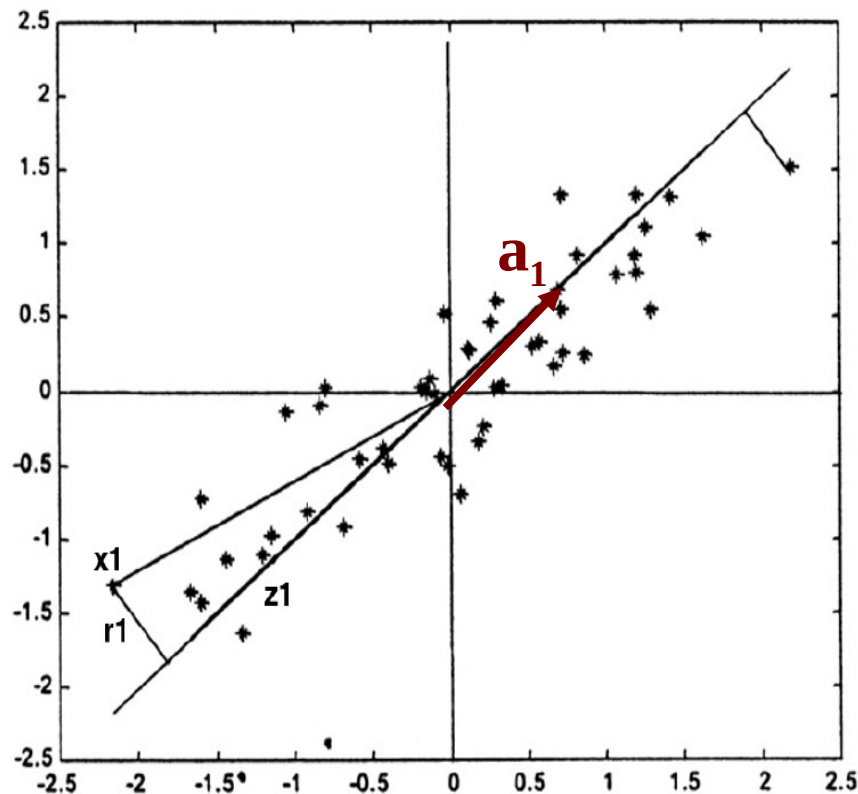
- Queremos una proyección lineal tal que los datos conserven lo más posible su estructura (relación de distancias).
- Ejemplo:
 - Datos en 2 dimensiones.
 - Proyectamos sobre una recta
 - Cuál es la recta que introduce menos distorsión?

Motivación



- La recta debe pasar cerca de la mayoría de los puntos.
- La distancia entre los puntos y entre sus proyecciones sobre la recta deben parecerse lo más posible

Notación



- Para que los **Z** y los **X** sean similares tenemos que minimizar a los **r**

- Versor paralelo a la recta
 $\mathbf{a}_1 = (a_{11}, a_{12}, \dots, a_{1p})$
- $Z_1 = \mathbf{a}_1' \mathbf{X}_1$
- $\mathbf{Z}_1 = Z_1 \mathbf{a}_1$
- Pitágoras:
 $\mathbf{x}_1' \mathbf{x}_1 = Z_1^2 + r_1^2$

$$\sum_{i=1}^n \mathbf{x}_i' \mathbf{x}_i = \sum_{i=1}^n z_i^2 + \sum_{i=1}^n r_i^2.$$

Motivación

$$\sum_{i=1}^n \mathbf{x}_i' \mathbf{x}_i = \sum_{i=1}^n z_i^2 + \sum_{i=1}^n r_i^2.$$

- El primer miembro es constante
- Para minimizar el último sumando tengo que maximizar el primer sumando.
- Como los Z tienen media cero, estamos:

MAXIMIZANDO LA VARIANZA DE LAS PROYECCIONES

Definición

- La primer componente principal es la dirección en la cual los datos tienen máxima varianza
 - Dirección=combinación lineal de las variables originales
- La segunda componente principal es la dirección ortogonal a la anterior en la cual hay máxima varianza
 - Es la primer componente del subespacio que queda
- hasta p...

Derivación

Buscamos el vector \mathbf{a}_1 : $\mathbf{z}_1 = \mathbf{X}\mathbf{a}_1$

tal que la $\text{var}(\mathbf{Z}_1)$ sea máxima.

Como \mathbf{Z}_1 tiene media 0: $\frac{1}{n}\mathbf{z}_1'\mathbf{z}_1 = \frac{1}{n}\mathbf{a}_1'\mathbf{X}'\mathbf{X}\mathbf{a}_1 = \mathbf{a}_1'\mathbf{S}\mathbf{a}_1$

donde \mathbf{S} es la matriz de covarianza de los datos.

Para que el problema tenga sentido pedimos que \mathbf{a} sea de modulo unitario: $\mathbf{a}_1'\mathbf{a}_1 = 1$

Derivación

- La restricción se introduce con un multiplicador de Lagrange:

$$M = \mathbf{a}'_1 \mathbf{S} \mathbf{a}_1 - \lambda (\mathbf{a}'_1 \mathbf{a}_1 - 1)$$

Queremos el máximo, derivamos:

$$\frac{\partial M}{\partial \mathbf{a}_1} = 2\mathbf{S} \mathbf{a}_1 - 2\lambda \mathbf{a}_1 = 0,$$

$$\mathbf{S} \mathbf{a}_1 = \lambda \mathbf{a}_1$$

Derivación

- Las direcciones principales son los autovectores de la matriz de covarianza
- Como: $\mathbf{a}_1' \mathbf{S} \mathbf{a}_1 = \lambda \mathbf{a}_1' \mathbf{a}_1 = \lambda$

La varianza de z es el autovalor. La primer DP es la del mayor autovalor. Las otras sucesivamente.

Complejidad

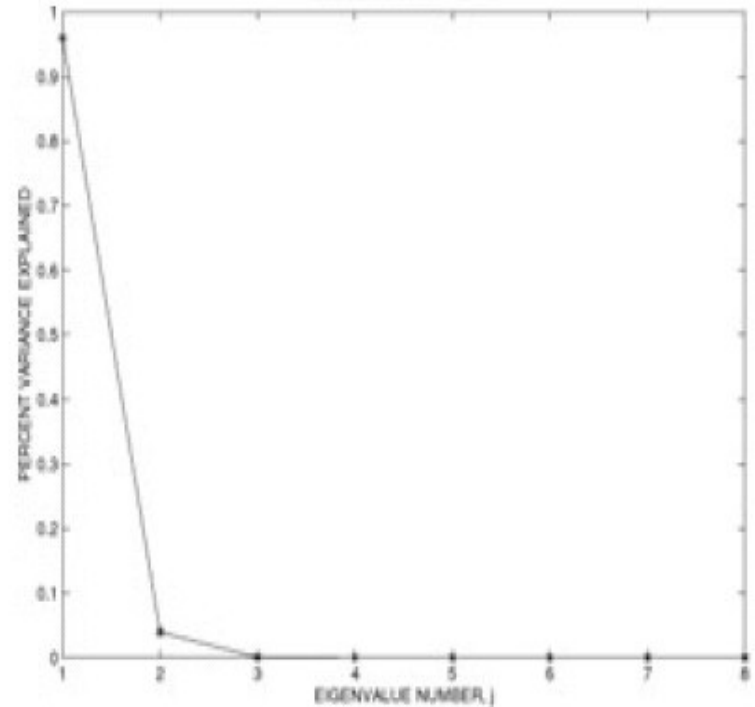
- Es del orden de $O(np^2 + p^3)$
- Para calcular S : np^2
- Para autovalores de S : p^3
- Lineal en n , pero cúbico en p !

Uso práctico

- Cuántas dimensiones necesito conservar?
- Varianza explicada: $\lambda / \sum \lambda$
- Dejar las direcciones que conservan “casi toda” la varianza

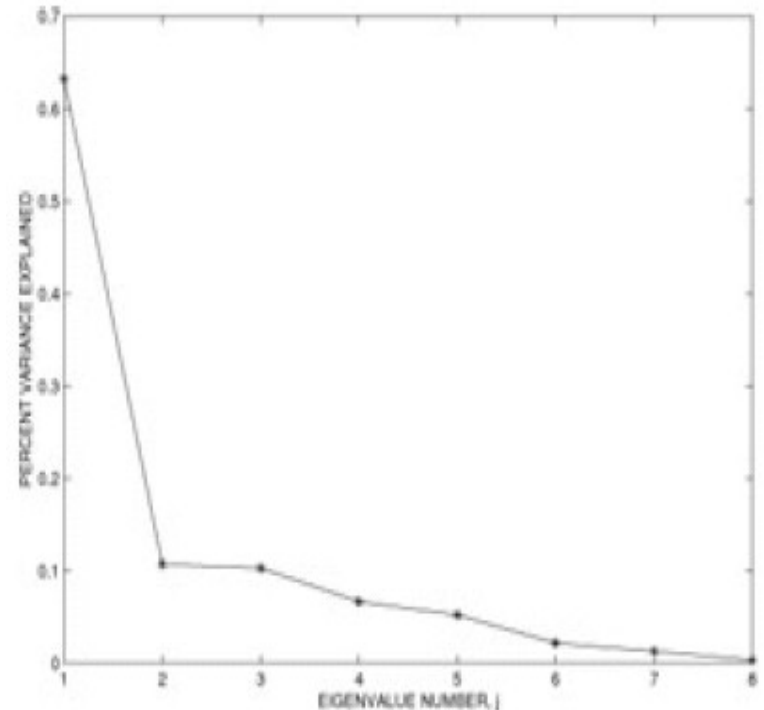
Correlación

- Datos sin normalizar – distintas varianzas para cada variable original
- Cuando las relaciones entre las variables tienen sentido físico-real



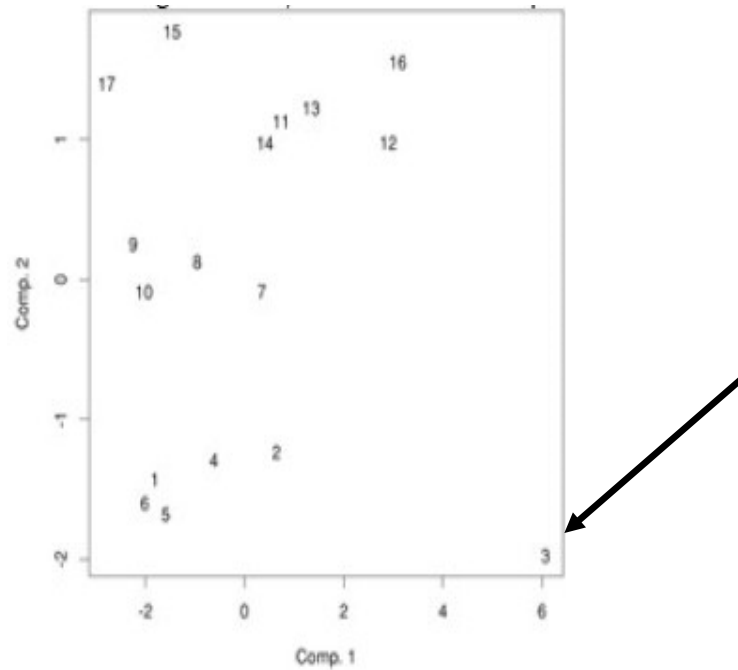
Covarianza

- Datos normalizados previamente (z-scores)
- Cuando las variables son inconmensurables (no tienen relación)
- La mayoría de los casos reales



Ejemplos

- Detección de outliers usando PCA



Ejemplos en R

```
datos<-crea.datos(2000,d=9,n.ce=0,C=1)
```

```
plot(datos[,1:5],col=datos[,10])
```

```
datos.pca<-prcomp(datos[,10],ret.x=T)
```

```
x11()
```

```
plot(datos.pca)
```

```
x11()
```

```
plot(datos.pca$x[,1:2],pch=17,col=datos[,10])
```

#en 100 dimensiones:

```
datos<-crea.datos(2000,d=100,n.ce=0,C=0.3)
```

```
x11()
```

```
plot(datos[,1:5],col=datos[,101])      #verificar que son las mismas variables
```

```
datos.pca<-prcomp(datos[,101],ret.x=T)
```

```
x11()
```

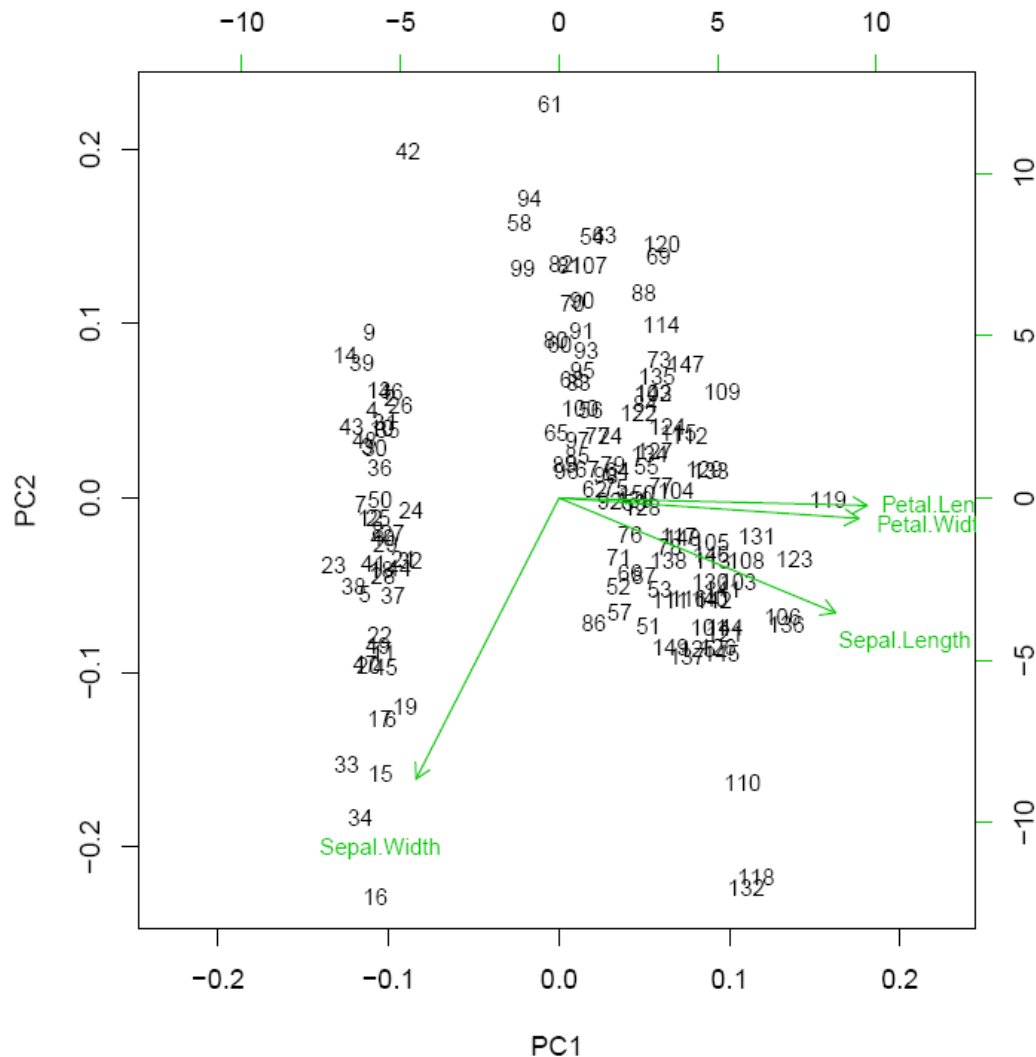
```
plot(datos.pca$x[,1:2],pch=17,col=datos[,101])
```

Interpretación

- PCA hace una proyección lineal, un giro de los ejes.
- Cada “eje principal” es una combinación lineal de las variables originales.
- Se puede buscar evidencia sobre la importancia de las variables originales evaluando su aporte a las PC.
- Se suele hacer gráficamente

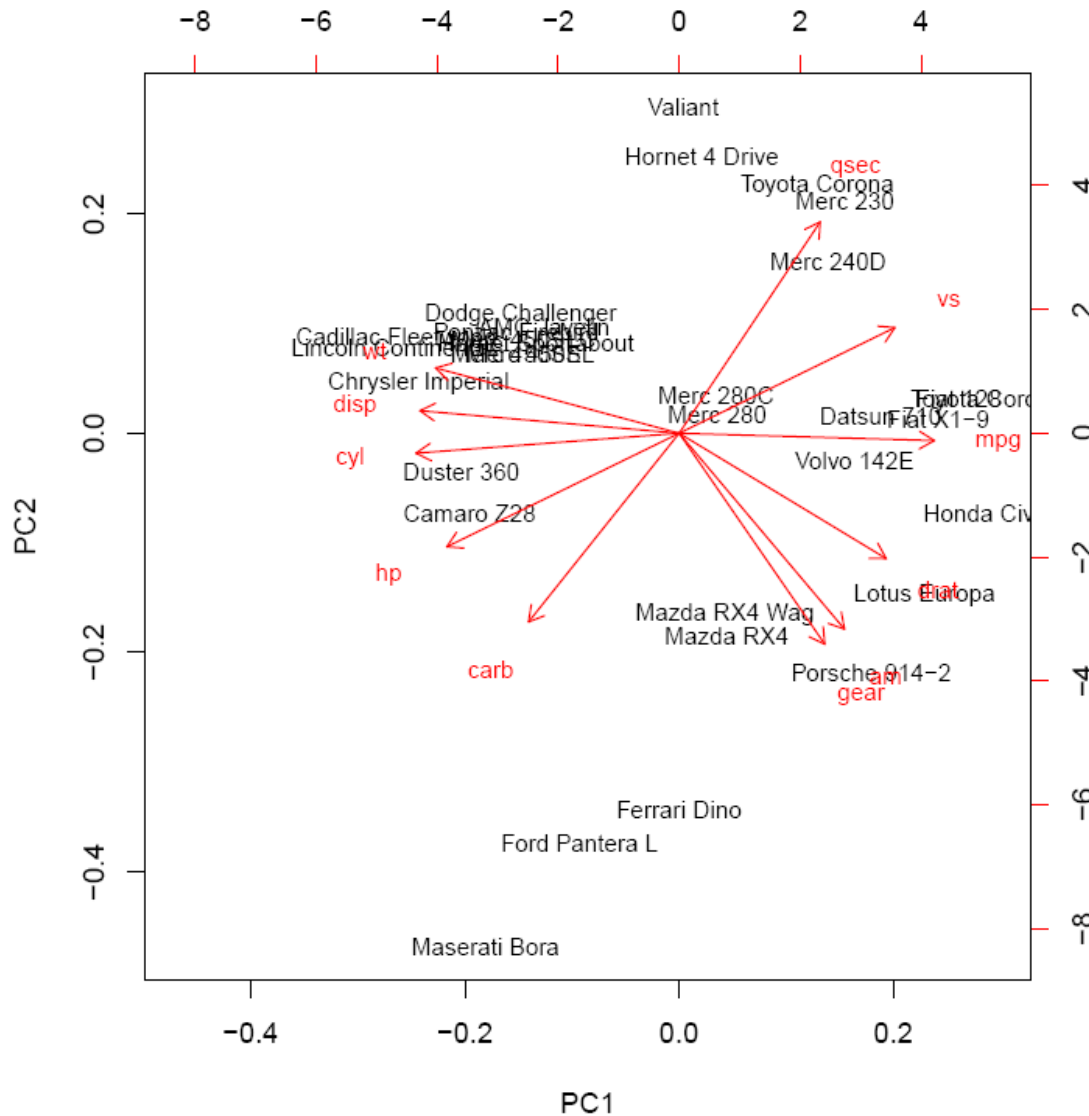
Interpretación: gráficas

```
>biplot(prcomp(iris[,-5], scale = TRUE))
```



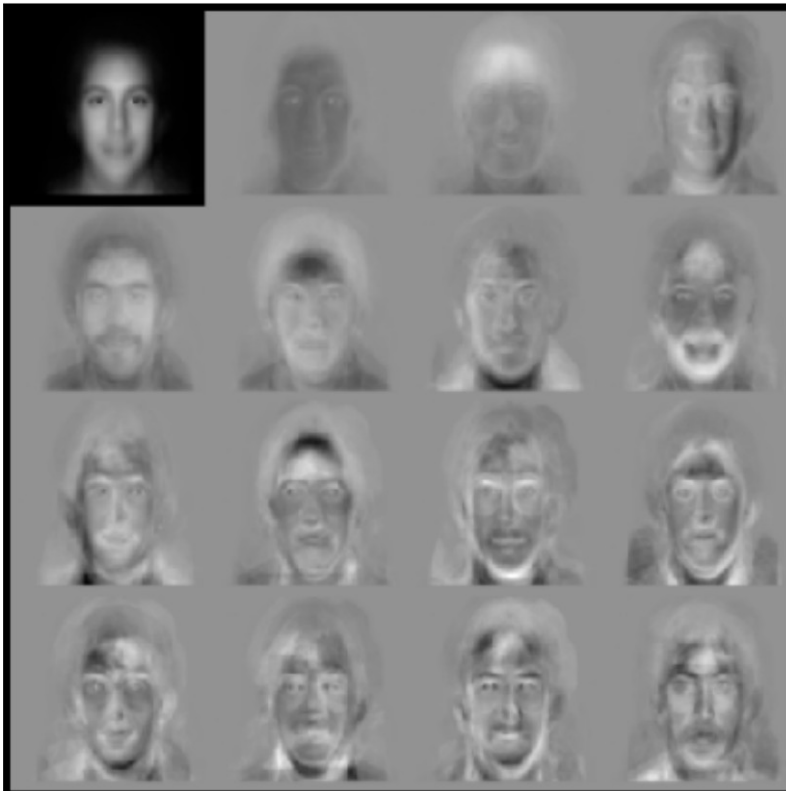
Interpretación: gráficas

```
>biplot(prcomp(mtcars, scale = TRUE),choices=c(1,2),cex=0.8)
```



Interpretación: eigenfaces

- Tomo imágenes de caras, centradas, en ByN.
- Cada cara es un vector en un espacio dado, cada coordenada es el valor del pixel correspondiente
- Puedo hacer PCA y buscar las direcciones principales



- La imagen completa es una combinación lineal de las “mascaras” de cada autovector
- Cada autovector de menor peso explica detalles menos importantes de la imagen

Resumen

- PCA: Proyecciones lineales en bajas dimensiones que explican los datos lo mejor posible.
- Las direcciones principales son los autovectores de la matriz de covarianza.
- La importancia de cada dirección esta relacionada con su varianza explicada (autovalores).

Extensión: Multi-Dimensional Scaling

- MDS. Método general.
- Busca representar los datos en un espacio de menor dimensión respetando las distancias entre los objetos.
- MDS se basa sólo en distancias entre puntos. No se necesita que los puntos estén en un espacio vectorial. Sólo las distancias. Por ejemplo:
 - grados arbitrarios de similitud entre vinos
 - cantidad de nodos entre dos computadoras

Extensión: Multi-Dimensional Scaling

- PCA es un caso muy particular de MDS
- MDS construye el equivalente a una matriz de covarianza, de la que se extraen los autovectores y autovalores, que son las componentes proyectadas

Ejemplo de MDS

- Código R:

```
loc <- cmdscale(eurodist)
```

```
x <- loc[,1]
```

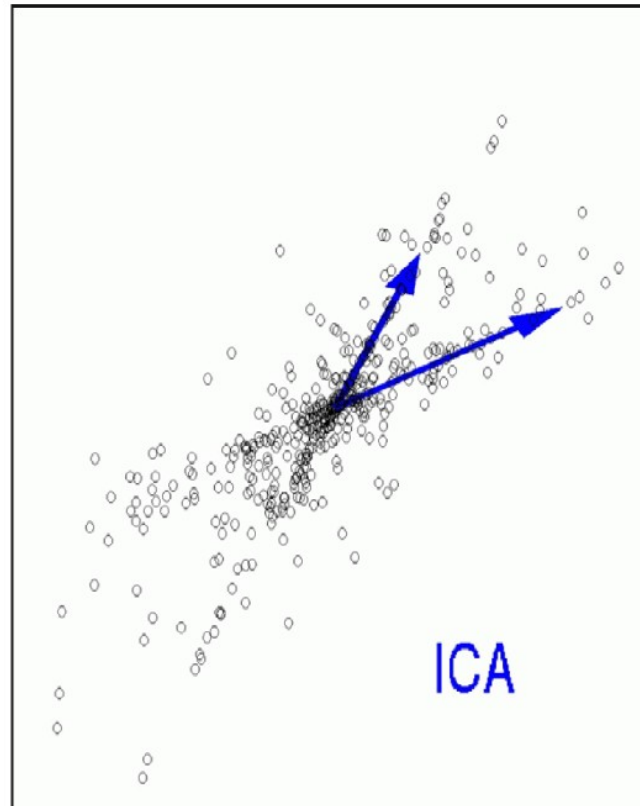
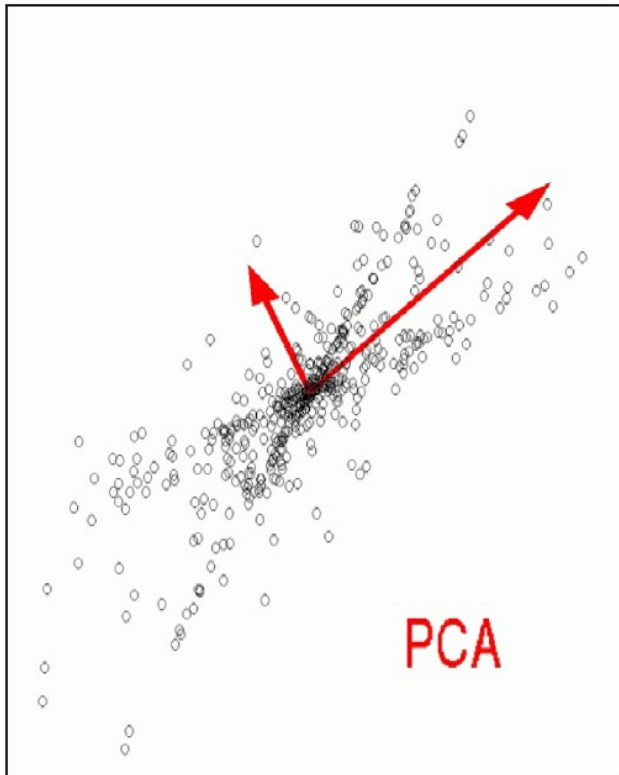
```
y <- -loc[,2]
```

```
plot(x, y, type="n", xlab="", ylab="",  
      main="cmdscale(eurodist)")
```

```
text(x, y, rownames(loc), cex=0.8)
```

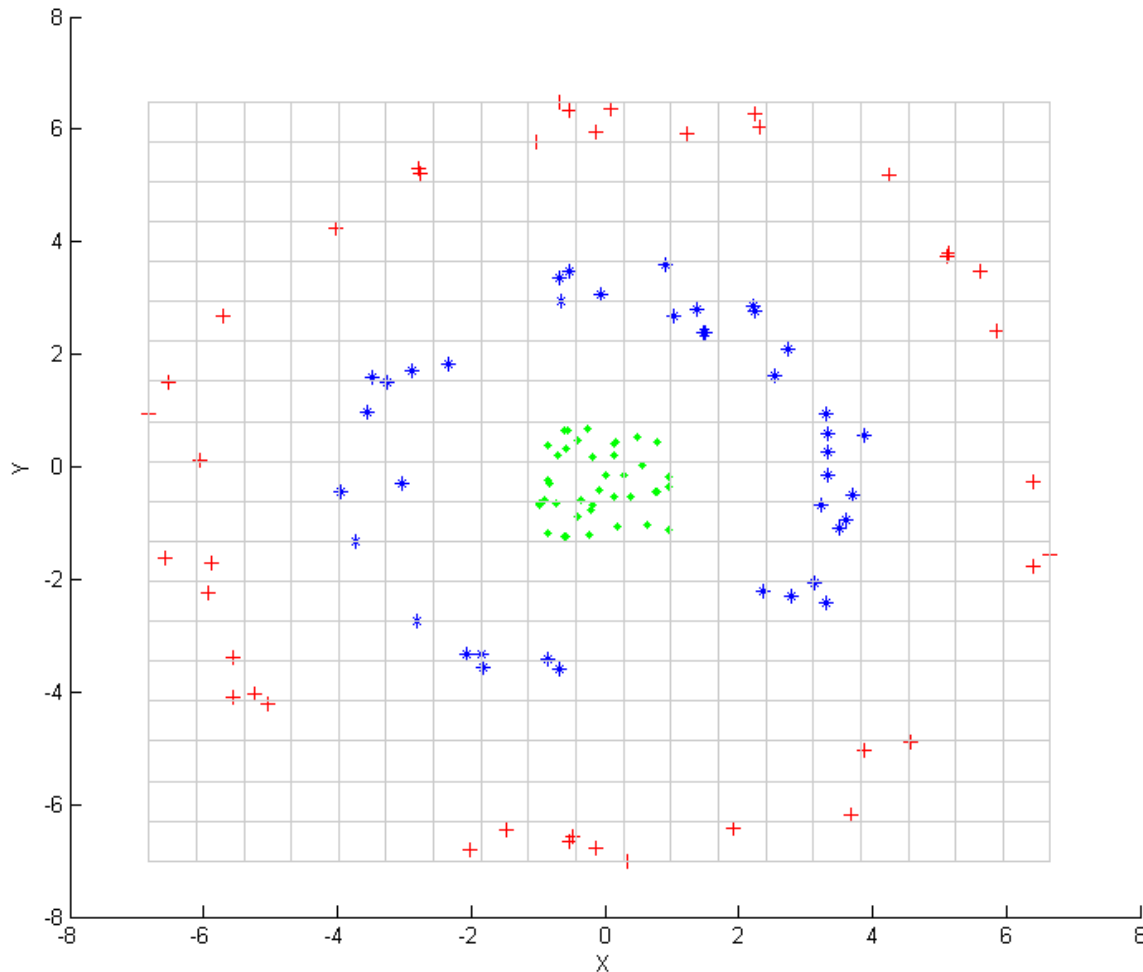
Otra opción: ICA

- Buscar la proyección lineal cuyas proyecciones sean independientes estadísticamente. Procedimiento similar.



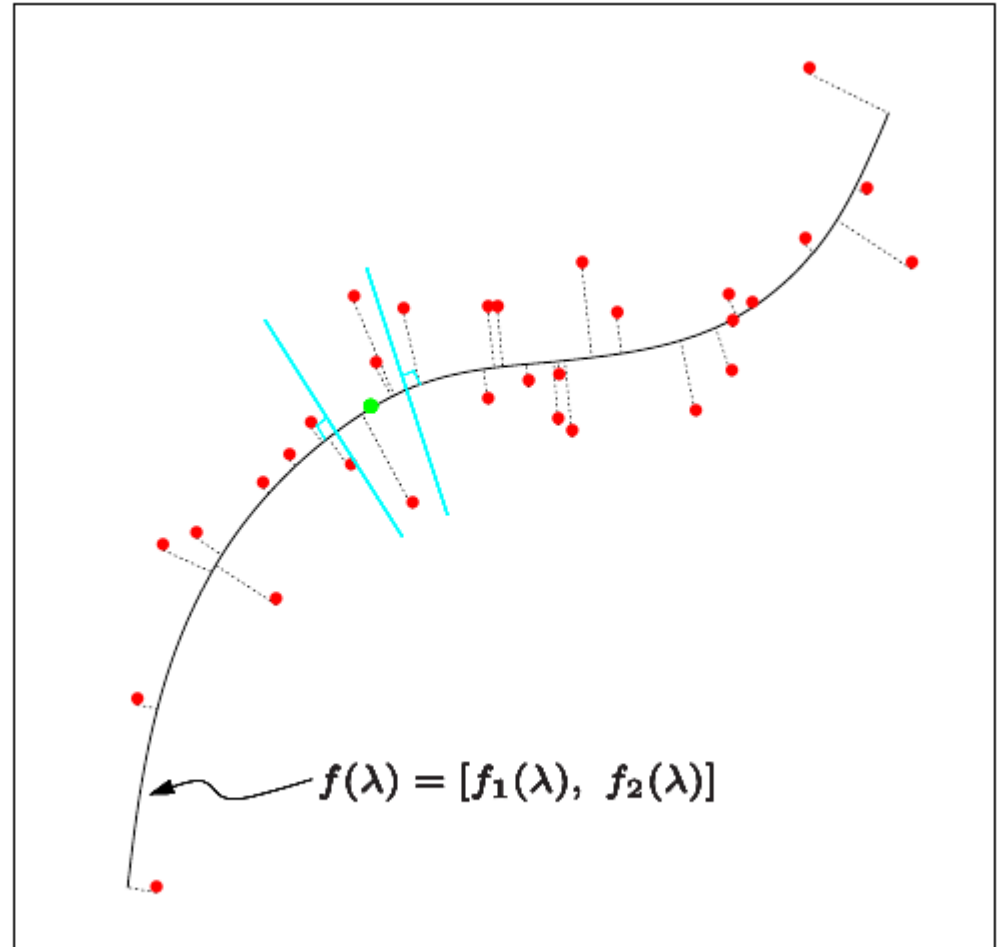
Proyecciones no-lineales

- Que hacemos si las proyecciones lineales no sirven?



Proyecciones no-lineales

- Buscar una superficie arbitraria de menor dimensión tal que:
 - la proyección conserve la estructura
 - la superficie sea suave



Proyecciones no-lineales

■ ISOMAP

- A global geometric framework for nonlinear dimensionality reduction, JB Tenenbaum, V Silva, JC Langford, Science 290 (5500), 2319-2323, 2002
- Grafo conectado de k-vecinos, luego MDS

■ t-SNE

- Visualizing Data using t-SNE, L van der Maaten, G Hinton, JMLR 9, 2579-2605, 2008
- Conserva relaciones de vecindad en alta y baja dimensión, optimización por gradiente, regulada por el parámetro perplexity