Electiva Aprendizaje automático
Análisis exploratorio de datos

Maestría en Gestión de Tecnologías de Información y Conocimiento
Facultad de Ingeniería
Universidad de Nariño

MSc Jimmy Mateo Guerrero Restrepo[1]
jimaguere@udenar.edu.co
3168211698

[1]Departamento de Sistemas
Facultad de Ingeniería

Aprendizaje automático, 2021

Contenido

Algunas nociones algebraicas previas: Concepto de Combinación Lineal

Dados uno o más vectores, es posible expresar un nuevo vector como combinación lineal de los primeros.

Dados dos vectores: $\vec{u}$ **y** $\vec{v}$**, y** dos números **reales**: **a** y **b cualesquiera**, el **vector**

$$\vec{w} = a\vec{u} + b\vec{v}$$

Se dice que es una combinación lineal **de** $\vec{u}$ y $\vec{v}$.

$$\begin{pmatrix} 20 \\ 12 \\ 37 \end{pmatrix} = 2\begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} + 3\begin{pmatrix} 6 \\ 2 \\ 9 \end{pmatrix}.$$

Algunas nociones algebraicas previas: Concepto de Combinación Lineal

Si un vector resultara combinación lineal de otros dos, el nuevo vector no contiene información nueva que no está contenida en los primeros dos. Es decir que su información es redundante.

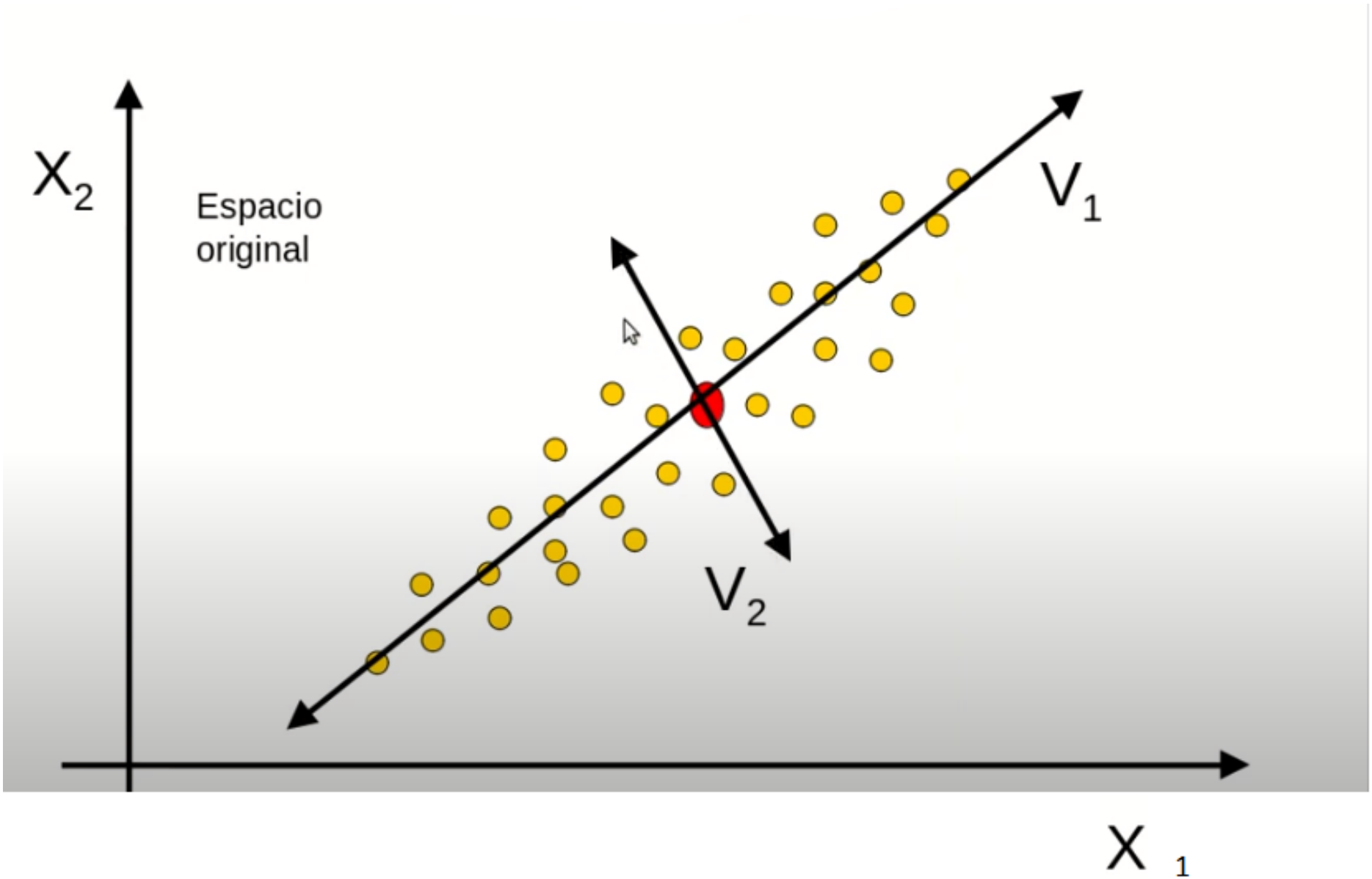
Análisis de Componentes Principales (ACP)

Estas técnicas fueron inicialmente desarrolladas por Pearson a finales del siglo XIX y posteriormente fueron estudiadas por Hotelling en los años 30 del siglo XX. Sin embargo, hasta la aparición de los ordenadores no se empezaron a popularizar.

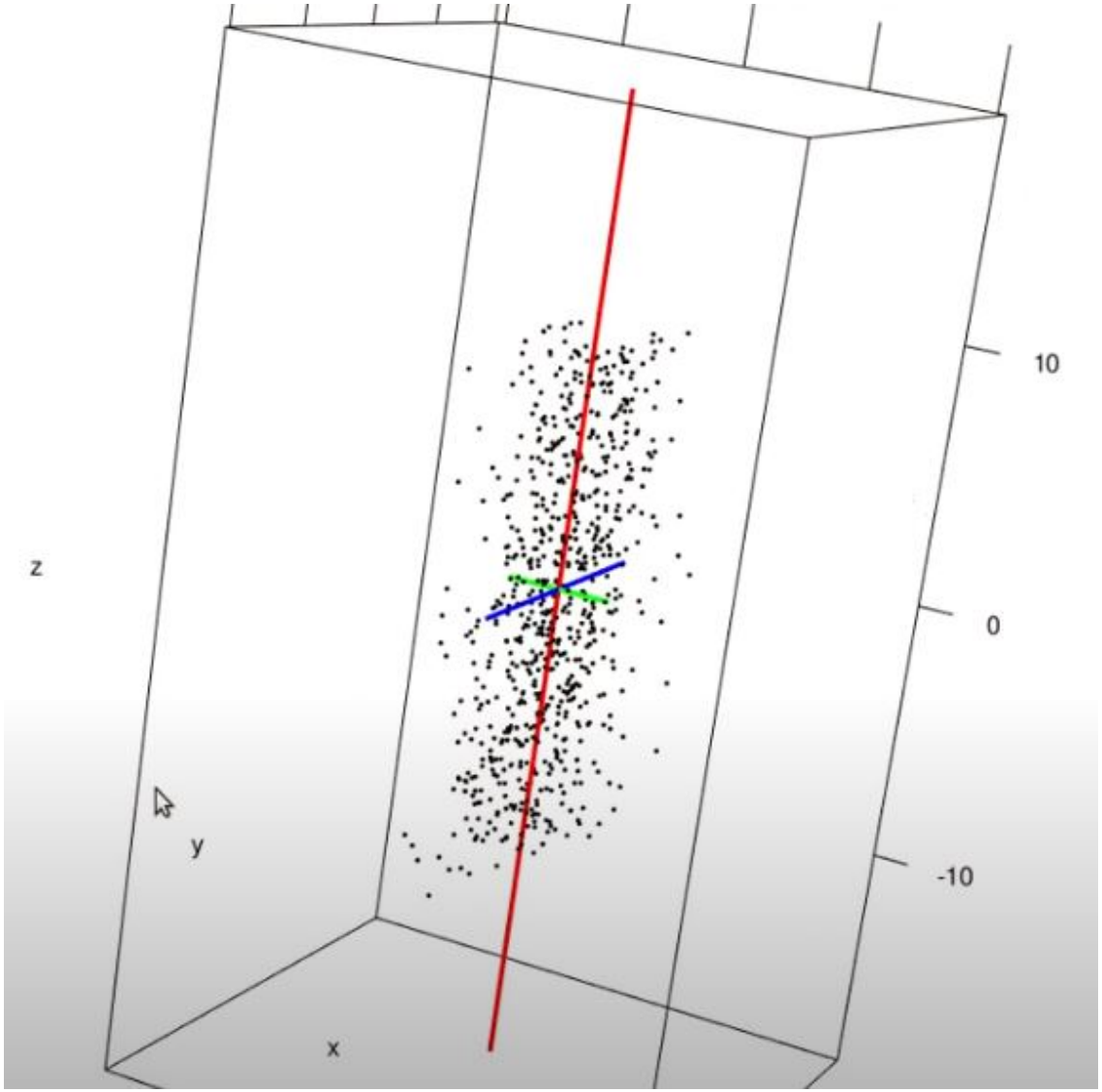
Análisis de Componentes Principales (ACP)

- Definición: El Análisis de Componentes principales consiste en encontrar combinaciones lineales de las variables originales para conseguir un nuevo conjunto de variables no correlacionadas, que denominaremos Componentes Principales, cuya obtención es en orden decreciente de importancia.
- El análisis de componentes principales (ACP) comprende un procedimiento matemático que transforma un conjunto de variables correlacionadas en un conjunto menor de variables no correlacionadas denominadas componentes principales.
- ACP (Análisis de Componentes Principales) es una técnica exploratoria que procura hallar aquellas combinaciones (lineales) de las variables originales que maximizan la varianza de la proyección de estas variables sobre el espacio que generan de las combinaciones lineales encontradas; es decir que minimizan la pérdida de la información inicial.

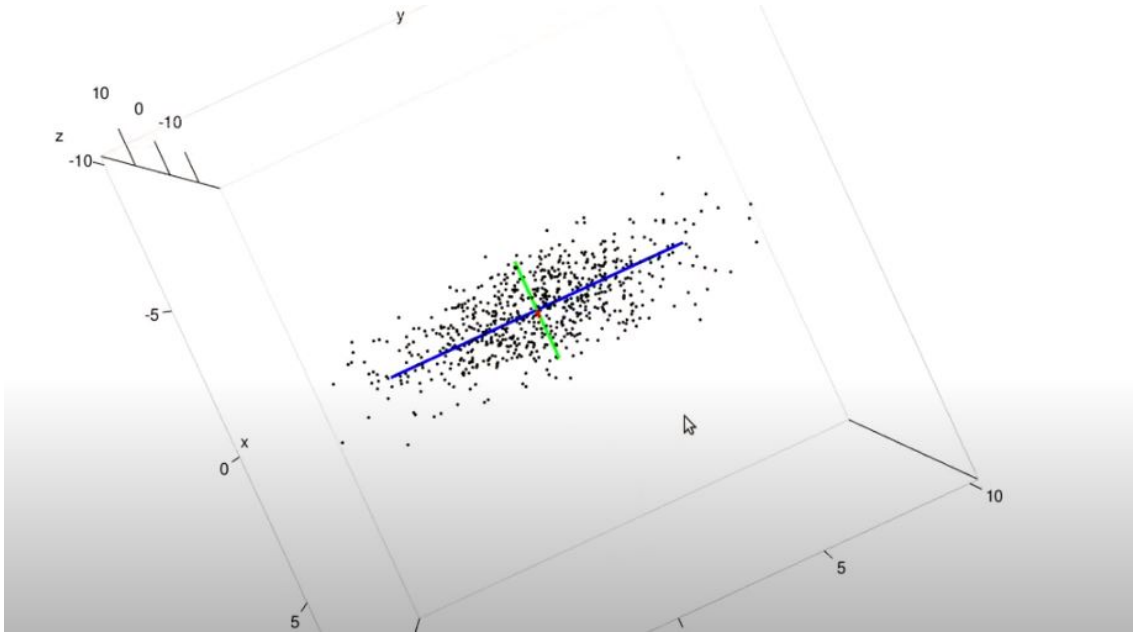
Análisis de Componentes Principales (ACP)



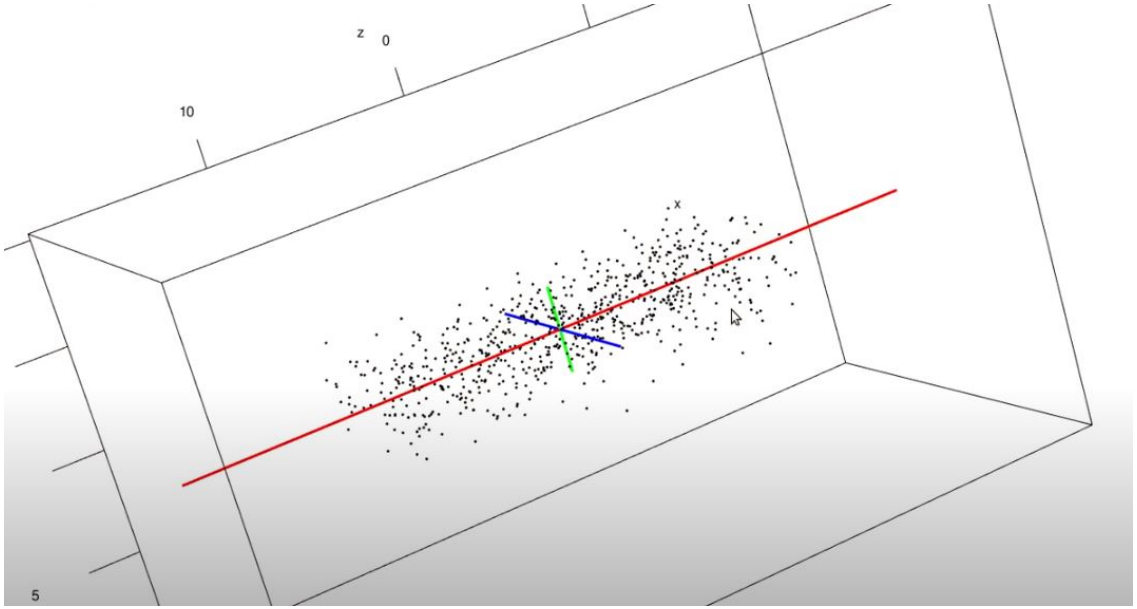
Análisis de Componentes Principales (ACP)



Análisis de Componentes Principales (ACP)



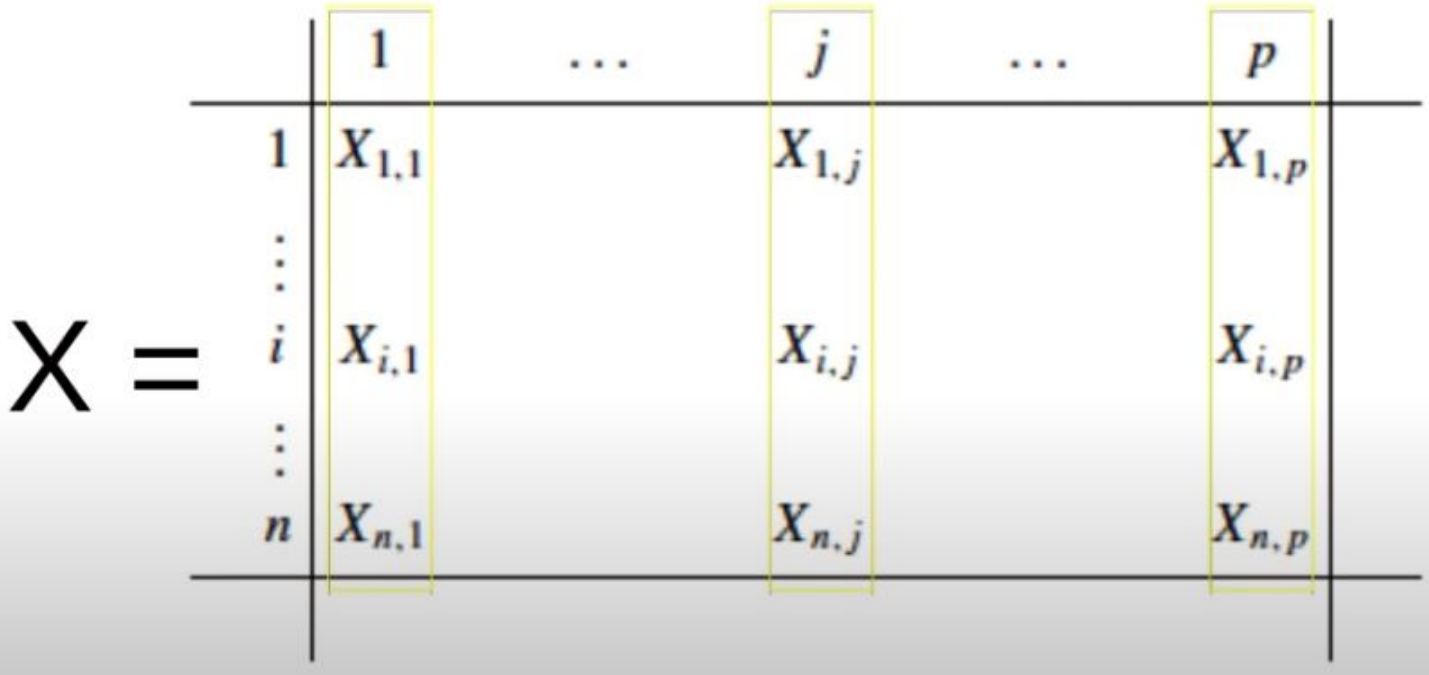
Análisis de Componentes Principales (ACP)



¿Cómo se define la primer componente principal?

Matriz de datos

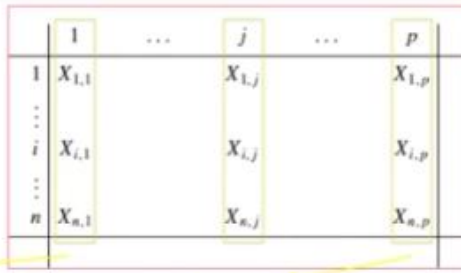
centrada



¿Cómo se define la primer componente principal?

Vamos a describir la idea original de Hotelling

La primer componente



$$\begin{pmatrix} Y_{11} \\ \vdots \\ Y_{n1} \end{pmatrix} = a_{11} \begin{pmatrix} x_{11} \\ \vdots \\ x_{n1} \end{pmatrix} + a_{21} \begin{pmatrix} x_{12} \\ \vdots \\ x_{n2} \end{pmatrix} + \cdots + a_{p1} \begin{pmatrix} x_{1p} \\ \vdots \\ x_{np} \end{pmatrix}$$

Componente

$$Y_1 = \sum_{j=1}^{p} a_{j1} X_j = \mathbf{X} a_1$$

Autovector 1 (Loading 1)

Tal que

$$a_1 = (a_{11}, a_{21}, \ldots, a_{p1})'$$

$$\max \mathrm{Var}(Y_1)$$

¿Cómo se define la primer componente principal?

Buscamos una variable Y_1 de la forma:

$$Y_1 = \sum_{j=1}^{p} a_{j1} X_j = a_{11}X_1 + a_{21}X_{12} + \cdots + a_{p1}X_p = \overrightarrow{a_1}' X$$

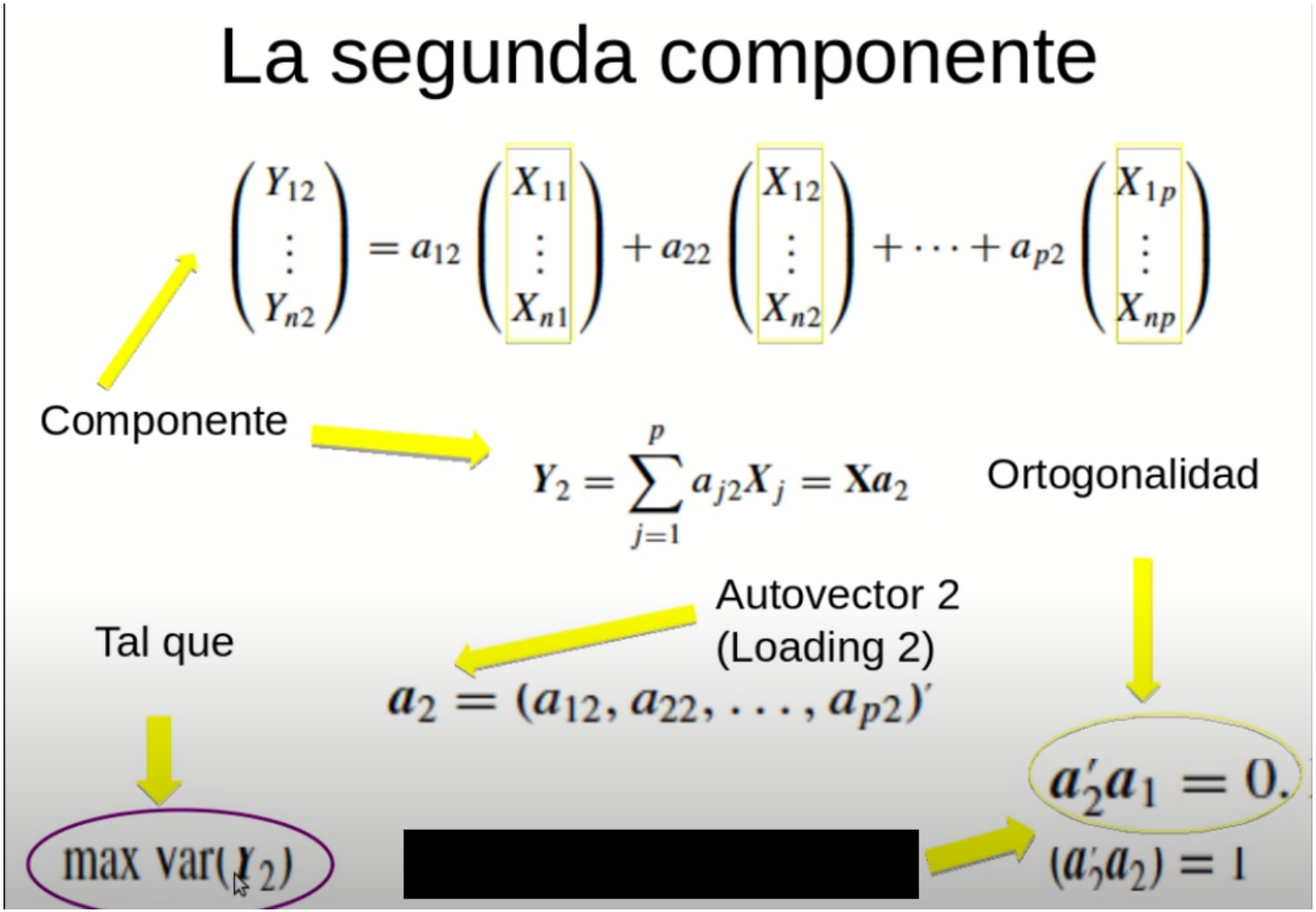
Con la condición de que $V(Y_1)$ sea máxima entre las varianzas de las combinaciones lineales de las variables originales con $\|\bar{a}_1\| = 1$

Los coeficientes del vector $\bar{a}_1$ se denominan **loadings**(*cargas*).

(a)=(a1 ,a2 ,a3 ,a4 ,aj ,..., an)

$$[a]: \sqrt{a_1^2 + a_2^2 + a_3^2 + \ldots + a_n^2}$$

¿Cómo se define la primer componente principal?



¿Tienen algún significado los loadings?

- Si la carga de una de las variables en la componente principal es positiva, indica que la variable y la componente tienen una correlación positiva. Es decir que un individuo que tenga una puntuación alta en esa variable tendrá valores más altos en esa componente que otro individuo que tenga los valores de las restantes variables similares al primer individuo pero un menor valor en esa variable.
- Si por el contrario la carga es negativa, indica que dicha variable correlaciona en forma negativa con la primera componente. Por lo tanto si dos individuos tienen puntuaciones similares en las restantes variables pero distinto valor en ésta; el que tenga puntuación más alta de los dos en esta variable se ubicará en un valor menor de la componente.

¿Cuál es la solución al problema planteado? ¿Cuáles son los valores de las cargas?

- Centralización de las variables: se resta a cada valor la media de la variable a la que pertenece. Con esto se consigue que todas las variables tengan media cero.
- Se resuelve un problema de optimización para encontrar el valor de los loadings con los que se maximiza la varianza. Una forma de resolver esta optimización es mediante el cálculo de eigenvector-eigenvalue de la matriz de covarianzas.

Ecuación característica

$A\vec{v} = \lambda\vec{v}$

o equivalentemente:

$(A-\lambda I)\ \vec{v} = \vec{0}$,

Lo cual indica que a excepción que $\vec{v} = \vec{0}$, esto indica necesariamente que la matriz

A-λI

es singular, o bien que su determinante[1] es nulo.

$P(\lambda) = |A - \lambda I| = 0$ donde $|A-\lambda I|$ es el determinante de la matriz **A-λI.**

Determinante es la asignación de un número a una matriz cuadrada, dicho número es la suma de los productos posibles tomando un elemento de cada fila y de cada columna con su signo de permutación.

Para una matriz de 2x2 el determinante se calcula: $\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$

Autovalores

matriz asociada es: $A = \begin{bmatrix} 2 & 3 \\ 3 & -6 \end{bmatrix}$

Planteamos la ecuación característica:

$$|A - \lambda I| = 0$$

$$\left| \begin{bmatrix} 2 & 3 \\ 3 & -6 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = 0$$

$$\begin{vmatrix} 2-\lambda & 3 \\ 3 & -6-\lambda \end{vmatrix} = (2-\lambda)(-6-\lambda) - 9 = 0$$

$-12+4\lambda+\lambda^2-9=0$ $\lambda^2+4\lambda-21=0$

Tiene como soluciones los autovalores: $\lambda=3$ y $\lambda=-7$.

Autovectores

$$\begin{bmatrix} 2 & 3 \\ 3 & -6 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = -7\begin{bmatrix} x \\ y \end{bmatrix} \qquad \begin{bmatrix} 2 & 3 \\ 3 & -6 \end{bmatrix}\begin{bmatrix} -1 \\ 3 \end{bmatrix} = -7\begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 \\ 3 & -6 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = 3\begin{bmatrix} x \\ y \end{bmatrix} \qquad \begin{bmatrix} 2 & 3 \\ 3 & -6 \end{bmatrix}\begin{bmatrix} 3 \\ 1 \end{bmatrix} = 3\begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

No sólo $\begin{bmatrix} -1 \\ 3 \end{bmatrix}$ y $\begin{bmatrix} 3 \\ 1 \end{bmatrix}$ son autovectores sino cualquier múltiplo de ellos también es autovector asociado al correspondiente autovalor.

Funciones importantes

- Dada la matriz A cuadrada, vale decir con igual cantidad de filas que de columnas, se pueden definir dos funciones importantes para el análisis multivariado, vinculadas a su "tamaño": la traza y el determinante.

Traza de A= tr(A) = $\sum_{i=1}^{n} a_{ii}$ = suma de los elementos diagonales de A

Determinante de A = det(A) = | A |

- Estas dos funciones están íntimamente vinculadas con autovalores de la matriz. Se puede demostrar que:

$Tr(A) = \sum_{i=1}^{n} \lambda_i$ (suma de los autovalores de A)

$Det(A) = \prod_{i=1}^{n} \lambda_i$ (producto de los autovalores de A)

Ejemplo

$$A = \begin{bmatrix} 2 & 3 \\ 3 & -6 \end{bmatrix}$$

- Tr(A) = 2 + (-6) = -4 = 3 + (-7)
- Det(A) = 2*(-6)-3*3 = -12 – 9 = -21= 3*(-7)

¿Cómo se hace en Python? (para no hacer las cuentas a mano!!)

- Ingresamos los valores y los guardamos en una matriz.
  import numpy as np
  from numpy import linalg as LA
  A=np.array([[2, 3], [3, -6]])
  print(A)

```
import numpy as np
from numpy import linalg as LA

A=np.array([[2, 3], [3, -6]])
print(A)
```

```
[[ 2  3]
 [ 3 -6]]
```

¿Cómo se hace en Python? (para no hacer las cuentas a mano!!)

- Podemos calcular la traspuesta de esa matriz.
  T=np.transpose(A)
  print(T)

```
T=np.transpose(A)
print(T)
```

```
[[ 2  3]
 [ 3 -6]]
```

¿Cómo se hace en Python? (para no hacer las cuentas a mano!!)

- Podemos calcular su traza.
  print(np.trace(A))

```
print(np.trace(A))
```

```
-4
```

- Es la misma la traza de su traspuesta?
  print(np.trace(np.transpose(A)))

```
print(np.trace(np.transpose(A)))
```

```
-4
```

¿Cómo se hace en Python? (para no hacer las cuentas a mano!!)

- Calculamos los autovalores y los autovectores de la matriz.
  w, v = LA.eig(A)
  print('autovalores:',w)
  print('autovectores 3:',v[:,0])
  print('autovectores -7:',v[:,1])

```
w, v = LA.eig(A)
print("autovalores:",w)
print("autovectores 3:",v[:,0])
print("autovectores -7:",v[:,1])
```

```
autovalores: [ 3. -7.]
autovectores 3: [0.9486833  0.31622777]
autovectores -7: [-0.31622777  0.9486833 ]
```

¿Cómo se hace en Python? (para no hacer las cuentas a mano!!)

- Verifiquemos que son autovalores y autovectores

```
#Av=xv
print("AV:\n",np.dot(A,v))
print("3V1:\n",np.multiply(w[0],v[:,0]))
print("7V2:\n",np.multiply(w[1],v[:,1]))
```

```
AV:
 [[ 2.84604989  2.21359436]
 [ 0.9486833  -6.64078309]]
3V1:
 [2.84604989 0.9486833 ]
7V2:
 [ 2.21359436 -6.64078309]
```

¿Cómo se hace en Python? (para no hacer las cuentas a mano!!)

- ¿Qué norma tienen?
  print("norma v1:",sum(v[:,0]*v[:,0]))
  print("norma v2:",sum(v[:,1]*v[:,1]))

```
print("norma v1:",sum(v[:,0]*v[:,0]))
print("norma v2:",sum(v[:,1]*v[:,1]))
```

```
norma v1: 0.9999999999999999
norma v2: 0.9999999999999999
```

- ¿Cómo son entre sí? sum(v[:,0]*v[:,1])

```
: sum(v[:,0]*v[:,1])
```

```
: 0.0
```

¿Cómo se hace en Python? (para no hacer las cuentas a mano!!)

- Calculemos la suma de los autovalores?
  print(sum(w))
  out: 4
  print(np.trace(A))
  out 4

¿Cómo se hace en Python? (para no hacer las cuentas a mano!!)

- Calculemos el producto de los autovalores
  print(w.prod())
  out: -21
  print(np.linalg.det(A))
  out : -21
- Ahora estamos en condiciones de analizar el problema de la reducción de dimensión

¿Cuál es la solución al problema planteado? ¿Cuáles son los valores de las cargas?

- Se puede demostrar que la variabilidad de la primera componente principal es máxima cuando el vector de cargas es el autovector de la matriz S, de varianzas y covarianzas de las variables originales, asociado al mayor autovalor de la misma.

$\bar{a}_1$ es el autovector de S asociado a λ_1, el mayor autovalor de S

- Es importante tener presente la siguiente propiedad

$$Var(AX+b) = AVar(X)A^T$$

X: es nuestra variable de dimension px1
A: es la matriz asociada a una transformacion de dimension pxp
B: vector de constantes de dimension px1

¿Cuál es la solución al problema planteado? ¿Cuáles son los valores de las cargas?

- ¿Cuánta es la variabilidad que logra captar la primer componente principal?

$$V(Y_1) = V\left(\sum_{j=1}^{p} a_{j1}X_j\right) = V(\overrightarrow{a_1}'X) = \overrightarrow{a_1}'V(X)\overrightarrow{a_1} = \overrightarrow{a_1}'S\overrightarrow{a_1} = \lambda_1$$

Donde λ_1 es el mayor autovalor de la matriz **S** de varianzas y covarianzas de las variables X; pues $\overrightarrow{a_1}$ es el autovector de norma uno asociado al mayor autovalor de la matriz de varianza y covarianzas.

¿Cuál es la solución al problema planteado? ¿Cuáles son los valores de las cargas?

- ¿Cómo se define la segunda componente principal?

$$Y_2 = \sum_{j=1}^{p} a_{j2} X_j = a_{12} X_1 + a_{22} X_{12} + \cdots + a_{p2} X_p = \overrightarrow{a_2}' X$$

Con la condición de que Y_2 tenga varianza máxima entre el conjunto de variables ortogonales a Y1, siendo nuevamente $\|\overrightarrow{a_2}\| = 1$

Es decir que en el espacio ortogonal a la primera componente principal, buscamos la segunda componente principal. En este procedimiento se va construyendo una nueva representación de variables, que pierde la menor información posible pero que, a la vez, construye un nuevo conjunto de variables no correlacionadas entre sí!.

- Se cumple además, que la variabilidad de las sucesivas componentes principales va disminuyendo:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_v \geq \cdots \geq \lambda_p$$

¿Qué proporción de la variabilidad total del conjunto logra captar cada componente?

- Si la variabilidad total del conjunto es la suma de las variabilidades de cada una de las variables involucradas en el problema, entonces la variabilidad total coincide con la traza de la matriz de varianzas y covarianzas S.

$$\text{Variabilidad Total} = \text{Traza}(S^2) = \lambda_1 + \lambda_2 + \cdots + \lambda_p$$

- Entonces cabe preguntarnos: ¿qué proporción de esa variabilidad logra captar cada una de las componentes principales consideradas?
- Las componentes son combinaciones lineales de las variables originales y se espera que, solo unas pocas (las primeras) recojan la mayor parte de la variabilidad de los datos, obteniéndose asi una reducción de la dimensión del conjunto de datos.

¿Qué proporción de la variabilidad total del conjunto logra captar cada componente?

- La proporción de la variabilidad que capta la primera componente principal es:

$$\frac{\lambda_1}{Traza(S)} = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \ldots + \lambda_p}$$

- La proporción de la variabilidad que capta la segunda componente principal es:

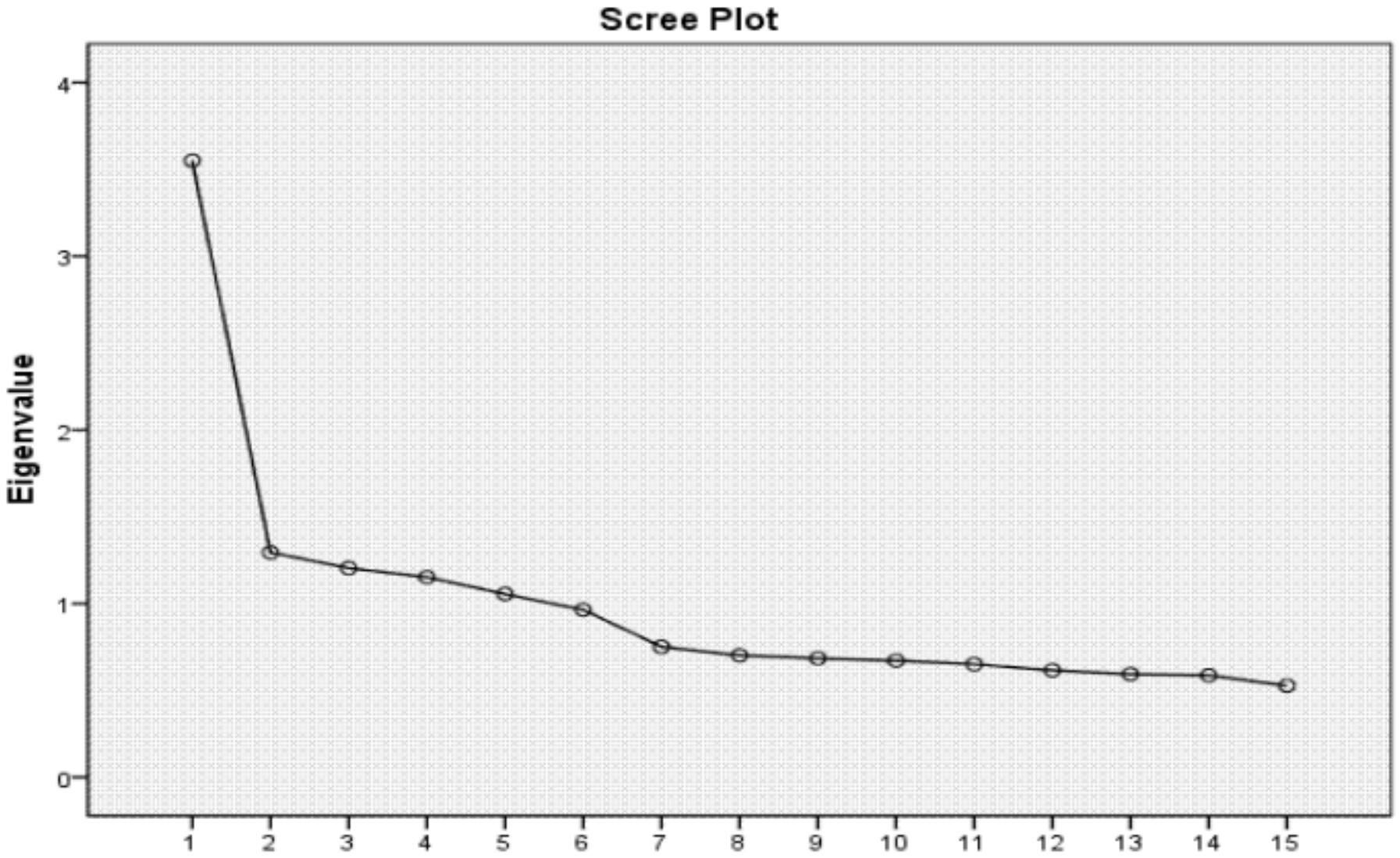
$$\frac{\lambda_2}{Traza(S)} = \frac{\lambda_2}{\lambda_1 + \lambda_2 + \ldots + \lambda_p}$$

- Como hemos visto, los valores van decreciendo, luego la proporción de variabilidad que cada componente logra captar de la variabilidad total, también va disminuyendo.
- En algún punto, dado que nuestro objetivo es reducir la dimensión del problema original, dejará de tener sentido seguir buscando componentes principales.

La pregunta es: ¿cuántas componentes corresponde considerar?

- Criterio 1: Porcentaje de la variabilidad mínimo que se desea explicar.
- Criterio 2: de Kaiser obtener las componentes principales a partir de la matriz de correlaciones R equivale a suponer que las variables observables tengan varianza 1. Por lo tanto una componente principal con varianza inferior a 1 explica menos variabilidad que una variable observable.
- Criterio 3: Criterio del bastón roto. Podemos representar un gráfico de sedimentación o Scree Plot con los valores propios como en la siguiente figura y considerar el número de componentes hasta que el descenso se estabiliza.

La pregunta es: ¿cuántas componentes corresponde considerar?



Escalas de medida

- Si las escalas de medida de las variables son muy diferentes, la variabilidad estaría dominada por las variables con magnitudes mayores de forma que las primeras componentes pueden mostrar simplemente las diferencias de escala.
- En ese caso conviene tomar la matriz estandarizada por columnas; centrando las variables y dividiéndolas por su desvío standard. En ese caso las componentes estarían calculadas sobre la matriz de correlaciones.
- Cuando las componentes principales se calculan a partir de las matrices de covarianzas, los factores de carga dependen de la escalas de medida de las variables por lo que son difíciles de interpretar.
- Cuando se calculan a partir de la matriz de correlaciones, los loadings son las correlaciones de las componentes y las variables originales. Los factores de carga suelen representarse en un gráfico que permite la interpretación visual de las relaciones.

Ejemplo

El conjunto de datos Sick, el cual registra enfermedades de la tiroides, proporcionados por: el Instituto Garavan, Instituto de Nueva Gales del Sur, Syndney, Australia. El archivo contiene estudios de pacientes, donde se pretende determinar si tienen o no una enfermedad. Clase (1) Enfermo Clase (0) No Enfermo
Se han registrado sobre 955 individuos las variables siguientes:

caso	Número de registro
edad	Edad del paciente
genero	Genero del paciente
depresion	El paciente tiene depresión
embarazada	Paciente en estado de embarazo
cirugía_de_tiroides	Tiene cirugías de tiroides
nivel_TSH	nivel TSH
nivel_T3	nivel T3
nivel_TT4	nivel TT4
nivel_T4U	nivel T4U
nivel_FTI	nivel FTI
clase	Enfermo tiroides 1

Ejemplo

```
import pandas as pd
import matplotlib.pyplot as plt
datos=pd.read_csv( "./Sick_pacientes.txt",sep=";")
datos.head(20)
```

	caso	edad	genero	depresion	embarazada	cirugía_de_tiroides	nivel_TSH	nivel_T3	nivel_TT4	nivel_T4U	nivel_FTI	clase
0	1	56	F	NO	NO	NO	1.900	2.044	129.00	1.0500	123.00	0
1	2	51	F	NO	NO	NO	0.250	1.900	101.00	1.0800	94.00	0
2	3	50	F	NO	NO	NO	2.000	2.500	133.00	1.0800	123.00	0
3	4	73	M	NO	NO	NO	1.800	2.100	103.00	0.9200	112.00	0
4	5	56	F	NO	NO	NO	3.774	3.900	141.00	1.1200	126.00	0
5	6	44	M	NO	NO	NO	1.900	2.200	103.00	1.2000	86.00	0
6	7	24	F	NO	NO	NO	1.700	2.400	126.00	0.9200	136.00	0
7	8	40	M	NO	NO	NO	0.250	2.300	134.00	0.8400	160.00	0
8	9	81	M	NO	NO	NO	1.300	1.200	147.00	1.0600	138.00	0
9	10	53	F	NO	NO	NO	0.850	2.000	103.00	0.9300	111.00	0
10	11	36	M	NO	NO	NO	1.700	2.600	129.00	1.1800	109.00	0
11	12	35	F	NO	NO	NO	1.200	2.000	89.00	0.9923	110.89	0
12	13	25	F	NO	NO	NO	1.500	2.100	133.00	0.9923	110.89	0
13	15	29	F	NO	NO	NO	2.100	2.200	101.00	0.7900	129.00	0
14	16	71	F	NO	NO	NO	0.920	2.100	84.00	0.8600	98.00	0
15	17	80	F	NO	NO	NO	0.005	2.700	101.00	0.8400	121.00	0
16	18	24	F	NO	NO	NO	0.380	2.300	152.00	0.9600	158.00	0
17	19	88	F	NO	NO	NO	13.000	2.044	123.00	0.9900	124.00	0
18	20	22	F	NO	NO	NO	1.000	2.700	150.00	1.2200	124.00	0
19	21	57	F	NO	NO	NO	3.774	2.044	108.33	0.9923	110.89	0

Ejemplo

- El vector medio del conjunto es:
  datos.drop(columns=[çaso”,çlase”]).mean()

```
edad          51.794764
nivel_TSH      3.773681
nivel_T3       2.039007
nivel_TT4    108.330639
nivel_T4U      0.992342
nivel_FTI    110.891749
dtype: float64
```

Ejemplo

- La Matriz de varianzas y covarianzas muestral es: datos.drop(columns=['caso','clase']).cov()

	edad	nivel_TSH	nivel_T3	nivel_TT4	nivel_T4U	nivel_FTI
edad	527.756576	-4.546745	-3.181298	-22.045582	-0.583977	39.685726
nivel_TSH	-4.546745	146.672617	-1.664949	-129.365019	0.223349	-127.945410
nivel_T3	-3.181298	-1.664949	0.510356	11.750985	0.058197	5.671876
nivel_TT4	-22.045582	-129.365019	11.750985	1150.310629	2.855427	764.893706
nivel_T4U	-0.583977	0.223349	0.058197	2.855427	0.036010	-1.102207
nivel_FTI	39.685726	-127.945410	5.671876	764.893706	-1.102207	911.760633

- La Matriz de varianzas y correlaciones es : datos.drop(columns=['caso','clase']).corr()

	edad	nivel_TSH	nivel_T3	nivel_TT4	nivel_T4U	nivel_FTI
edad	1.000000	-0.016342	-0.193843	-0.028294	-0.133957	0.057211
nivel_TSH	-0.016342	1.000000	-0.192438	-0.314945	0.097185	-0.349872
nivel_T3	-0.193843	-0.192438	1.000000	0.484987	0.429289	0.262936
nivel_TT4	-0.028294	-0.314945	0.484987	1.000000	0.443660	0.746884
nivel_T4U	-0.133957	0.097185	0.429289	0.443660	1.000000	-0.192358
nivel_FTI	0.057211	-0.349872	0.262936	0.746884	-0.192358	1.000000

Ejemplo

- Los vectores y valores propios de la matriz de varianzas y covarianzas son:

```
from numpy import linalg as LA
w, v = LA.eig(datos.drop(columns=['caso','clase']).cov())
i=1
for x in w:
print('Autovalor:',i)
print(x)
print('Autovector:')
print(v[:,i-1])
i=i+1
```

Ejemplo

```
Autovalor: 1
1824.9600009098904
Autovector:
[-0.00734277  0.10753622 -0.0069574  -0.75447763 -0.00077423 -0.64737667]
Autovalor: 2
534.9078373616317
Autovector:
[ 0.98718477 -0.01275769 -0.00703461 -0.10986581 -0.001907    0.11480353]
Autovalor: 3
251.04750136273637
Autovector:
[-0.1593136  -0.09773086 -0.01092273 -0.64667388 -0.01032179  0.73936114]
Autovalor: 4
125.76276097688718
Autovector:
[-2.23266498e-03  9.89291840e-01 -4.96050821e-03  1.67563838e-02
  8.75951809e-04  1.44881058e-01]
Autovalor: 5
0.3630984885086398
Autovector:
[ 0.00514799  0.00442713  0.99916672 -0.0132772   0.03761199  0.00536767]
Autovalor: 6
0.005622987290445085
Autovector:
[ 4.08403338e-05 -1.98442851e-03 -3.77366504e-02 -6.98911646e-03
  9.99236607e-01  7.02580702e-03]
```

Ejemplo

- Los vectores y valores propios de la matriz de varianzas y covarianzas son:

```
from sklearn.decomposition import PCA
variables=['edad','nivel_TSH','nivel_T3',
'nivel_TT4','nivel_T4U','nivel_FTI']
pca = PCA(n_components=6).fit(datos[variables])
i=1
for x in pca.explained_variance_:
print('Autovalor:',i)
print(x)
print('Autovector:')
print(pca.components_[i-1,:])
i=i+1
```

Ejemplo

```
Autovalor: 1
1824.9600009098929
Autovector:
[ 0.00734277 -0.10753622  0.0069574   0.75447763  0.00077423  0.64737667]
Autovalor: 2
534.9078373616335
Autovector:
[ 0.98718477 -0.01275769 -0.00703461 -0.10986581 -0.001907    0.11480353]
Autovalor: 3
251.04750136273682
Autovector:
[ 0.1593136   0.09773086  0.01092273  0.64667388  0.01032179 -0.73936114]
Autovalor: 4
125.76276097688694
Autovector:
[-2.23266498e-03  9.89291840e-01 -4.96050821e-03  1.67563838e-02
  8.75951809e-04  1.44881058e-01]
Autovalor: 5
0.363098488508649 34
Autovector:
[ 0.00514799  0.00442713  0.99916672 -0.0132772   0.03761199  0.00536767]
Autovalor: 6
0.0056229872904450 55
Autovector:
[ 4.08403338e-05 -1.98442851e-03 -3.77366504e-02 -6.98911646e-03
  9.99236607e-01  7.02580702e-03]
```

Ejemplo

- Varianza total
  import numpy as np
  print('Varianza total Traza:', np.trace(datos[variables].cov()))
  print('Varianza total suma
  autovalores:',sum(pca.explained_variance_))

```
import numpy as np
print("Varianza total Traza:", np.trace(datos[variables].cov()))
print("Varianza total suma autovalores:",sum(pca.explained_variance_))
```

```
Varianza total Traza           : 2737.0468220869448
Varianza total suma autovalores: 2737.046822086949
```

Ejemplo

- Varianza explicada por cada componente
  i=1
  acum=0
  for x in pca.explained_variance_:
  acum=acum+x/sum(pca.explained_variance_)
  print('Autovalor (',i,'):',
  x/sum(pca.explained_variance_),'Acumulada:',acum)
  i=i+1

```
i=1
acum=0
for x in pca.explained_variance_:
    acum=acum+x/sum(pca.explained_variance_)
    print('Autovalor (',i,'):', x/sum(pca.explained_variance_),'Acumulada:',acum)
    i=i+1
```

```
Autovalor ( 1 ): 0.6667624339427244 Acumulada: 0.6667624339427244
Autovalor ( 2 ): 0.1954324759975337 Acumulada: 0.862194909940258
Autovalor ( 3 ): 0.09172203388589371 Acumulada: 0.9539169438261518
Autovalor ( 4 ): 0.04594834109596821 Acumulada: 0.99986528492212
Autovalor ( 5 ): 0.00013266067850158052 Acumulada: 0.9999979456006216
Autovalor ( 6 ): 2.054399378581923e-06 Acumulada: 1.0000000000000002
```

Ejemplo

- Cargas componentes Principal
  cargas=pd.DataFrame()
  cargas['variables']=variables
  i=1
  for x in pca.explained_variance_:
  cargas['CP'+str(i)]=pca.components_[i-1,:]
  i=i+1
  cargas

```
cargas=pd.DataFrame()
cargas['variables']=variables
i=1
for x in pca.explained_variance_:
    cargas['CP'+str(i)]=pca.components_[i-1,:]
    i=i+1
cargas
```

	variables	CP1	CP2	CP3	CP4	CP5	CP6
0	edad	0.007343	0.987185	0.159314	-0.002233	0.005148	0.000041
1	nivel_TSH	-0.107536	-0.012758	0.097731	0.989292	0.004427	-0.001984
2	nivel_T3	0.006957	-0.007035	0.010923	-0.004961	0.999167	-0.037737
3	nivel_TT4	0.754478	-0.109866	0.646674	0.016756	-0.013277	-0.006989
4	nivel_T4U	0.000774	-0.001907	0.010322	0.000876	0.037612	0.999237
5	nivel_FTI	0.647377	0.114804	-0.739361	0.144881	0.005368	0.007026

Ejemplo

- Cargas de la Primera componente Principales

```
Y1=(0.00734*edad)+(-0.1075*nivel_TSH)+(0.00695*nivel_T3)+(0.75447*nivel_TT4)+(0.000774*nivel_T4U)+(0.64737*nivel_FTI)
```

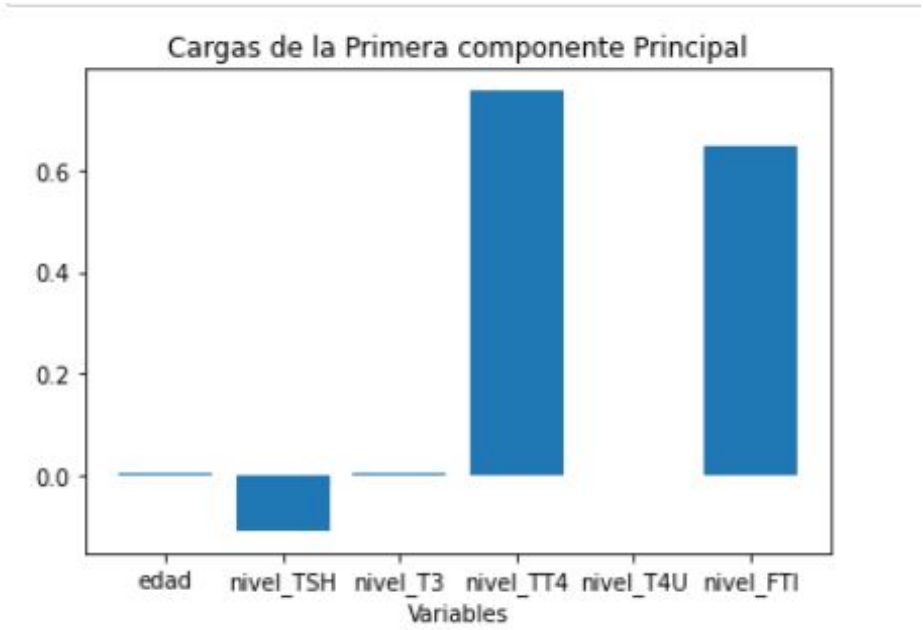
- Cargas de la Segunda Componente Principal

```
Y2=(0.98718*edad)+(-0.01275*nivel_TSH)+(-0.00703*nivel_T3)+(-0.1098*nivel_TT4)+(-0.001907*nivel_T4U)+(0.11480353*nivel_FTI)
```

Ejemplo

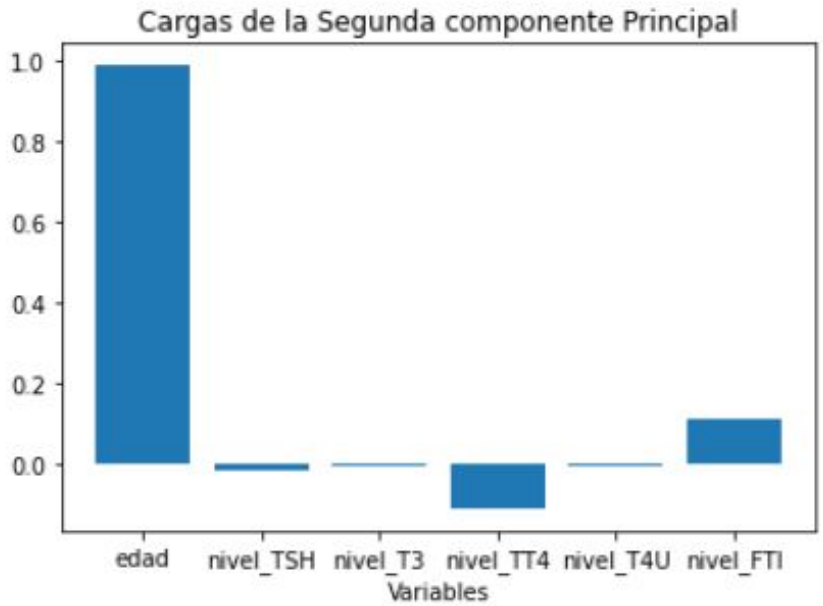
- Cargas de la Primera componente Principales

```
import matplotlib.pyplot as plt
plt.bar(cargas.variables, cargas.CP1)
plt.xlabel('Variables')
plt.title('Cargas de la Primera componente Principal')
plt.show()
```



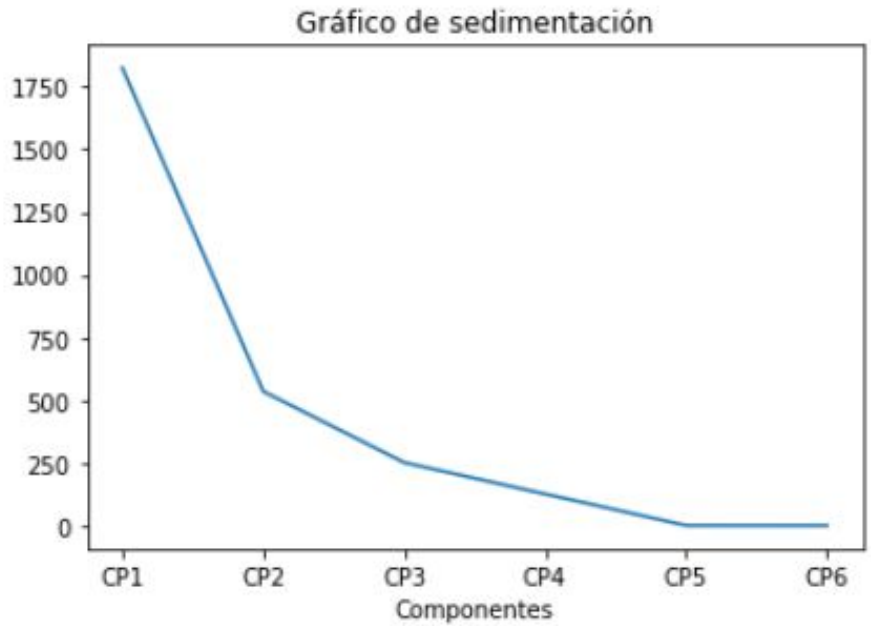
Ejemplo

- Cargas de la Segunda Componente Principal



Ejemplo

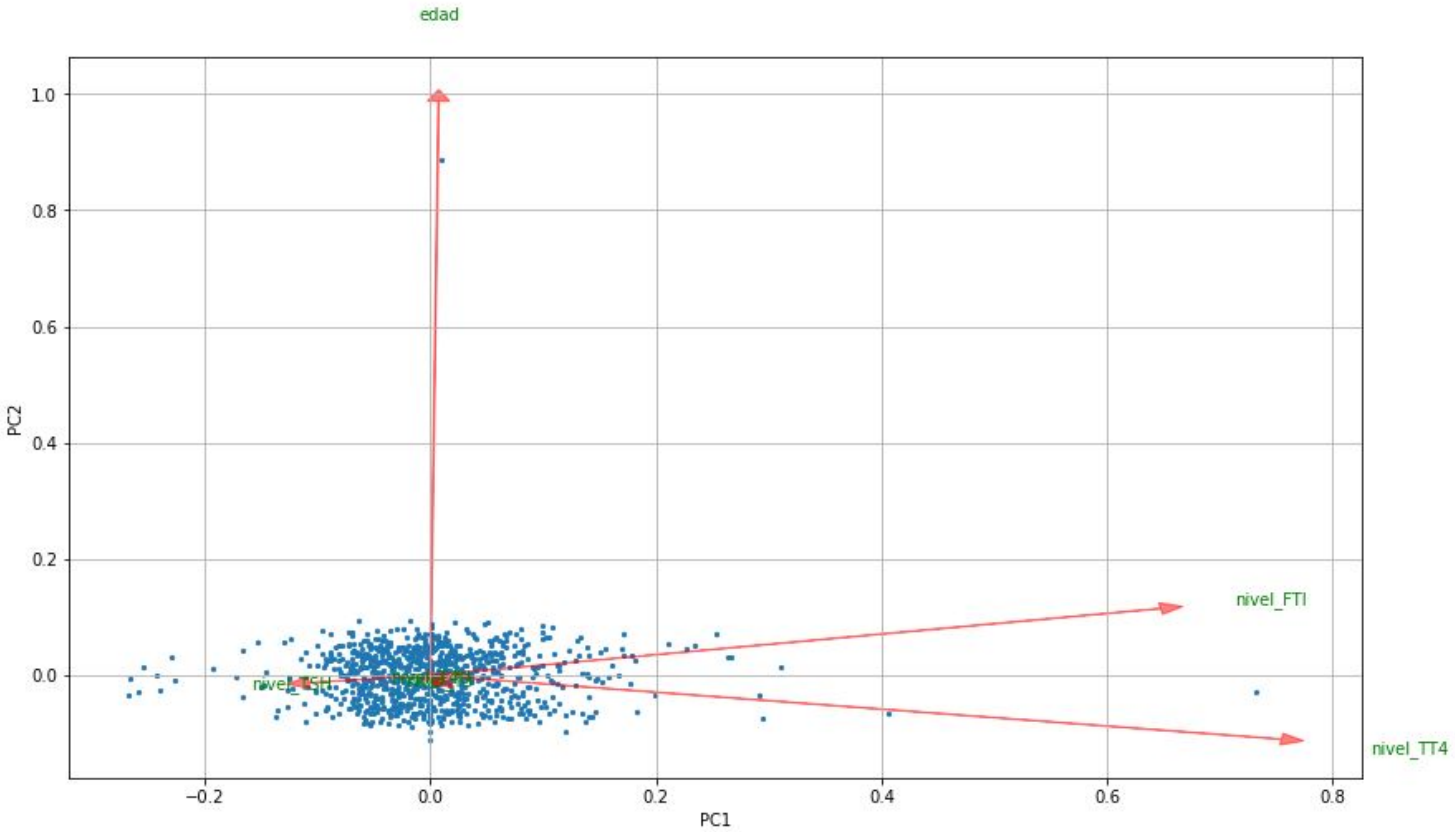
- Gráfico de sedimentación
  plt.plot(cargas.drop(columns='variables').columns,
  pca.explained_variance_)
  plt.xlabel('Componentes')
  plt.title('Gráfico de sedimentación')
  plt.show()



Biplot

- Un Biplot es una representación gráfica de datos multivariantes. De la misma manera que un diagrama de dispersión muestra la distribución conjunta de dos variables, un Biplot representa tres o más variables.
- El prefijo "bi"se refiere a la superposición, en la misma representación, de individuos y variables.
- Los biplots son útiles para describir gráficamente los datos o para mostrar los resultados proporcionados por modelos más formales.

Biplot


edad
nivel_FTI
nivel_TSH
nivel_TT4
PC1
PC2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8

Biplot

- En el biplot se aprecian las relaciones entre las variables y entre los individuos.
- Las variables que forman ángulos muy pequeños significan que están muy correlacionadas.
- Cuando dos variables son ortogonales (perpendiculares) indica que no están correlacionadas.

Análisis de Componentes Principales (ACP)

el propósito fundamental de la técnica consiste en la reducción de la dimensión de los datos con el fin de SIMPLIFICAR el problema en estudio,importante destacar entonces que:

- Permite descartar información redundante.
- Nos permite lograr una representación gráfica de información multidimensional.
- Si las variables originales no están fuertemente correlacionadas, no tiene sentido buscar componentes principales.
- Las unidades de medición pueden influir en la variabilidad de las componentes principales, luego para seleccionar el método conviene considerar la matriz de correlaciones y no la de covarianzas.
- Debemos establecer un criterio para seleccionar la cantidad de componentes principales a seleccionar.

Bibliografía I

Tom M. Mitchel

Machine Learning, cap 5

McGraw-Hill Science/Engineering/Math; (March 1, 1997)