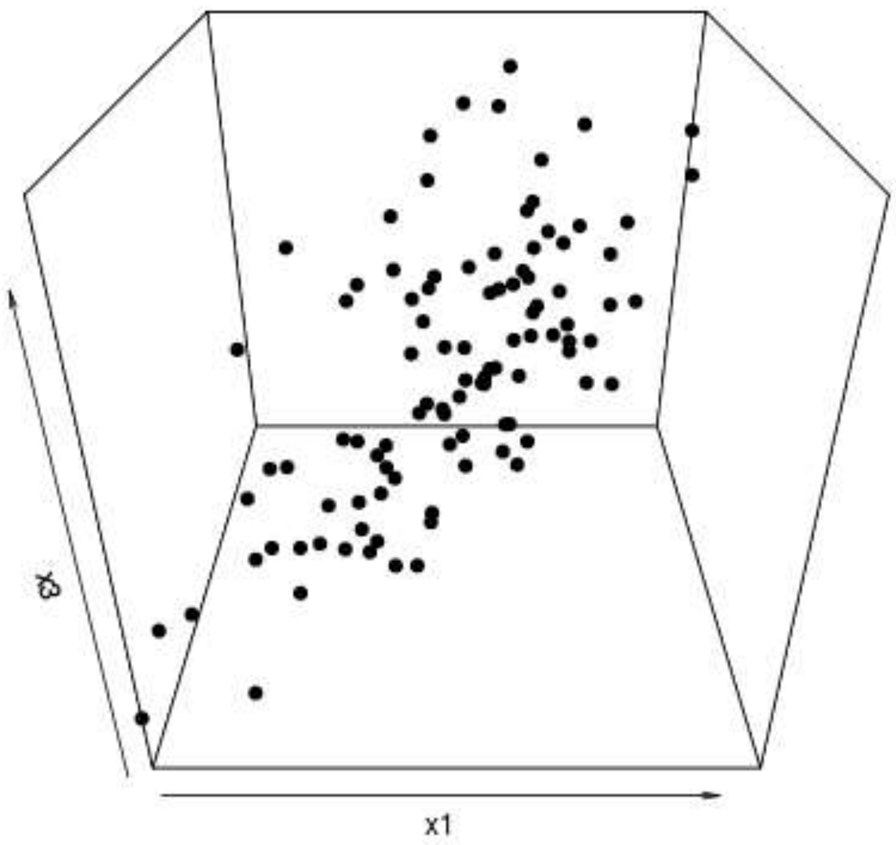
PCA

Principal Compoment Analysis

PCA

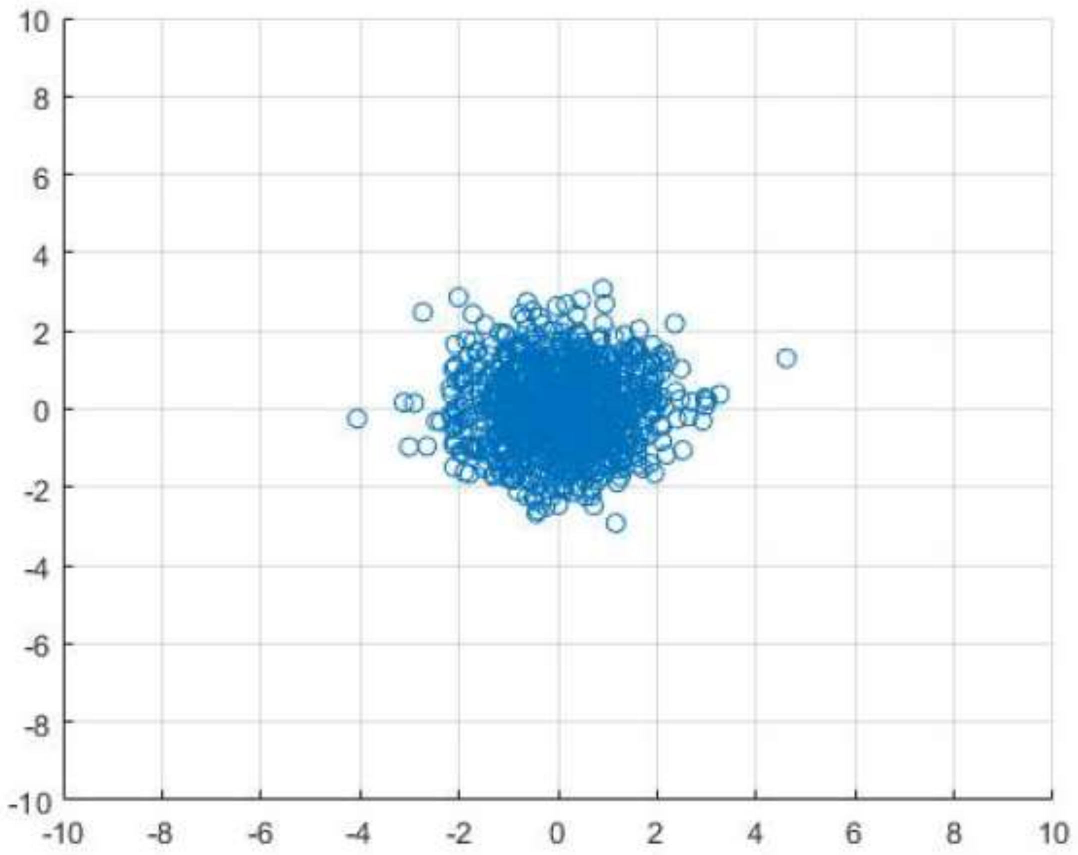
Análisis de Componente Principal

- Técnica para describir un dataset en nuevos términos (componentes)
- Los nuevos componentes no serán correlacionadas (ortogonal)
- Estarán ordenadas de forma descendente según la varianza
- Útil para reducir dimensiones



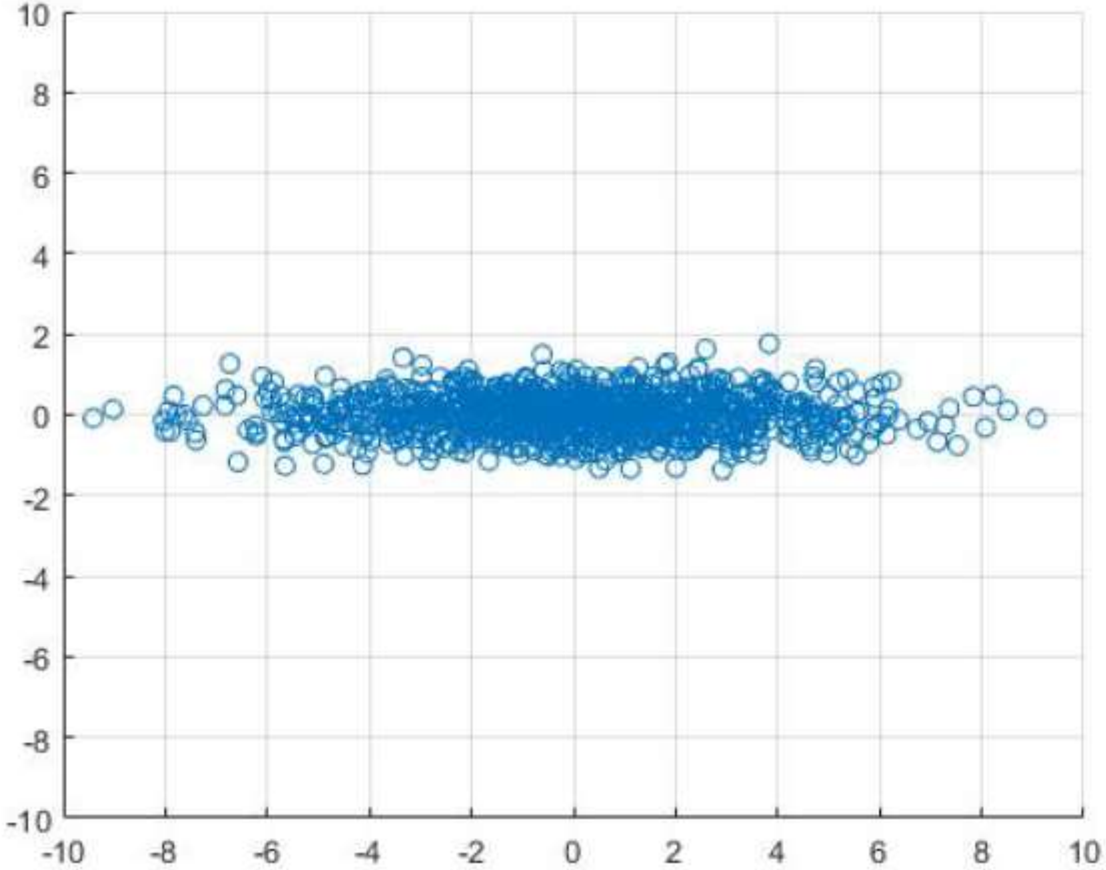
Visualizando

- Data en 3D (3 features)
- Determinar los nuevos ejes
- Mapear los puntos de datos a los nuevos ejes



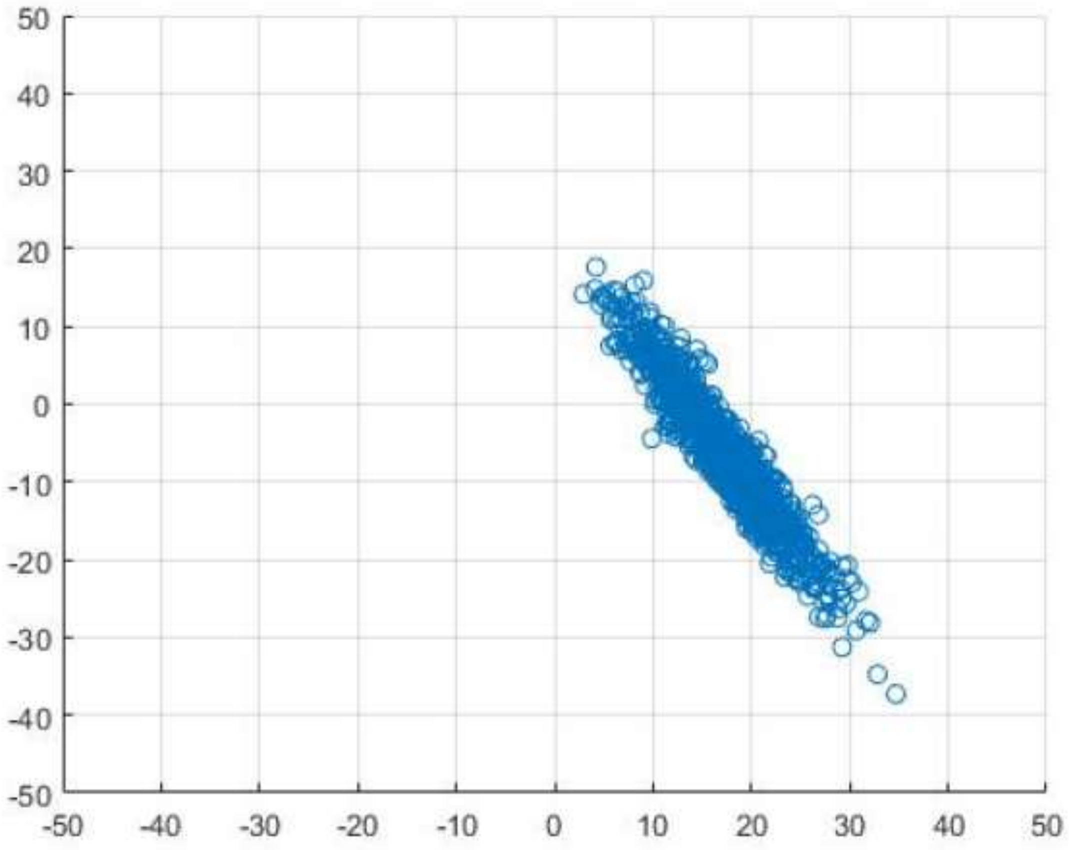
Ejemplo 2D Normal

- $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- 2D Gaussiano, centrado en el origen



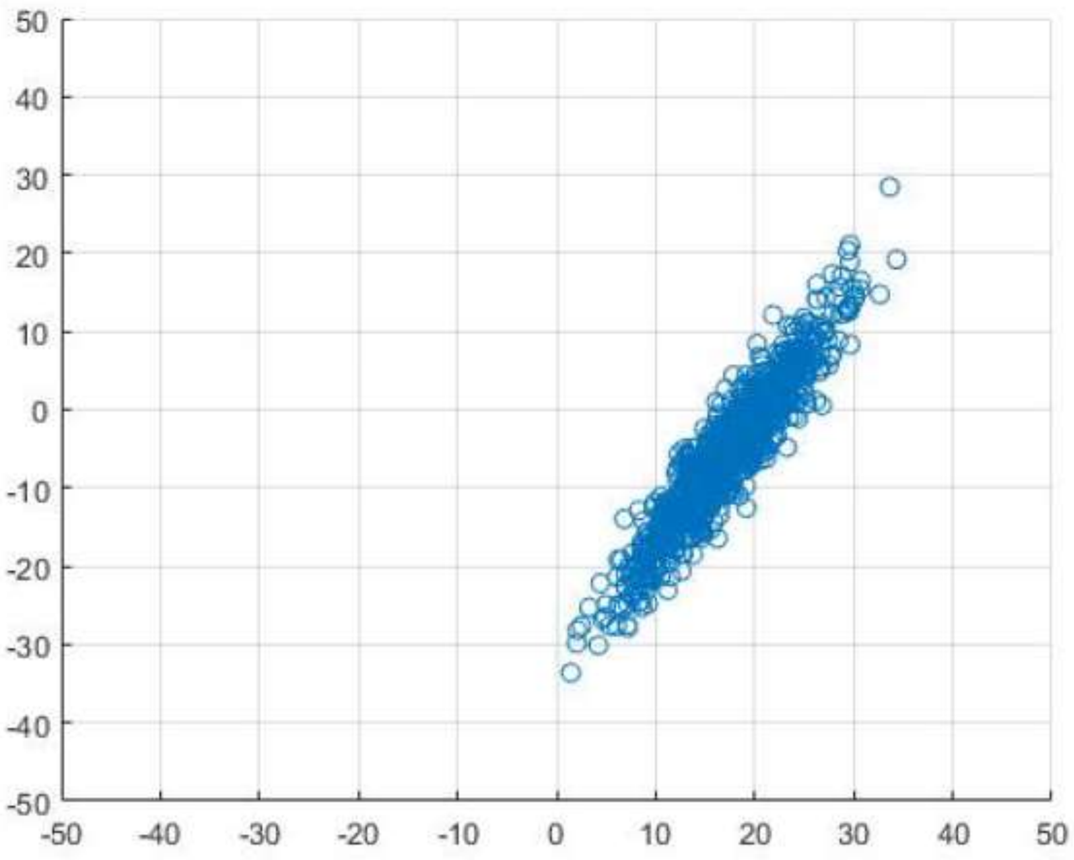
Ejemplo 2D Elíptico

- $\Sigma = \begin{bmatrix} 8.9 & 0 \\ 0 & 0.24 \end{bmatrix}$
- 2D elíptico, centrado en el origen



Ejemplo Covarianza -

- $\Sigma = \begin{bmatrix} 27.8 & -43.8 \\ -43.8 & 78.3 \end{bmatrix}$
- $\mu = \begin{bmatrix} 17 \\ -5 \end{bmatrix}$
- Covarianza negativa



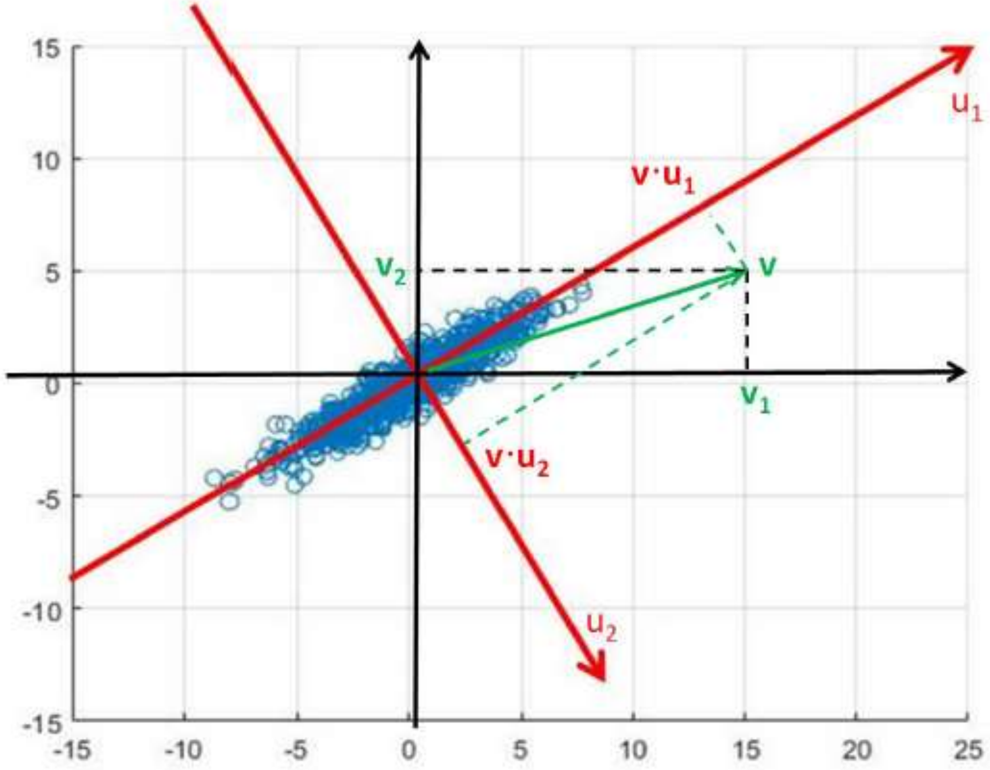
Ejemplo Covarianza +

- $\Sigma = \begin{bmatrix} 26.8 & 42.4 \\ 42.4 & 76.3 \end{bmatrix}$
- $\mu = \begin{bmatrix} 17 \\ -5 \end{bmatrix}$
- Covarianza positiva

- La covarianza nos explica la relación de una variable vs otra variable. Y la medida de “dispersión” de cada una de las variables

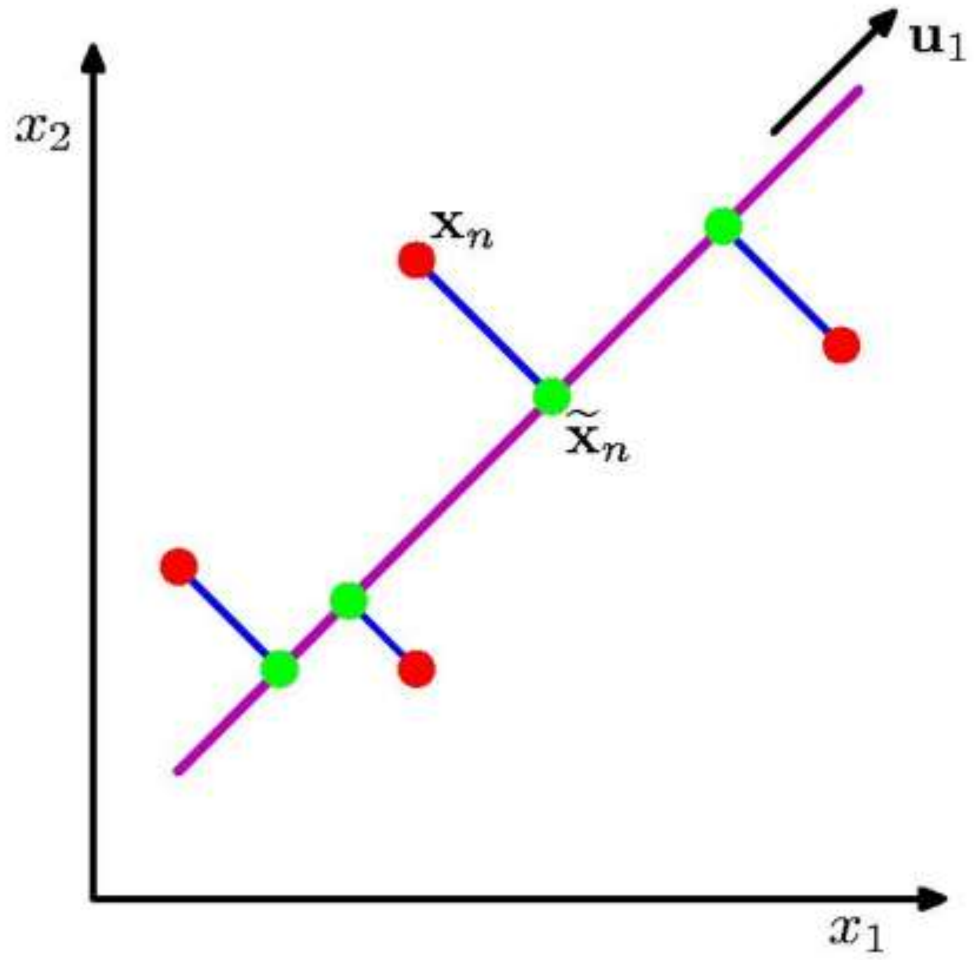
Motivación

- Tenemos covarianza positiva respecto al sistema de coordenadas “original”
- Covarianza es mínima en el sistema de coordenadas rojo.
- Objetivo: identificar el sistema de coordenadas rojo automáticamente



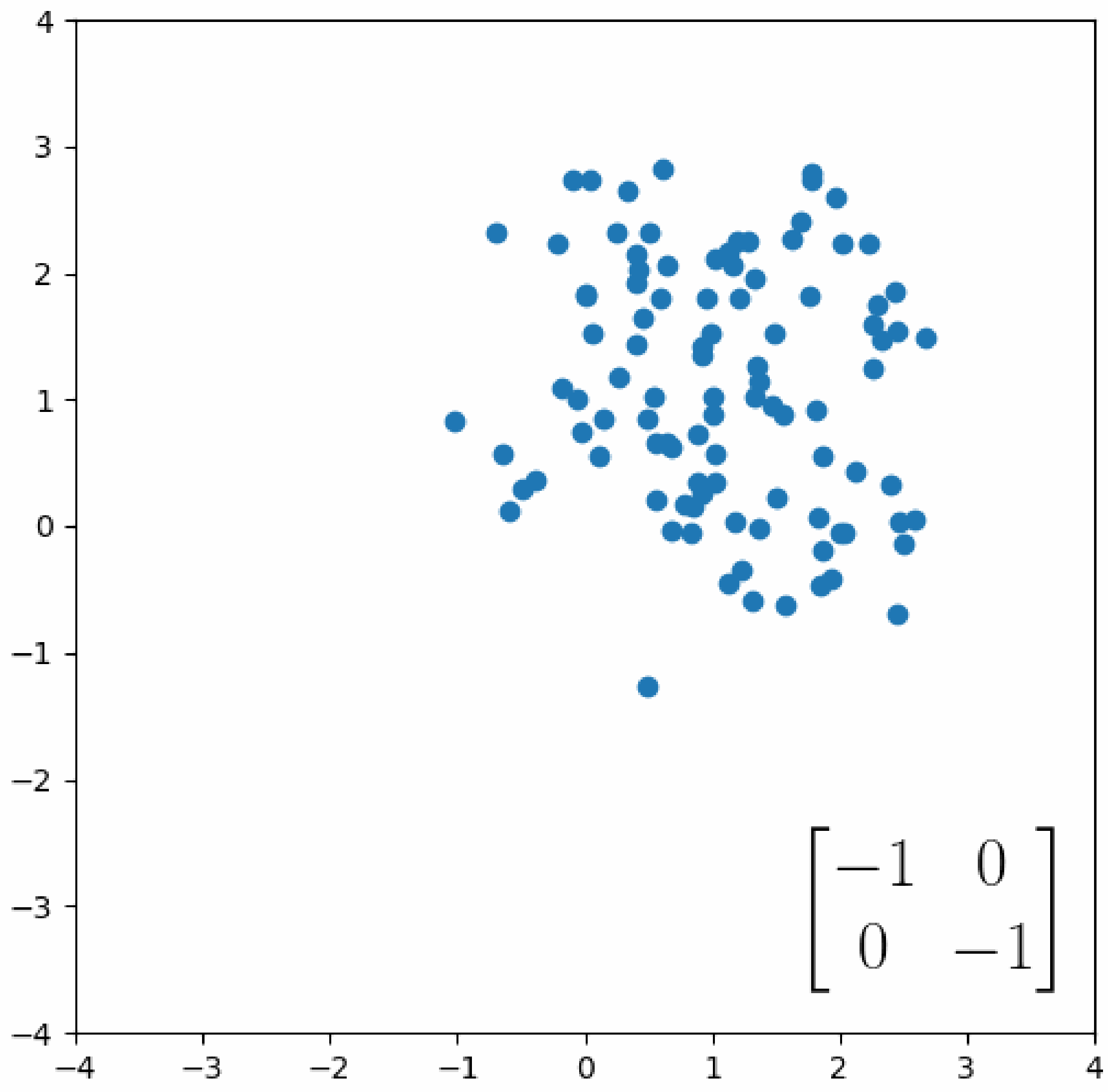
Dos visiones

- Para el primer eje, queremos maximizar la varianza
- También se puede minimizar el “error” de los puntos hacia el eje



Transformación Lineal

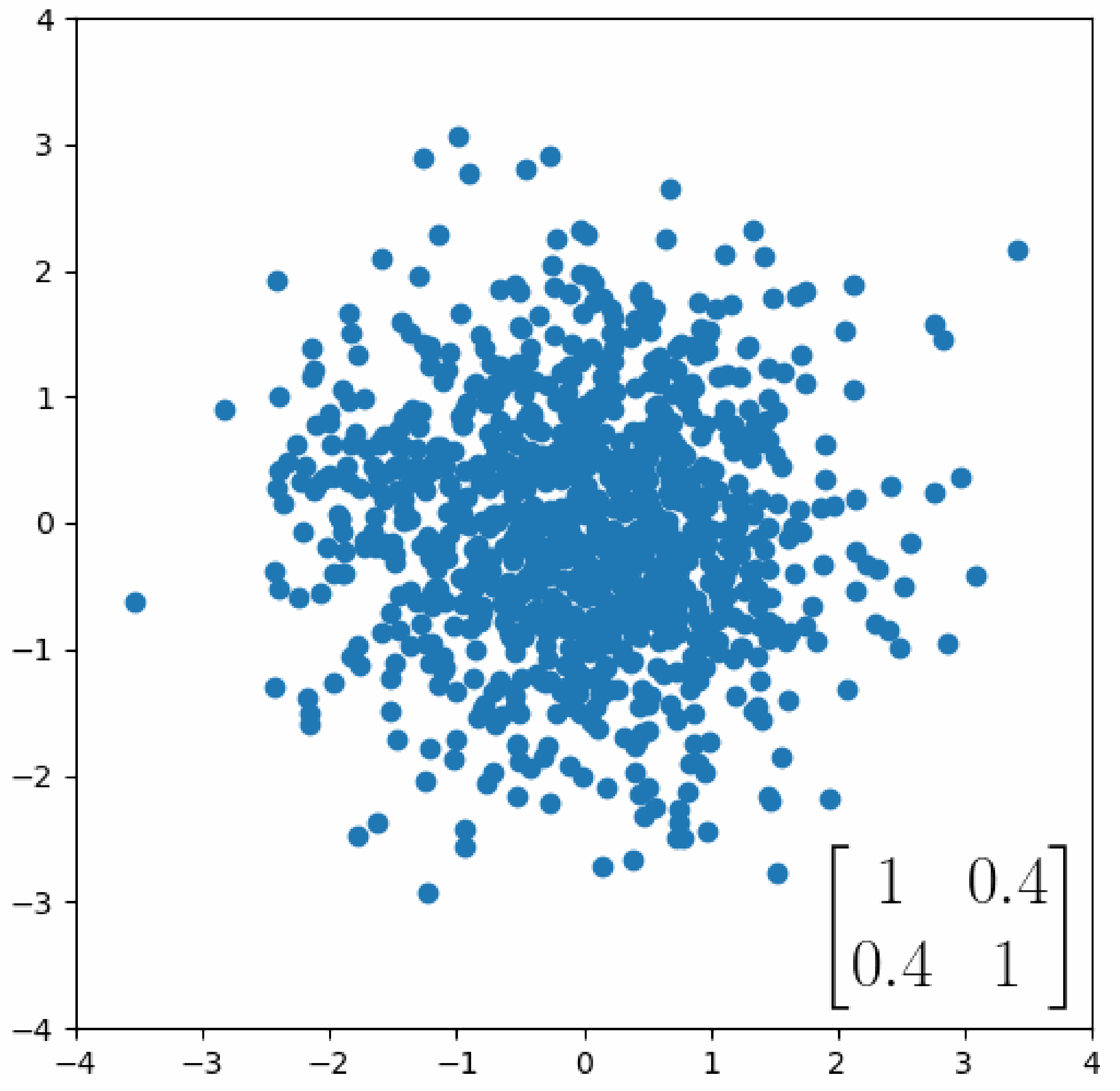
- Mover los puntos de un lado a otro siguiendo instrucciones
- Las instrucciones se escriben en matrices de N*N dimensiones
- Los puntos se multiplican por la matriz y se obtiene la data transformada
- $\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$



http://www.billconnelly.net/?p=697

Transformación Lineal

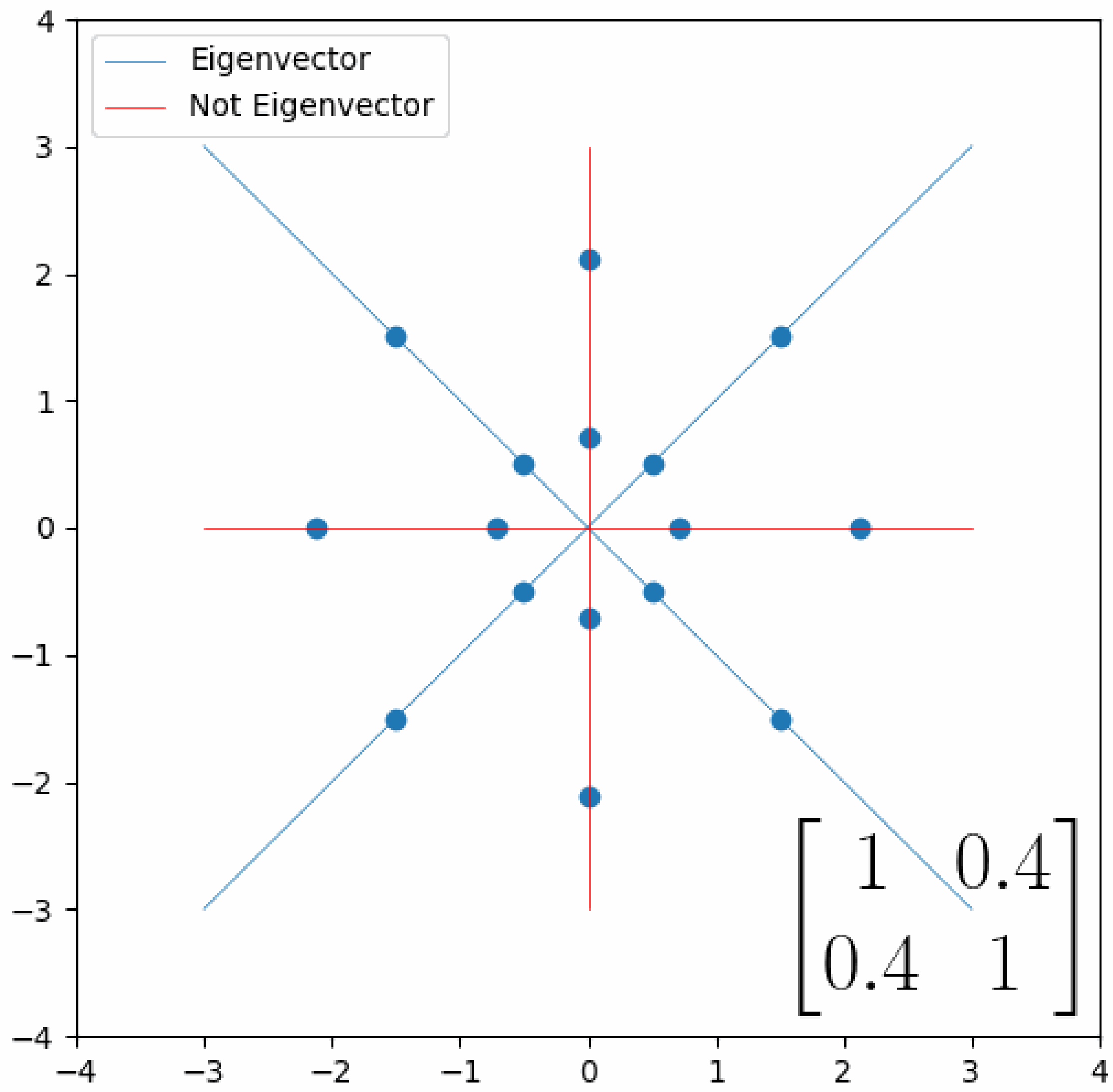
- La transformación que nos va a interesar es el "estiramiento" (shearing)
- $\begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
- Notar que en la imagen, los puntos en una de las diagonales se mantiene relativamente fijo a las coordenadas
  - No hay una traslación de data



http://www.billconnelly.net/?p=697

Transformación Lineal

- Esa línea se lo conoce como vector propio (eigenvector) de la matriz.
  - Nos dice solo la dirección hacia donde estirar
  - El vector propio tiene magnitud 1
- La magnitud del "estiramiento" es conocido como valor propio (eigenvalue)
  - Si un punto empieza en (1,1) y termina en (2,2), el valor propio es 2



http://www.billconnelly.net/?p=697

Relación con la matriz de covarianza

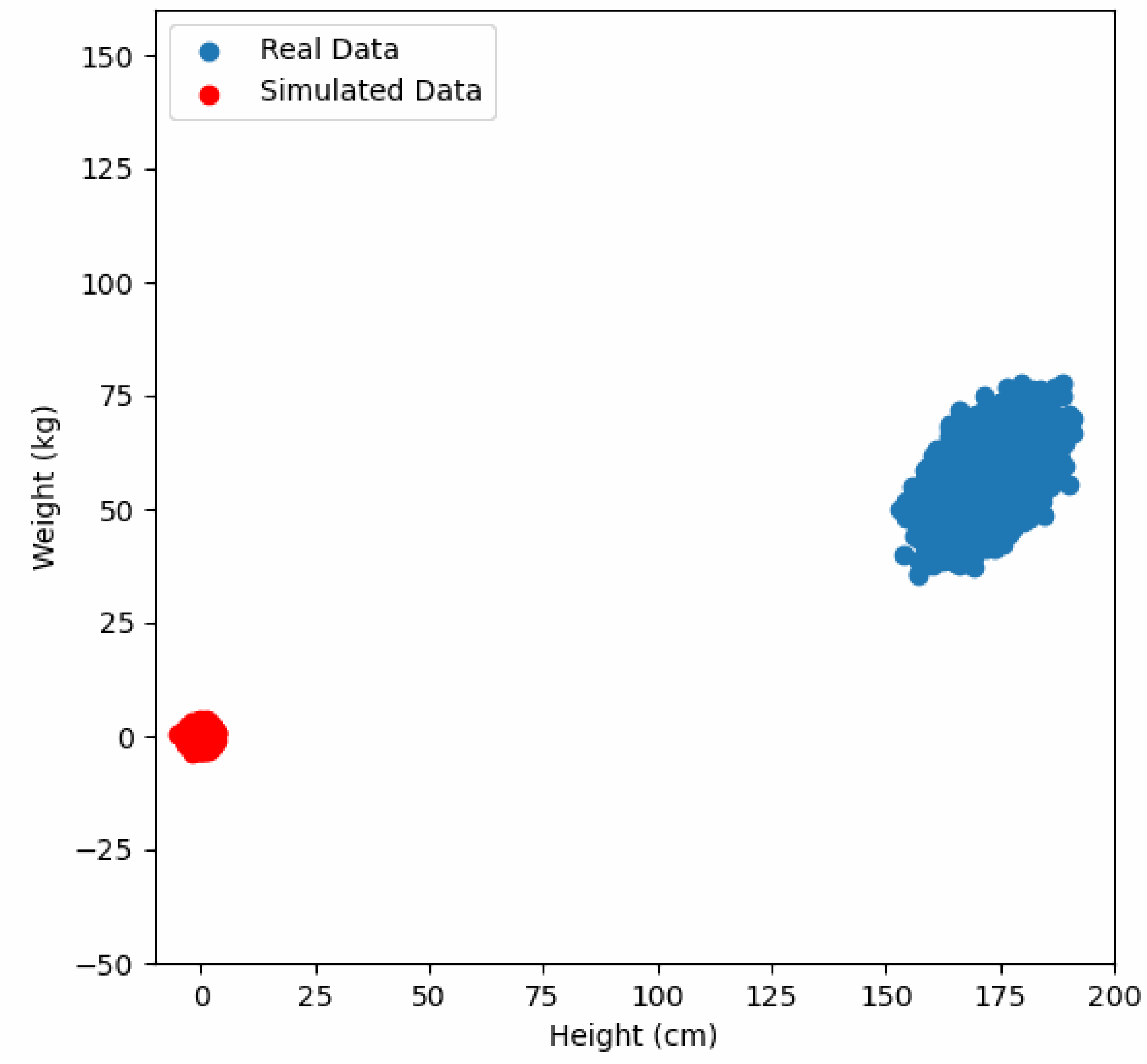
- Podemos imaginar la matriz de covarianza como una transformación lineal.

- $T = cov(data)$
- $cov(data) = cov\left(\sqrt{T} \times DataAleatoria\right)$

- Si tenemos puntos aleatorios distribuidos normalmente, y le aplicamos la raíz cuadrada de la matriz de covarianza de una data específica, obtendremos algo similar a esa data específica

Gráficamente

*notar que también se está sumando la media de la data original para trasladar la nube de puntos simulado a la ubicación de la data real



http://www.billconnelly.net/?p=697

Determinando los componentes principales

- La matriz de covarianza tiene propiedades particulares
  - Diagonal
  - Cuadrada
  - Simétrica.
- Podemos descomponerla

$$\Sigma = \begin{bmatrix} var\ (x_1) & cov(x_1, x_2) \\ cov(x_2, x_1) & var\ (x_2) \end{bmatrix}$$

Descomposición de la matriz de covarianza

- Descomponer una matriz

$$\Sigma = U \cdot D \cdot U^{-1}$$

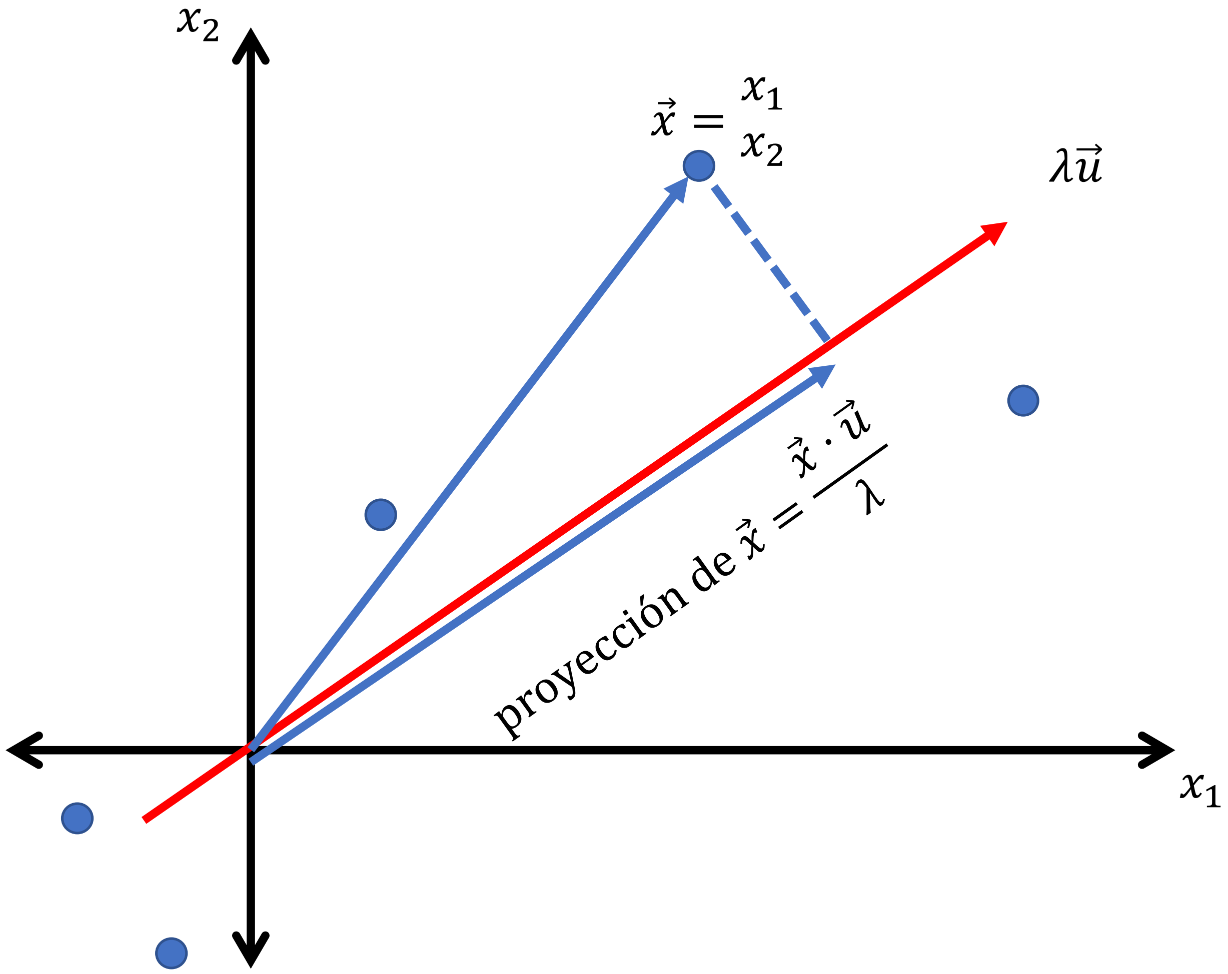
- Donde D es una matriz diagonal compuesta de los valores propios y U son los vectores propios en columnas

$$U \cdot D \cdot U^{-1} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \overrightarrow{u_1} & \overrightarrow{u_2} & \overrightarrow{u_3} \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \vdots & \vdots & \vdots \\ \overrightarrow{u_1} & \overrightarrow{u_2} & \overrightarrow{u_3} \\ \vdots & \vdots & \vdots \end{bmatrix}^{-1}$$

Por propiedad, U es una matriz ortogonal (90°) y unitaria

$$U \cdot U^{-1} = U \cdot U^T = 1$$

Proyectando valores en un componente



- Cada eje original contribuye proporcionalmente en la proyección.
- La proporción es dictada por el vector propio
- Se puede interpretar como la contribución de cada feature al componente

Calcular D y U

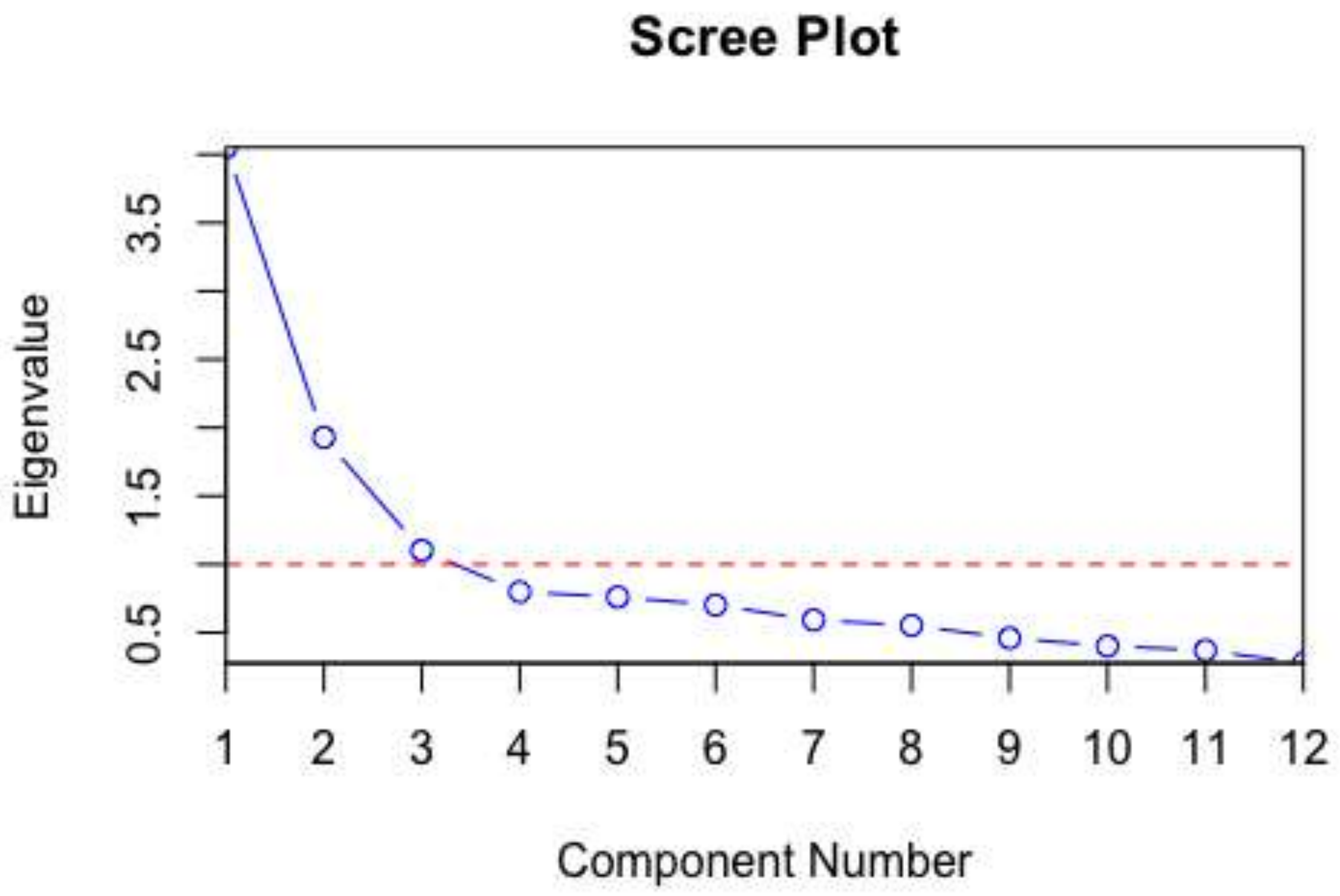
- D y U se calculan usando cálculo de determinantes y resolviendo ecuaciones polinomiales que no están en el alcance del curso.
- Código resuelve el problema.
- Podemos colocar la matriz de covarianza y obtener los valores y vectores propios
- O usar directamente librerías

Nota importante

- Es necesario que la data se encuentre estandarizada para evitar que la magnitud de un feature particular sea considerada como “importancia”

El resultado del PCA

- De la matriz U obtenemos los vectores propios que nos indican la dirección de los ejes del nuevo sistema de coordenadas.
- De la matriz D obtenemos los valores propios que nos indican la cantidad de variación que tiene ese eje. La “importancia”
- Ordenandolo descendentemente y graficando cada eje se obtiene el gráfico Scree



Aplicaciones Prácticas

- Visualización del altas dimensiones
- Reducción de dimensiones
- Ofuscación de datos sensibles