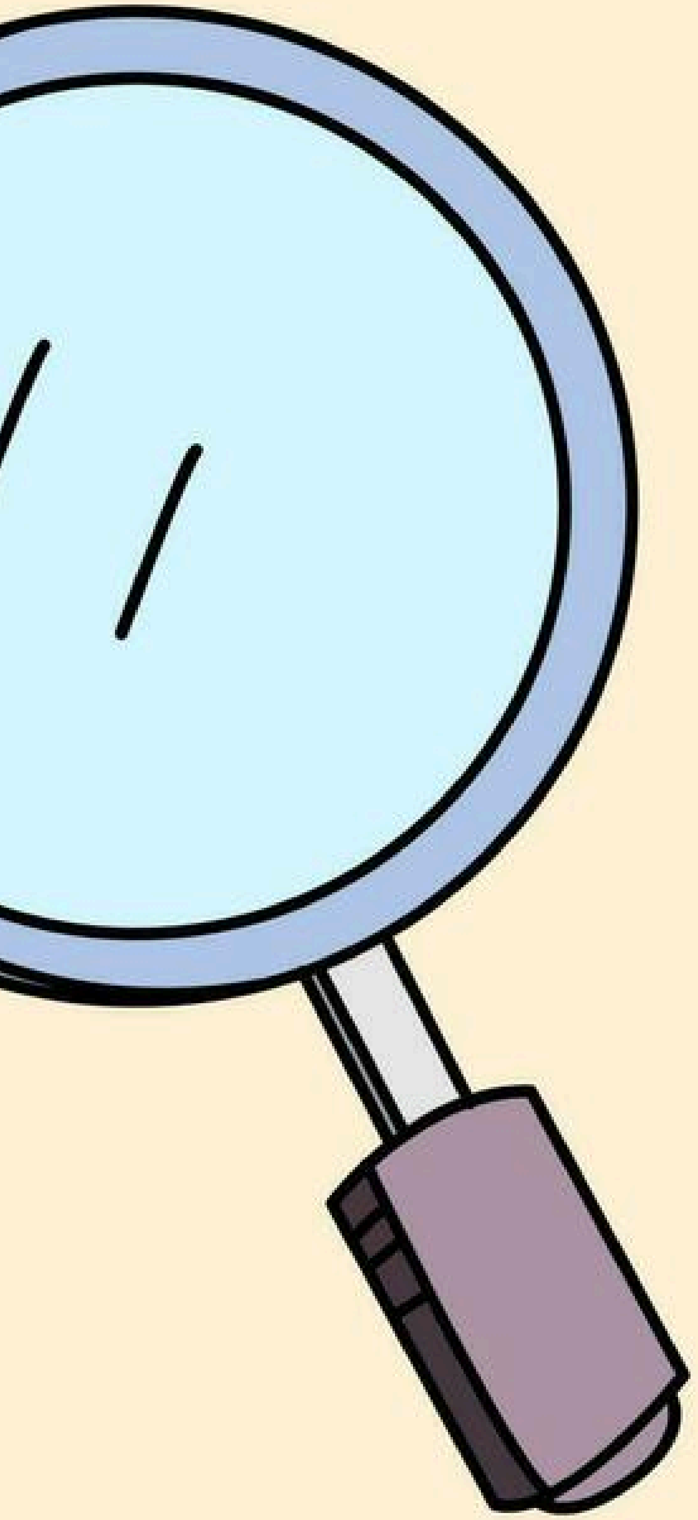
# MINERÍA DE DATOS

**Maximiliano Ojeda**

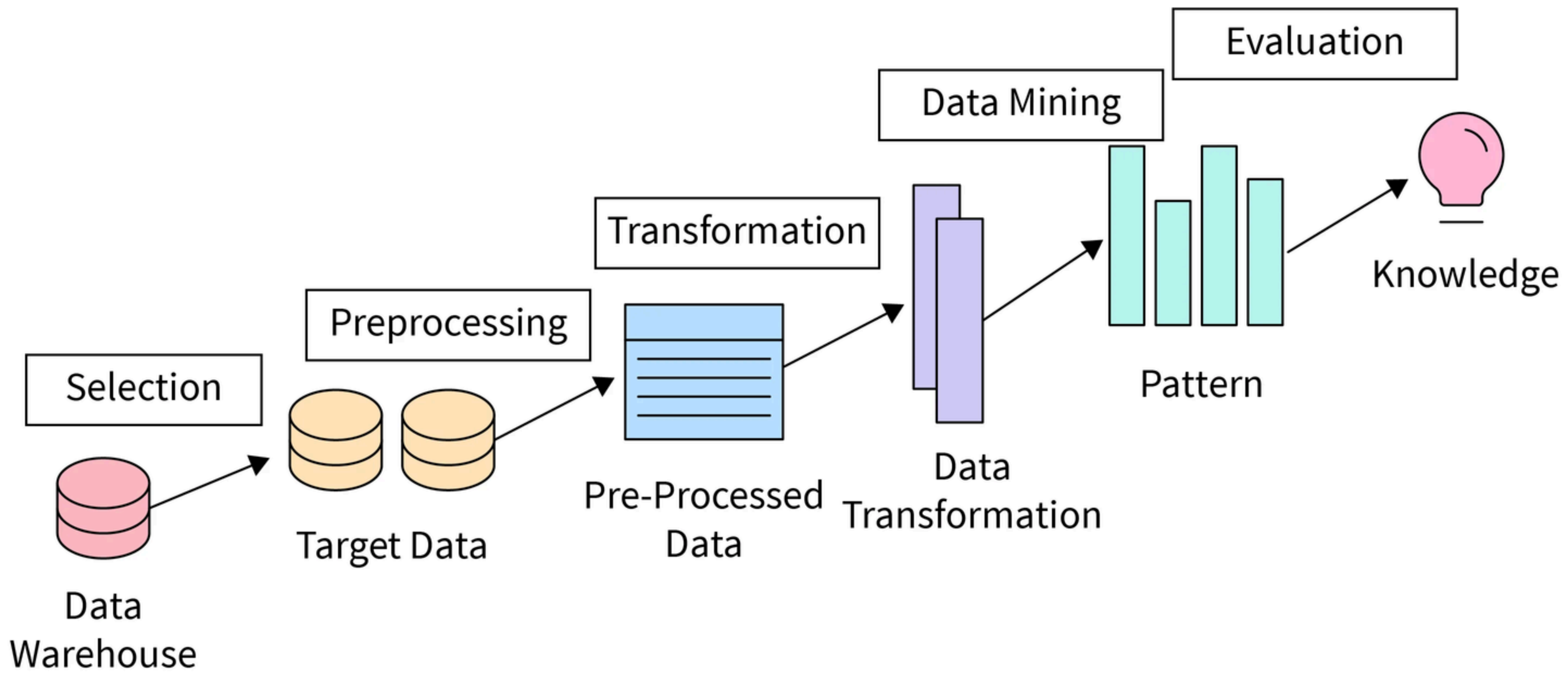
muojeda@uc.cl

IIC-2433



# Reducción de Dimensionalidad

# Knowledge Discovery in Databases (KDD)



# Data Reduction

- Representar los datos de forma más compacta
- Cuando los datos son más pequeños es más fácil aplicar algoritmos costosos computacionalmente
- Pérdida de información

# Data Reduction

## Sampling

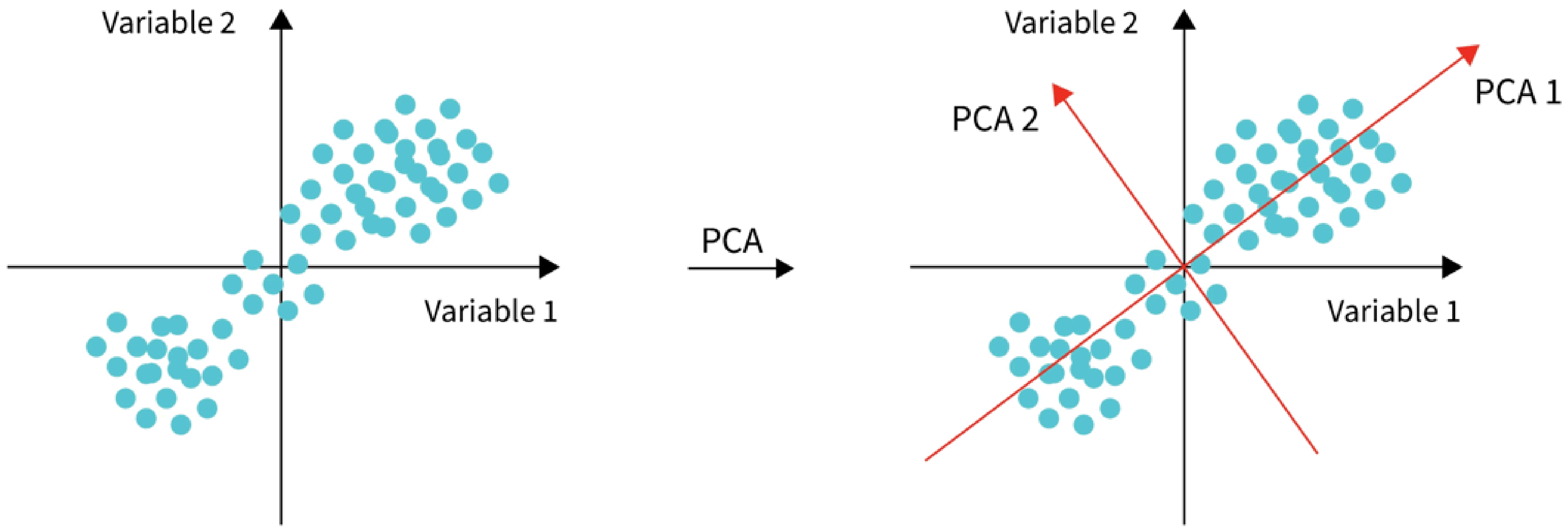
- Muestreo sin remplazo de un dataset de ***n*** filas es simplemente tomar aleatoriamente un total de $\lceil n \cdot f \rceil$ filas

- En el *stratified sampling* se selecciona una misma cantidad de ejemplo por “clase”


# PCA (Principal Component Analysis)

## Objetivo

- Capturar la **máxima varianza** de los datos en el menor número de dimensiones posible.
- Simplificar el dataset para análisis, visualización o como preprocesamiento antes de aplicar otros algoritmos de Machine Learning.



# Correlación (Recordatorio)

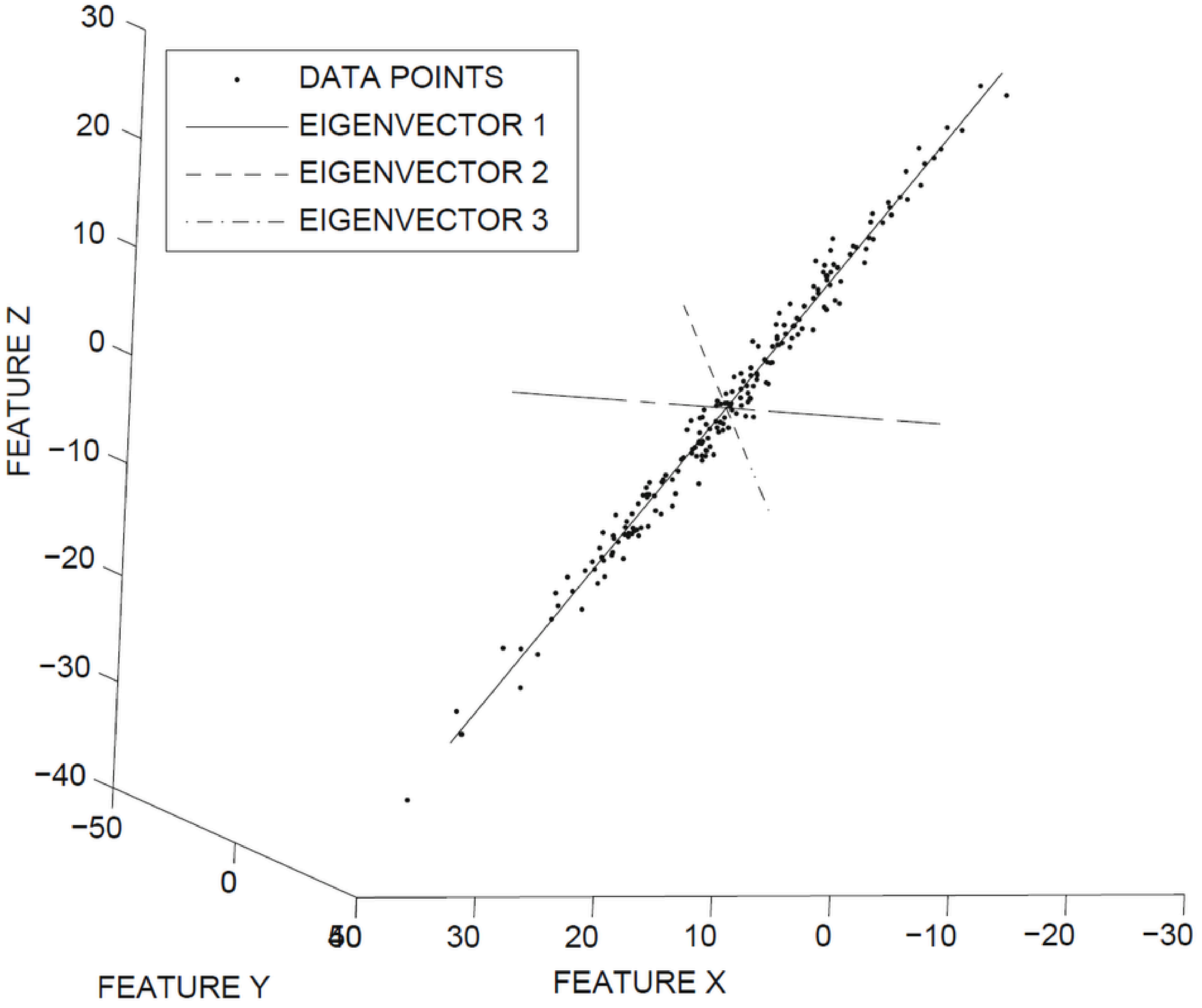
## Pearson

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

# PCA (Principal Component Analysis)

## Idea central

- Reducir la dimensionalidad de los datos (menos variables).
- Mantener la mayor cantidad posible de información (en términos de varianza).
- Eliminar redundancia debida a correlaciones entre variables.



# Proceso PCA

Tenemos un conjunto de datos:

$$D \in \mathbb{R}^{n \times d}$$

*n* cantidad de filas y *d* número de dimensiones (columnas)

## 1. Centrado de los datos

$$\mu = (\mu_1, \mu_2, \ldots, \mu_d), \quad \mu_j = \frac{1}{n} \sum_{k=1}^{n} x_j^{(k)}$$

Centrar matriz con las medias

$$X = D - \mathbf{1}\mu$$

# Proceso PCA

## 2. Matriz de covarianza

$$C = \frac{1}{n} X^T X$$

$$\begin{bmatrix} \mathrm{Cov}(x,x) & \mathrm{Cov}(x,y) & \mathrm{Cov}(x,z) \\ \mathrm{Cov}(y,x) & \mathrm{Cov}(y,y) & \mathrm{Cov}(y,z) \\ \mathrm{Cov}(z,x) & \mathrm{Cov}(z,y) & \mathrm{Cov}(z,z) \end{bmatrix}$$

## 3. Descomposición en autovalores y autovectores

$$C = P \Lambda P^T$$

- **P** → autovectores (direcciones de los nuevos ejes).
- **λ** → autovalores (cuánta varianza captura cada autovector).
- Los autovalores están ordenados de mayor a menor.

# Proceso PCA

## 4. Construcción de componentes principales

Sea la matriz

$$P_k = [v_1 \; v_2 \; \ldots \; v_k]$$

La proyección de los datos centrados sobre los primeros **k** componentes es:

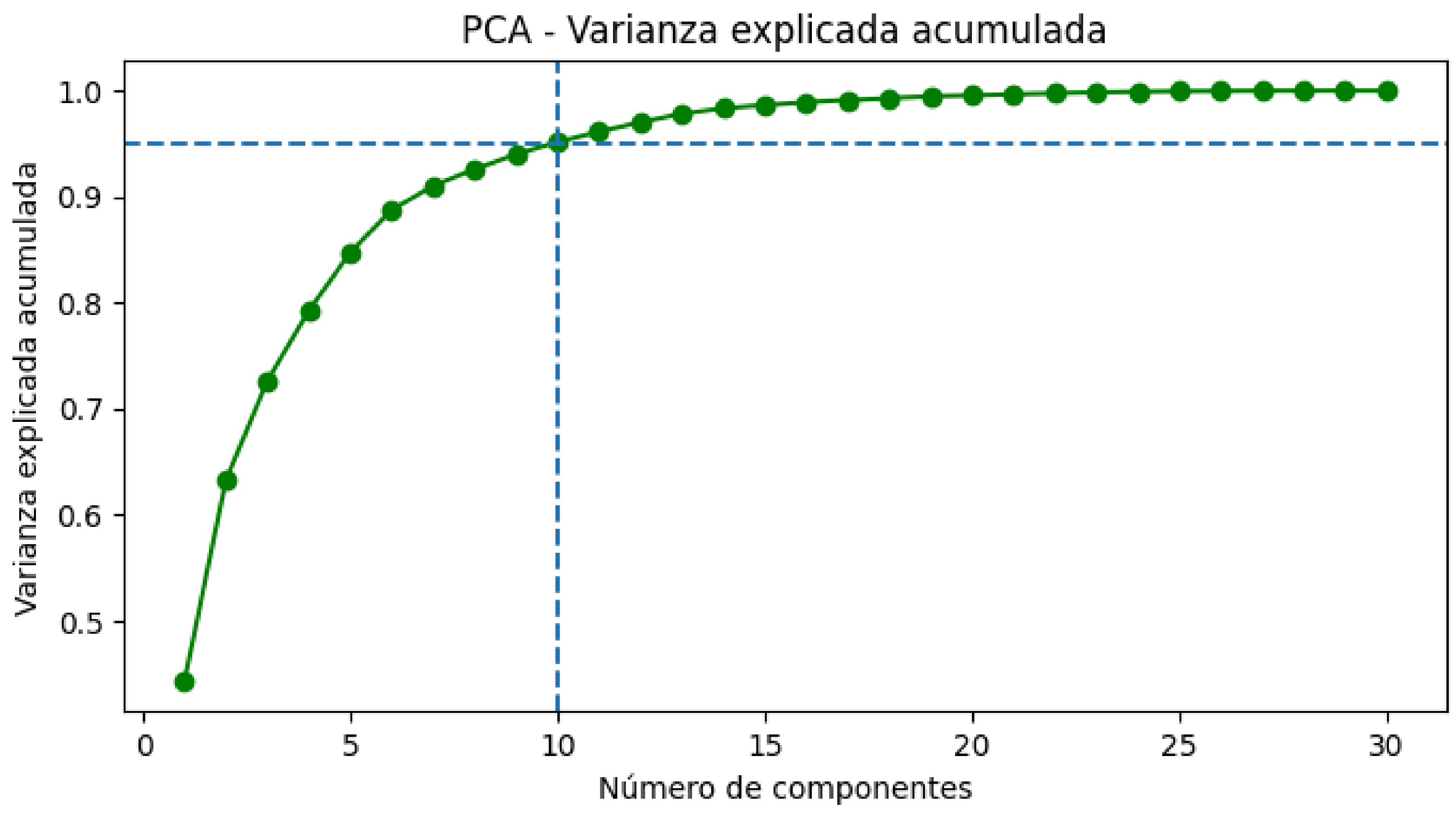
$$Z = XP_k$$

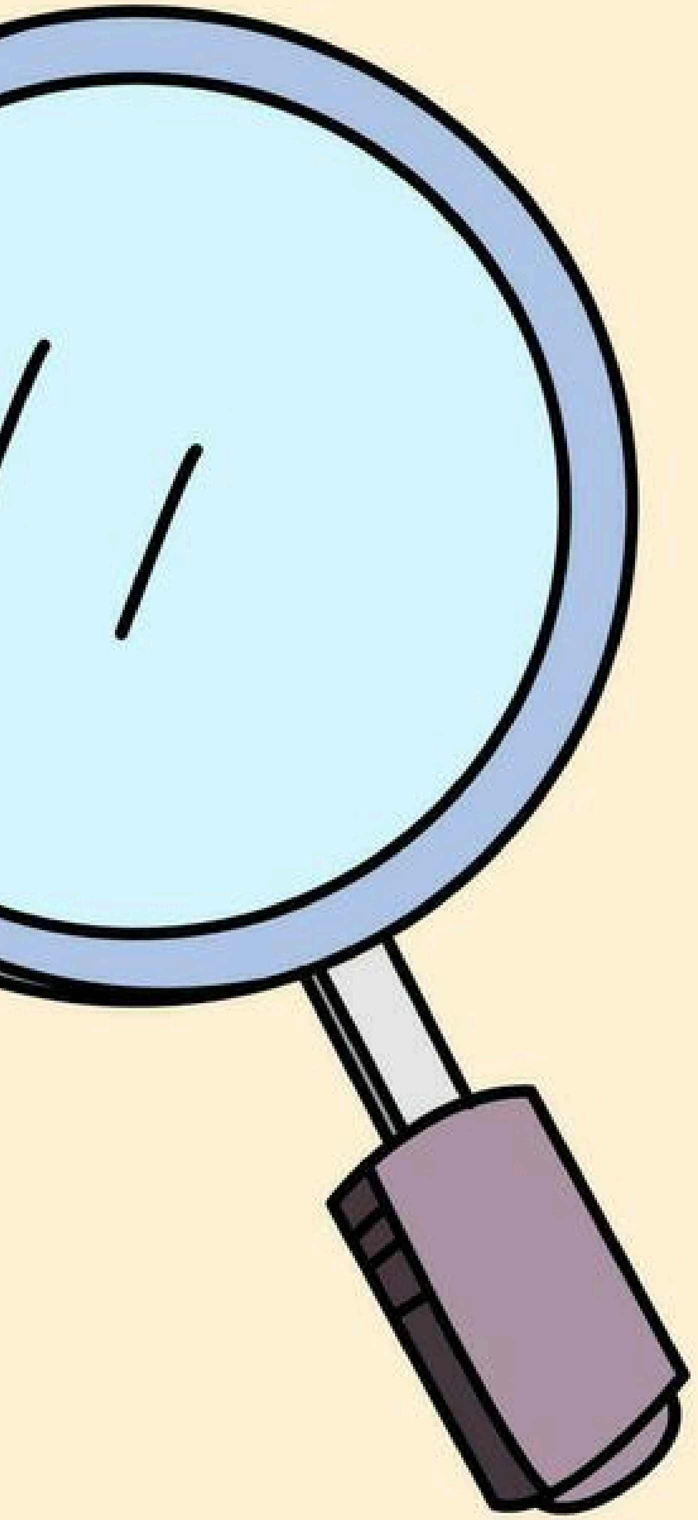
# Ejemplo PCA

```
data = load_breast_cancer()
X = data.data  # (filas, columnas)

X_std = StandardScaler().fit_transform(X)

# PCA (todas las componentes)
pca = PCA(n_components=X_std.shape[1], svd_solver='full')
X_pca = pca.fit_transform(X_std)
```





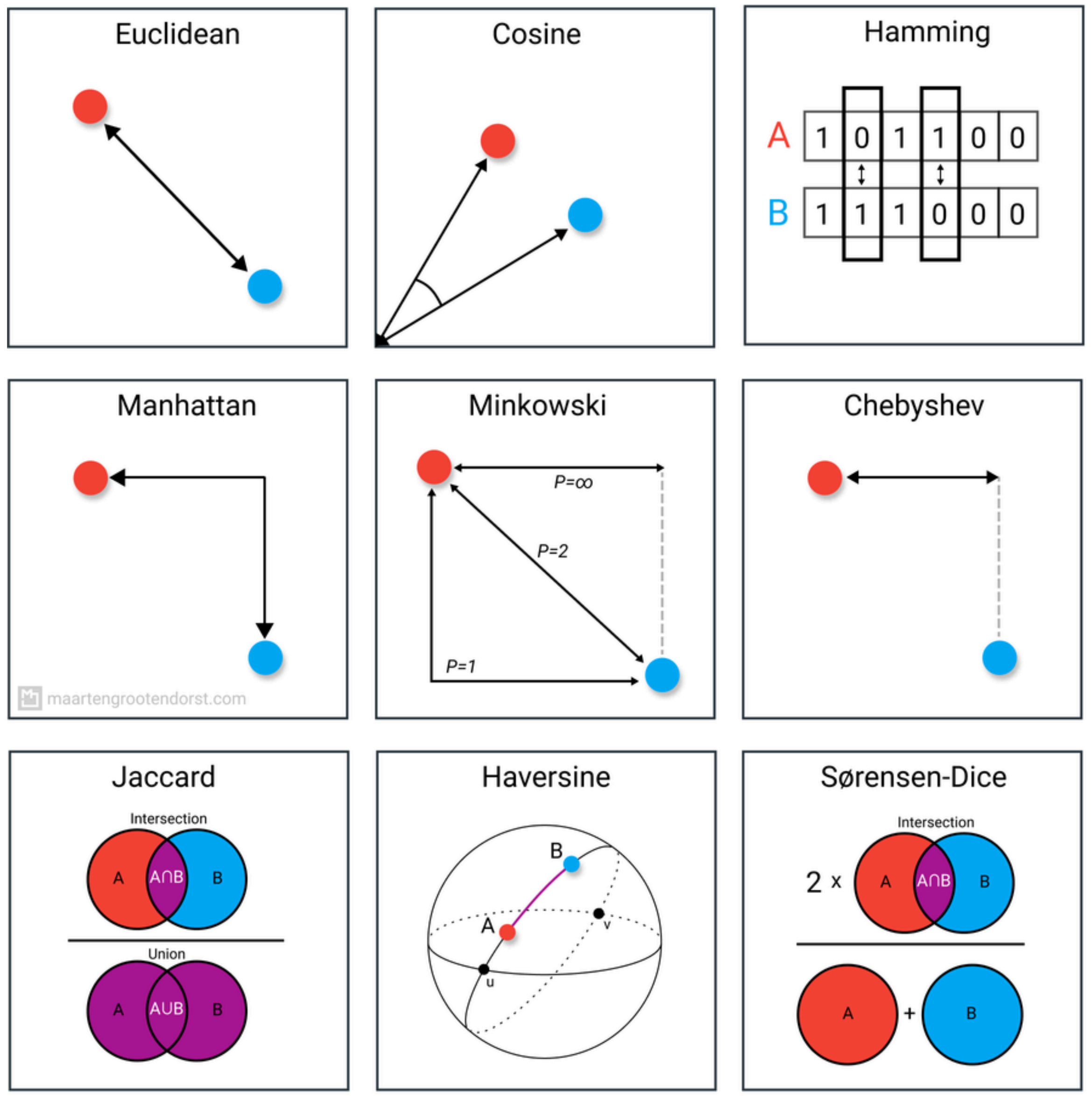
# Distancias y Similitud

# Similitud o Distancia en Data Mining

Dada una pareja de objetos $O_1$ y $O_2$, calcular

**Similitud:** $Sim(O_1, O_2)$ (valores altos → más similares)
**Distancia:** $Dist(O_1, O_2)$ (valores bajos → más similares)

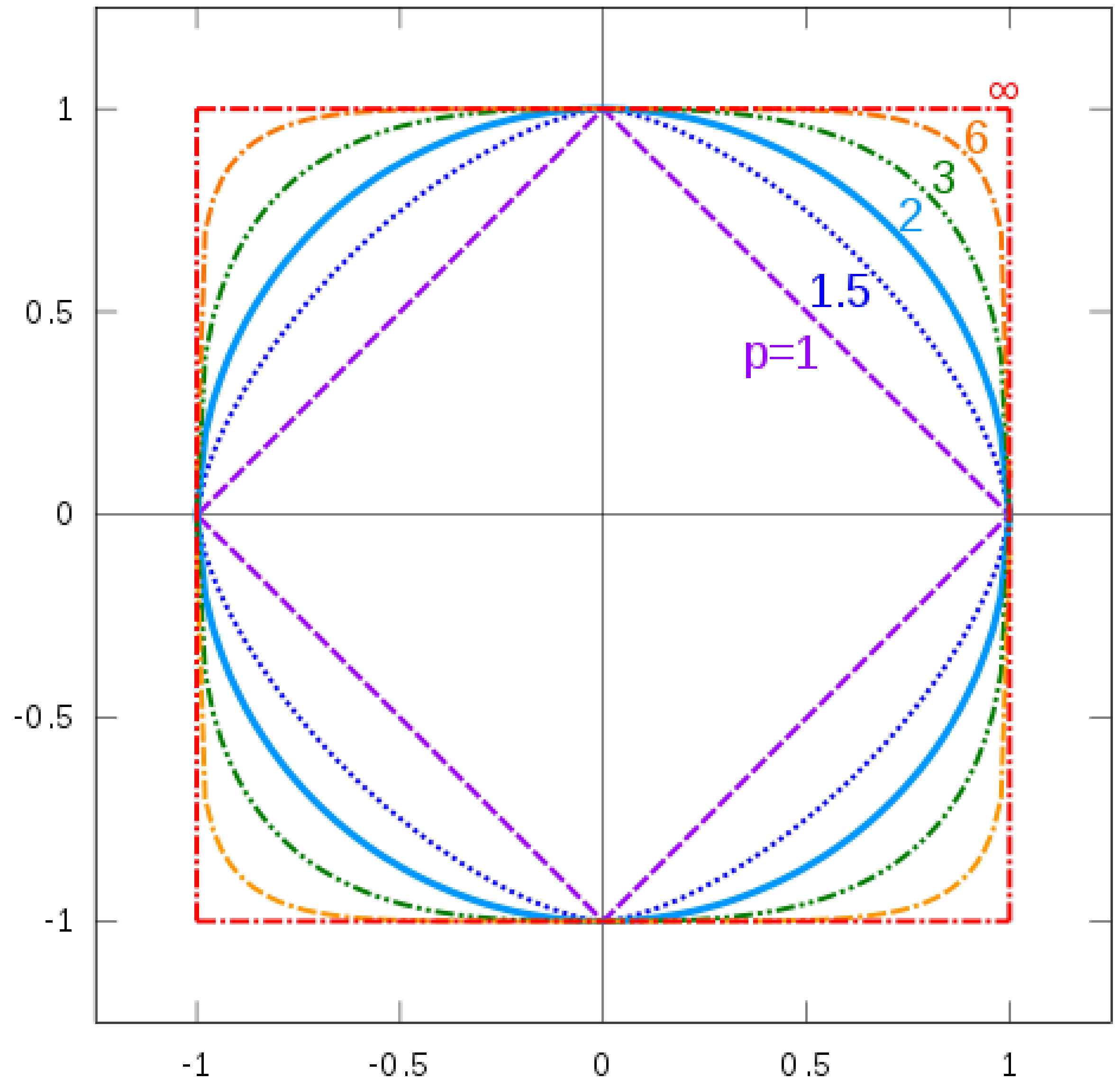
- Es clave para identificar objetos o patrones similares/diferentes.
- La elección entre función de distancia o similitud depende del dominio (datos espaciales, texto, series temporales, etc.).



# $L_p$-norm (Minkowski)

La distancia más común para data cuantitativa es $L_p$-Norm

$$Dist(\overline{X}, \overline{Y}) = \left( \sum_{i=1}^{d} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

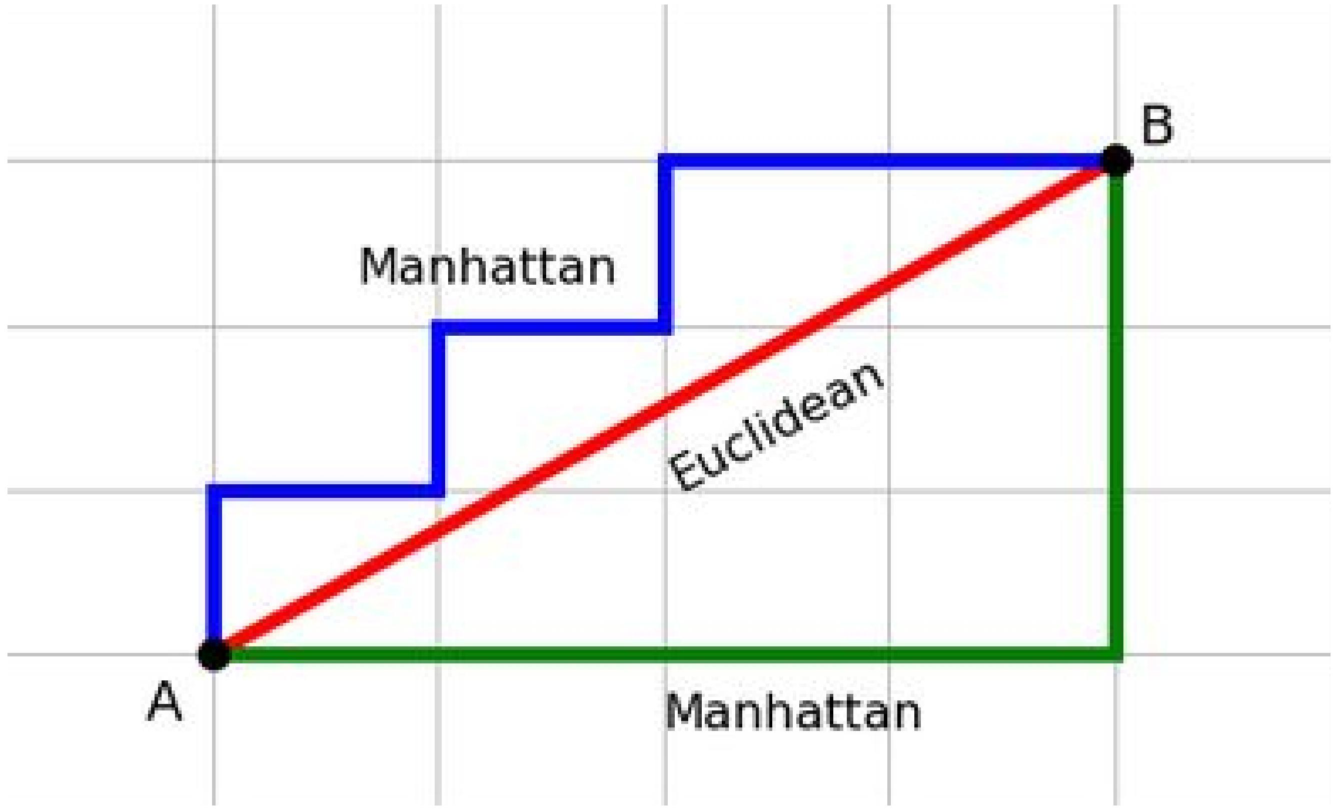


# Distancia Manhatann y Euclidiana

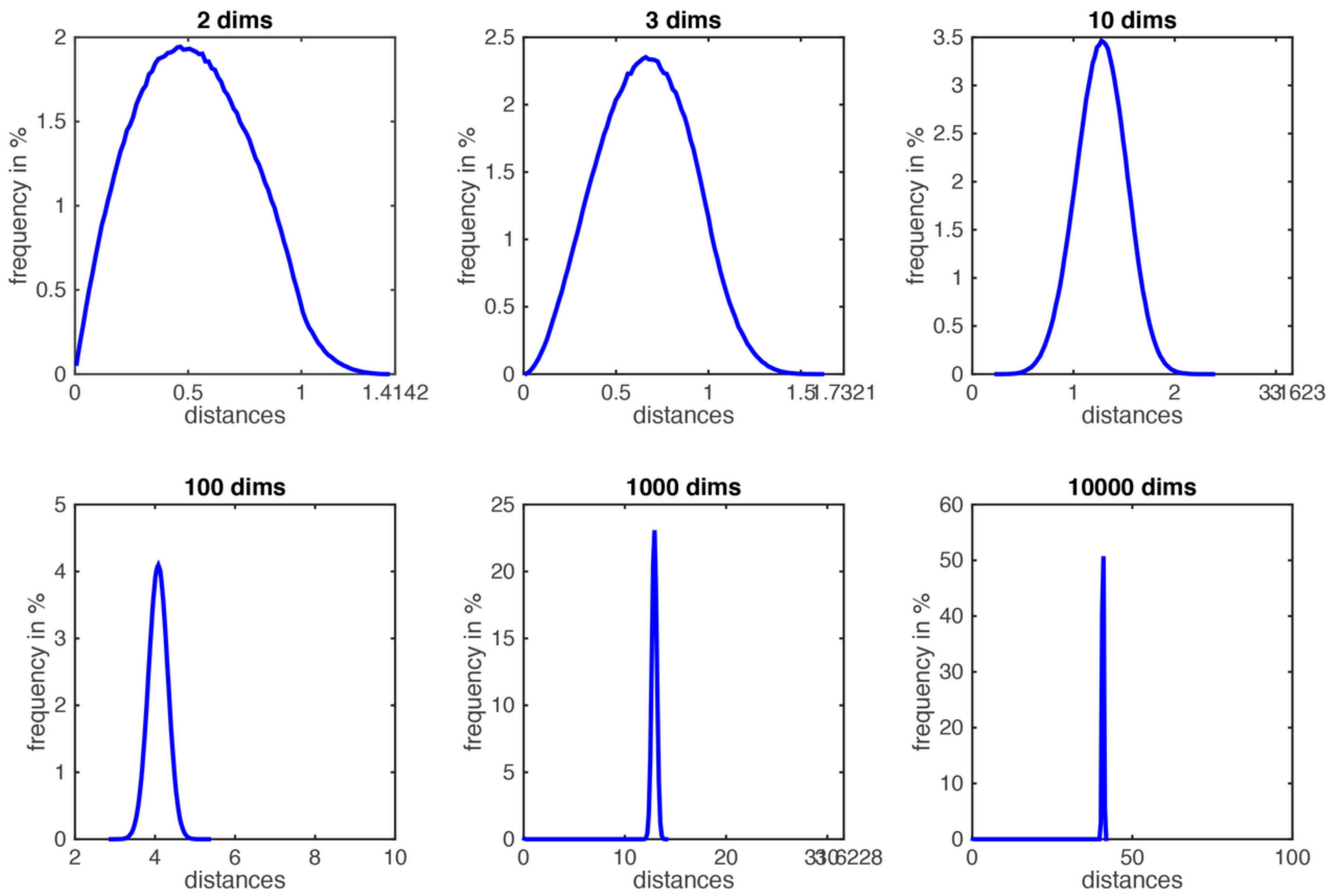
Casos especiales de $L_p$-Norm

$$Dist_{\text{Manhattan}}(X,Y) = \sum_{i=1}^{d} |x_i - y_i|$$

$$Dist_{\text{Euclidiana}}(X,Y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

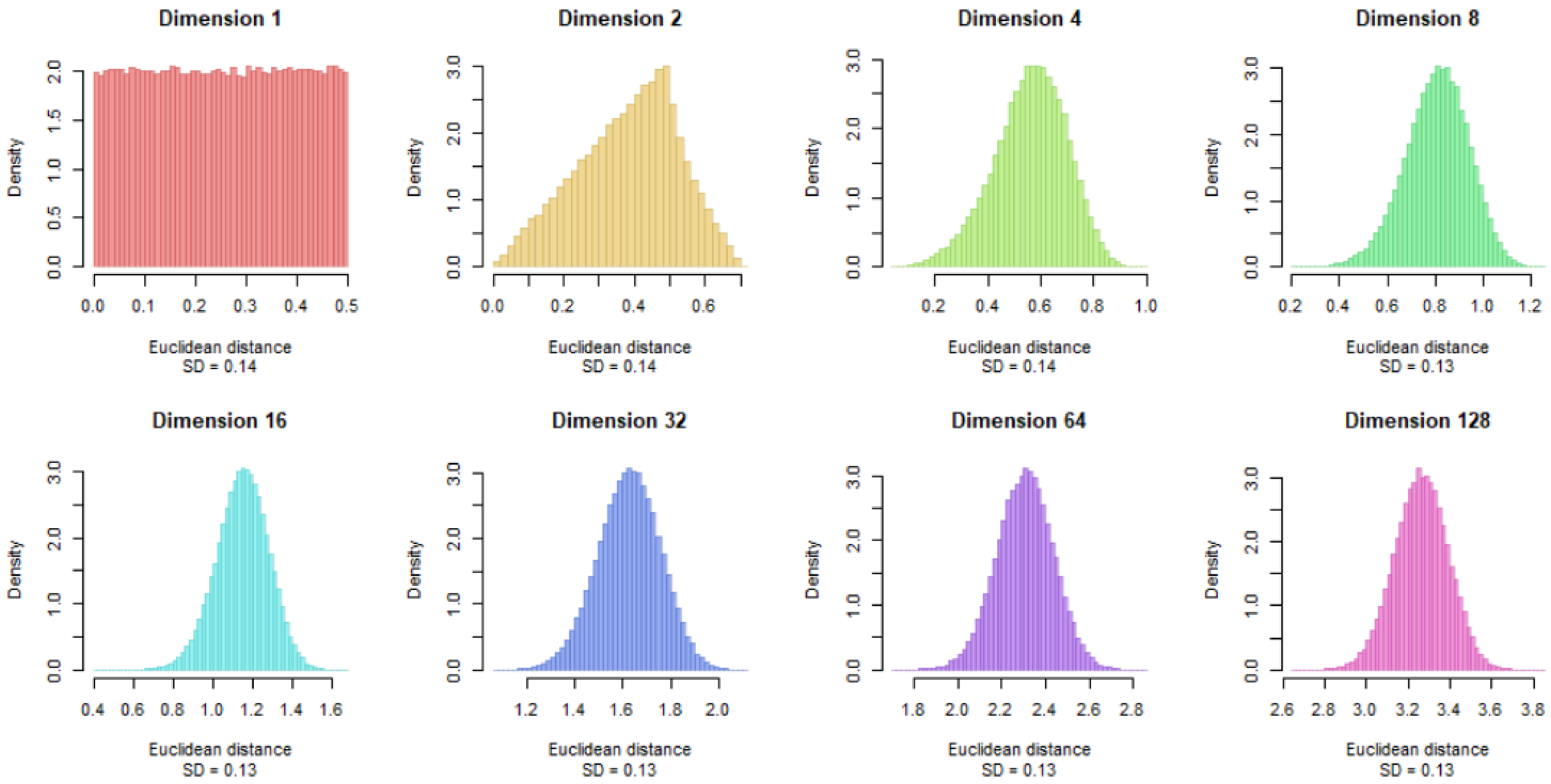


# Distancia Euclidiana




Maldición de la dimensionalidad

# Distancia Euclidiana




Maldición de la dimensionalidad

# Similitud Coseno

Para dos vectores de 100 dimensiones cada uno

$$X = [x_0, x_1, \ldots, x_{99}], \quad Y = [y_0, y_1, \ldots, y_{99}]$$

El producto interno se calcula como,

$$\langle X, Y \rangle = x_0 \cdot y_0 + x_1 \cdot y_1 + \cdots + x_{99} \cdot y_{99}$$

Geométricamente,

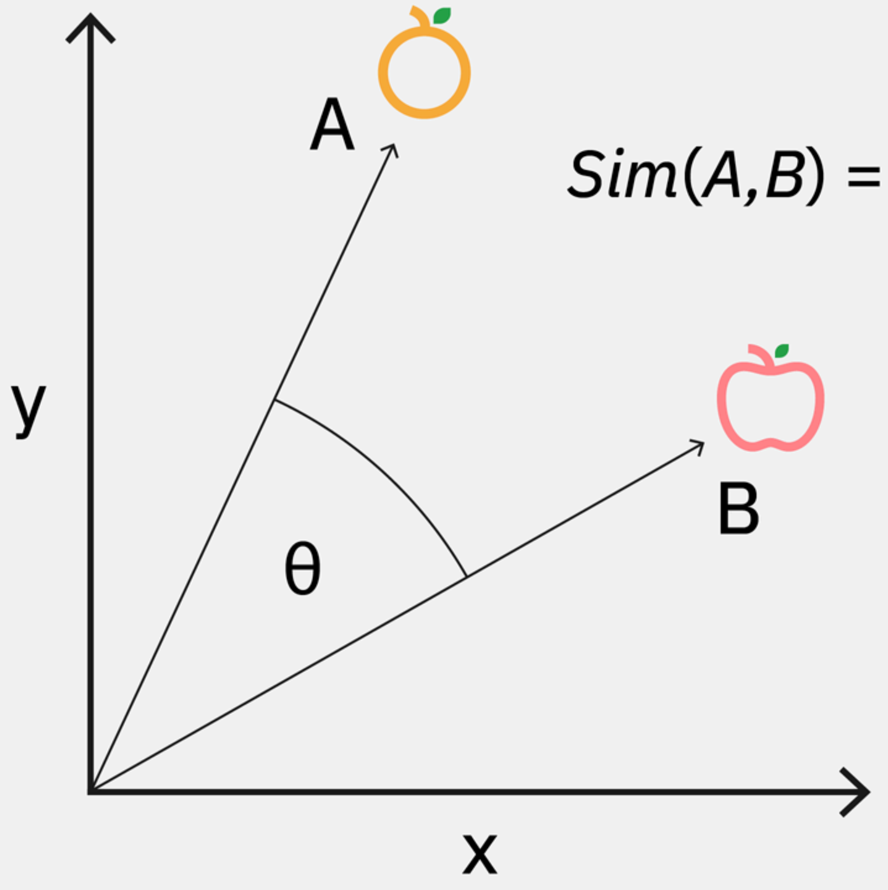
$$\langle X, Y \rangle = |X| \cdot |Y| \cdot \cos(\theta)$$

Despejando obtenemos,

$$\mathrm{Sim}_{cos}(X, Y) = \cos(\theta) = \frac{\langle X, Y \rangle}{|X| \cdot |Y|}$$

# Similitud Coseno

## Cosine Similarity




$$Sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\,\|B\|}$$

# MINERÍA DE DATOS

**Maximiliano Ojeda**

muojeda@uc.cl

IIC-2433