



# Co pamiętamy z poprzednich zajęć



# Gdzie znaleźć dane

# kaggle



# Etapy czyszczenia danych

- ✓ ogarnięcie NaN-ów
- ✓ poprawa niespójności (np. Male/M/man)
- ✓ usunięcie duplikatów
- ✓ usunięcie zbędnych kolumn
- ✓ poprawa typów danych (float, datetime)
- ✓ ogarnięcie outlierów
- ✓ walidacja logiczna
- ✓ normalizacja/standaryzacja





# Przygotowanie Danych

## STANDARDYZACJA



$\text{średnia} = 0$

$\text{odchylenie} = 1$

# Standaryzacja Danych

Aby sprowadzić dane do rozkładu normalnego:

- Wczytujemy dane,
- Z całego przedziału danych odczytujemy średnią i odchylenie standardowe,
- Wykonujemy standaryzację:  $S = (\text{dane} - \text{średnia}) / \text{odchylenie}$ .

Aby przywrócić dane do stanu przed standaryzacją:

- Wczytujemy parametry użyte w standaryzacji
- $\text{dane} = S * \text{odchylenie} + \text{średnia}$

# Standaryzacja Danych

```
import numpy as np
import numpy.typing as npt

# dane = []
# for _ in range(10):
#     dane.append(np.random.randint(-4,10))

# Generujemy losowe liczby całe z zakresu [-4; 10)
dane: list[int] = [np.random.randint(-4, high=10) for _ in range(10)] # ← Komprehensja list. Analog w komentarzu wyżej
dane_array: npt.NDArray = np.array(dane) # Konwersja na np.NDArray


def standaryzacja(dane: npt.NDArray) → tuple[npt.NDArray, tuple[np.float64, np.float64]]: 1 usage
    średnia: np.float64 = dane.mean()
    odchylenie: np.float64 = dane.std()
    return (dane - średnia) / odchylenie, (średnia, odchylenie)

def przywrocenie_standaryzacji(standaryzowane: npt.NDArray, średnia: np.float64, odchylenie: np.float64) → npt.NDArray:
    return standaryzowane * odchylenie + średnia


print(f"Dane wejściowe: {dane_array}")
standaryzowane, parametry = standaryzacja(dane_array)
print(f"Dane standaryzowane: {standaryzowane}\nŚrednia: {parametry[0]}\nSTD: {parametry[1]}")
przywrocone_dane = przywrocenie_standaryzacji(standaryzowane, *parametry)
print(f"Przywrócone dane: {przywrocone_dane}")
```

Przykład standaryzacji, Jakub Susoł

# Standaryzacja

średnia 0 z odchyleniem 1

KIEDY?

różne jednostki lub zakresy żeby zachować porównywalny wpływ (normalizacja spłaszcza różnice)

outliery są isotne

metoda zakłada rozkład normalny lub używa odległości euklidesowej

np. HCA, PCA, regresja, metody oparte o macierze kowariancji

# Normalizacja

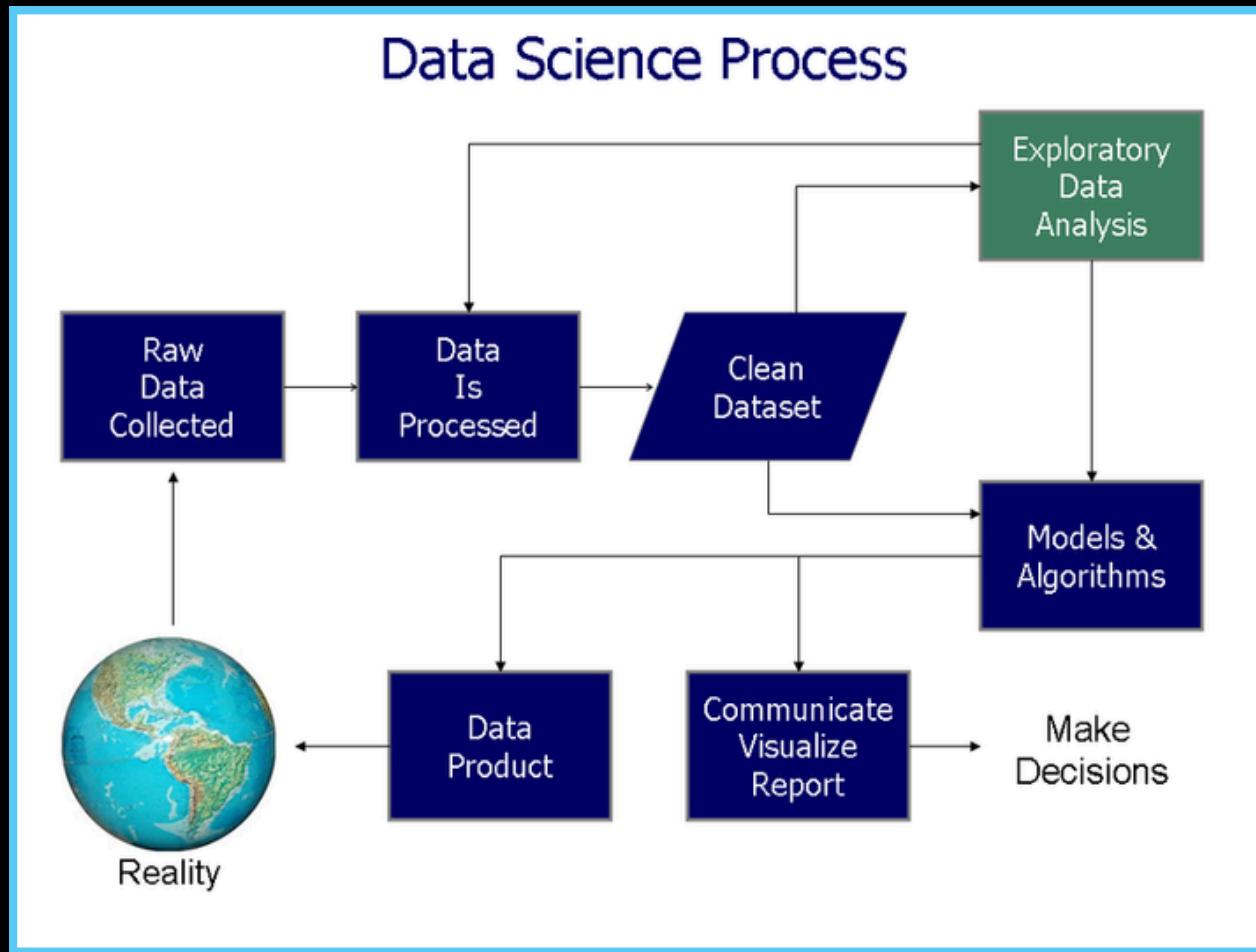
wartości w przedziale [0, 1]

ważne rzeczywiste proporcje wartości (np. obrazy)

metody oparte na odległości Manhattan

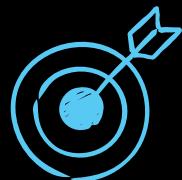
np. sieci neuronowe (szybsza nauka), KNN, gradient boosting

# Przepływ danych w świecie



# EDA

## (Exploratory Data Analysis)



CEL:  
zrozumienie rozkładu,  
zależności i problemów



# EDA

## (Exploratory Data Analysis)



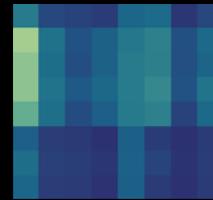
statystyki opisowe



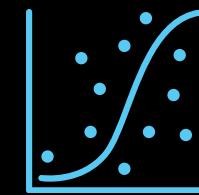
rozkłady, histogramy



rozkład klas

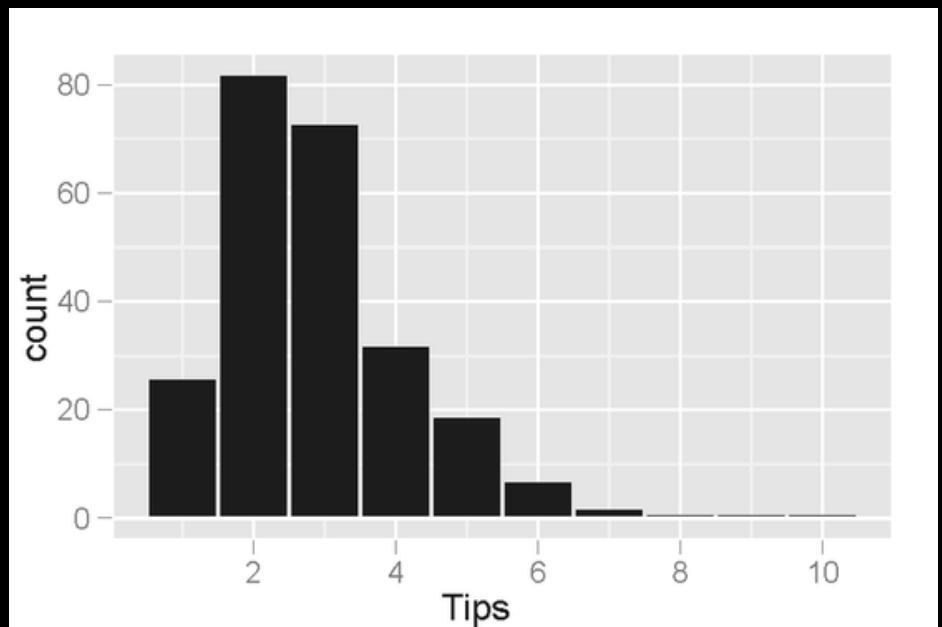
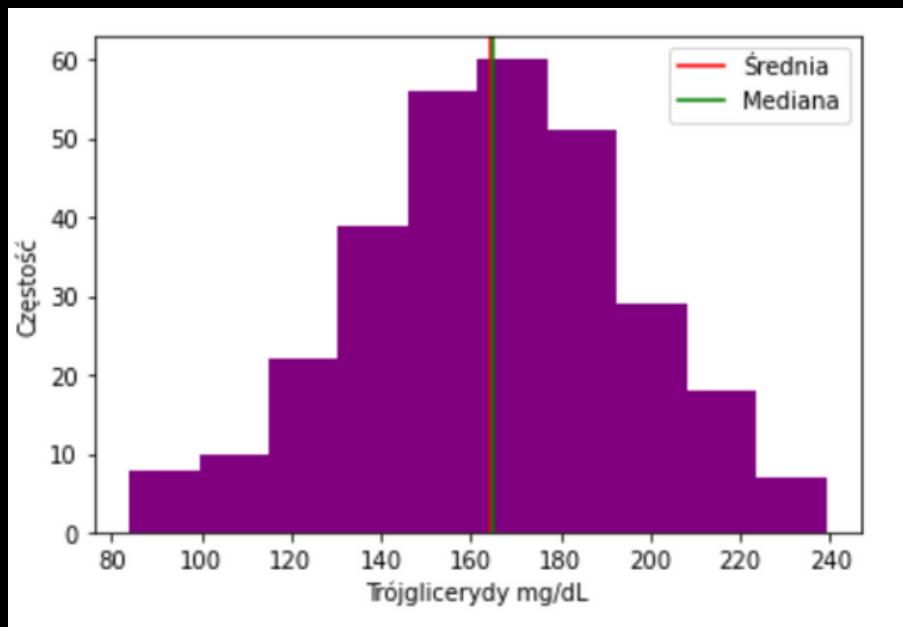


korelacje



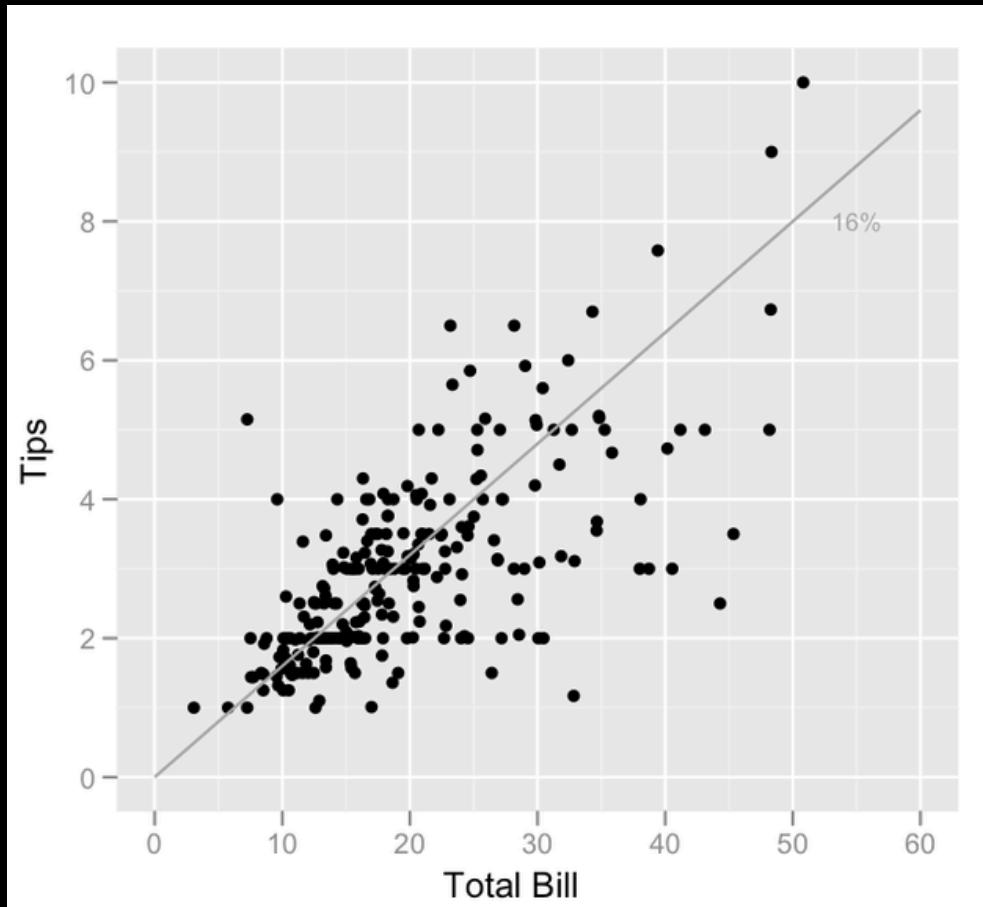
wizualizacje outlierów  
(boxplot, scatter plot)

# Histogramy



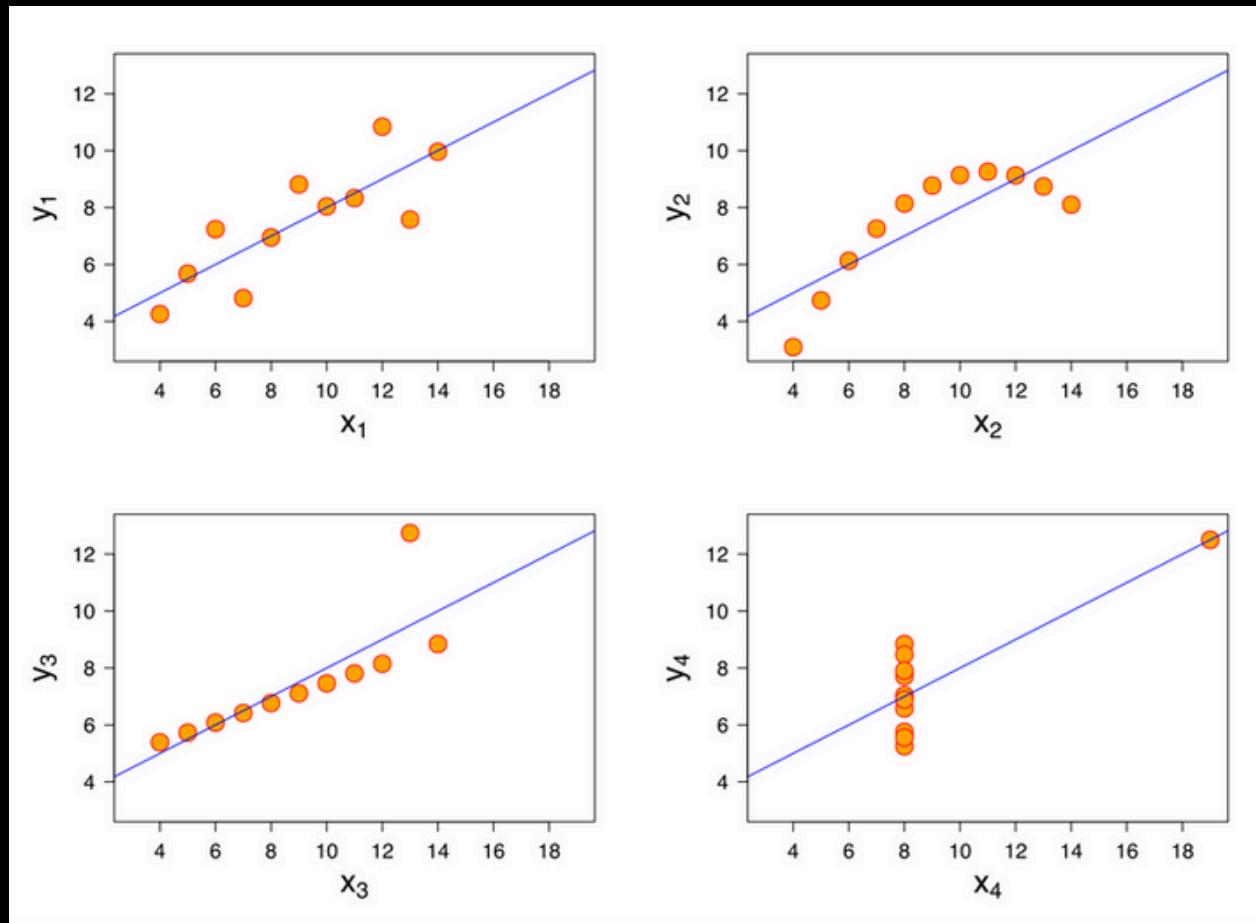
W jaki sposób dane są rozłożone w danej serii, np. ilość napiwków, gęstość trójkątów itp. Zwykle są wykonywane tylko dla jednej zmiennej

# Scatter Plot



Wykres relacji dla dwóch zmiennych, przedstawiający wzajemną zależność pomiędzy danymi.

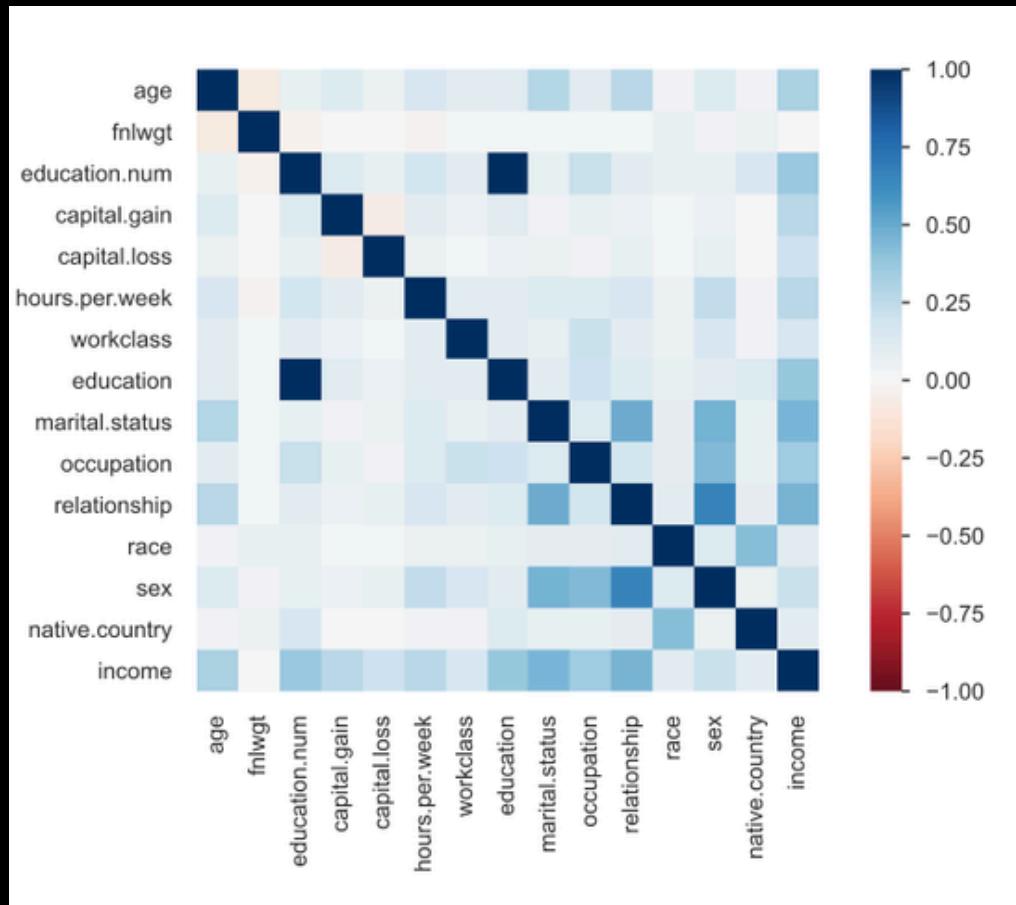
# Scatter Plot



Kwartet Anscomba - jak rozkład danych może się różnić dla zbiorów o takiej samej średniej, odchyleniu, równaniu regresji i współczynniku korelacji

[https://pl.wikipedia.org/wiki/Kwartet\\_Anscombe'a](https://pl.wikipedia.org/wiki/Kwartet_Anscombe'a)

# Korelacje



Jak zmiana jednej zmiennej wpływa na wartości drugiej

<https://towardsdatascience.com/a-data-scientists-essential-guide-to-exploratory-data-analysis-25637eee0cf6>

k-means be like:

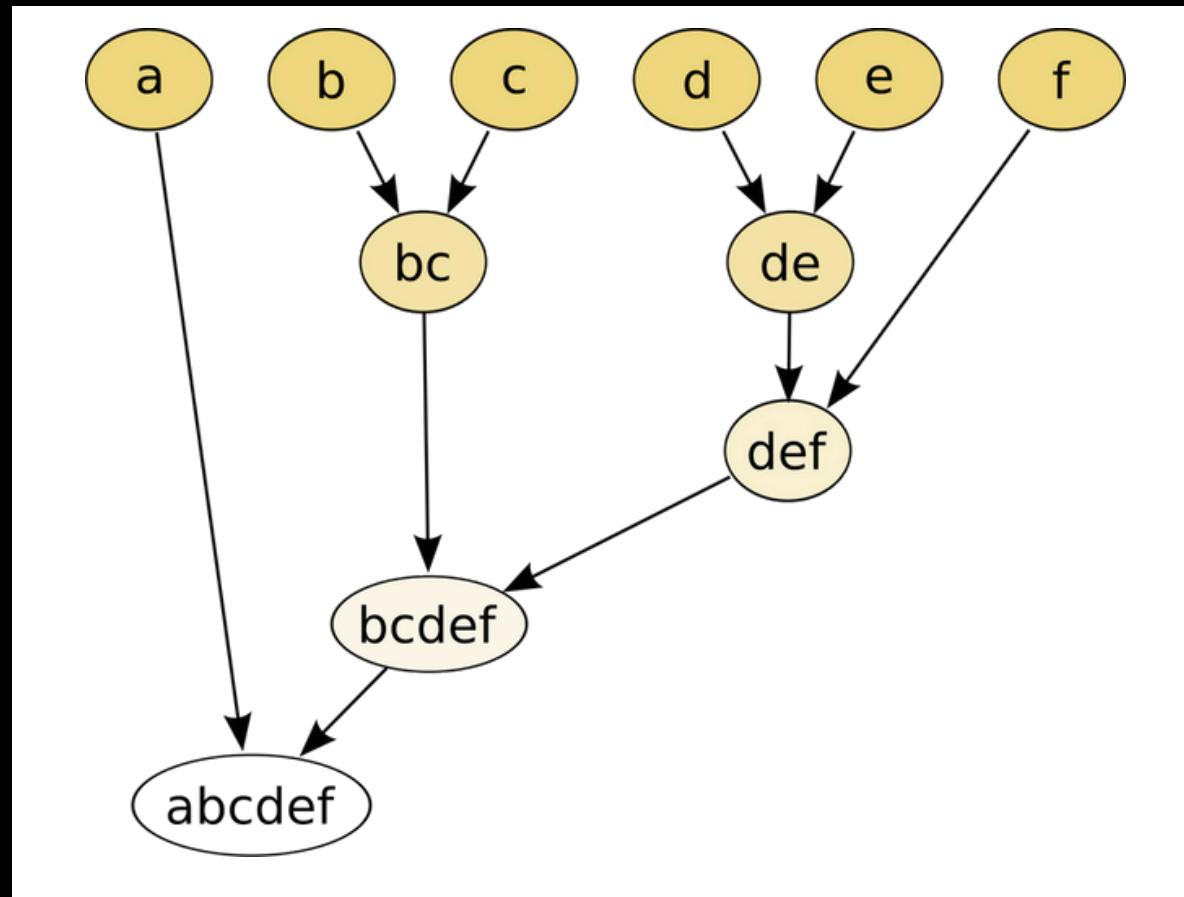


Klastrowanie

# Hierarchical Cluster Analysis (HCA)

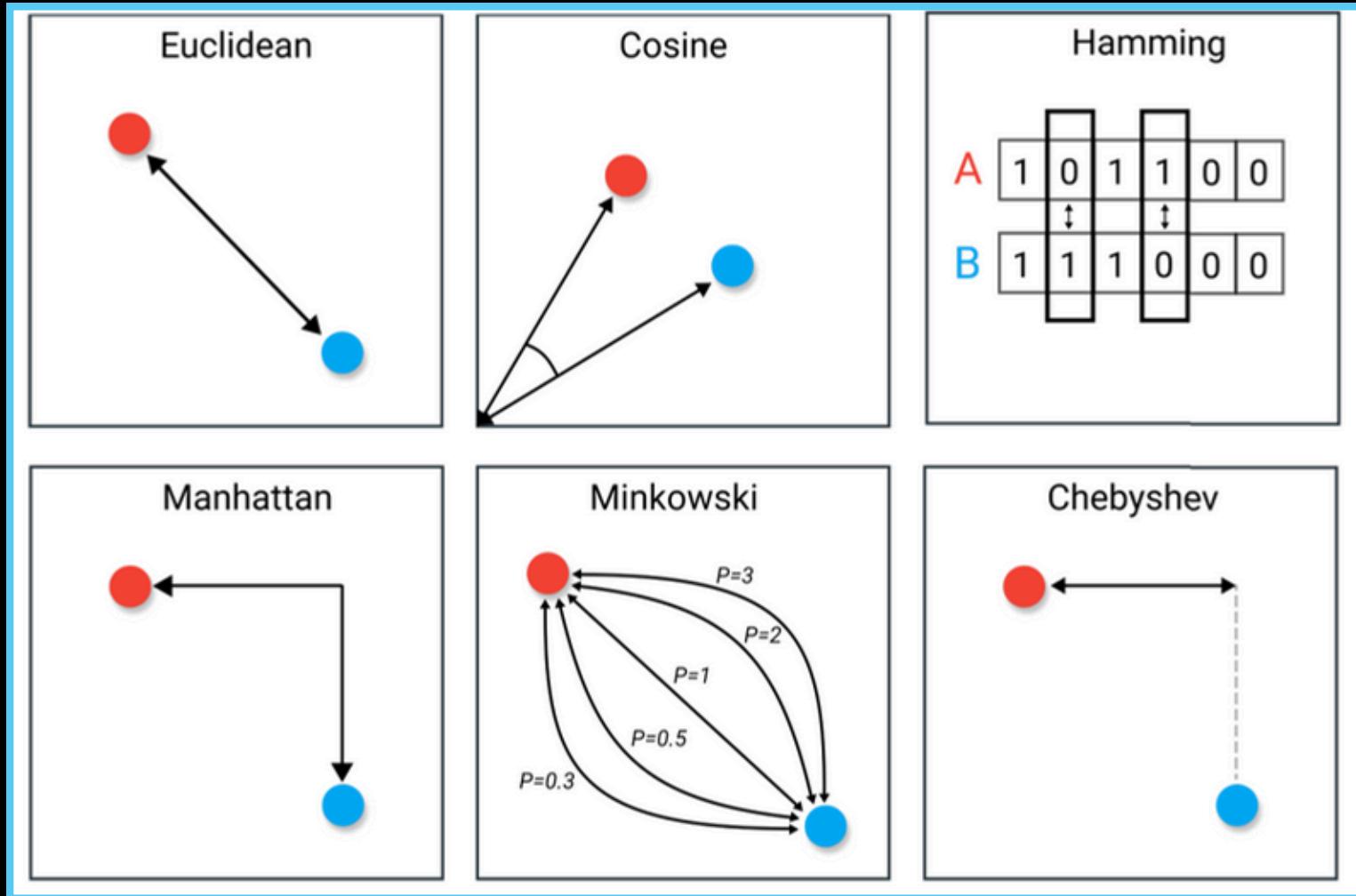


# HCA - Klastrowanie Hierarchiczne



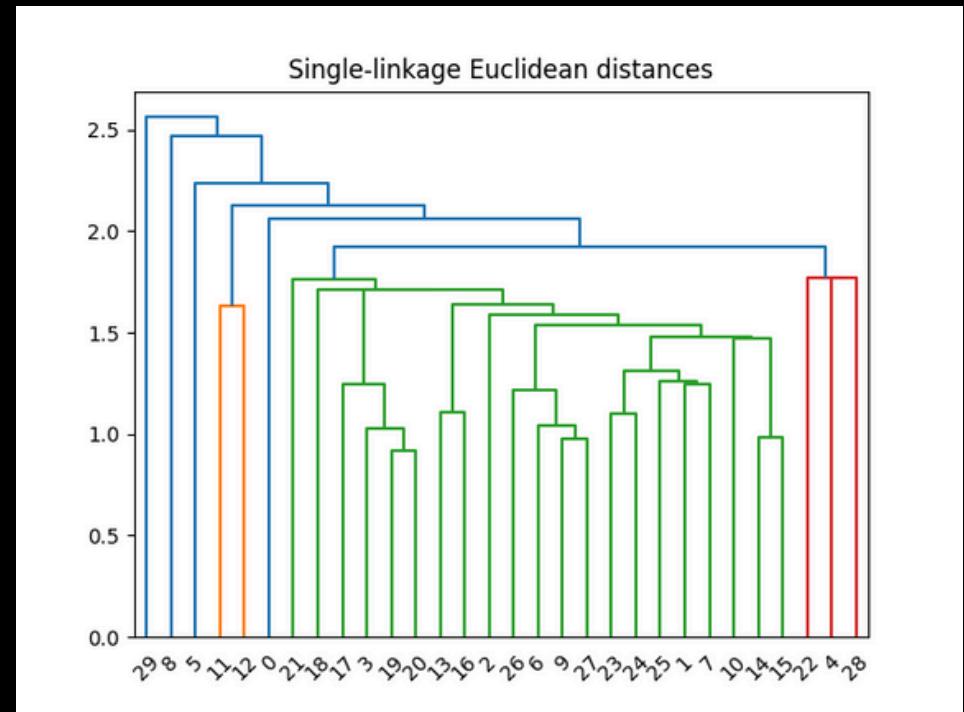
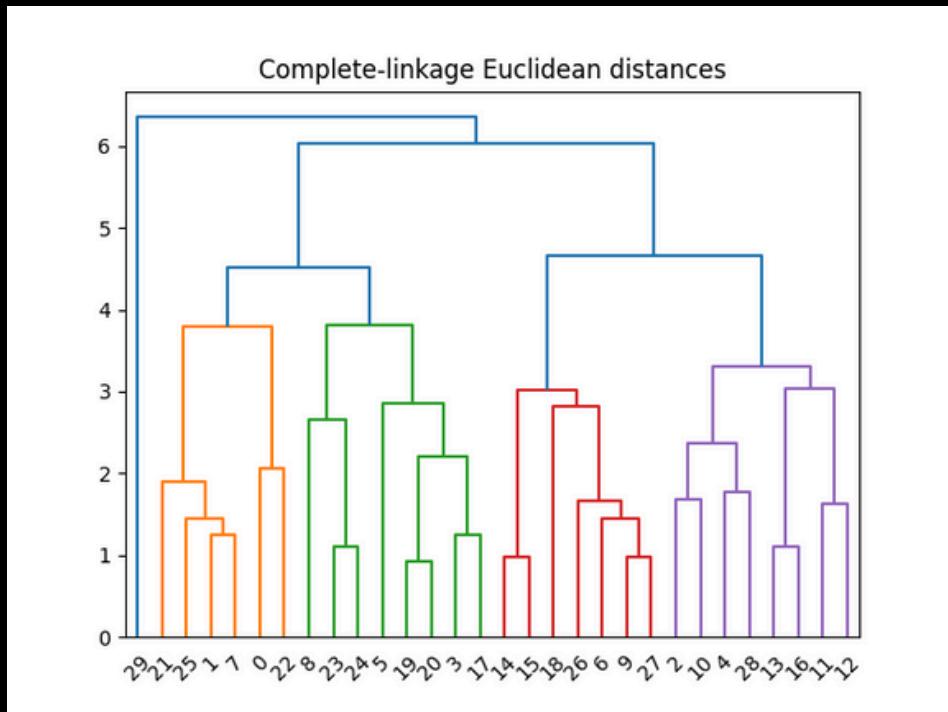
To nic innego jak grupowanie osobników o podobnych cechach, zwykle obliczając dystans między punktami.

# HCA - Miary Odległości



<https://www.maartengrootendorst.com/blog/distances/>

# HCA - Dendrogramy



węzły - skupienia  
liście - obiekty

# HCA - Metody łączenia

**Single linkage** (metoda najbliższego sąsiada) – odległość między dwoma klastrami to najmniejsza odległość między ich elementami.

**Complete linkage** (metoda najdalszego sąsiada) – używa największej odległości między elementami klastrów.

**Metoda Warda** – minimalizuje wzrost wariancji wewnętrz klastrów po połączeniu.

# HCA - Metody łączenia

Names	Formula
Maximum or complete-linkage clustering	$\max_{a \in A, b \in B} d(a, b)$
Minimum or single-linkage clustering	$\min_{a \in A, b \in B} d(a, b)$
Unweighted average linkage clustering (or UPGMA)	$\frac{1}{ A  \cdot  B } \sum_{a \in A} \sum_{b \in B} d(a, b).$
Weighted average linkage clustering (or WPGMA)	$d(i \cup j, k) = \frac{d(i, k) + d(j, k)}{2}.$
Centroid linkage clustering, or UPGMC	$\ \mu_A - \mu_B\ ^2$ where $\mu_A$ and $\mu_B$ are the centroids of $A$ resp. $B$ .
Median linkage clustering, or WPGMC	$d(i \cup j, k) = d(m_{i \cup j}, m_k)$ where $m_{i \cup j} = \frac{1}{2} (m_i + m_j)$
Versatile linkage clustering <sup>[9]</sup>	$\sqrt[p]{\frac{1}{ A  \cdot  B } \sum_{a \in A} \sum_{b \in B} d(a, b)^p}, p \neq 0$
Ward linkage, <sup>[10]</sup> Minimum Increase of Sum of Squares (MISSQ) <sup>[11]</sup>	$\frac{ A  \cdot  B }{ A \cup B } \ \mu_A - \mu_B\ ^2 = \sum_{x \in A \cup B} \ x - \mu_{A \cup B}\ ^2 - \sum_{x \in A} \ x - \mu_A\ ^2 - \sum_{x \in B} \ x - \mu_B\ ^2$
Minimum Error Sum of Squares (MNSSQ) <sup>[11]</sup>	$\sum_{x \in A \cup B} \ x - \mu_{A \cup B}\ ^2$
Minimum Increase in Variance (MIVAR) <sup>[11]</sup>	$\frac{1}{ A \cup B } \sum_{x \in A \cup B} \ x - \mu_{A \cup B}\ ^2 - \frac{1}{ A } \sum_{x \in A} \ x - \mu_A\ ^2 - \frac{1}{ B } \sum_{x \in B} \ x - \mu_B\ ^2$ $= \text{Var}(A \cup B) - \text{Var}(A) - \text{Var}(B)$
Minimum Variance (MNVAR) <sup>[11]</sup>	$\frac{1}{ A \cup B } \sum_{x \in A \cup B} \ x - \mu_{A \cup B}\ ^2 = \text{Var}(A \cup B)$
Hausdorff linkage <sup>[12]</sup>	$\max_{x \in A \cup B} \min_{y \in A \cup B} d(x, y)$
Minimum Sum Medoid linkage <sup>[13]</sup>	$\min_{m \in A \cup B} \sum_{y \in A \cup B} d(m, y)$ such that $m$ is the medoid of the resulting cluster
Minimum Sum Increase Medoid linkage <sup>[13]</sup>	$\min_{m \in A \cup B} \sum_{y \in A \cup B} d(m, y) - \min_{m \in A} \sum_{y \in A} d(m, y) - \min_{m \in B} \sum_{y \in B} d(m, y)$
Medoid linkage <sup>[14][15]</sup>	$d(m_A, m_B)$ where $m_A, m_B$ are the medoids of the previous clusters
Minimum energy clustering	$\frac{2}{nm} \sum_{i,j=1}^{n,m} \ a_i - b_j\ _2 - \frac{1}{n^2} \sum_{i,j=1}^n \ a_i - a_j\ _2 - \frac{1}{m^2} \sum_{i,j=1}^m \ b_i - b_j\ _2$

Metod klastrowania danych w HCA jest cała masa, każda przydaje się w różnym zastosowaniu zależnym od typu danych czy potrzebnej analizy.

# HCA - Metody łączenia

**Single linkage** - dobre, gdy klastry są wydłużone albo nieregularne, ale potrafi połączyć wszystko w jedno

**Complete linkage** - super, gdy chcesz mieć ładne, zwarte grupy i wyraźne granice; jeśli masz odstające punkty to działa gorzej

**Average linkage** - coś pomiędzy single a complete, sensowne, zbalansowane klastry, bezpieczny wybór

**Ward** - najlepsza, gdy masz dane liczbowe i chcesz, żeby klastry były jak najbardziej jednorodne, często wybierana w praktyce, bo daje czyste, równe grupy

# HCA - Klastrowanie Hierarchiczne



# HCA - w służbie policji

Andrzej Porębski<sup>1</sup>

**Analiza skupień zorganizowanych grup przestępczych  
jako przykład zastosowania metod statystycznych  
w kryminologii**



