

1 Curs 5. Tipuri de date. Calitatea si preprocesarea datelor. Statistici descriptive

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import matplotlib
matplotlib.style.use('ggplot')
import pandas as pd

print(f'NumPy version: {np.__version__}')
print(f'pandas version: {pd.__version__}')
print(f'matplotlib version: {matplotlib.__version__}')
```

executed in 855ms, finished 17:57:11 2021-03-22

NumPy version: 1.19.2
pandas version: 1.2.3
matplotlib version: 3.3.4

1.1 Tipuri de date

- Un set de date este o colectie de obiecte-data (eng: data objects) si de attribute.
- Sinonime pentru obiecte-data: înregistrare, punct, vector, pattern (termen ce poate induce confuzie), eveniment, caz, esantion (termen ce poate induce confuzie), observatie, entitate.
- Obiectele sunt descrise prin attribute
- Sinonime pentru atribut: **variabila**, **caracteristica**, **trasatura (feature)**, **dimensiune** (a nu se confunda cu omonimul din algebra).

Atribute

Date	Id	Refund	Marital Status	Taxable Income	Cheat
	1	Yes	Single	125K	No
	2	No	Married	100K	No
	3	No	Single	70K	No
	4	Yes	Married	120K	No
	5	No	Divorced	95K	Yes
	6	No	Married	60K	No

Definitie: Atribut: proprietate sau caracteristica a unui obiect ce poate sa varieze fie de la un



obiect la altul, fie de la un moment de timp la altul.

- Exemple: culoarea ochilor, temperatura.
- Trebuie facuta diferenta între proprietatile atributelor si proprietatile valorilor atributelor:
 - acelasi atribut poate avea valori diferite: înaltimea poate fi masurata în metri sau picioare
 - diferite atribute pot fi masurate cu acelasi tip de date, dar proprietatile atributelor pot fi diferite: pentru niste persoane, atributele "înaltime" si "id" sunt reprezentate prin numere întregi; în timp ce are sens sa faci media înaltimilor, nu are nicio noima media id-urilor; operatiile ce se pot face pentru înaltime (medie, max etc.) nu se aplica si pentru id-uri; id-urile nu au un maxim, în timp ce înaltimea - da.

▼ 1.1.1 Tipuri de atribute

Exista diferite tipuri de atribute:

- Categoriale (calitative)
 - **nominale**: valori diferite care permit recunoasterea diferentelor; exemple: cod postal, id-uri, culoarea ochilor, genul; operatii permisibile: =, \neq ;
 - **ordinale**: valorile permit ordonarea obiectelor; exemple: scara duritatii mineralelor, grade (militare etc.), gradul de satisfactie pentru un anumit produs; operatii permise: =, \neq , <, >; functii aplicabile: mediana, percentile etc.
- Numerice (cantitative)
 - **interval**: se poate face diferenta între valori (i.e. exista unitati de masura asociate); exemple: date calendaristice, temperaturi in grade Celsius sau Fahrenheit; pe lânga operatiile de mai sus admit si adunare, scadere; functii aplicabile: media, deviatia standard, corelatia
 - **multiplabile**: permit împartiri si înmultiri; exemple: temperatura în Kelvin, cantitati monetare, numar de elemente, vârsta, greutate; operatii: cele de mai sus si înmultirea, împartirea; functii aplicabile: media geometrica, variatie procentuala.

▼ 1.1.2 Transformari la nivel de atribute

Exista niste transformari care se pot efectua sau nu asupra unor atribute:

- pentru atribute nominale: orice asociere unu-la-unu (bijectie), de exemplu permutari; daca toti angajatii au un id, reassignarea lor nu ar modifica esenta datelor;
- pentru atribute ordinale: orice modificare de valori care respecta ordinea datelor (transformare monotona): $val_noua = f(val_veche)$, unde $f(\cdot)$ functie monoton crescatoare; {bun, mai bun, cel mai bun} poate fi reprezentat prin {1, 2, 3} sau la fel de bine prin {0.3, 12, 14};
- pentru atribute interval: transformari de forma $a * val_veche + b$ unde a si b sunt constante; ex: transformarea din Celsius în Fahrenheit;
- pentru atribute multiplabile: $val_veche/val_noua = r$; ex: raportul greutatii lui x si y este 2.

▼ 1.1.3 Descrierea atributelor prin numarul de valori

- Atribute discrete:

- o multime cel mult numarabila de valori;
- exemple: coduri postale, cuvinte într-un document
- se reprezinta cel mai frecvent ca numere naturale
- caz special — attribute binare: {prezent, absent}
- Attribute continue:
 - valorile sunt exprimate prin numere reale
 - exemple: temperatura, masa
 - dpdv practic reprezentarea se face cu o precizie finita
 - reprezentare actuala: valori în virgula mobila
- Valori asimetrice:
 - doar prezenta unei trasaturi (i.e. valoare non-zero) este importanta
 - exemple: vectorul care reprezinta daca niste cuvinte sunt prezente (eventual: de cate ori) într-un document
 - daca se iau in considerare doi astfel de vectori, conteaza mai mult cuvintele pe care le au în comun decât cuvintele care lipsesc din ambele documente, simultan

▼ 1.1.4 Tipuri de seturi de date

- Seturi de date de tip: înregistrare, de tip graf si de tip secventa
- Caracteristici generale:
 - **dimensionalitatea** = numarul de attribute pe care obiectele-data le au. Un numar de dimensiuni prea mare duce la “blestemul dimensionalitatii”; pentru multe dimensiuni se pot aplica tehnici de reducere a numarului de dimensiuni;
 - **caracterul rarefiat al datelor** = procentul de date utile; de exemplu, pentru date asimetrice este numarul de valori nenule. Specularea acestui caracter poate reduce drastic necesarul de memorie sau timpul de calcul;
 - **rezolutia** = scara la care se face raportarea valorilor; e posibil ca scari diferite sa releve (sau sa ascunda) pattern-uri; ex: masuratori meteo raportate pe zile pot arata iminenta unei furtuni, dar la scala de saptamâni asa ceva nu mai e vizibil

▼ 1.1.4.1 Seturi de date de tip înregistrare

- cel mai des furnizate si frecvent utilizate in aplicatii: multime de obiecte cu un set predefinit de attribute
- nu exista legatura între înregistrari distincte
- stocare: fisiere text (e.g. CSV), Excel, baze de date relationale – views

Cazuri remarcabile de seturi de date înregistrare:

- Tranzactii, date specifice cosurilor de cumparaturi:
 - exemplu: într-un magazin, setul de produse cumparate de un client in timpul unei sesiuni de cumparaturi = continutul cosului de cumparaturi
 - se analizeaza asocierea între produsele individuale din tranzactii
 - posibilitate de reprezentare: indicator boolean care arata daca un produs anume face sau nu parte dintr-un cos de cumparaturi
 - variatie: cate exemplare din produs au fost achizitionate (0, 1, . . .)

<i>TID</i>	<i>Items</i>
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

- Matrice de date:
 - pentru cazul in care datele au acelasi set fix de attribute *numerice*
 - fiecare data in parte poate fi considerata un punct in spatiu multidimensional
 - fiecare atribut considerat este o dimensiune
 - este tipul de date standard pentru analiza statistica
 - nota: intre conceptul de dimensiune asa cum e definit in matematica si cel de dimensiune —atribut pot exista diferente

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

* Matrice de date rarefiate: * caz special al datelor matrice * date asimetrice: in putine cazuri anumite trasaturi sunt prezente, predominanta este lipsa trasaturilor * exemplu: tranzactii din cosuri de cumparaturi in care modelarea se face: obiectul (nu) a fost cumparat * exemplu: documente cu continut dintr-un anumit lexic; pentru un document se creeaza un vector numeric, care la fiecare cuvant are precizat daca apare (eventual: de cate ori apare) sau nu = matricea document–termen * in practica, tipuri de date specializate pentru date rare sunt benefice; [Sparse matrices]
(<https://docs.scipy.org/doc/scipy/reference/sparse.html>)

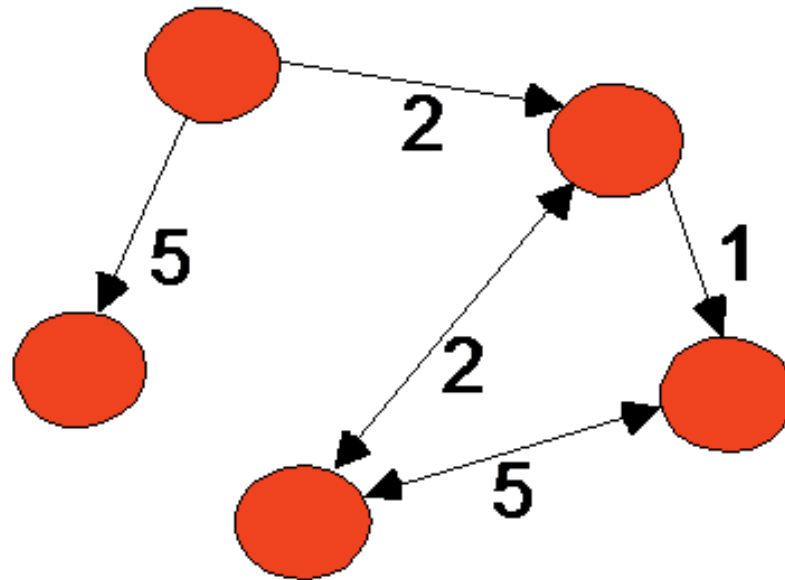
	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

1.1.4.2 Seturi de date de tip graf

- reprezentare convenabila pentru cazurile:
 - graful reprezinta relatii între obiecte
 - obiectele însele sunt reprezentate ca graf

Caz 1: datele reprezinta relatii între obiecte

- obiectele sunt reprezentate ca noduri în graf
- relatiile dintre obiecte sunt reprezentate sub forma de arce sau muchii
- exemplu: pagini web care contin legaturi catre alte pagini
- exemplu de algoritim ce foloseste structura de graf: [algoritmul PageRank](http://en.wikipedia.org/wiki/PageRank) (<http://en.wikipedia.org/wiki/PageRank>)



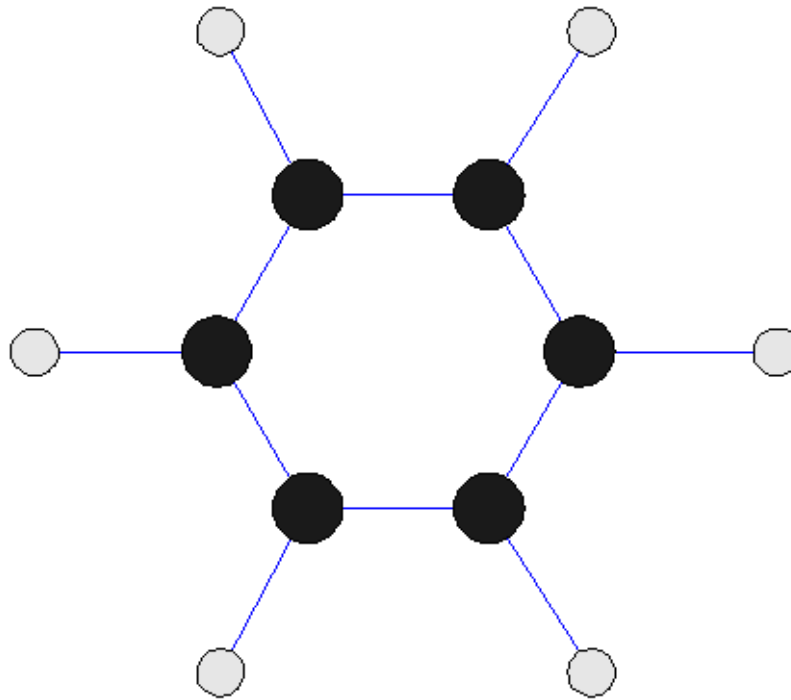
```

<a href="papers/papers.html#bbbb">
Data Mining </a>
<i>
<a href="papers/papers.html#aaaa">
Graph Partitioning </a>
<i>
<a href="papers/papers.html#aaaa">
Parallel Solution of Sparse Linear System of Equations </a>
<i>
<a href="papers/papers.html#ffff">
N-Body Computation and Dense Linear System Solvers

```

Caz 2: obiectele-data sunt grafuri

- obiectele pot contine subobiecte care sunt legate între ele
- uneori nu doar legaturile sunt importante, ci si forma lor: unghiul dintre muchii poate avea aceeaasi relevanta ca si legaturile însele;
- exemplu: formulele chimice - benzen = C_6H_6
- utilitate: se poate detecta care substructura apare mai des; sau daca prezenta sau absenta unor astfel de substructuri este legata de prezenta/absenta anumitor proprietati chimice.
- domeniu aparte: "mineritul" substructurilor

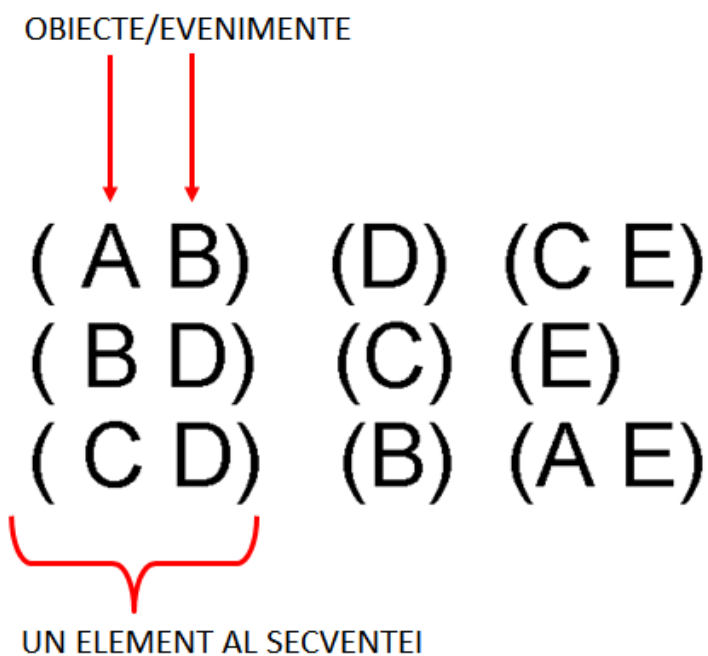


▼ 1.1.4.3 Seturi de tip secventa

- attributele au relatii care implica ordonare în timp sau spatiu
- subtipuri: date secventiale, secventa, serii de timp si date spatiale

Caz: date secventiale

- numite si date temporale
- fiecare înregistrare are un atribut suplimentar de timp asociat



Caz: date secventa

- setul de date entitati individuale, precum secvente de cuvinte sau de litere;
- similare cu cele secventiale, dar fara timp inclus
- pozitia din secventa este importanta
- exemplu: informatia genetica este o secventa de nucleotide (gene)
- aplicatie: predictia similaritatilor în structura si functia genelor pe baza similaritatii dintre secvente

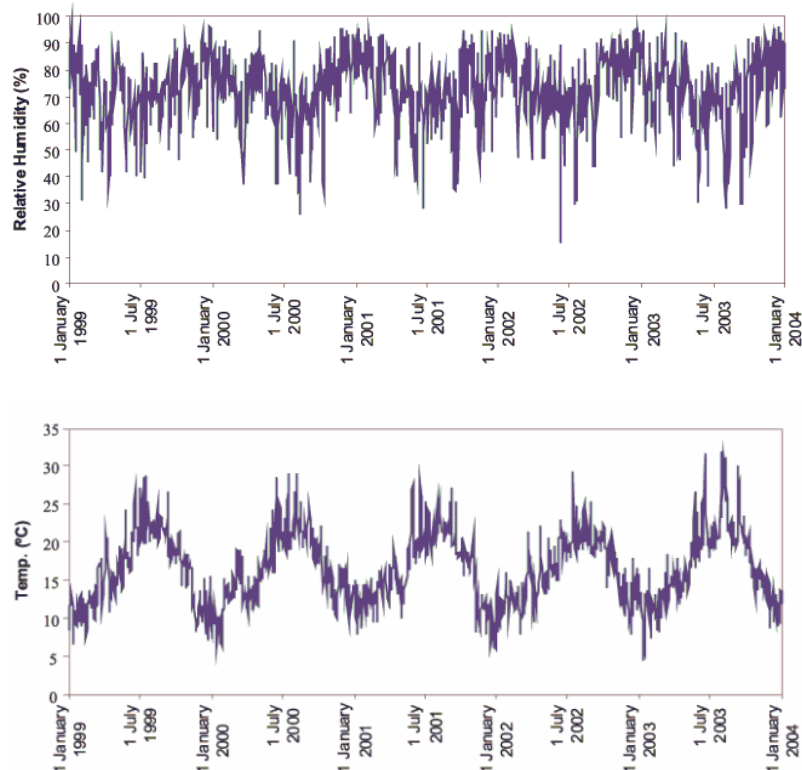
```

GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG

```

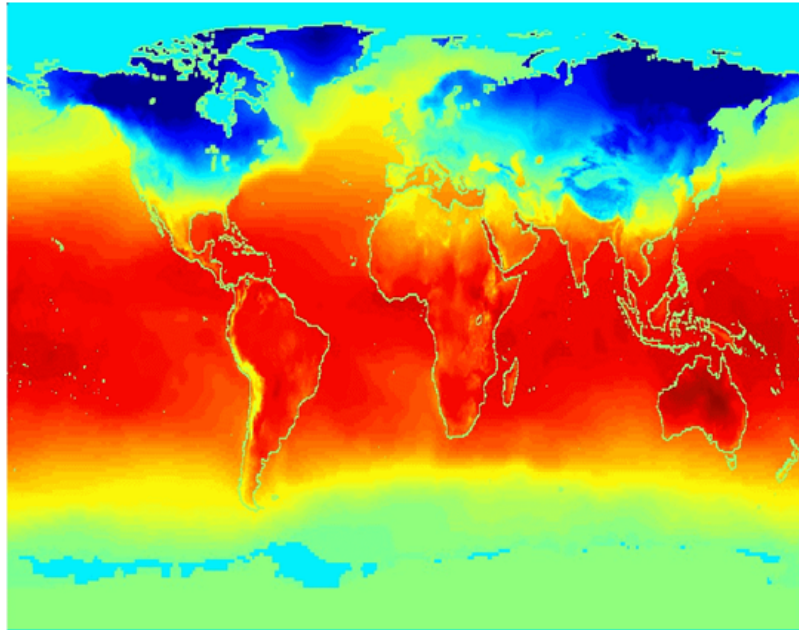
▼ Caz: serii de timp

- fiecare înregistrare e o serie de timp = o serie de masuratori efectuate la anumite momente de timp
- exemplu: seturi de date de tip financiar, reprezentând valorile unor stocuri
- exemplu: date meteo masurate lunar



Caz: Date spatiale

- cazul datelor care au attribute spatiale sau areale
- exemplu: date climatice raportate pe regiuni
- exemplu: date adunate pentru scurgerea unui fluid — pozitia diferitelor puncte este inregistrata



1.2 Calitatea datelor

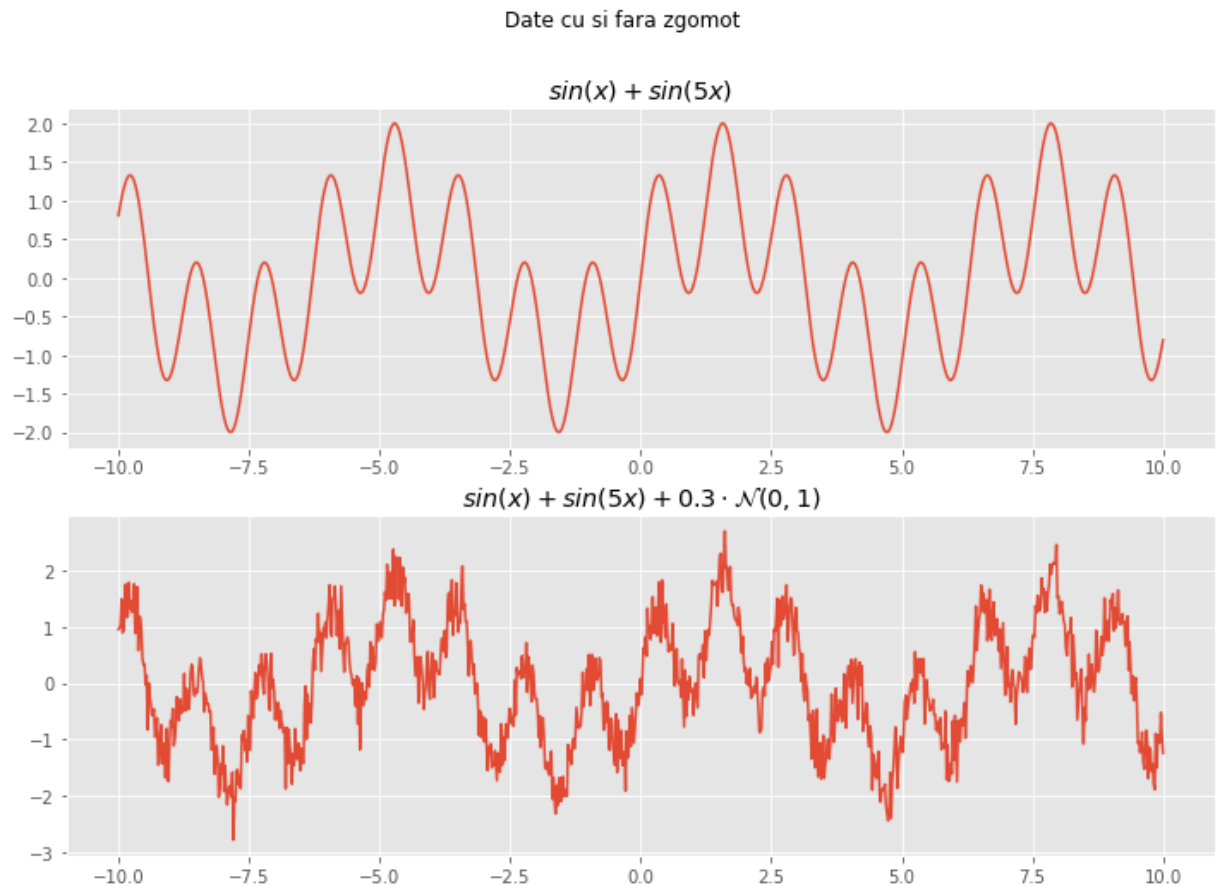
1.2.1 Probleme legate de masurarea si colectarea datelor

- Presupunerea ca datele pe baza carora se face DS sunt de calitate perfecta este naiva
- Prevenirea problemelor care duc la scaderea calitatii datelor nu este o optiune pentru un data scientist
- Abordari:
 - detectarea si corectarea erorilor = curatarea datelor
 - construirea de algoritmi care sa tolereze o calitate slaba a datelor
- surse de probleme în calitatea datelor:
 - procesele de masurare
 - aplicatiile folosite
- Zgomotul
 - componenta aleatoare care se adauga unui proces de masurare
 - Exemplu: distorsiunea vocii unei persoane pe o linie telefonica slaba
 - Daca eroarea apare mereu în acelasi loc: artefact


```
In [2]: x = np.linspace(-10, 10, 1000)
y = np.sin(x) + np.sin(5*x)

fig, (ax1, ax2) = plt.subplots(2, figsize=(12, 8))
fig.suptitle('Date cu si fara zgomot')
ax1.set_title(r'$sin(x) + sin(5x)$')
ax1.plot(x, y)
ax2.set_title(r'$sin(x) + sin(5x) + 0.3 \cdot \mathcal{N}(0, 1)$')
ax2.plot(x, y + 0.3 * np.random.randn(len(x)));
```

executed in 787ms, finished 17:57:12 2021-03-22



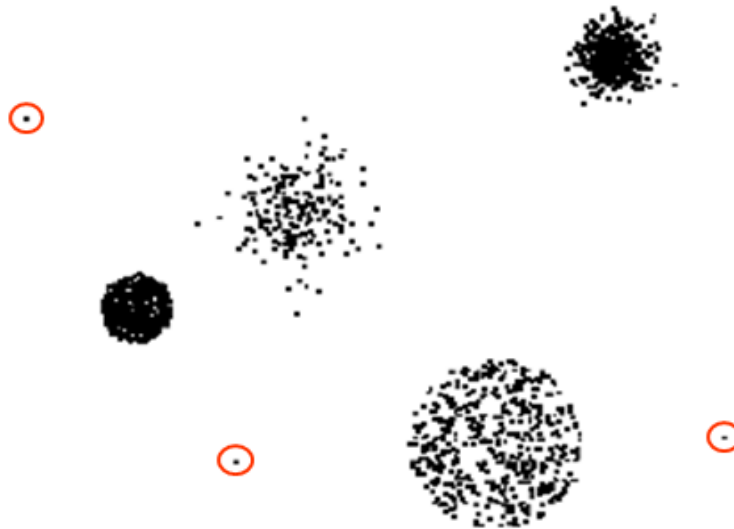
1.2.2 Calitatea datelor: precizie, abatere, acuratete

- **Precizie:** apropierea valorilor rezultate prin masuratori repetate ale aceleiasi cantitati.
- **Abatere (eng: bias):** o variatie sistematica a masuratorilor fata de cantitatea reala.
- Exemplu: se masoara o cantitate de 1 gram. Valorile obtinute sunt: {1.015, 0.990, 1.013, 1.001, 0.986}. Media este 1.001, deci abaterea este $|1.001 - 1| = 0.001$.
- Precizia este considerata abaterea standard:

- $\sigma = \sqrt{E[(X - E(X))^2]}$ deci pentru datele de mai sus precizia este 0.013.
- **Acuratetea:** apropierea masuratorilor fata de valoarea adevarata ce se vrea a fi masurata.

▼ 1.2.3 Calitatea datelor: anomalii

- Anomaliile sunt obiecte cu caracteristici considerabil diferite fata de majoritatea obiectelor din setul de date
- Anomaliile (outliers) nu sunt zgomote, ci obiecte legitime
- Utilitate: detectarea de nise pe piata, detectarea fraudelor



▼ 1.2.4 Calitatea datelor: valori lipsa

- Cazuri: una sau mai multe valori de atribut lipsesc
- Motive pentru lipsa valorilor:
 - informatia nu este colectata — oamenii nu vor sa spuna vârsta sau greutatea
 - attributele nu se pot aplica tot timpul tuturor obiectelor: copiii nu au venituri
- Operarea în aceste situatii:
 - eliminarea obiectelor-data sau a atributelor cu valori lipsa
 - estimarea valorilor lipsa
 - ignorarea valorilor lipsa în timpul analizei

▼ 1.2.5 Calitatea datelor: valori inconsistente; duplicare

- Valori inconsistente:
 - Exemplu: oras si cod postal precizate, dar codul postal corespunde altui oras
 - Exemplu: typos, marimi cu valori improprii (greutate negativa)
 - Operare: detectarea valorilor gresite si corectarea folosind interventie umana
 - E necesara utilizarea surselor de date redundante sau a cunostintelor specifice domeniului

- Date duplicate:
 - Duplicarea poate sa fie exacta sau aproape exacta
 - Exemplu: aceeași persoană cu adrese de email diferite
 - Procesul de curățare = deduplicare

▼ 1.2.6 Calitatea datelor din perspectiva aplicațiilor

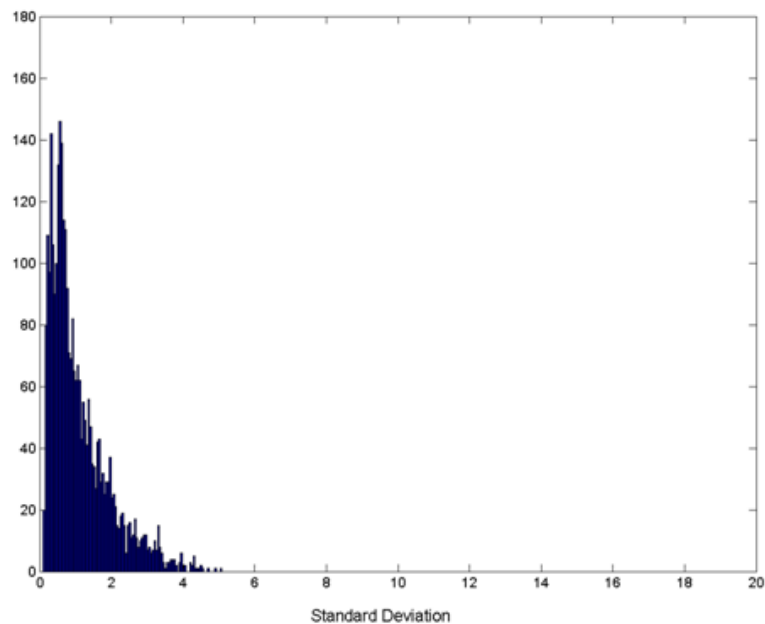
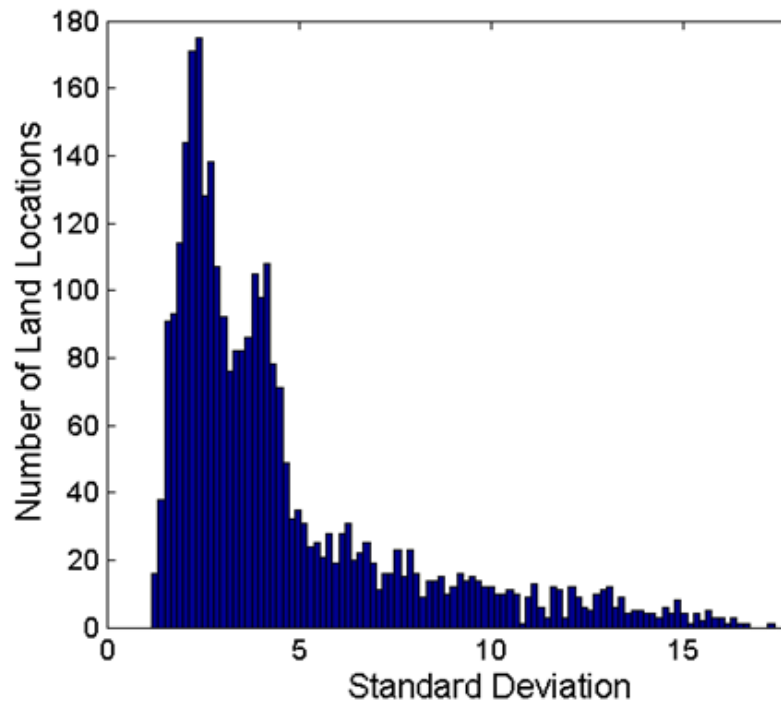
- Din perspectiva aplicațiilor, “datele au calitate bună dacă sunt potrivite pentru utilizarea intenționată”
- Caracterul oportun al datelor — dacă datele sunt perimate, atunci modele și pattern-urile obținute sunt depășite
- Relevanța — în cazul în care se vrea crearea de modele pentru accidente rutiere, omiterea genului și vârstei conducătorilor duce modele cu acuratețe mică; alta situație este data de esanționarea neadecvată
- Cunoștințele apriori despre date — de exemplu, faptul că anumite atribute sunt puternic corelate sau construite unele pe baza altora poate fi utilizată pentru reducerea redundanței și a dimensionalității; cunoașterea preciziei datelor, a caracterului lor oportun sau a scalei de măsură e de cele mai multe ori esențială.

▼ 1.3 Preprocesarea datelor

- Strategii și tehnici complexe, ce pot cere până la 60% din timpul total al procesului de extragere de cunoștințe
- Două variante:
 1. selectarea obiectelor-data și a atributelor
 2. crearea/schimbarea de atribute
- Variante de preprocesare:
 - agregare
 - esanționare
 - reducerea dimensionalității
 - selectarea unui subset de atribute
 - crearea de atribute
 - discretizare și binarizare
 - transformarea variabilelor

▼ 1.3.1 Agregare

- Scop: combinarea a două sau mai multe atribute (sau obiecte) într-un singur atribut (sau obiect)
- Utilitate:
 - reducerea cantității de date
 - schimbarea scalei: orașele sunt agregate în regiuni, state, continente
 - date mai stabile: datele agregate au tendința de a avea variabilitate mai mică



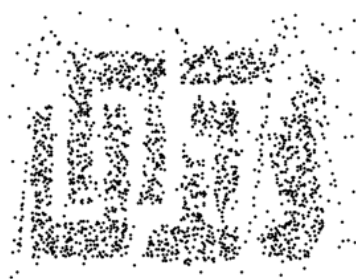
1.3.2 Esantionare

- principala tehnica folosita pentru selectarea datelor
- în statistica, a fost folosita atât pentru investigatii preliminare cât si pentru analiza finala a datelor
- este folosita pentru ca obtinerea întregului set de date este imposibila sau costisitoare
- în DS esantionarea este folosita pentru ca procesarea întregului set de date este consumatoare de timp
- un esantion este reprezentativ daca are aproximativ aceleasi proprietati de interes ca si setul original

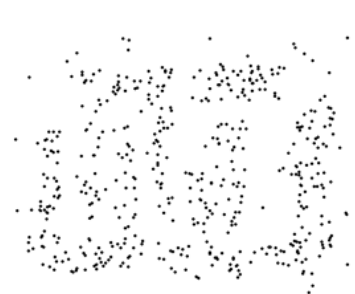
- utilizarea unui esantion reprezentativ e aproape la fel de exacta ca folosirea întregului set de date
- tipuri de esantionare:
 - esantionare aleatoare uniforma: avem o probabilitate egala de alegere a unui obiect anume
 - esantionare fara înlocuire: daca un obiect este selectat, atunci el este scos din populatie
 - esantionare cu înlocuire: obiectele nu sunt scoase din populatie atunci când sunt selectate; un acelasi obiect poate fi selectat mai multe ori
 - esantionarea stratificata: se divid datele în partiti (ex: femei/barbati); din fiecare partiie se extrage apoi esantion, a.i. proportia din straturi sa fie aceeasi cu proportia din multimea initiala.



Esantion de 8000 de puncte

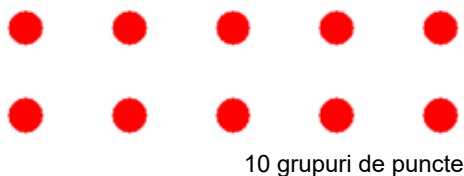


Esantion de 2000 de puncte

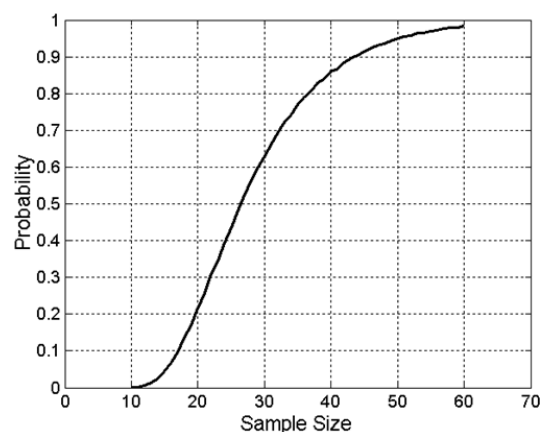


Esantion de 500 de puncte

- Determinarea marimii esantionului
 - problema: avem un set de date care consta dintr-un numar mic de grupuri, în fiecare grup e aproximativ acelasi numar de obiecte
 - trebuie extras un esantion astfel încât din fiecare grup sa fie cel puțin un element selectat
 - "garantarea" se exprima probabilist: care e numarul de date din esantion astfel încât probabilitatea ca sa fie adevarata proprietatea de la punctul anterior sa depaseasca un anumit prag?



10 grupuri de puncte



Probabilitatea de regasire a macar unui punct din fiecare grup, în functie de marimea esantionului

▼ 1.3.3 Reducerea dimensionalitatii

- seturile de date pot avea un numar mare de attribute apare fenomenul de “blestem al dimensionalitatii”: datele sunt rare în spatiu multidimensional
- valorile pentru densitate de probabilitate si distante dintre puncte – critice pentru clustering si detectarea de anomalii — devin nerelevante
- solutie: reducerea dimensionalitatii fara a pierde prea mult din informatie
- beneficii: algoritmi de DS lucreaza mai eficient pe dimensiuni mai putine; modelele rezultate sunt mai simple, inteligibile
- Scop:
 - evitarea blestemului dimensionalitatii
 - reducerea timpului de rulare necesar algoritmilor de DS
 - datele devin mai usor de vizualizat
 - poate ajuta la eliminarea trasaturilor irelevante sau reducerea zgomotului
- Tehnici folosite:
 - analiza componentelor principale (Principal Component Analysis)
 - descompunerea valorilor principale (Singular Value Decomposition)
 - alte metode: supervizate si transformari neliniare

▼ 1.3.4 Selectarea subsetului de attribute (feature selection)

- motivatie: de multe ori, în seturile de date pot exista attribute redundante (e.g. pret si pret cu taxe incluse) sau irelevante (e.g. id-uri)
- unele din aceste attribute pot fi eliminate prin “bun simt” sau prin cunoastere apriori a (datelor) problemei
- varianta ideala de lucru: se încearca toate combinatiile de trasaturi; numarul de încercari ar fi deci: $C_n^1 + C_n^2 + \dots + C_n^n = 2^n - 1$ deci prohibitiv prin brute force
- variante:
 1. metode încorporate — algoritmul de DS însusi decide ce attribute sa se foloseasca: e.g. arborii de decizie
 2. metode de filtrare — attributele sunt selectate înainte de a se aplica algoritmul: e.g. selectarea atributelor pentru care corelatia este minima sau PCA
 3. metode bazate pe încercare — se foloseste un algoritm oarecare de DS si se încearca diferite subseturi de trasaturi – dar nu brute force;
- Pentru selectarea subsetului de trasaturi e nevoie de:
 - o masura pentru evaluarea unui subset de attribute
 - o metoda de cautare pentru generarea unui nou subset de attribute
 - un criteriu de oprire
 - o procedura de validare
- Alternativa pentru selectarea de attribute: ponderarea atributelor
 - attribute importante -> ponderi mari; attribute nerelevante -> ponderi mici
 - ponderile se pot asocia pe baza cunostintelor asupra domeniului sau prin metode automate
 - utilizare de ponderi: daca pentru dimensiunea (atributul) de indice i avem calculata ponderea $p_i \geq 0$, putem de eemplu sa folosim o distanta de tip Euclidian ponderata:

$$d_E(x, y; p) = \sqrt{p_1 \cdot (x_1 - y_1)^2 + p_2 \cdot (x_2 - y_2)^2 + \dots + p_n \cdot (x_n - y_n)^2}$$

Comparati aceasta formula cu distanta Euclidian standard:

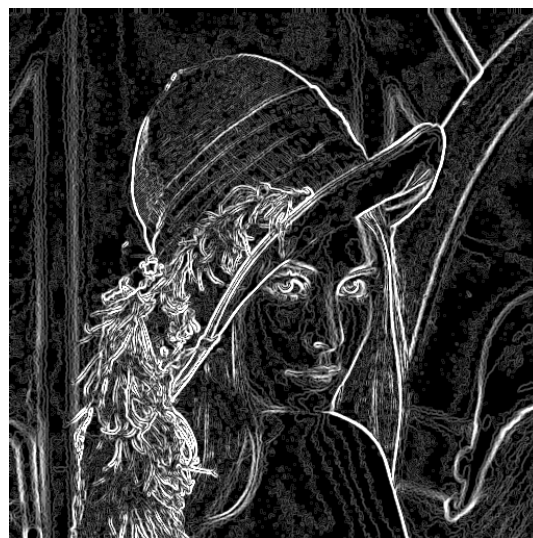
$$d_E(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$$

▼ 1.3.5 Crearea de trasaturi (feature extraction)

- Scop: crearea de noi atribute pe baza celor existente (e.g. raportul dintre venituri si cheltuieli, dintre masa corporala si patratul înaltimii = BMI)
- Noile atribute pot releva mai eficient informatia ascunsa în date
- Numarul atributelor noi obtinute poate fi mai mic decât la plecare
- Metode:
 - Extragere de trasaturi
 - Transformarea datelor, utilizarea unui alt spatiu
 - Construirea de trasaturi
- Exemplu: extragerea de trasaturi
 - crearea unui nou set de atribute pe baza celor existente
 - exemplu: fotografie -> detectarea muchiilor
 - metodele sunt strâns legate de domeniu — algoritmi de procesare grafica, utilizarea unor variabile economice derivate etc.

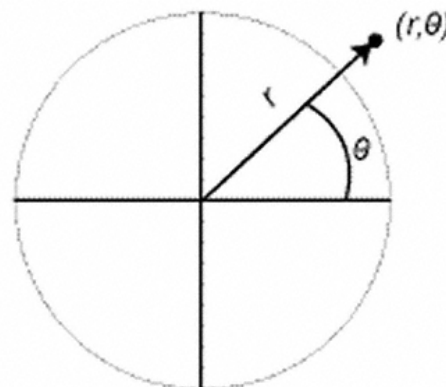
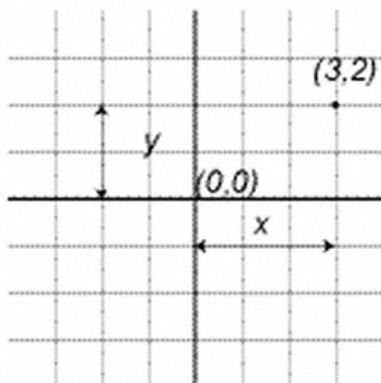


Imagine initiala

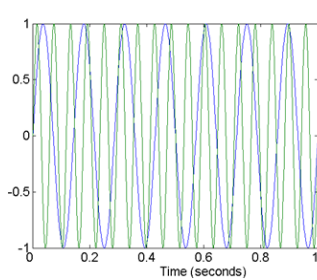


Imagine dupa "edge detection"

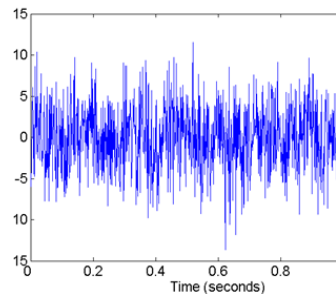
- Exemplu: transformarea datelor
 - trecerea de la coordonatele carteziene la cele polare



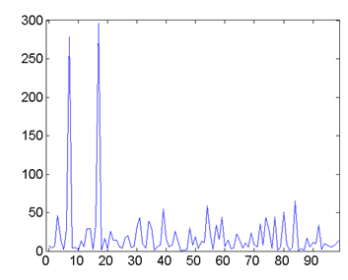
- transformata Fourier



Doua functii sinusoidale de frecvente diferite



Doua functii sinusoidale plus zgomot

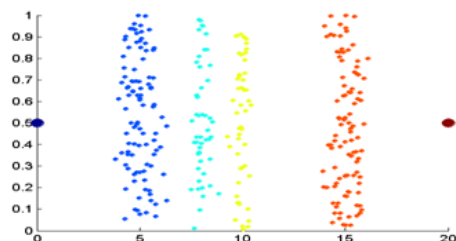


Spectrul de putere, obtinut dupa aplicarea transformatei Fourier"

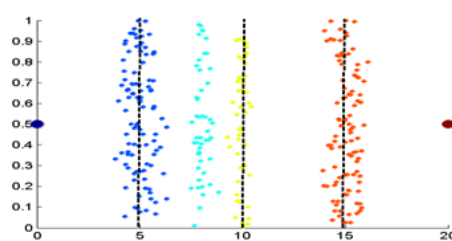
- construirea de noi attribute
 - uneori, attributele originare pot sa aiba informatia necesara dar formatul acestor date este neadecvat pentru algoritmul de DS
 - exemplu: pentru rețele neurale intrarea este numerica, dar datele curente au si valori nominale
 - exemplu de trasatura derivata: densitatea = masa / volumul
 - crearea de noi trasaturi necesita cunoasterea domeniului problemei

1.3.6 Discretizare si binarizare

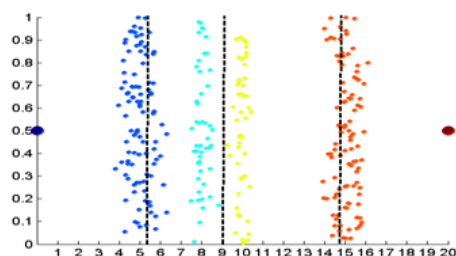
- unii algoritmi cer ca datele sa fie în forma de attribute categoriale
- algoritmi de determinare a asocierilor cer date binare
- apare nevoia de a transforma attribute continue în attribute discrete (discretizare) sau chiar binare (binarizare)
- Discutie pentru binarizare:
 - m valori \rightarrow fiecare valoare initiala se transforma într-un numar întreg din intervalul $[0, m - 1]$ si apoi se foloseste transcrierea în baza 2
 - daca attributele sunt ordinale, atunci trebuie pastrata ordinea prin transformare
 - exemplu: {slab, mediu, bun} $\rightarrow \{(x_1 = 0, x_2 = 0), (x_1 = 0, x_2 = 1), (x_1 = 1, x_2 = 0)\}$
 - $\lceil \log_2 m \rceil$ biti folositi
 - one hot encoding: pentru m clase se folosesc m biti; fiecare bit arata prezenta sau absenta unei trasaturi exemplu: {slab, mediu, bun} $\rightarrow (x_1 = 0, x_2 = 0, x_3 = 1), (x_1 = 0, x_2 = 1, x_3 = 0), (x_1 = 1, x_2 = 0, x_3 = 0)$
- Discretizare a atributelor continue
 - procesul de discretizare este dependent de date si de tipul problemei
 - pasi:
 - se decide numarul de categorii ce se produc;
 - se decide modul de asociere între valorile continue si cele discrete
 - exemplu: se sorteaza datele, se efectueaza $n - 1$ taieturi în interval si pentru fiecare subinterval se asigneaza o eticheta
- Discretizare nesupervizata
 - intervalele pot fi de latime egala sau considerate astfel încât sa
 - prezinte frecvente egale
 - metode de clustering, e.g. K-means
 - inspectare vizuala a datelor



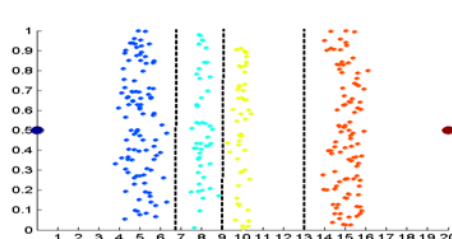
Datele originale



Discretizare cu latime egala



Discretizare cu frecventa egala



Discretizare cu K-means

- Discretizare supervizata
 - utilizarea de informatie suplimentara – etichetele de clasa – poate duce la o discretizare mai buna
 - conceptual: discretizarea sa se faca astfel încât sa se maximizeze “puritatea” intervalelor
 - metode statistice: intervale mai mici se unesc astfel încât sa nu se depaseasca un prag de puritate
- Atribute categoriale cu prea multe atribute
 - exemplu: o multitudine de departamente
 - se pot unifica pe domenii: inginerie, stiinte sociale, biologie
 - altfel: gruparea poate fi facuta în conjunctie cu un algoritm de ML în care se urmareste îmbunatatirea rezultatului de clasificare

▼ 1.3.7 Transformarea de variabile

- transformare care se aplica la toate valorile unei variabile
- exemplu: daca doar amplitudinea unei valori este importanta, atunci se poate considera valoarea absoluta a variabilei
- functii folosite: $\exp(x)$, $\log(x)$, \sqrt{x} , $1/x$, functia putere
- functii complexe: normalizare
- pentru datele care prezinta domeniu mare de valori: scara logaritmica $x \rightarrow \frac{1}{x}$ poate fi mai adecvata
 - exemplu: transferuri de date de 109, 108, 1000, 10 octeti; prin transformare magnitudinea nu mai e importanta, se ajunge la valori pe scala mai ingusta: 9, 8, 3, 1.
- aplicarea trebuie facuta în cunostinta de cauza: transformarea $x \rightarrow \frac{1}{x}$ e o functie de monotonicie descrescatoare;

▼ 1.3.8 Exemplu de transformare: standardizarea

- în statistica: daca \bar{x} este media unui atribut si s_x e abaterea standard a valorilor atributului:

$$s_x = \sqrt{E[(x - \bar{x})^2]}$$

atunci transformarea $x \rightarrow \frac{x - \bar{x}}{s_x}$ creeaza o noua variabila cu media zero si abaterea 1; operatia se numeste standardizare.

- exemplu: persoane cu variabilele: venitul anual si înaltimea; diferentele între înaltime sunt mici comparativ cu diferentele între venituri; diferentele între venituri ar domina distanta între doi oameni, daca s-ar calcula direct pe baza valorilor; standardizarea reduce asemenea diferente de magnitudine
- problema: abaterea patratica medie e influentata prea usor de date extreme, outliers; uneori se considera abaterea absoluta medie

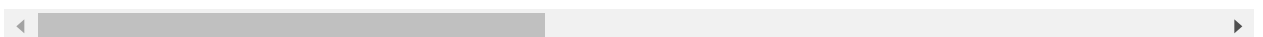
$$\sum_{i=1}^m \frac{|x_i - \mu|}{m}$$

unde μ e media sau mediana

▼ 1.4 Explorarea datelor: statistici de sumarizare pentru date univariate

▼ 1.4.1 Explorarea datelor

- Explorarea datelor reprezintă investigarea preliminară a datelor, cu scopul de a obține o înțelegere a caracteristicilor lor
- Pasul de explorare poate fi de folos în alegerea pașilor de preprocesare sau analiză
- Se poate folosi abilitatea naturală a oamenilor de a recunoaște pattern-uri
- Domeniul a fost introdus de către statisticianul John Tukey: Exploratory Data Analysis, Addison-Wesley
- AED este domeniu opus lui "Confirmatory Data Analysis", care are ca scop testarea ipotezelor statistice, calculul intervalelor de încredere etc.
- Curs de AED: [aici \(https://www.coursera.org/learn/data-analysis-with-python?specialization=ibm-data-science&utm_source=gg&utm_medium=sem&campaignid=2087860785&utm_campaign=10-IBM-Data-Science-ROW&utm_content=10-IBM-Data-Science-ROW&adgroupid=116274867101&device=c&keyword=&matchtype=b&network=g&devicemodel=yWJrboZGvrVzGL9QL0-8PazMsYrrQaAnG4EALw_wcB\)](https://www.coursera.org/learn/data-analysis-with-python?specialization=ibm-data-science&utm_source=gg&utm_medium=sem&campaignid=2087860785&utm_campaign=10-IBM-Data-Science-ROW&utm_content=10-IBM-Data-Science-ROW&adgroupid=116274867101&device=c&keyword=&matchtype=b&network=g&devicemodel=yWJrboZGvrVzGL9QL0-8PazMsYrrQaAnG4EALw_wcB)



- În AED, așa cum este definit de Tukey:
 - Focus-ul este pe vizualizare
 - Gruparea (clustering) și detectarea de anomalii sunt văzute ca tehnici exploratorii
 - Acestea două sunt subdomenii aparte ale DS, dincolo de analiză exploratorie

▼ 1.4.2 Setul de date Iris

- Constă în date măsurate pentru 150 de flori de iris, din 3 specii (Iris Setosa, Iris Versicolour, Iris Virginica, câte 50 de exemplare pe specie)

- Măsurătorile sunt pentru lungimea/lăţimea petalelor/sepalelor în centimetri (4 coloane)
- A cincea coloană este specia florii – atribut nominal
- Datele se pot descărca [de aici \(http://archive.ics.uci.edu/ml/datasets/Iris\)](http://archive.ics.uci.edu/ml/datasets/Iris)

```
In [3]: iris_url = 'http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.'
iris_columns = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
data = pd.read_csv(iris_url, header = None)
data.columns = iris_columns
data.describe(include='all')
```

executed in 885ms, finished 17:57:12 2021-03-22

Out[3]:

	sepal_length	sepal_width	petal_length	petal_width	class
count	150.000000	150.000000	150.000000	150.000000	150
unique	NaN	NaN	NaN	NaN	3
top	NaN	NaN	NaN	NaN	Iris-versicolor
freq	NaN	NaN	NaN	NaN	50
mean	5.843333	3.054000	3.758667	1.198667	NaN
std	0.828066	0.433594	1.764420	0.763161	NaN
min	4.300000	2.000000	1.000000	0.100000	NaN
25%	5.100000	2.800000	1.600000	0.300000	NaN
50%	5.800000	3.000000	4.350000	1.300000	NaN
75%	6.400000	3.300000	5.100000	1.800000	NaN
max	7.900000	4.400000	6.900000	2.500000	NaN

iris setosa



petal

sepal

iris versicolor



petal

sepal

iris virginica



petal

sepal

Sursa: morioh.com

1.4.3 Statistici de sumarizare

- Statisticile de sumarizare sunt numere care schițează caracteristicile unui set de valori

- Reprezintă manifestarea cea mai vizibilă a statisticii
- Exemple: frecvența, media, dispersia

▼ 1.4.3.1 Frecvența și valoarea modală

Pentru un set de m date categoricale cu valorile $\{v_1, \dots, v_i, \dots, v_k\}$ frecvența relativă a unei valori v_i este:

$$frecventa(v_i) = \frac{\text{numarul de obiecte cu valoarea } v_i}{m}$$

iar frecvența absolută este numărătorul fracției de mai sus.

```
In [4]: from collections import Counter
freq_iris = Counter(data['class'])
rel_freq_iris = {iris_type: freq / len(data) for iris_type, freq in freq_iris.items()}

executed in 6ms, finished 17:57:12 2021-03-22
```

```
In [5]: print(f'Frequency: {freq_iris}')
print(f'Relative Frequency: {rel_freq_iris}')

executed in 10ms, finished 17:57:12 2021-03-22

Frequency: Counter({'Iris-setosa': 50, 'Iris-versicolor': 50, 'Iris-virginica': 50})
Relative Frequency: {'Iris-setosa': 0.3333333333333333, 'Iris-versicolor': 0.3333333333333333, 'Iris-virginica': 0.3333333333333333}
```

- Valoarea modală (sau moda) este valoarea cu cea mai mare frecvență:

$$moda = \arg \max_{v_i} frecventa(v_i)$$

- Atenție la situația când o anumită valoare este folosită pentru a semnifica lipsa datelor: null-ul poate apărea ca modă
- Pot exista seturi de date pentru care frecvența maximă să fie atinsă pentru mai multe valori = seturi multimodale
- Pentru valori continue, conceptele de modă/frecvență nu sunt utile, cu excepția cazului când se aplică un pas de discretizare

▼ 1.4.3.2 Percentile

- Pentru cazul valorilor ordonate se pot considera percentilele
- Pentru un atribut continuu sau ordinal x și un număr p întreg între 0 și 100, a p -a percentilă x_p este o valoare din șirul de valori ale lui x astfel încât p din aceste valori sunt mai mici decât x_p
- Nu există o definiție standardizată pentru percentile, cea de mai sus este luată pentru fixare
- Pentru cazul în care se calculează percentile pentru set mare de date, diferențele datorate diferitelor moduri de definire devin neesențiale
- Tradițional se consideră $x_{0\%} = \min(x)$ iar din definiție se poate arăta că $x_{100\%} = \max(x)$
- Mod de calcul pentru determinarea celei de a p -a percentile: pentru un set de m date se calculează valoarea întreagă k cea mai apropiată de $\frac{m}{100} \cdot p + \frac{1}{2}$ și se ia valoarea

corespunzătoare acestui rang k în şirul x sortat

- Funcție utilizabila: [numpy.percentile](https://numpy.org/doc/stable/reference/generated/numpy.percentile.html)
(<https://numpy.org/doc/stable/reference/generated/numpy.percentile.html>) sau funcția pandas [pandas.DataFrame.quantile](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.quantile.html)
(<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.quantile.html>)

In [6]: `# help(np.percentile)`

executed in 5ms, finished 17:57:12 2021-03-22

In [7]: `_25th_percentile_sepal_length = np.percentile(data.sepal_length, 25)`

`print(f'Cate valori sunt strict mai mici decat a 25-a percentila: {np.sum(data.sepal_length < _25th_percentile_sepal_length)}')`
`print(f'Cate valori sunt strict mai mici sau egale decat a 25-a percentila: {np.sum(data.sepal_length <= _25th_percentile_sepal_length)}')`
`print(f'lista valorilor sortate: {list(data.sepal_length.sort_values())}')`

`# se poate intelege pentru aceste date de ce definitia exacta a percentilelor n`

executed in 10ms, finished 17:57:12 2021-03-22

Cate valori sunt strict mai mici decat a 25-a percentila: 32
Cate valori sunt strict mai mici sau egale decat a 25-a percentila: 41
lista valorilor sortate: [4.3, 4.4, 4.4, 4.4, 4.5, 4.6, 4.6, 4.6, 4.6, 4.7, 4.7, 4.8, 4.8, 4.8, 4.8, 4.8, 4.9, 4.9, 4.9, 4.9, 4.9, 4.9, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.1, 5.1, 5.1, 5.1, 5.1, 5.1, 5.1, 5.1, 5.1, 5.2, 5.2, 5.2, 5.2, 5.3, 5.4, 5.4, 5.4, 5.4, 5.4, 5.4, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.7, 5.7, 5.7, 5.7, 5.7, 5.7, 5.7, 5.7, 5.8, 5.8, 5.8, 5.8, 5.8, 5.8, 5.9, 5.9, 5.9, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.1, 6.1, 6.1, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.5, 6.5, 6.5, 6.5, 6.5, 6.6, 6.6, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.8, 6.8, 6.8, 6.9, 6.9, 6.9, 6.9, 7.0, 7.1, 7.2, 7.2, 7.2, 7.3, 7.4, 7.6, 7.7, 7.7, 7.7, 7.7, 7.9]

In [8]: `data['sepal_length'].quantile(q=0.25)`

executed in 8ms, finished 17:57:12 2021-03-22

Out[8]: 5.1

1.4.3.3 Media si mediana

- Pentru un set de valori x_1, x_2, \dots, x_m valoarea medie este:

$$\bar{x} = E(x) = \text{mean}(x) = \frac{1}{m} \sum_{i=1}^m x_i$$

- Mediana este a 50-a percentila. O varianta bruta este sortarea valorilor in ordine crescatoare, obtinandu-se permutarea (reordonarea) $\{x_{(1)}, x_{(2)}, \dots, x_{(m)}\}$; mediana este:

$$\text{mediana}(x) = \begin{cases} x_{r+1} & \text{daca } m = 2r + 1 \\ \frac{x_{(r)} + x_{(r+1)}}{2} & \text{daca } m = 2r \end{cases}$$

O varianta mai rafinata (si care se poate aplica si la calculul de percentile) este utilizarea de statistici de ordine (complexitate liniara in loc de $O(n \log n)$).

Obtinere: [numpy.median \(https://numpy.org/doc/stable/reference/generated/numpy.median.html\)](https://numpy.org/doc/stable/reference/generated/numpy.median.html)
sau [pandas.DataFrame.median \(https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.median.html\)](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.median.html).

In [9]: `np.median(data.sepal_length.values)`

executed in 9ms, finished 17:57:12 2021-03-22

Out[9]: 5.8

In [10]: `data.median()`

executed in 8ms, finished 17:57:12 2021-03-22

Out[10]:

sepal_length	5.80
sepal_width	3.00
petal_length	4.35
petal_width	1.30
dtype:	float64

Diferente între medie și mediană:

- Media este valoare "de mijloc" doar dacă distribuția datelor este simetrică: arată la fel în stânga și în dreapta unei valori - care e chiar "mijlocul"
- Dacă distribuția este asimetrică, atunci mediana este un indicator mai bun pentru valoare de mijloc
- Media este influențată de outliers, în timp ce mediana - nu
- Medie retezată (eng: trimmed mean) se utilizează pentru a exclude anomaliile: se fixează un procent p între 0 și 100; se elimină primele și ultimele $(p/2)\%$ din date; se calculează media pentru ceea ce rămâne
- media standard se obține din media retezată cu $p = 0$

Exemple:

- Considerăm valorile {1, 2, 3, 4, 5, 90}. Media este 17.5, mediana este 3.5. Valoarea de trimmed mean pentru $p = 40\%$ este 3.5, considerabil diferită față de media setului întreg de date
- Media, medianele și valoarea de trimmed mean pentru iris sunt:

Măsura	Lungimea sepalelor	Lungimea sepalelor	Lungimea petalelor	Lungimea petalelor
Media	5.84	3.05	3.76	1.20
Mediana	5.80	3.00	4.35	1.30
Trimmed mean (20%)	5.79	3.02	3.72	1.12

- Exercițiu: dacă valoarea medianei este mai mică decât media, ce puteți spune despre date?

▼ 1.4.3.4 Măsurări ale împrăstierii datelor

- Sunt măsuri care cuantifică concentrarea datelor

- Diametrul domeniului de valori (eng: range) al unui set de date $\{x_1, x_2, \dots, x_m\}$ corespunzător atributului x este

$$range(x) = \max(x) - \min(x) = x_{(m)} - x_{(1)}$$

- Range-ul este nerelevant, deoarece putem avea că majoritatea datelor sunt concentrate într-o zonă îngustă, dar câteva valori outlier măresc artificial diametrul setului
- Varianța (dispersia; eng: variance) unui set de date de m valori este:

$$variana(x) = s_x^2 = \frac{1}{m-1} \sum_{i=1}^m (x_i - \bar{x})^2$$

- Utilizarea numitorului $m - 1$ în loc de m este numită [corecția Bessel](http://en.wikipedia.org/wiki/Bessel%27s_correction) (http://en.wikipedia.org/wiki/Bessel%27s_correction) și are ca scop corectarea abaterii din estimarea varianței de populație
- Abaterea standard este $s_x = \sqrt{s_x^2}$ și are aceeași unitate de măsură ca și atributul x
- Python: [numpy.std](https://numpy.org/doc/stable/reference/generated/numpy.std.html) (<https://numpy.org/doc/stable/reference/generated/numpy.std.html>)

In [11]: `np.std(data.petal_length), np.std(data.sepal_length)`

executed in 7ms, finished 17:57:13 2021-03-22

Out[11]: (1.7585291834055201, 0.8253012917851409)

- Deoarece media poate să fie distorsionată de outliers, rezultă că dispersia poate fi și ea influențată
- Se preferă considerarea altor trei măsuri:
 - absolute average deviation, AAD:

$$AAD(x) = \frac{1}{m} \sum_{i=1}^m |x_i - \bar{x}|$$

- median absolute deviation, MAD:

$$MAD(x) = median(\{|x_1 - \bar{x}|, \dots, |x_m - \bar{x}|\})$$

- interquartile range:

$$IQR(x) = x_{75\%} - x_{25\%}$$

Verificam ca prin standardizare de date, media devine 0 si deviatia standard 1:

```
In [12]: med_petal_length = np.mean(data.petal_length)
# centram datele: media lor devine 0
centered_petal_length = data.petal_length - med_petal_length
# assert 0 == np.mean(centered_petal_length) # fails, FP small values ~ 0
assert np.allclose(0, np.mean(centered_petal_length))
std_petal_length = np.std(data.petal_length)
std_centered_petal_length = np.std(centered_petal_length)
assert np.allclose(std_petal_length, std_centered_petal_length)
standardized_petal_length = centered_petal_length/std_centered_petal_length
# verificare: media 0...
assert np.allclose(0, np.mean(standardized_petal_length))
# si standard deviation ~ 1
assert np.allclose(1, np.std(standardized_petal_length))
```

executed in 11ms, finished 17:57:13 2021-03-22

1.4.4 Statistici de sumarizare a datelor multivariate

- Date multivariate: date cu mai multe atribute
- Pentru atributul x_i calculăm media \bar{x}_i
- Media setului de obiecte este $\mathbf{x} = (\bar{x}_1, \dots, \bar{x}_n)$
- Analog se poate calcula dispersia, mediana etc. pe fiecare dimensiune
- Matricea de covarianță: elementul s_{ij} este covarianța de atributelor x_i și x_j :

$$s_{ij} = \text{covarianța}(x_i, x_j) = \frac{1}{m-1} \sum_{k=1}^m (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)$$

unde x_{pq} este a p -a valoare a atributului x_q

- s_{ij} este măsură a gradului în care două atribute variază împreună (mai precis: care este gradul lor de dependență liniară) și depinde de magnitudinea valorilor atributelor
- $s_{ij} = 0$ arata ca atributele s_i și s_j nu sunt liniar dependente
- numpy: [numpy.cov](https://numpy.org/doc/stable/reference/generated/numpy.cov.html) (<https://numpy.org/doc/stable/reference/generated/numpy.cov.html>)

In [13]: `# matricea de date initiala are pe linii inregistrarile de masuratori. Pentru a
np.cov(data.iloc[:, :-1].values.T)`

executed in 19ms, finished 17:57:13 2021-03-22

Out[13]: `array([[0.68569351, -0.03926846, 1.27368233, 0.5169038],
 [-0.03926846, 0.18800403, -0.32171275, -0.11798121],
 [1.27368233, -0.32171275, 3.11317942, 1.29638747],
 [0.5169038 , -0.11798121, 1.29638747, 0.58241432]])`

In [14]: `df_covariance = pd.DataFrame(np.cov(data.iloc[:, :-1].values.T), index=iris_col
df_covariance.columns = iris_columns[:-1]
df_covariance`

executed in 17ms, finished 17:57:13 2021-03-22

Out[14]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	0.685694	-0.039268	1.273682	0.516904
sepal_width	-0.039268	0.188004	-0.321713	-0.117981
petal_length	1.273682	-0.321713	3.113179	1.296387
petal_width	0.516904	-0.117981	1.296387	0.582414

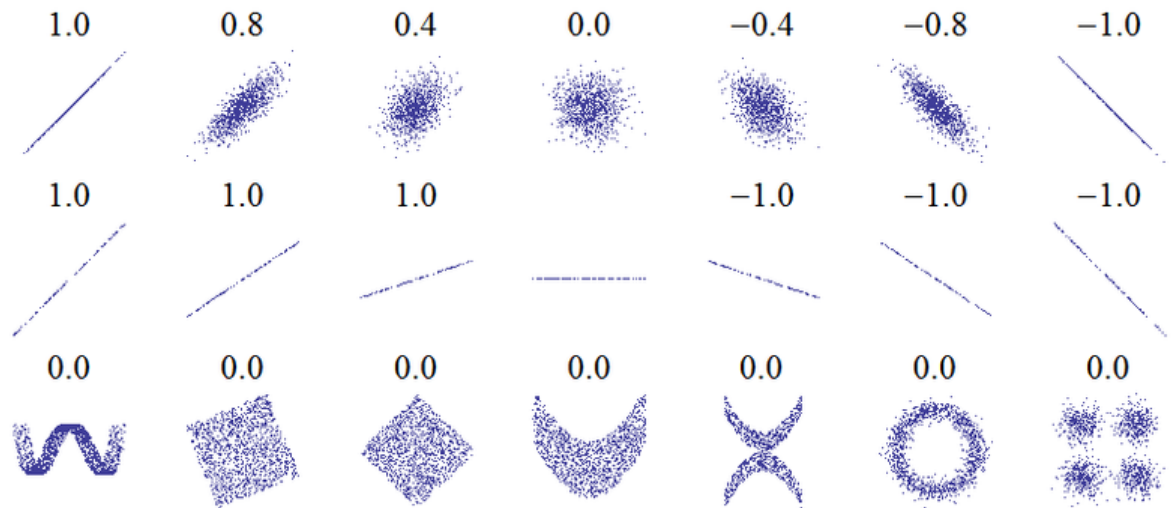
- pentru a elimina dependenta de magnitudinea datelor, definim matricea de corelatie:

$$r_{ij} = \text{corelatia}(x_i, x_j) = \frac{\text{covarianța}(x_i, x_j)}{s_i s_j} \in [-1, 1]$$

- r_{ij} se mai numeste corelatia Pearson a atributelor x_i și x_j
- $r_{ij} = \pm 1$ indica faptul ca x_i e in relatie liniara cu x_j :

$$x_{ki} = a \cdot x_{kj} + b$$

cu $\text{sgn}(a) = \text{sgn}(r_{ij})$



Seturi de date (x, y) împreună cu coeficientul de corelație. Coeficientul de corelație surprinde gradul în care un nor de puncte poate fi aproximat printr-o dreaptă (sus) precum și modul în care ele sunt legate liniar (creștere simultană sau evoluții în sensuri diferite), dar nu și panta acestei legături (figurile din mijloc) sau relații mai complexe între date (rândul de jos). Sursa: Wikipedia. *

Funcție NumPy: `[numpy.corrcoef]`

(<https://numpy.org/doc/stable/reference/generated/numpy.corrcoef.html>)

```
In [15]: np.corrcoef(data.iloc[:, :-1].values.T)
```

executed in 10ms, finished 17:57:13 2021-03-22

```
Out[15]: array([[ 1.          , -0.10936925,  0.87175416,  0.81795363],
                [-0.10936925,  1.          , -0.4205161 , -0.35654409],
                [ 0.87175416, -0.4205161 ,  1.          ,  0.9627571 ],
                [ 0.81795363, -0.35654409,  0.9627571 ,  1.          ]])
```

```
In [16]: df_correlation = pd.DataFrame(np.corrcoef(data.iloc[:, :-1].values.T), index=iris_columns,
df_correlation.columns = iris_columns[:-1]
df_correlation
```

executed in 14ms, finished 17:57:13 2021-03-22

Out[16]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.109369	0.871754	0.817954
sepal_width	-0.109369	1.000000	-0.420516	-0.356544
petal_length	0.871754	-0.420516	1.000000	0.962757
petal_width	0.817954	-0.356544	0.962757	1.000000

Putem folosi functii din Pandas DataFrame: [pandas.DataFrame.corr](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html)

(<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html>)

In [17]: `data.corr()`

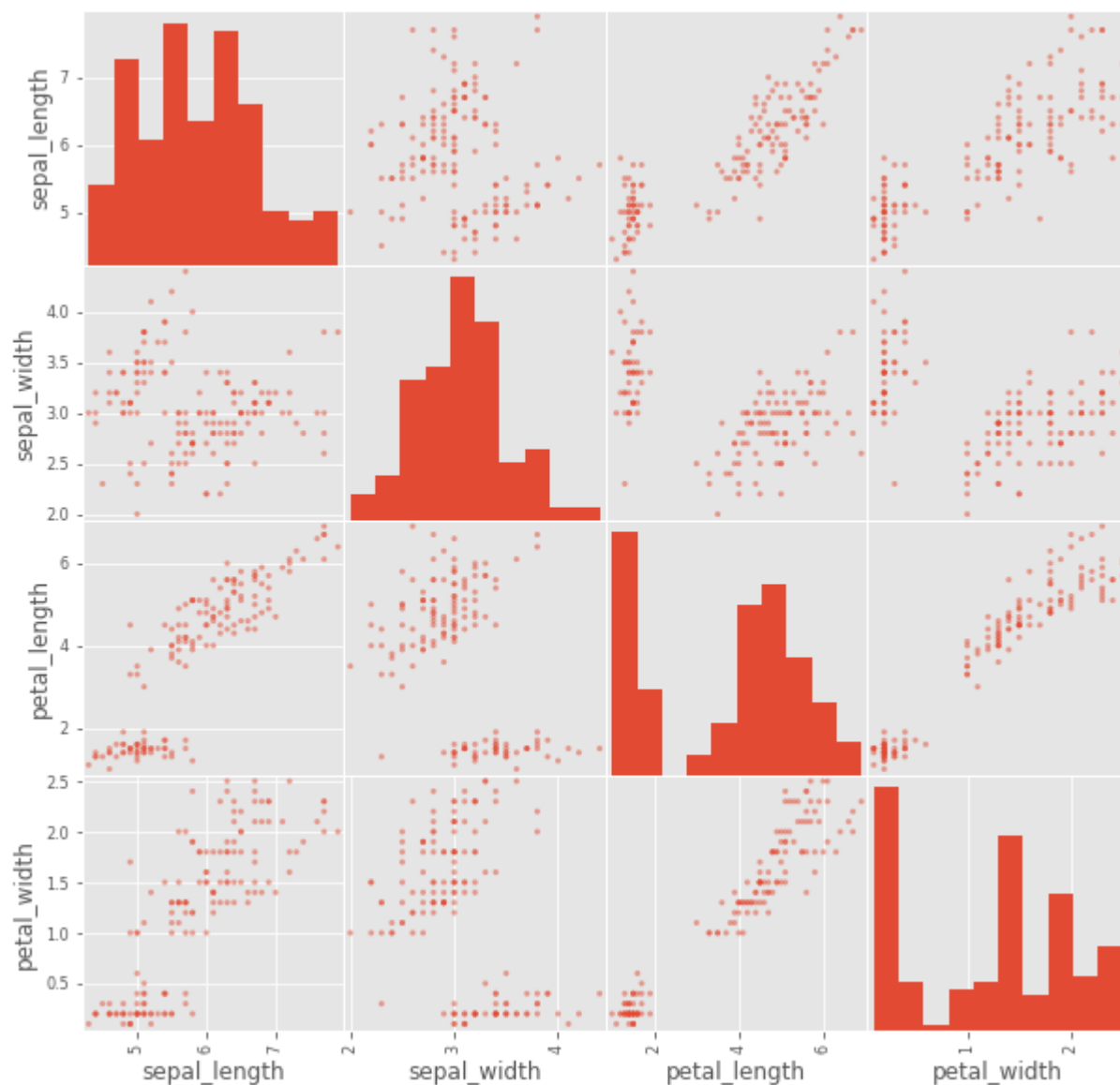
executed in 14ms, finished 17:57:13 2021-03-22

Out[17]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.109369	0.871754	0.817954
sepal_width	-0.109369	1.000000	-0.420516	-0.356544
petal_length	0.871754	-0.420516	1.000000	0.962757
petal_width	0.817954	-0.356544	0.962757	1.000000

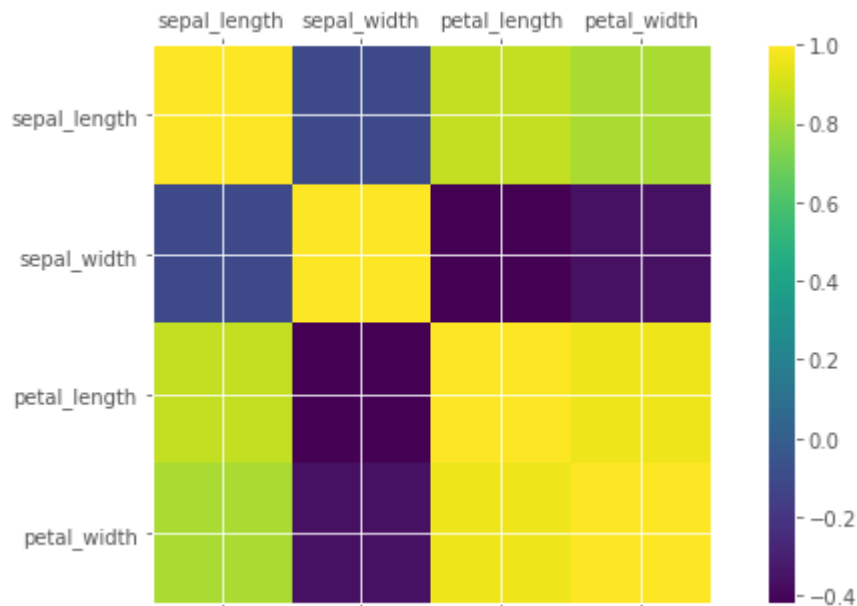
In [18]: `pd.plotting.scatter_matrix(data, figsize=(10, 10))`
`plt.show()`

executed in 1.23s, finished 17:57:14 2021-03-22



```
In [19]: plt.figure(figsize=(10,5))
plt.matshow(data.corr(), fignum=1)
plt.xticks(range(4), data.columns[:-1])
plt.yticks(range(4), data.columns[:-1])
plt.colorbar()
plt.show()
```

executed in 292ms, finished 17:57:14 2021-03-22



Alt exemplu: generam un set de date aleator + o modificare liniara a lui, cu zgomot aleator:

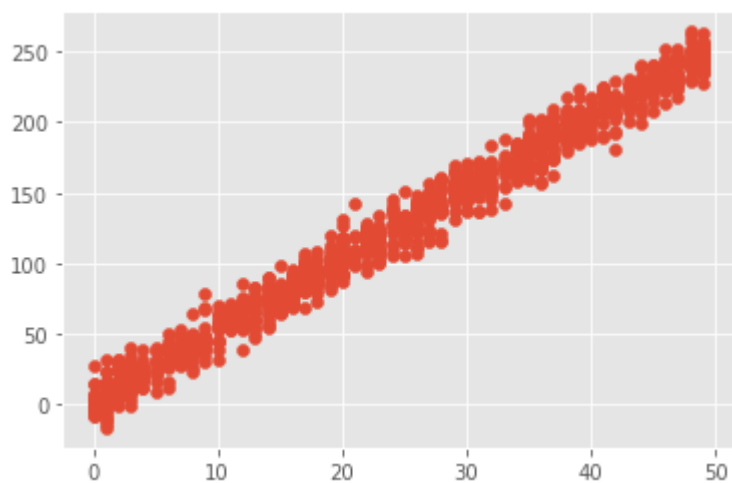
```
In [20]: np.random.seed(100)
x = np.random.randint(0, 50, 1000)
# corelatie pozitiva intre x si y
y = 5*x + 3 + np.random.normal(0, 10, 1000)
np.corrcoef(x, y)
```

executed in 17ms, finished 17:57:14 2021-03-22

```
Out[20]: array([[1.          , 0.99053295],
                [0.99053295, 1.          ]])
```

```
In [21]: plt.scatter(x, y)
plt.show()
```

executed in 221ms, finished 17:57:14 2021-03-22



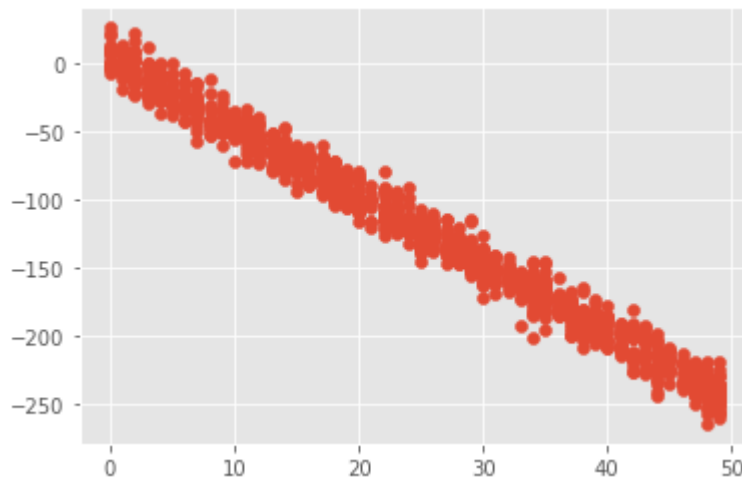
```
In [22]: # corelatie negativa intre x si y
y = -5*x + 3 + np.random.normal(0, 10, 1000)
np.corrcoef(x, y)
```

executed in 11ms, finished 17:57:14 2021-03-22

```
Out[22]: array([[ 1.          , -0.99088942],
                [-0.99088942,  1.          ]])
```

```
In [23]: plt.scatter(x, y)
plt.show()
```

executed in 201ms, finished 17:57:15 2021-03-22



```
In [24]: # corelatie liniara slaba:
x = np.random.randint(0, 50, 1000)
y = np.random.randint(0, 50, 1000)

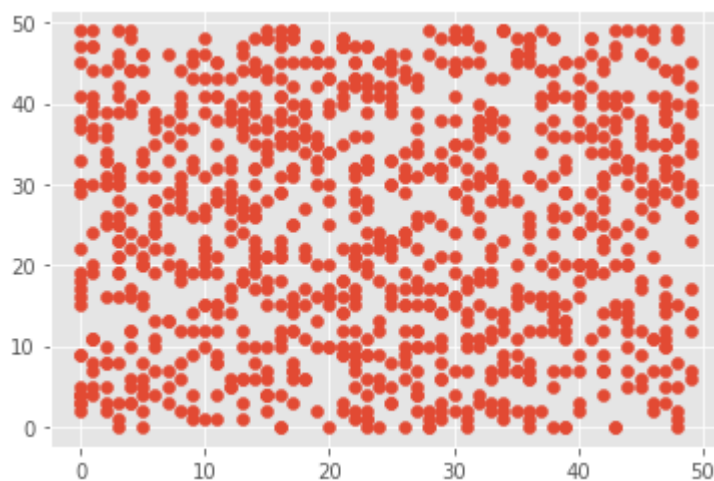
np.corrcoef(x, y)
```

executed in 8ms, finished 17:57:15 2021-03-22

```
Out[24]: array([[1.          , 0.02742447],
               [0.02742447, 1.          ]])
```

```
In [25]: plt.scatter(x, y)
plt.show()
```

executed in 206ms, finished 17:57:15 2021-03-22

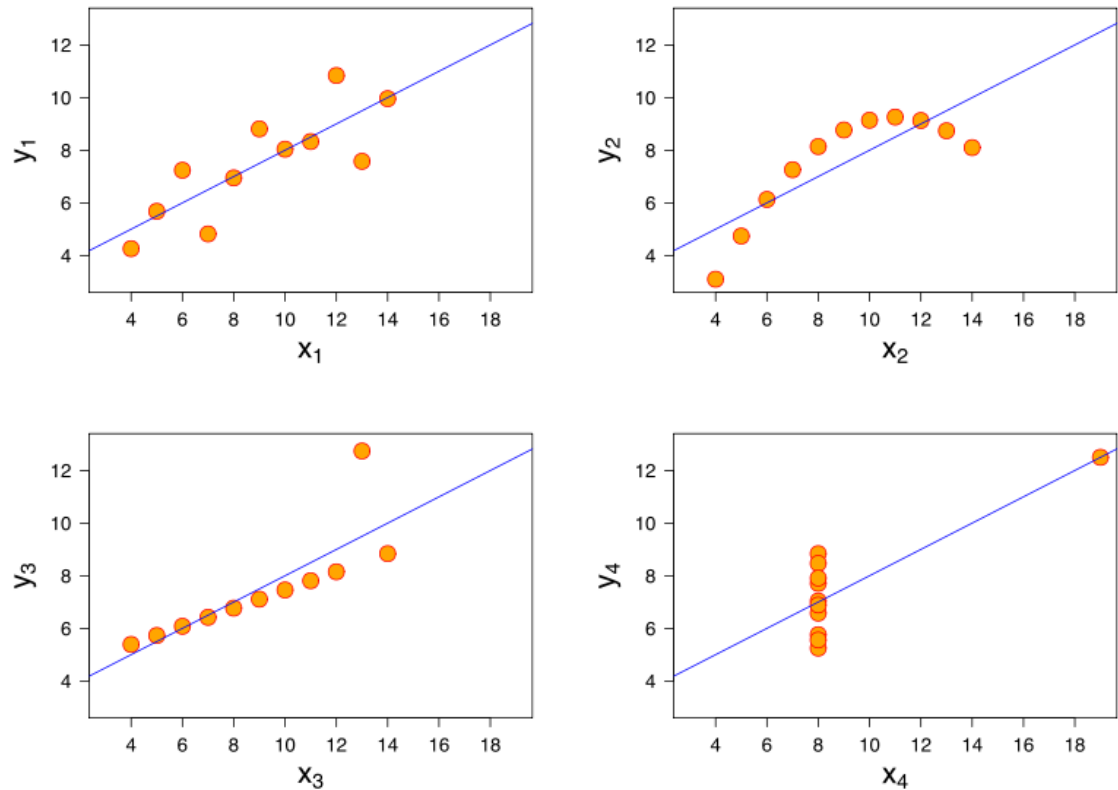


Legat de coeficientul de corelație, câteva observații:

- “Corelația nu înseamnă cauzalitate” – nu se poate folosi o valoare absolută apropiată de 1 ca argument că între două atribute există o relație de cauzalitate. Corelație mare poate fi o condiție necesară pentru legătură de cauzalitate, dar nu asigură și suficiența. Cu toate

acestea, corelația mare poate fi folosită ca punct de pornire în cercetarea unei legături între diferite fenomene.

- Corelația și liniaritatea – coeficientul Pearson reprezintă puterea unei relații liniare între două seturi de valori, dar nu caracterizează complet relația dintre date.
- Exemplu: 4 seturi de date cu două atribute; în toate situațiile media și dispersia lui y este aceeași, de asemenea avem același coeficient de corelație în fiecare caz (0.816); cu toate acestea, legătura dintre x și y e extrem de diferită de la un caz la altul:



Date cu caracteristici numerice identice (medie, dispersie, corelație), dar esențial diferite ca natură: cvartetul lui Anscombe. Sursa: Wikipedia