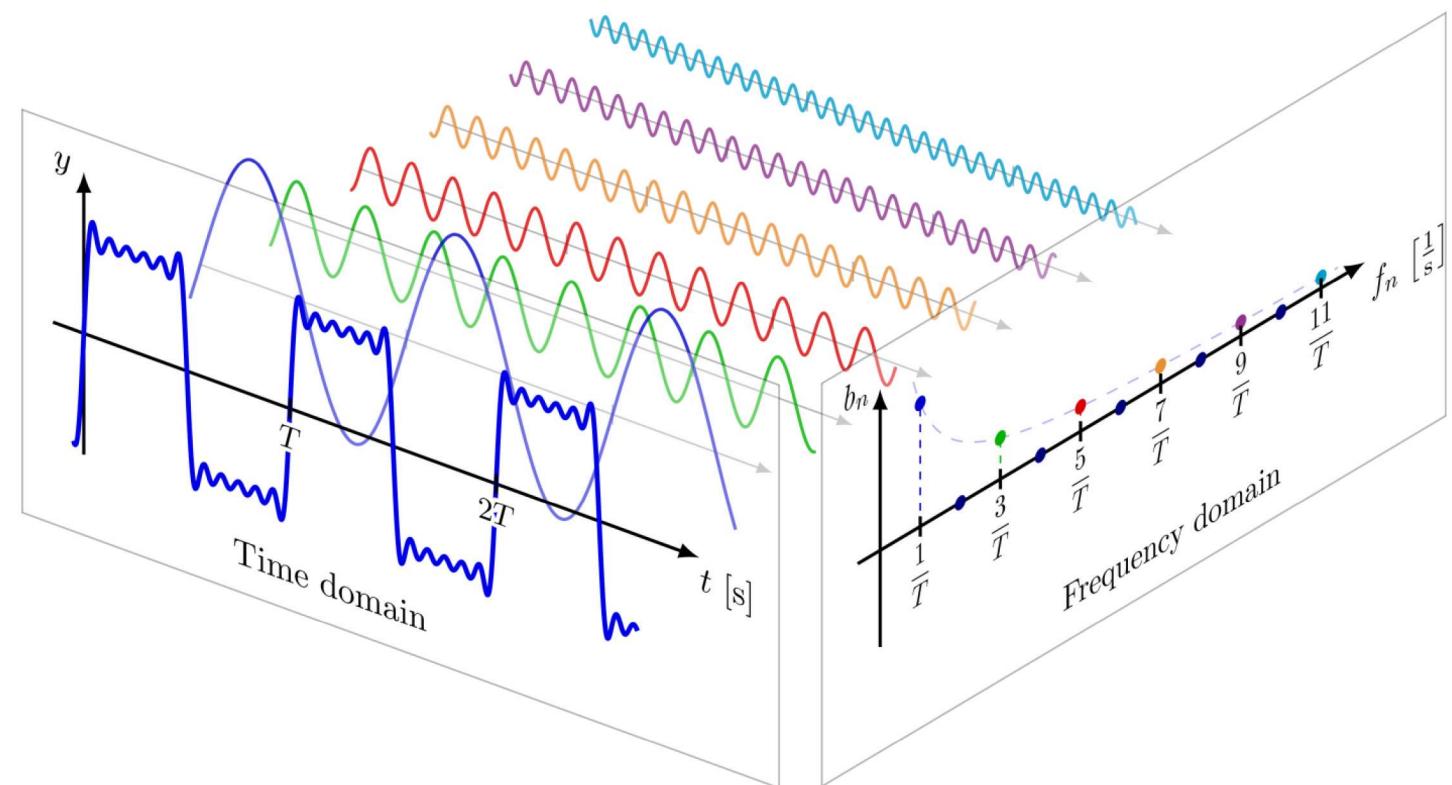


CURS 7 PD

Analiza statistică a semnalelor



Introducere

Histograma

Indicatori statistici

TCLab

1. Introducere

2. Histograma

3. Indicatori statistici

4. TCLab

Introducere

TCLab

Histograma

Indicatori statistici

- **Analiza statistică** a unui semnal este o metodă de studiu ce utilizează instrumente statistice pentru a extrage informații relevante și pentru a caracteriza un semnal.
- Semnalul, o secvență de valori numerice ce variază în timp sau spațiu, este analizat din punct de vedere al distribuției valorilor.
- Analiza statistică a unui semnal se face prin calcularea unor indicatori statistici ai semnalului, indicatori ce pot fi clasificați în una din următoarele categorii:
 1. indicatori ai tendinței centrale – descriu tendința medie a valorilor semnalului;
 2. indicatori ai împrăștierii – măsoară gradul de diversificare al valorilor;
 3. indicatori ai formei distribuției – descriu forma distribuției valorilor.

1. Introducere

2. Histograma

3. Indicatori statistici

4. TCLab

Introducere

TCLab

Histograma

Indicatori statistici

1. Introducere

2. Histograma

3. Indicatori statistici

4. TCLab

Introducere

Indicatori statistici

Histograma

TCLab

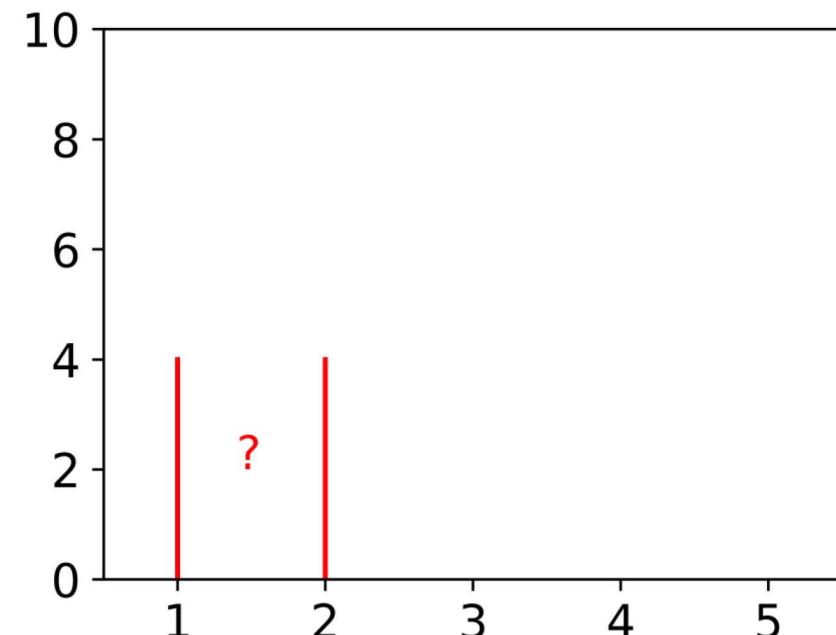
Definiție

Histograma este un grafic ce oferă o interpretare vizuala a semnalelor numerice prin indicarea numărului de eșantioane care se află în anumite intervale de valori. Aceste intervale de valori se numesc **clase**. Frecvența datelor care se încadrează în fiecare clasă este descrisă de dimensiunea barei.

Exemplu

Fie semnalul definit de eșantioanele: $x = [1.1 \ 1.3 \ 1.8 \ 2.1 \ 2.7 \ 3.1 \ 3.2 \ 4.3 \ 4.1 \ 3.9 \ 3.2 \ 2.5 \ 1.7 \ 1.2 \ 1.1]$

- Presupunem că prima clasă de valori este în intervalul [1 2].
Câte valori avem în acest interval?



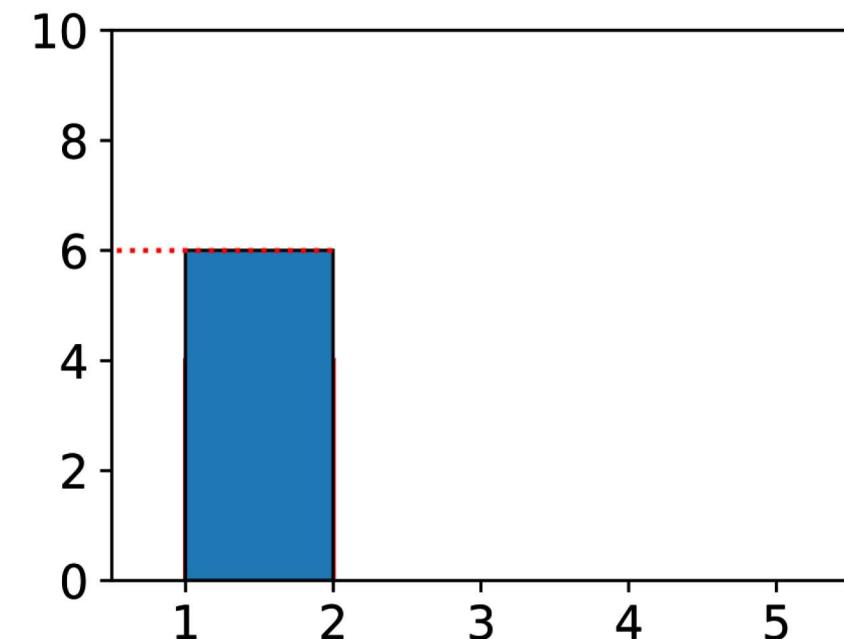
Definiție

Histograma este un grafic ce oferă o interpretare vizuala a semnalelor numerice prin indicarea numărului de eșantioane care se află în anumite intervale de valori. Aceste intervale de valori se numesc **clase**. Frecvența datelor care se încadrează în fiecare clasă este descrisă de dimensiunea barei.

Exemplu

Fie semnalul definit de eșantioanele: $x = [1.1 \quad 1.3 \quad 1.8 \quad 2.1 \quad 2.7 \quad 3.1 \quad 3.2 \quad 4.3 \quad 4.1 \quad 3.9 \quad 3.2 \quad 2.5 \quad 1.7 \quad 1.2 \quad 1.1]$

- Presupunem că prima clasă de valori este în intervalul [1 2].
Câte valori avem în acest interval?
- Presupunem că a doua clasă de valori este în intervalul [2 3].
Câte valori avem în acest interval?



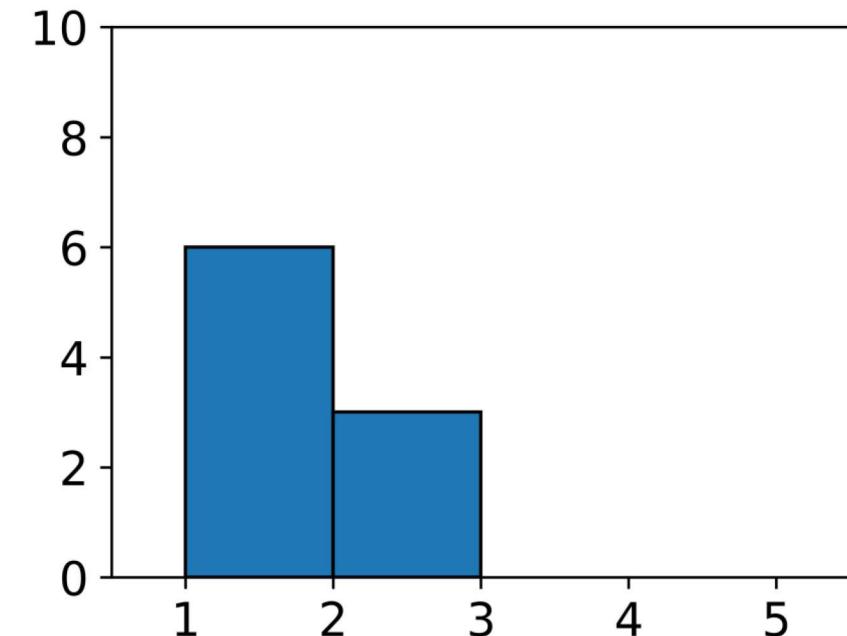
Definiție

Histograma este un grafic ce oferă o interpretare vizuala a semnalelor numerice prin indicarea numărului de eșantioane care se află în anumite intervale de valori. Aceste intervale de valori se numesc **clase**. Frecvența datelor care se încadrează în fiecare clasă este descrisă de dimensiunea barei.

Exemplu

Fie semnalul definit de eșantioanele: $x = [1.1 \text{ } 1.3 \text{ } 1.8 \text{ } 2.1 \text{ } 2.7 \text{ } 3.1 \text{ } 3.2 \text{ } 4.3 \text{ } 4.1 \text{ } 3.9 \text{ } 3.2 \text{ } 2.5 \text{ } 1.7 \text{ } 1.2 \text{ } 1.1]$

- Presupunem că prima clasă de valori este în intervalul [1 2].
Câte valori avem în acest interval?
- Presupunem că a doua clasă de valori este în intervalul [2 3].
Câte valori avem în acest interval?
- Presupunem că a treia clasă de valori este în intervalul [3 4].
Câte valori avem în acest interval?



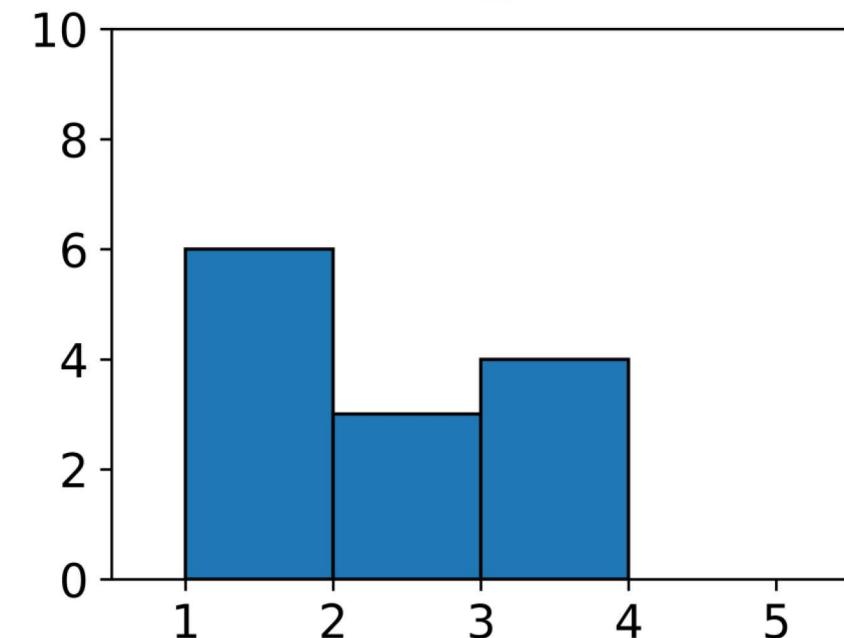
Definiție

Histograma este un grafic ce oferă o interpretare vizuala a semnalelor numerice prin indicarea numărului de eșantioane care se află în anumite intervale de valori. Aceste intervale de valori se numesc **clase**. Frecvența datelor care se încadrează în fiecare clasă este descrisă de dimensiunea barei.

Exemplu

Fie semnalul definit de eșantioanele: $x = [1.1 \text{ } 1.3 \text{ } 1.8 \text{ } 2.1 \text{ } 2.7 \text{ } 3.1 \text{ } 3.2 \text{ } 4.3 \text{ } 4.1 \text{ } 3.9 \text{ } 3.2 \text{ } 2.5 \text{ } 1.7 \text{ } 1.2 \text{ } 1.1]$

- Presupunem că prima clasă de valori este în intervalul [1 2].
Câte valori avem în acest interval?
- Presupunem că a doua clasă de valori este în intervalul [2 3].
Câte valori avem în acest interval?
- Presupunem că a treia clasă de valori este în intervalul [3 4].
Câte valori avem în acest interval?
- Presupunem că a patra clasă de valori este în intervalul [4 5].
Câte valori avem în acest interval?



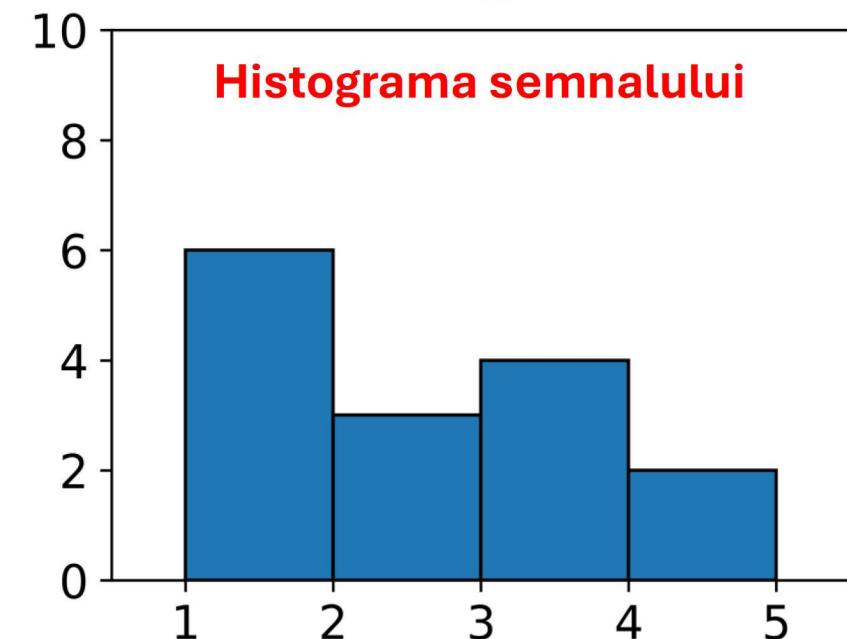
Definiție

Histograma este un grafic ce oferă o interpretare vizuala a semnalelor numerice prin indicarea numărului de eșantioane care se află în anumite intervale de valori. Aceste intervale de valori se numesc **clase**. Frecvența datelor care se încadrează în fiecare clasă este descrisă de dimensiunea barei.

Exemplu

Fie semnalul definit de eșantioanele: $x = [1.1 \text{ } 1.3 \text{ } 1.8 \text{ } 2.1 \text{ } 2.7 \text{ } 3.1 \text{ } 3.2 \text{ } 4.3 \text{ } 4.1 \text{ } 3.9 \text{ } 3.2 \text{ } 2.5 \text{ } 1.7 \text{ } 1.2 \text{ } 1.1]$

- Presupunem că prima clasă de valori este în intervalul [1 2].
Câte valori avem în acest interval?
- Presupunem că a doua clasă de valori este în intervalul [2 3].
Câte valori avem în acest interval?
- Presupunem că a treia clasă de valori este în intervalul [3 4].
Câte valori avem în acest interval?
- Presupunem că a patra clasă de valori este în intervalul [4 5].
Câte valori avem în acest interval?



- Afişarea histogramei semnalului x se va face apelând funcția `hist` din librăria `matplotlib.pyplot`, dându-i ca parametru semnalul x : `plt.hist(x)`.
- Implicit atât barele cât și contururile acestora vor avea culoarea albastră, ceea ce poate îngreuna vizualizarea histogramei.
- Doi parametri utili pentru funcția `plt.hist` se referă la aspectul vizual al histogramei, și anume la culoarea barelor (`color`) și a bordurilor barelor (`edgecolor`), culorile fiind specificate cu aceleași coduri ca la funcția `plt.plot`.

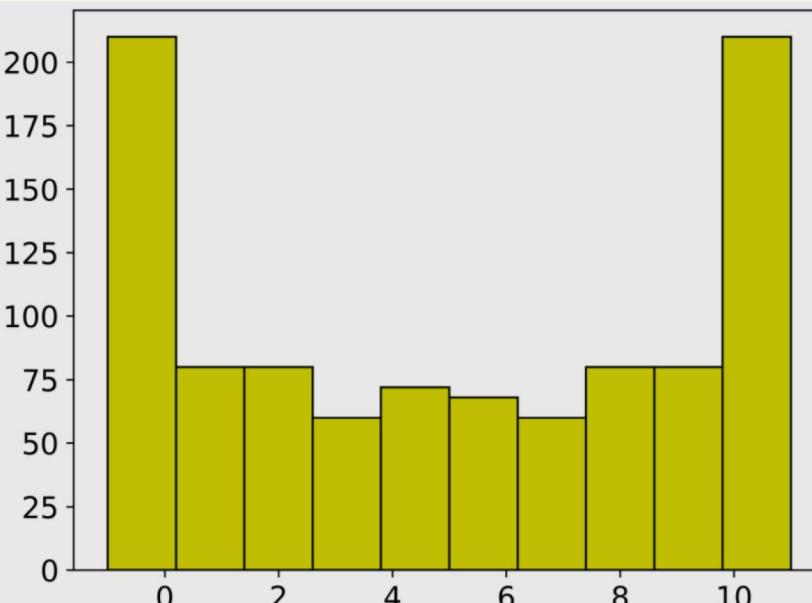
EXEMPLU**main.py**

```
import numpy as np
import matplotlib.pyplot as plt

tf = 2
fs = 500

t = np.arange(0, tf, 1 / fs)
x = 5 + 6 * np.sin(2 * np.pi * t * 5)

plt.hist(x, color="y", edgecolor="k")
plt.show()
```

rezultat

- Afişarea histogramei semnalului x se va face apelând funcția `hist` din librăria `matplotlib.pyplot`, dându-i ca parametru semnalul x : `plt.hist(x)`.
- În mod implicit `plt.hist` va afișa grafic histograma semnalului x , împărțind intervalul de valori al semnalului x în 10 clase de dimensiune egală, lucru care nu este întotdeauna convenient.

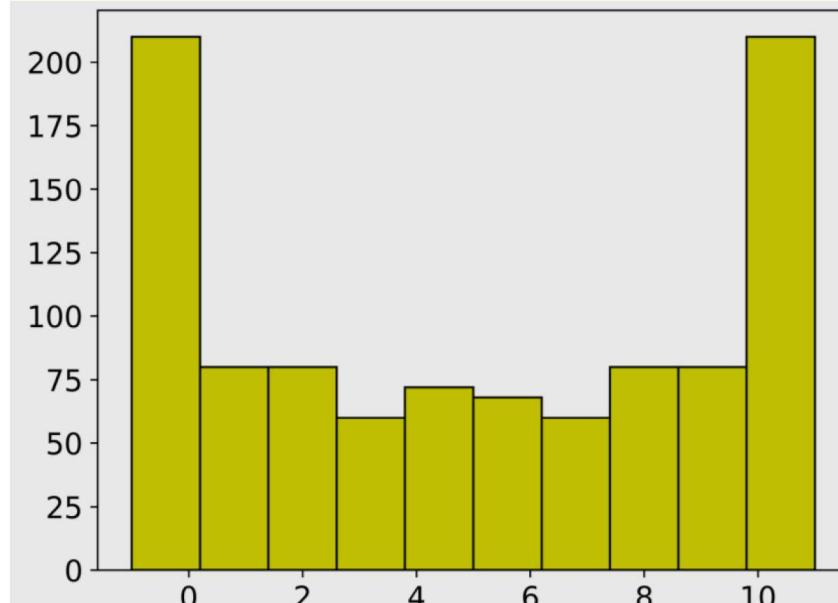
EXEMPLU**main.py**

```
import numpy as np
import matplotlib.pyplot as plt

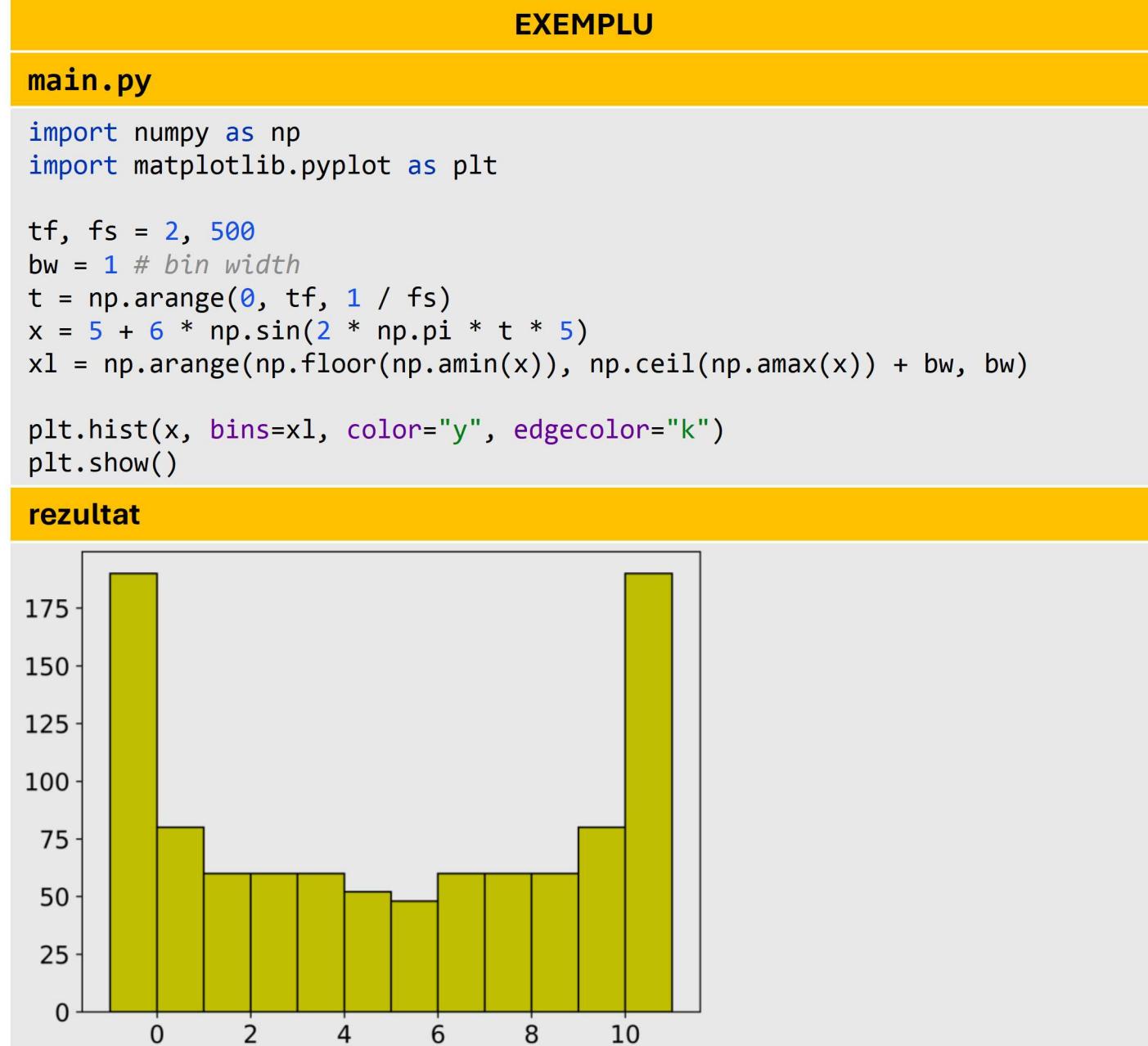
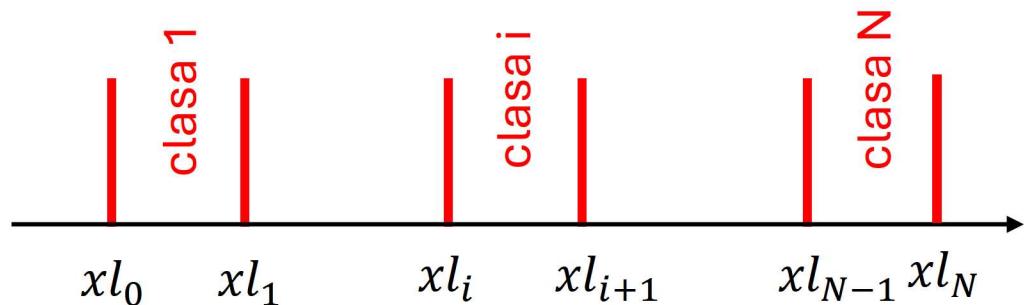
tf = 2
fs = 500

t = np.arange(0, tf, 1 / fs)
x = 5 + 6 * np.sin(2 * np.pi * t * 5)

plt.hist(x, color="y", edgecolor="k")
plt.show()
```

rezultat

- În mod implicit plt.hist va afișa grafic histograma semnalului x, împărțind intervalul de valori al semnalului x în 10 clase de dimensiune egală, lucru care nu este întotdeauna convenient.
- De obicei, va trebui să specificăm noi numărul de clase prin a da ca parametru funcției și o variabilă care să conțină limitele claselor histogramei: plt.hist(x, bins=xl) unde variabila $xl = [xl_0, xl_1, \dots, xl_N]$ reprezintă limitele celor N clase. Astfel, limitele clasei i vor fi xl_{i-1} și xl_i .



- **Distribuția de probabilitate discretă** poate fi reprezentată de o histogramă prin normalizarea valorilor histogramei, dând parametrul `density=True` funcției `plt.hist`.
- De data asta, aria barelor va reprezenta probabilitatea ca, selectând aleatoriu un eșantion din semnalul dat, acesta să se afle în clasa definită de bara respectivă.
- Suma ariilor barelor va fi 1.

EXEMPLU

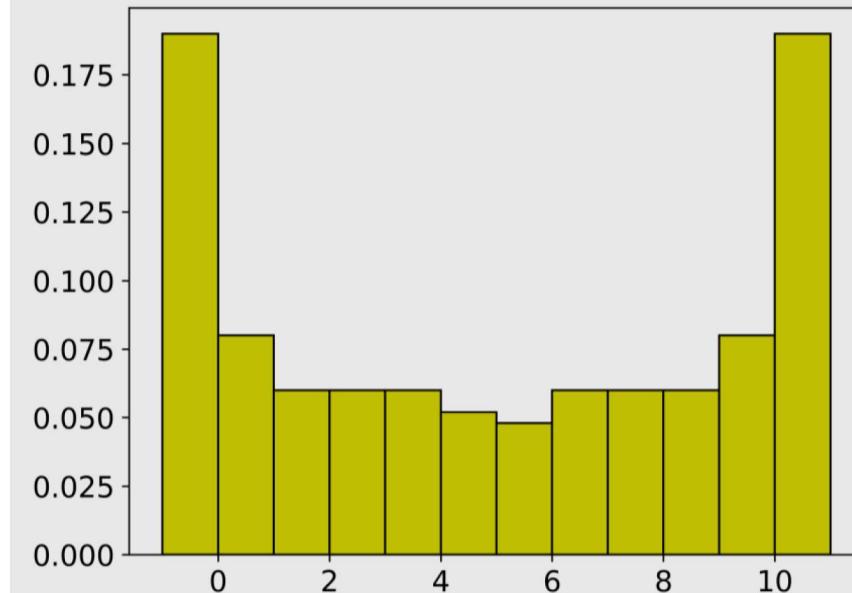
`main.py`

```
import numpy as np
import matplotlib.pyplot as plt

tf, fs = 2, 500
bw = 1 # bin width
t = np.arange(0, tf, 1 / fs)
x = 5 + 6 * np.sin(2 * np.pi * t * 5)
xl = np.arange(np.floor(np.amin(x)), np.ceil(np.amax(x)) + bw, bw)

plt.hist(x, bins=xl, color="y", edgecolor="k", density=True)
plt.show()
```

rezultat



Dorim afișarea comparativă a distribuției temperaturilor medii înregistrate zilnic în 2011-2021 în Galați și Brașov
(sursă date: <https://www.kaggle.com/code/shudhanshurp/romania-weather-eda>)

main.py

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('weather_2011-2021_Romania.csv')
temp_gl = df.query("Address == 'Galati, Romania'")["Temperature"]
temp_bv = df.query("Address == 'Brasov, Romania'")["Temperature"]
xl = np.arange(-25, 35, 1)

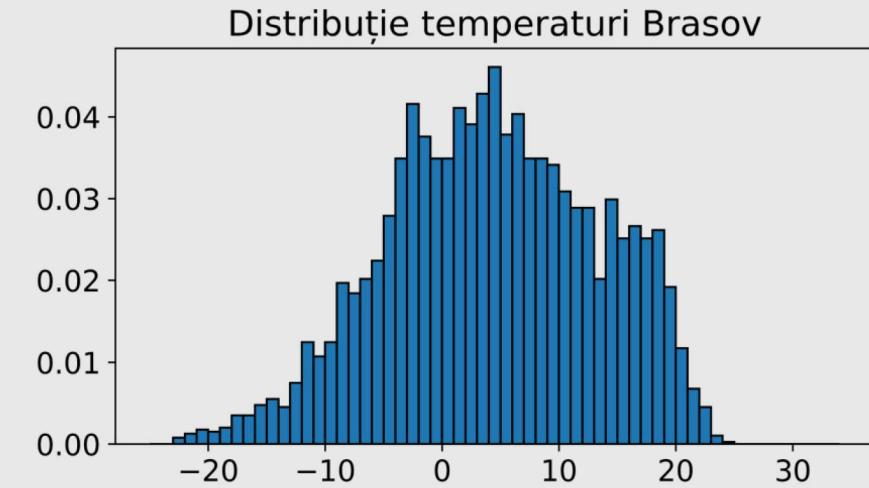
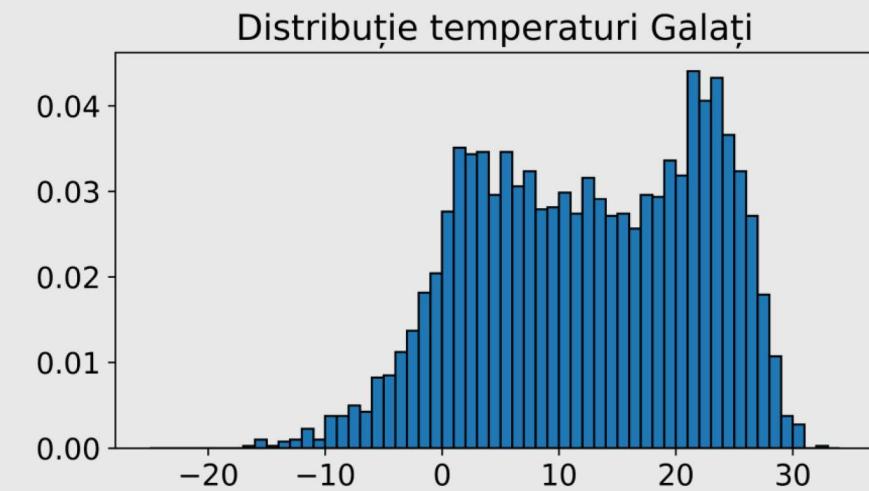
plt.figure(figsize=(6, 7))

plt.subplot(2, 1, 1)
plt.hist(temp_gl, bins=xl, edgecolor="k", density=True)
plt.title("Distribuție temperaturi Galați")

plt.subplot(2, 1, 2)
plt.hist(temp_bv, bins=xl, edgecolor="k", density=True)
plt.title("Distribuție temperaturi Brasov")

plt.tight_layout()
plt.show()
```

rezultat



1. Introducere

2. Histograma

3. Indicatori statistici

4. TCLab

Introducere

Indicatori statistici

Histograma

TCLab

1. Introducere

2. Histograma

3. Indicatori statistici

4. TCLab

Introducere

Histograma

Indicatori statistici

TCLab

- Analiza statistică a unui semnal se face prin calcularea unor indicatori statistici ai semnalului, indicatori ce pot fi clasificați în una din următoarele categorii:
 1. indicatori ai tendinței centrale – descriu tendința medie a valorilor semnalului;
 2. indicatori ai împrăștierii – măsoară gradul de diversificare al valorilor;
 3. indicatori ai formei distribuției – descriu forma distribuției valorilor.
- Pentru calculul indicatorilor statistici ce vor fi prezențați în continuare vom folosi funcții din librăria **NumPy** și modulul **stats** al librăriei **SciPy**.
- **SciPy** este o librărie **Python** pentru calcul științific și tehnic, conținând module pentru calcul statistic, optimizări, algebră liniară, integrare, interpolare, procesare de semnale etc. Instalarea acestei librării se face în mod asemănător cu instrucțiunile prezentate în lucrările anterioare.
- O parte a indicatorilor se pot calcula și pentru obiectele de tip **Series** (definite utilizând librăria **Pandas**), utilizând metodele prezentate în cursul dedicat acestei librării.

- Indicatorii tendinței centrale descriu distribuția de probabilitate $x = [1 \ 1 \ 1 \ 2 \ 2 \ 3 \ 3 \ 3 \ 4 \ 4 \ 3 \ 3 \ 3 \ 2 \ 1 \ 1 \ 1]$ în întregul ei:

- modul** – valoarea cu frecvența cea mai mare
valoarea care apare cel mai des în semnal

```
x_mode = scipy.stats.mode(x)[0]
```

$$x = [1 \underline{1} \underline{1} \ 2 \ 2 \ 3 \ 3 \ 3 \ 4 \ 4 \ 3 \ 3 \ 3 \ 2 \ \underline{1} \ \underline{1} \ 1]$$

- mediana** – valoarea din mijlocul unei distribuții
valoarea din mijlocul sirului de eșantioane obținut prin sortarea în ordine crescătoare a eșantioanelor

```
x_median = np.median(x)
```

$$\{1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 2 \ \underline{2} \ 2 \ 3 \ 3 \ 3 \ 3 \ 4 \ 4\}$$

- media** – valoarea medie a eșantioanelor semnalului

media aritmetică a valorilor eșantioanelor $\bar{x} = \frac{1}{N} \cdot \sum_{n=0}^{N-1} x[n]$

```
x_mean = np.mean(x)
```

$$\bar{x} = 2.13333$$

- Indicatorii tendinței centrale descriu distribuția de probabilitate $x = [1 \ 1 \ 1 \ 2 \ 2 \ 3 \ 3 \ 3 \ 4 \ 4 \ 4 \ 3 \ 3 \ 3 \ 2 \ 1 \ 1 \ 1]$ în întregul ei:

- modul** – valoarea cu frecvența cea mai mare
valoarea care apare cel mai des in semnal

```
x_mode = scipy.stats.mode(x)[0]
```

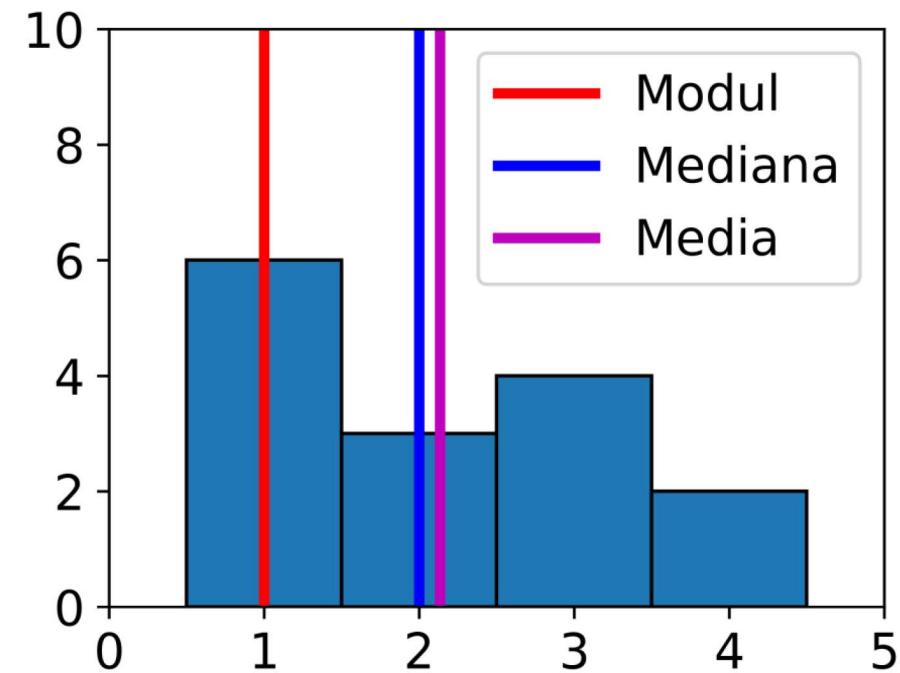
- mediana** – valoarea din mijlocul unei distribuții
valoarea din mijlocul sirului de eșantioane obținut prin sortarea în ordine crescătoare a eșantioanelor

```
x_median = np.median(x)
```

- media** – valoarea medie a eșantioanelor semnalului

media aritmetică a valorilor eșantioanelor $\bar{x} = \frac{1}{N} \cdot \sum_{n=0}^{N-1} x[n]$

```
x_mean = np.mean(x)
```



- Indicatorii împrăştierii măsoară gradul de diversificare a valorilor:

1. amplitudinea absolută

diferența dintre valoarea maximă și valoarea minimă a unei distribuții

$$R = \text{npamax}(x) - \text{npanmin}(x)$$

2. amplitudinea relativă

raportul procentual dintre amplitudine și medie $R_p = \frac{R}{\bar{x}} \cdot 100$

$$Rp = (\text{npamax}(x) - \text{npanmin}(x)) / \text{np.mean}(x) * 100$$

3. abaterea standard (deviația standard)

depărtarea valorilor individuale de la media lor

$$s = \text{np.std}(x)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} |x[n] - \bar{x}|^2}$$

4. dispersia (varianța)

depărtarea valorilor individuale de la media lor $\sigma^2 = \frac{1}{N} \sum_{n=0}^{N-1} |x[n] - \bar{x}|^2$

$$s2 = \text{np.var}(x)$$

- Indicatorii formei distribuției descriu forma distribuției:

1. indicele de asimetrie (skewness)

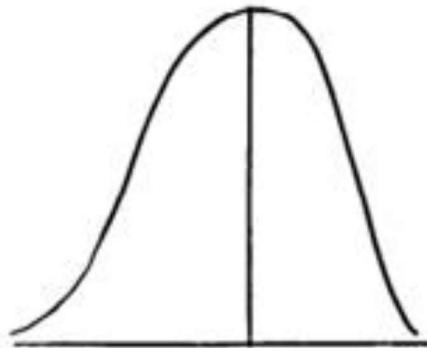
măsoară gradul de asimetrie

$$\gamma_1 = \frac{\sum_{n=0}^{N-1} (x[n] - \bar{x})^3}{N \cdot \sigma^3}$$

În Python

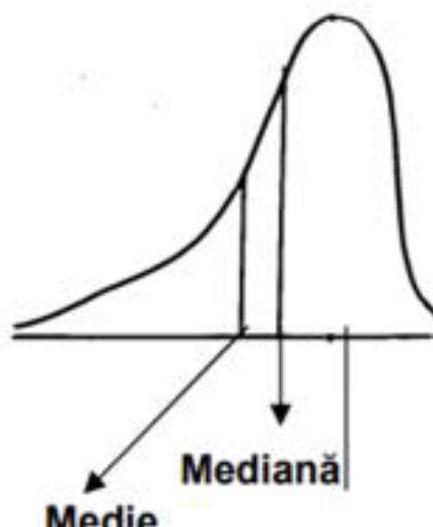
```
gamma1 = scipy.stats.skew(x)
```

Distribuție: simetrică
skewness=0



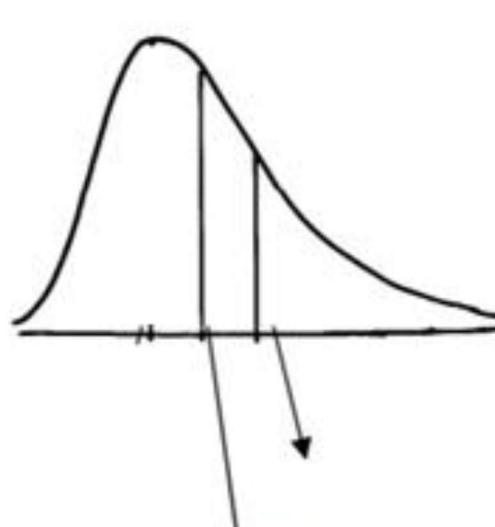
Medie
Mediană
Mod

asimetrică negativ
skewness<0



Medie
Mediană

asimetrică pozitiv
skewness>0



Medie
Mediană

- Indicatorii formei distribuției descriu forma distribuției:

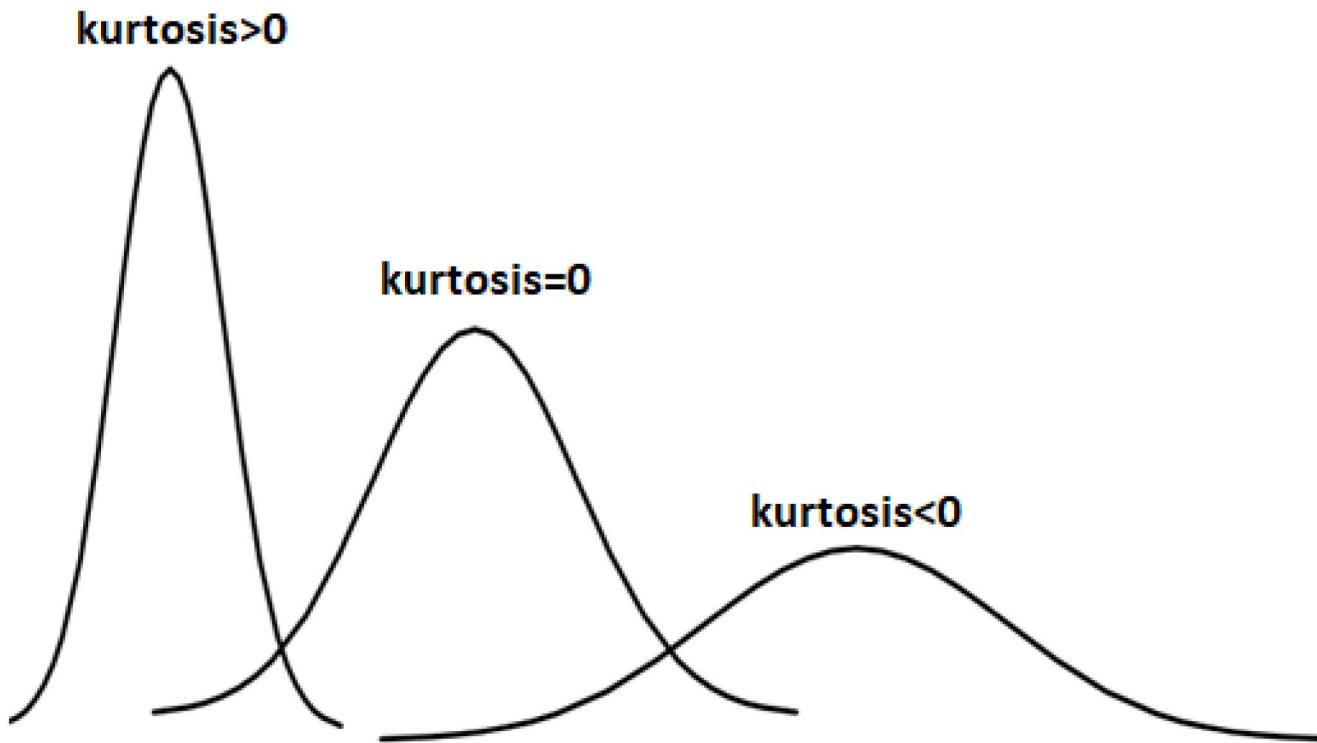
2. indicele de aplatizare (curtoza / kurtosis)

gradul de boltire / aplatizare al distribuției

În Python

```
gamma2 = scipy.stats.kurtosis(x)
```

$$\gamma_2 = \frac{\sum_{n=0}^{N-1} (x[n] - \bar{x})^4}{N \cdot \sigma^4}$$



Dorim afișarea valorilor indicilor statistici calculați pentru temperaturie medii înregistrate zilnic în 2011-2021 în Galați
(sursă date: <https://www.kaggle.com/code/shudhanshurp/romania-weather-eda>)

main.py

```
import numpy as np
import pandas as pd
import scipy

df = pd.read_csv('weather_2011-2021_Romania.csv')
temp_g1 = df.query("Address == 'Galati, Romania'")["Temperature"]

date_statistice = pd.Series({
    "modul": scipy.stats.mode(temp_g1)[0],
    "mediana": np.median(temp_g1),
    "media": np.mean(temp_g1),
    "amplitudinea absolută": npamax(temp_g1)-npamin(temp_g1),
    "amplitudinea relativă": (npamax(temp_g1)-npamin(temp_g1))/np.mean(temp_g1),
    "abaterea standard": np.std(temp_g1),
    "dispersia": np.var(temp_g1),
    "skewness": scipy.stats.skew(temp_g1),
    "kurtosis": scipy.stats.kurtosis(temp_g1)
})

print(date_statistice)
```

rezultat

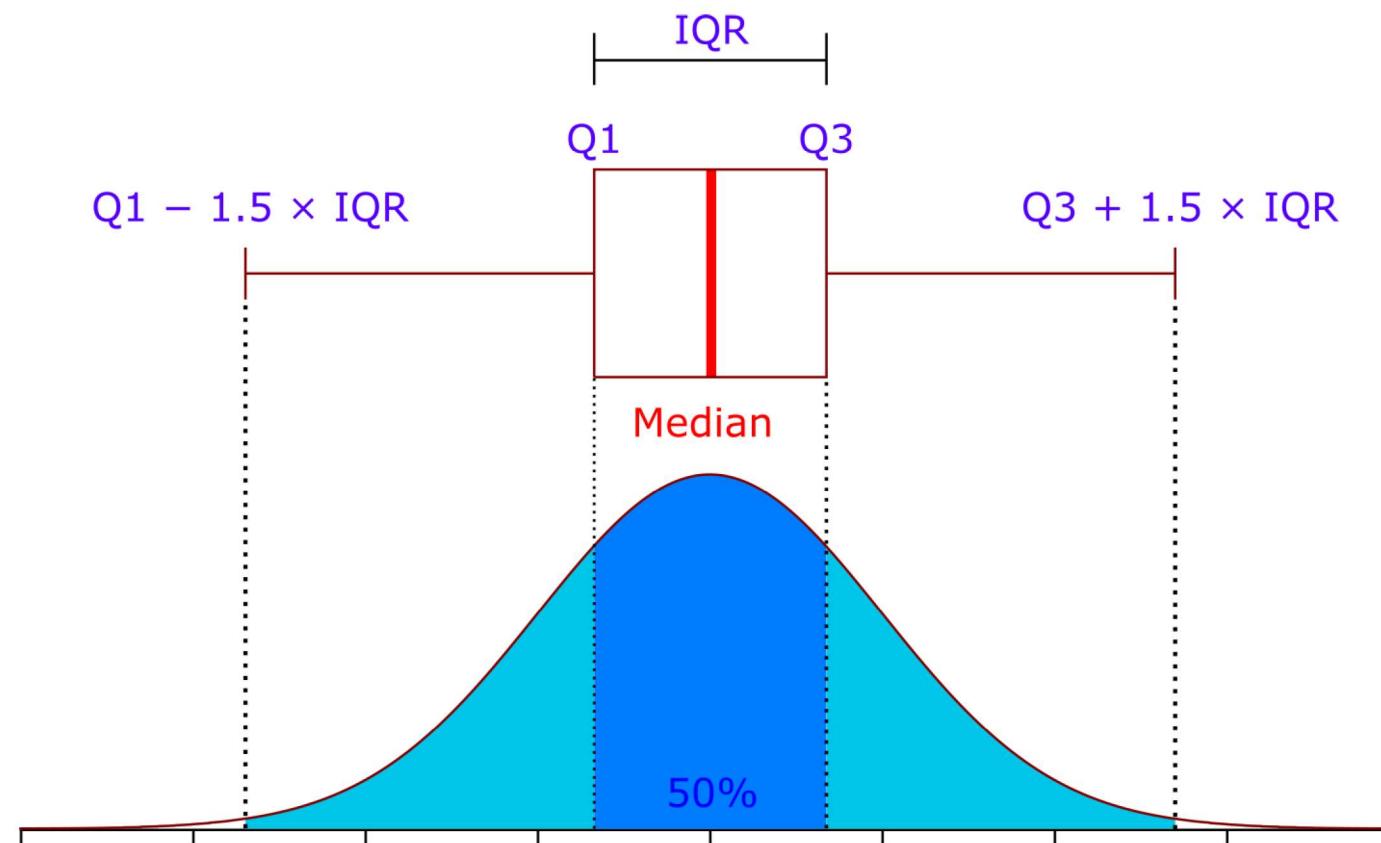
modul	21.800000
mediana	12.700000
media	12.431508
amplitudinea absolută	48.500000
amplitudinea relativă	3.901377
abaterea standard	9.693988
dispersia	93.973407
skewness	-0.212069
kurtosis	-0.921068

dtype: float64

Definiție

Graficul de tip **boxplot** este un tip de reprezentare grafică a datelor care oferă o vizualizare clară a distribuției și a variației unui set de valori. Boxplot-ul este folosit pentru a arăta caracteristicile esențiale ale unui set de date, într-un format compact.

- Cutia, partea centrală a graficului, reprezintă intervalul dintre primul quartil (Q1) și al treilea quartil (Q3), interval ce se numește **interval interquartilic** (IQR) și reprezintă 50% din date.
- Linia din interiorul cutiei reprezintă mediana (împarte datele în două jumătăți)
- Liniile orizontale din fiecare parte a cutiei (numite „mustăți”) se extind până la o distanță de 1,5 ori IQR în ambele direcții față de cutie.
- Valorile care depășesc acest interval sunt considerate *outliers*.

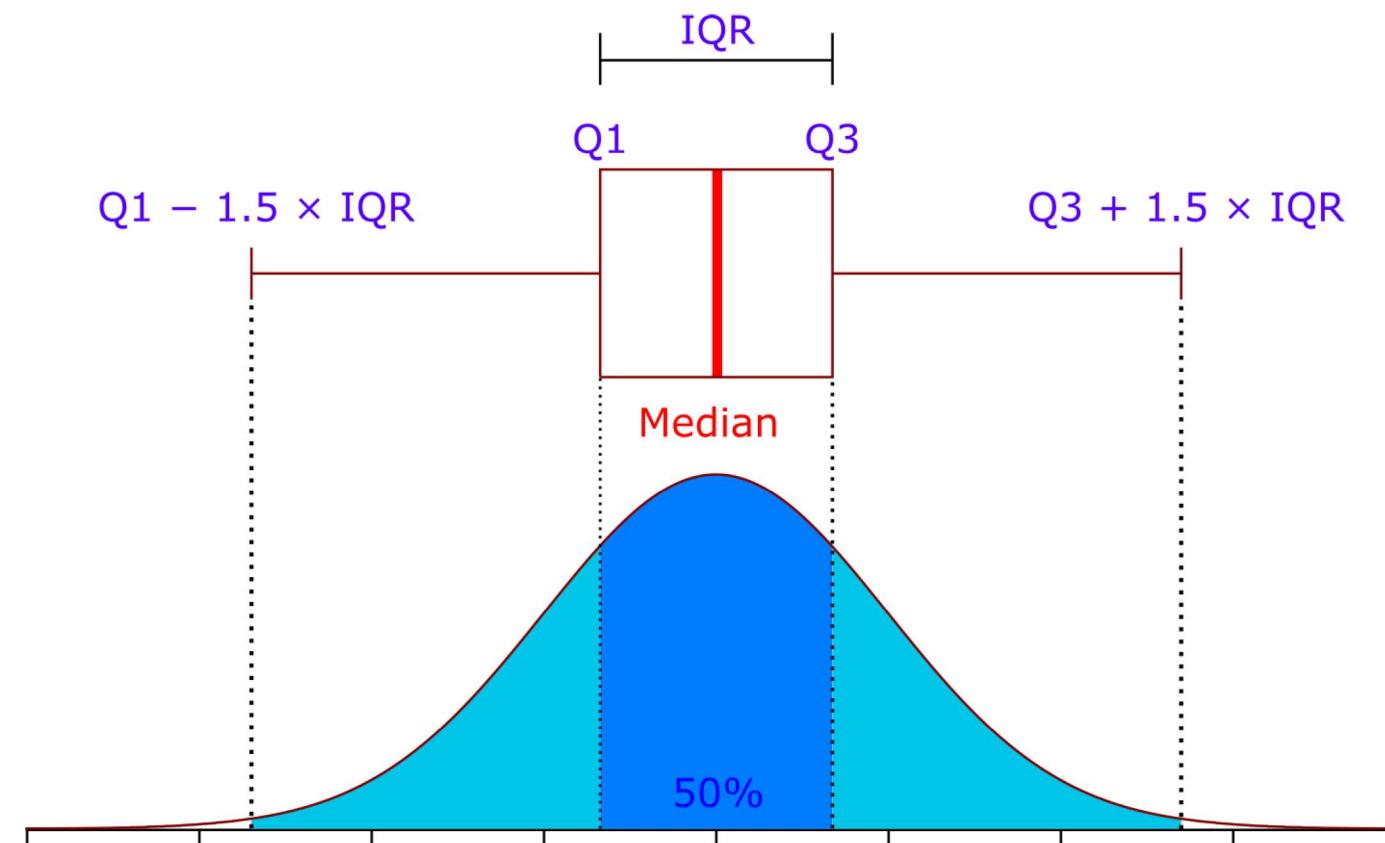


Definiție

Graficul de tip **boxplot** este un tip de reprezentare grafică a datelor care oferă o vizualizare clară a distribuției și a variației unui set de valori. Boxplot-ul este folosit pentru a arăta caracteristicile esențiale ale unui set de date, într-un format compact.

În Python:

- Pentru a realiza grafice de tip **boxplot** se utilizează din **Matplotlib** funcția `plt.boxplot`.
- Această funcție primește ca prim parametru o listă de iterabile și pentru fiecare iterabil, realizează, în aceeași figură, câte un grafic de tip **boxplot**.
- Optional se poate da și parametrul `tick_labels`, o listă ce va specifica etichetele fiecărei distribuții



Dorim afișarea comparativă a statisticilor temperaturilor medii înregistrate zilnic în 2011-2021 în Galați și Brașov utilizând boxplot
(sursă date: <https://www.kaggle.com/code/shudhanshurp/romania-weather-eda>)

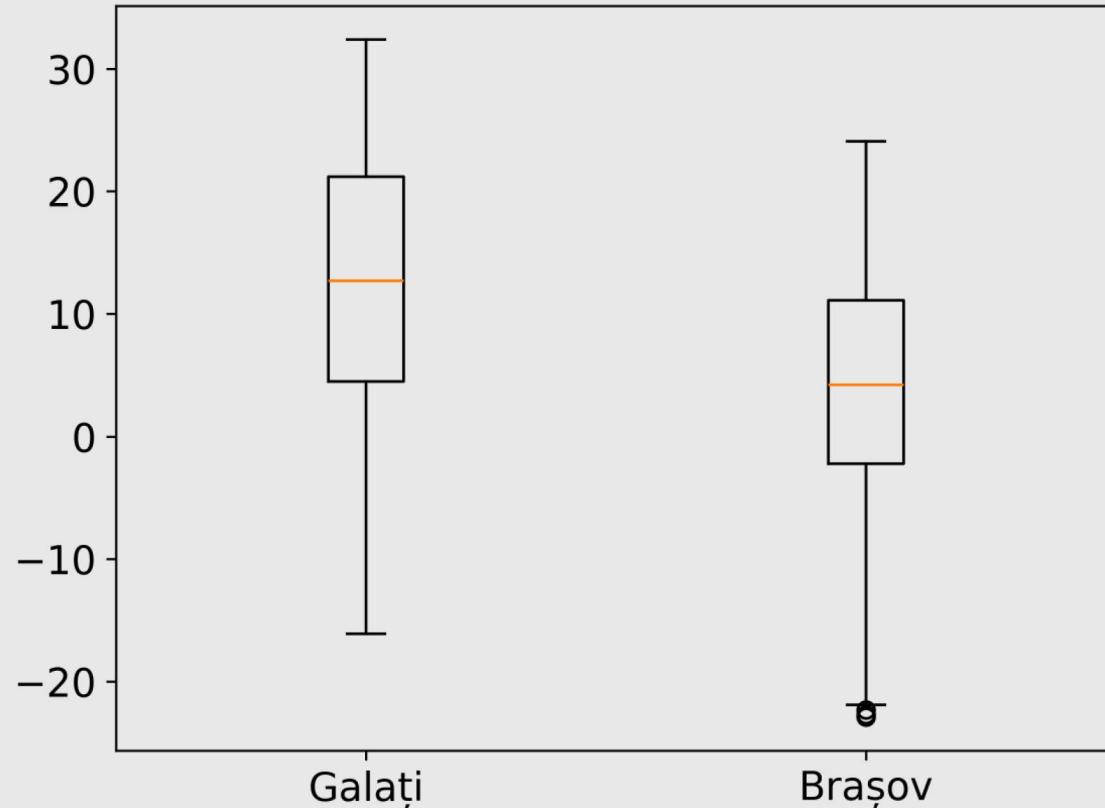
main.py

```
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('weather_2011-2021_Romania.csv')
temp_gl = df.query("Address == 'Galati, Romania'")["Temperature"]
temp_bv = df.query("Address == 'Brasov, Romania'")["Temperature"]

plt.boxplot([temp_gl, temp_bv], tick_labels=["Galați", "Brașov"])
plt.show()
```

rezultat



1. Introducere

2. Histograma

3. Indicatori statistici

4. TCLab

Introducere

Histograma

Indicatori statistici

TCLab

1. Introducere

2. Histograma

3. Indicatori statistici

4. TCLab

Introducere

Histograma

Indicatori statistici

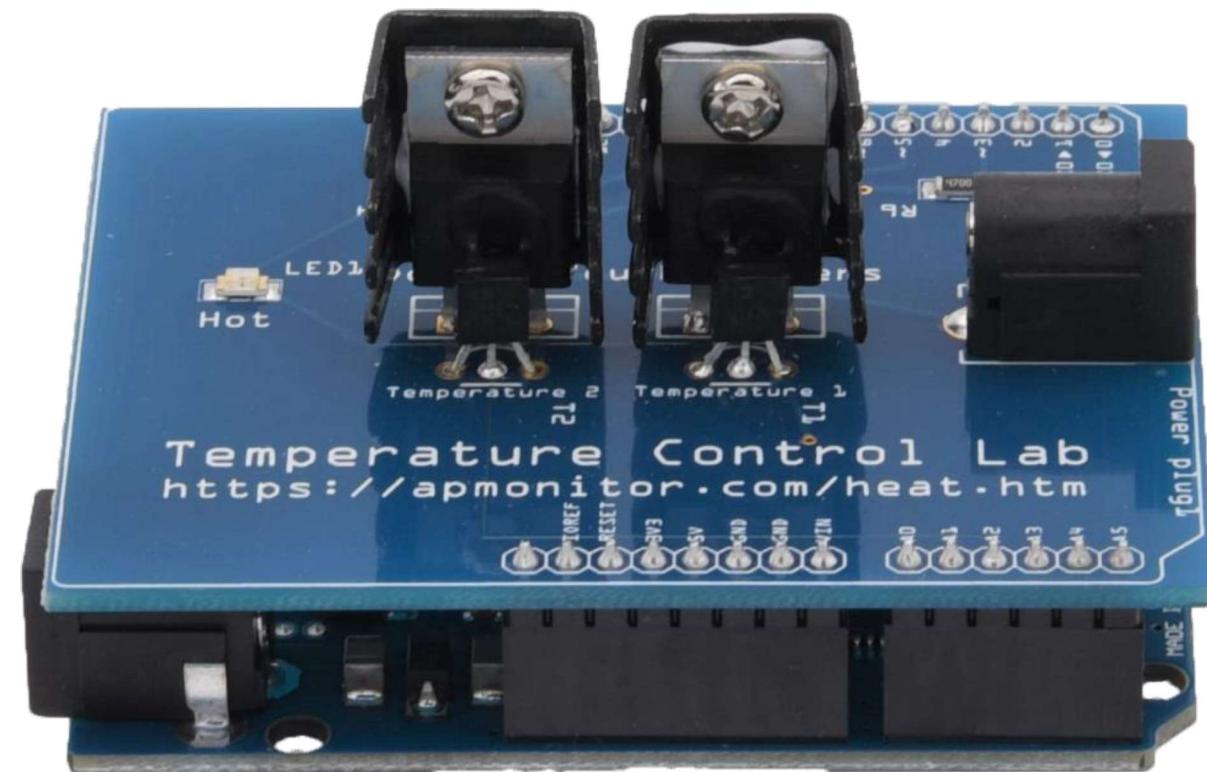
TCLab

„**Temperature Control Lab**” este o aplicație de control cu:

- un Arduino
- două încălzitoare
- doi senzori de temperatură.

Puterea de ieșire a încălzitorului este reglată pentru a menține un punct de referință dorit de temperatură. Energia termică de la încălzitor este transferată prin conducție, convecție și radiație către senzorul de temperatură. Căldura este, de asemenea, transferată de la dispozitiv către mediul înconjurător.

Semnalele provenite de la senzorii de temperatură sunt eșantionate cu o frecvență de eșantionare ce depinde de versiunea „**Temperature Control Lab**”. Pentru versiunea pe care o vom folosi în continuare, frecvența de eșantionare este $f_s = 14.8 \text{ Hz}$.



- Pentru a ne putea interfața cu „**Temperature Control Lab**” este necesar să instalăm și să importăm librăria **tclab**.
- Cât timp suntem conectați cu hardware-ul putem citi valorile celor două temperaturi folosind atributele T1 și T2 ale variabilei lab (lab.T1 și lab.T2). Aceste atribută pot fi doar citite. Orice încercare de a seta altă valoare acestor atribută va genera o eroare.
- Încălzitoarele pot fi controlate folosind funcțiile membre Q1 și Q2 care primesc ca parametru o valoare între 0 și 100 reprezentând procentajul de putere pe care îl dăm acestora. Spre exemplu, dacă dorim să setăm cel de-al doilea încălzitor la 80 % vom scrie lab.Q2(80).

Mod de lucru

```
import tclab

with tclab.TCLab() as lab: # ne conectăm și definim variabila Lab
    ...
    # comunicăm cu hardware-ul folosind variabila Lab
    # ex: setăm valori la încălzitoare, citim temperaturi

    # în afara structurii "with" nu mai putem comunica cu hardware-ul
```

Dorim să setăm primul încălzitor la 100% și să achiziționăm timp de 100 de secunde valorile temperaturii de la senzorul aferent acestui încălzitor

main.py

```
import tclab
import numpy as np
import matplotlib.pyplot as plt
import math

fs = 14.8 # frecvența de esantionare
tf = 100 # câte secunde achiziționam date

N = math.ceil(fs * tf + 1) # numărul total de eșanțioane

t = np.linspace(0, tf, N) # vectorul de timp
x = np.zeros((N,)) # inițializam vectorul valorilor

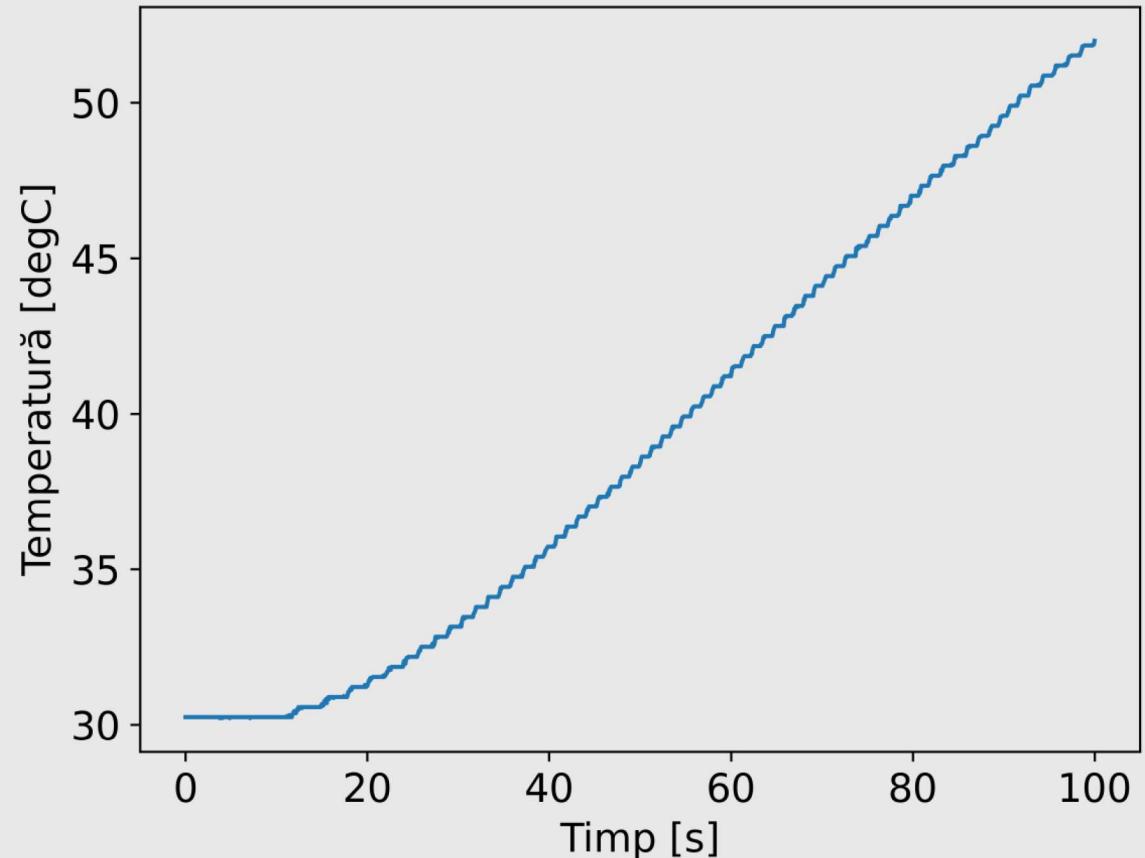
with tclab.TCLab() as lab:
    lab.Q1(100) # pornim încălzitorul la 80 %

    for k in range(0, N):
        x[k] = lab.T1

    lab.Q1(0) # oprim încălzitorul

plt.plot(t, x)
plt.xlabel("Timp [s]")
plt.ylabel("Temperatură [degC]")
plt.show()
```

rezultat



1. Introducere

2. Histograma

3. Indicatori statistici

4. TCLab

Introducere

Histograma

Indicatori statistici

TCLab

1. Introducere

2. Histograma

3. Indicatori statistici

4. TCLab

Introducere

Histograma

Indicatori statistici

TCLab