

Lucrarea 7. Analiza statistică a semnalelor

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.

Înainte de a trece la prezentarea acestor indicatori statistici, este necesar să cunoaștem câteva aspecte legate de histograma unui semnal.

6.1 Histograma

O histogramă este un grafic care oferă o interpretare vizuala a semnalelor numerice prin indicarea numărului de eșantioane care se află într-un interval de valori. Aceste intervale de valori se numesc clase. Frecvența datelor care se încadrează în fiecare clasă este descrisă de dimensiunea barei.

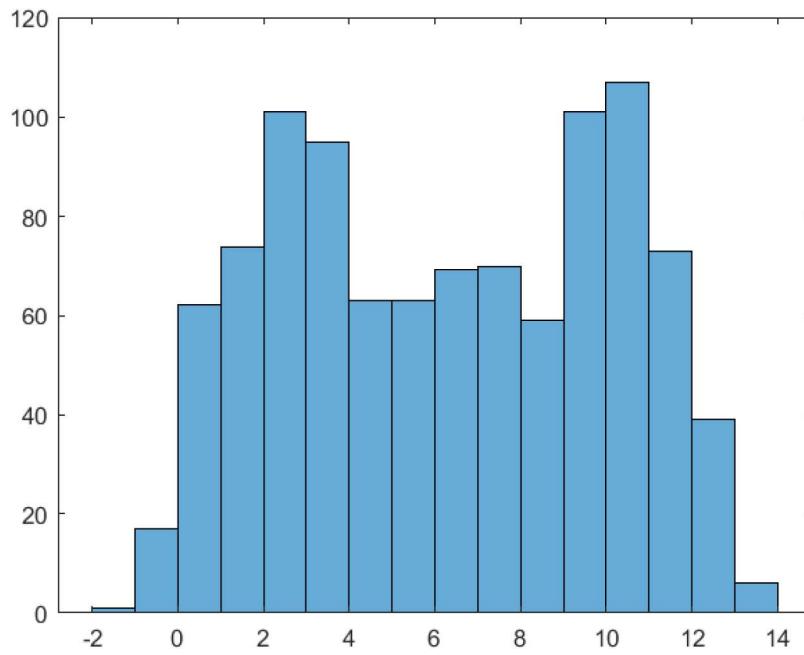


Figura 1 – Histograma

Pentru a afișa în Python histograma semnalului x vom apela funcția `hist` din librăria **matplotlib.pyplot**, dându-i ca parametru semnalul x : `plt.hist(x)` [6].

Apelarea funcției doar cu parametrul `x` (`plt.hist(x)`) va afișa grafic histograma semnalului `x`, împărțind intervalul de valori al semnalului `x` în 10 clase de dimensiune egală. Totuși, acest lucru nu ne este întotdeauna convenient, deci, 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 = [xl0, xl1, ... xlN]` 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: `plt.hist(x, bins=xl, density=True)`.

Alți 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 sunt specificate folosind aceleași coduri de culori ca la funcția `plt.plot`.

Exemplu:

main.py

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

tf = 2
fs = 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.subplot(3, 1, 1)
plt.plot(t, x, "b")
plt.subplot(3, 1, 2)
plt.hist(x, bins=xl, color="r", edgecolor="k")
plt.subplot(3, 1, 3)
plt.hist(x, bins=xl, density=True, edgecolor="k")
plt.show()
```

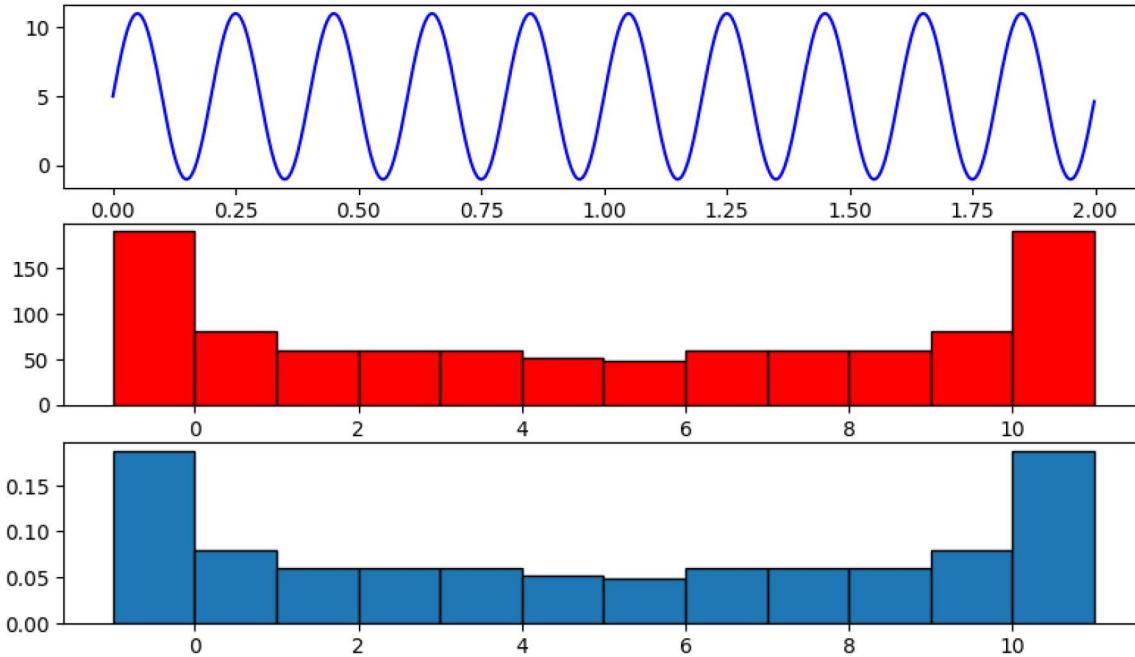


Figura 2 – Exemplu de grafic afișat

6.2 Indicatori statistici

Pentru calculul indicatorilor statistici ce vor fi prezentati 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.

6.2.1 Indicatori ai tendinței centrale

Indicatorii tendinței centrale descriu distribuția de probabilitate în întregul ei.

Indicatorii tendinței centrale [10]:

1. modul – valoarea cu frecvența cea mai mare
2. mediana – valoarea din mijlocul unei distribuții
3. media – valoarea medie a eșantioanelor semnalului

$$\bar{x} = \frac{1}{N} \cdot \sum_{n=0}^{N-1} x[n]$$

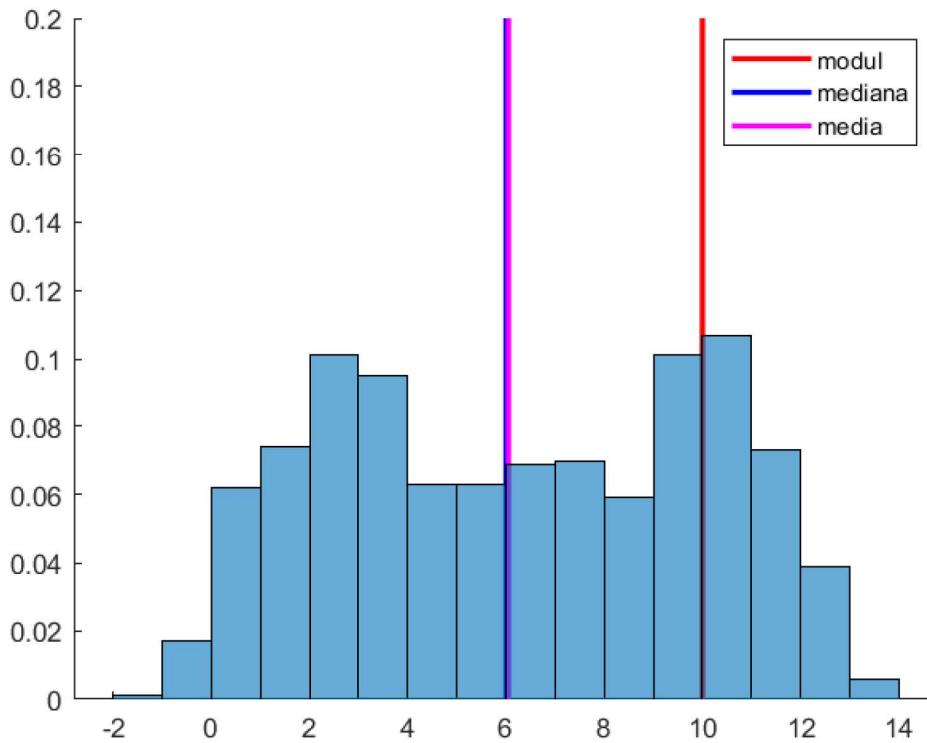


Figura 3 – Indicatori ai tendinței centrale

În Python vom folosi, din librăria **SciPy** funcția `mode` și din librăria **NumPy** funcțiile `median` și `mean` [6]:

1. `x_mode = scipy.stats.mode(x) [0]`
2. `x_median = np.median(x)`
3. `x_mean = np.mean(x)`

6.2.2 Indicatori ai împrăștierii

Indicatorii împrăștierii măsoară gradul de diversificare a valorilor [11]:

1. amplitudinea absolută (diferența dintre valoarea maximă și valoarea minimă a unei distribuții $R = np.amax(x) - np.amin(x)$)
2. amplitudinea relativă (raportul procentual dintre amplitudine și medie) –
$$R_p = \frac{R}{\bar{x}} \cdot 100$$
3. abaterea standard (depărtarea valorilor individuale de la media lor $s=np.std(x)$)

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

4. dispersia (depărtarea valorilor individuale de la media lor $s^2 = np.var(x)$)

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

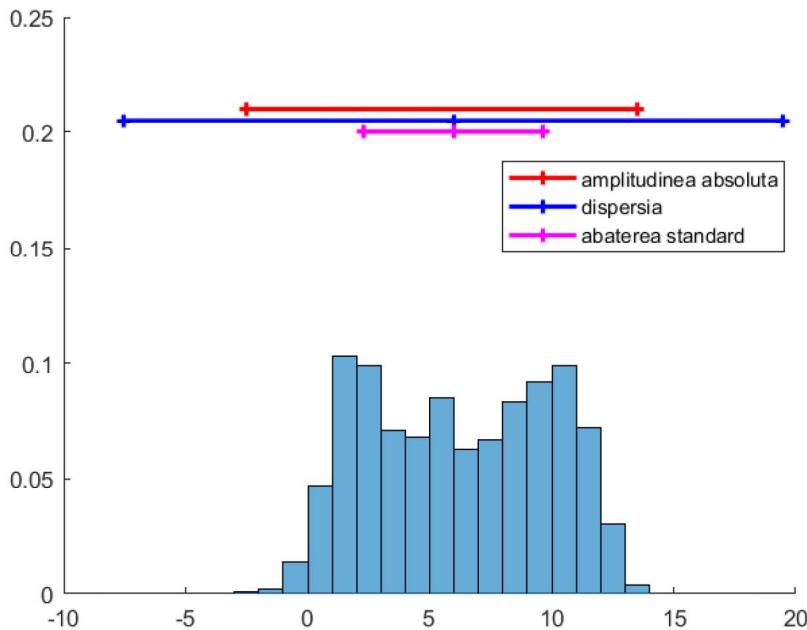


Figura 4 – Indicatori ai împrăștierii

6.2.3 Indicatori ai formei distribuției

Indicatorii formei distribuției descriu forma distribuției [11]:

1. Indicele de asimetrie (skewness) măsoară gradul de asimetrie.
 $is = scipy.stats.skew(x)$

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

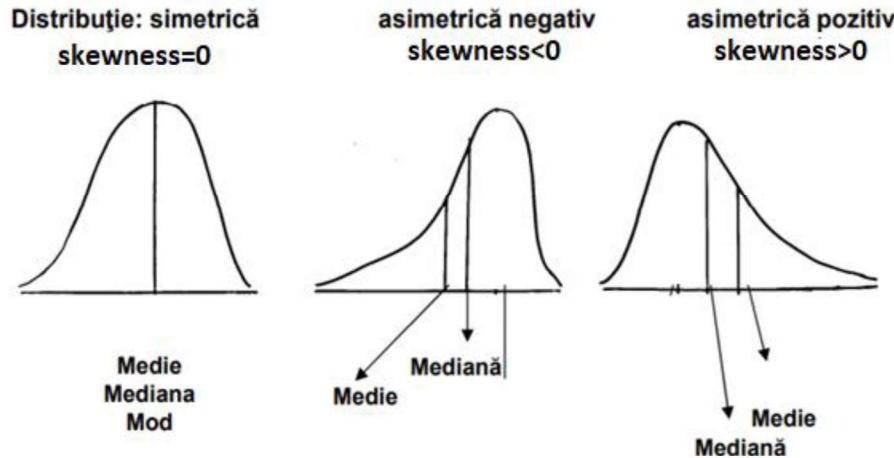


Figura 5 – Gradul de asimetrie al unei distribuții [11]

2. Indicele de aplatizare (curtoza / kurtosis) gradul de boltire / aplatizare al distribuției.

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

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

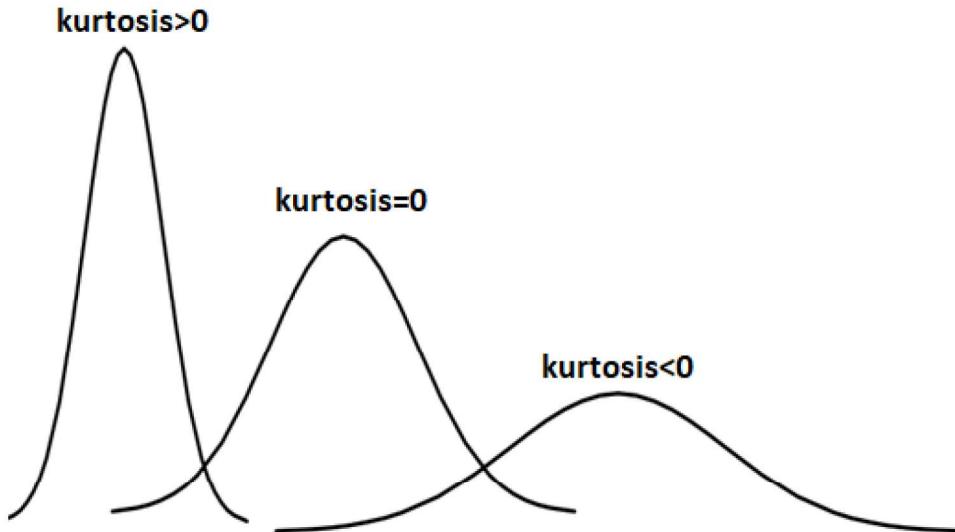


Figura 6 – Gradul de boltire / aplatizare al unei distribuții [11]

6.2.4 Graficul de tip boxplot

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 **mediană** (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**.

Figura următoare prezintă un astfel de grafic de tip **boxplot** în comparație cu o distribuție:

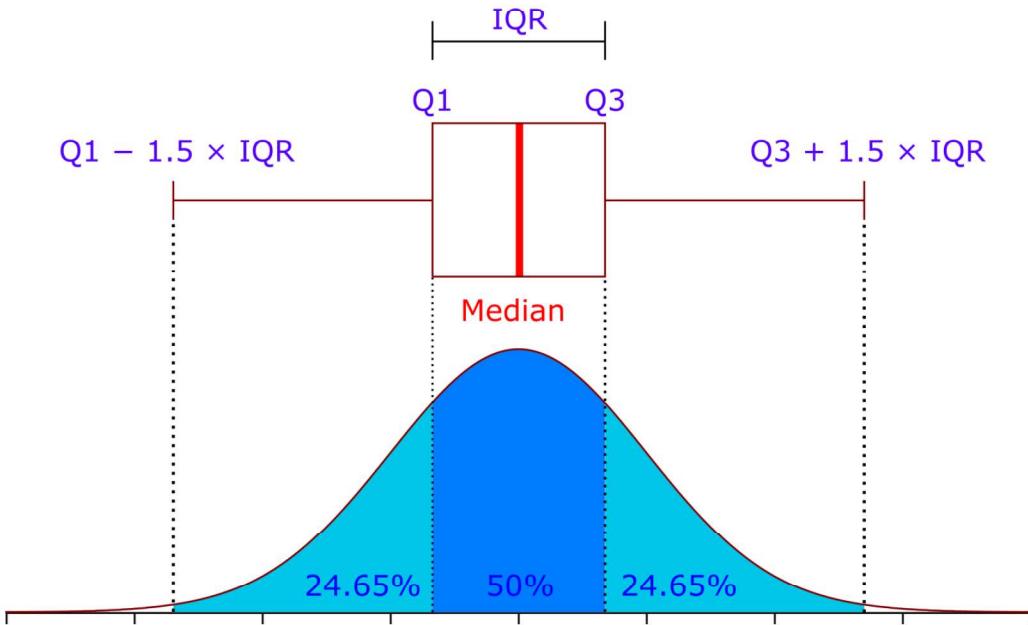


Figura 7 – Exemplu boxplot

Pentru a realiza grafice de tip **boxplot** în **Python** 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**. Opțional se poate da și parametrul `tick_labels`, o listă ce va specifica etichetele fiecărei distribuții.

Exemplu: 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>).

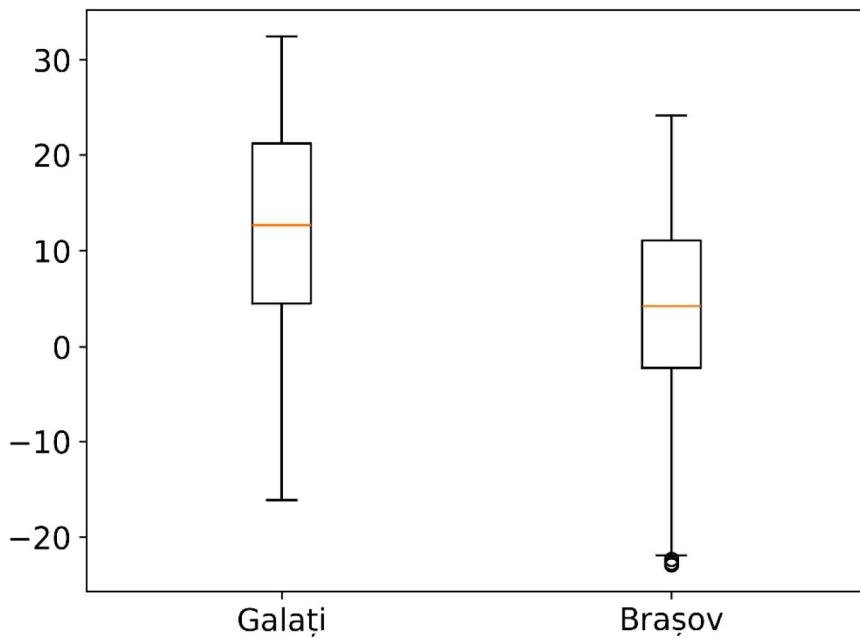
main.py

```
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



6.3 Temperature Control Lab

„Temperature Control Lab” [12] 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.



Figura 8 – Temperature Control Lab [12]

Acest „laborator de buzunar” a fost creat ca o resursă pentru a fi folosită învățarea tehniciilor de identificare a modelului unui sistem și a tehniciilor de control automat. În continuare, vom folosi „Temperature Control Lab” pentru lucrări practice ce țin de procesarea datelor: achiziția și procesarea semnalelor, mai exact ale semnalelor provenite de la senzorii de temperatură [12].

Un aspect important este faptul că 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 $f_s = 14.8 \text{ Hz}$.

6.3.1 Interfața cu Python

Pentru a ne putea interfața cu „Temperature Control Lab” este necesar să instalăm și să importăm librăria **tclab**. Modul de lucru cu această librărie:

main.py

```
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
```

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 atrbute pot fi doar citite. Orice încercare de a seta altă valoare acestor atrbute va genera o eroare.

Încălzitoarele pot fi controlate folosind funcțiiile 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).

Exemplu de achiziție date – 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 achiziționăm date

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

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

with tclab.TCLab() as lab:
    lab.Q1(80) # 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()

```

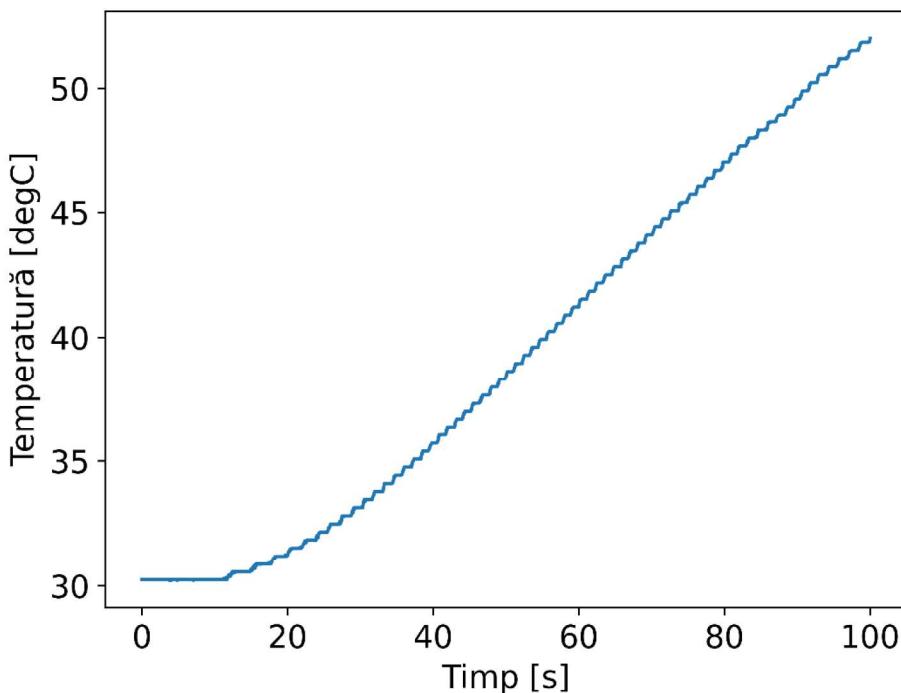


Figura 8 – Exemplu de grafic afișat

6.4 Exerciții

1. Fie semnalul:

$$x(t) = 6 + 5 \cdot \frac{\sin(2 \cdot \pi^2 \cdot t)}{2 \cdot \pi^2 \cdot t}$$

eșantionat cu o frecvență $f_s = 500 \text{ Hz}$ pe o perioadă de 2 secunde și cuantizat cu un pas de cuantizare $q = 0.5$:

- Să se genereze și să se reprezinte grafic eșantioanele semnalului înainte de cuantizare și după cuantizare;
- Să se reprezinte grafic histograma eșantioanelor semnalului și distribuția de probabilitate discretă;
- Să se calculeze și afișeze, pentru semnalul dat, toți indicatorii statistici prezentați.

2. Fie semnalul:

$$x(t) = 6 + 5 \cdot \frac{\sin(2 \cdot \pi^2 \cdot t)}{2 \cdot \pi^2 \cdot t}$$

afectat de zgomot (în Python vom adăuga semnalului x valorile unei matrici unidimensionale `zgomot = np.random.randn(len(t))`) eșantionat cu o frecvență $f_s = 500 \text{ Hz}$ pe o perioadă de 2 secunde și cuantizat cu un pas de cuantizare $q = 0.5$:

- a. Să se genereze și să se reprezinte grafic eșantioanele semnalului înainte de cuantizare și după cuantizare;
 - b. Să se reprezinte grafic histograma eșantioanelor semnalului și distribuția de probabilitate discretă;
 - c. Să se calculeze și afișeze, pentru semnalul dat, toți indicatorii statistici prezentați.
3. Pornind de la **DataFrame**-ul `df1` rezultat în urma rezolvării cerinței numărul 6 din Lucrarea 4, rezolvați:
- a. Afipați grafic în 2 sub-ferestre histogramele temperaturilor medii și temperaturilor maxime înregistrate de stația „Vârfu Omu”. Adăugați etichete axelor și titlu graficului.
 - b. Afipați un grafic de tip boxplot pentru toate temperaturile medii înregistrate, grupate după numele stației meteorologice.
4. Achiziționați de la „Temperature Control Lab” valorile de temperatură de la cel de-al doilea senzor pentru 20 de secunde, fără a porni încălzitorul:
- a. Să se reprezinte grafic semnalul achiziționat;
 - b. Să se reprezinte grafic histograma eșantioanelor semnalului achiziționat și distribuția de probabilitate discretă;
 - c. Să se calculeze și afișeze, pentru semnalul achiziționat, toți indicatorii statistici prezentați.
5. Achiziționați de la „Temperature Control Lab” valorile de temperatură de la primul senzor pentru 90 de secunde, timp în care încălzitorul va fi setat la 100 %:
- a. Să se reprezinte grafic semnalul achiziționat;
 - b. Să se reprezinte grafic histograma eșantioanelor semnalului achiziționat și distribuția de probabilitate discretă;
 - c. Să se calculeze și afișeze, pentru semnalul achiziționat, toți indicatorii statistici prezentați.