



# **BỘ GIÁO DỤC ĐÀO TẠO** **TRƯỜNG ĐẠI HỌC SÀI GÒN**

**Khoa Công nghệ Thông tin**

**PHÂN TÍCH DỮ LIỆU - BUỔI 02**  
**Tiền Xử Lý Dữ Liệu – THỰC HÀNH**

**CBGD: Phan Thành Huấn**

✉ ⚡ : [pthuan112358@gmail.com](mailto:pthuan112358@gmail.com)

📞 : 097 882 8842

## Nội dung

1. Cài đặt môi trường Python
2. Một số thư viện dùng trong PTDL;
3. Websites chứa dữ liệu dùng trong môn PTDL;
4. Cấu trúc tập dữ liệu CSV (Comma-Separated Values);
5. Thực hành tiền xử lý dữ liệu.



DA-SGU2023

# 1-Cài đặt môi trường Python

## Cài đặt môi trường Python trên Windows:

**Bước 1:** Tải Python tại <https://www.python.org/downloads/>

- Chọn phiên bản Python phù hợp với hệ điều hành (Python 3.9.6);
- Tải trình cài đặt Python và chạy tệp cài đặt;

**Bước 2:** Cài đặt Python

- Chạy trình cài đặt Python đã tải xuống;
- Chọn "**Add Python to PATH**" và nhấp vào "**Install Now**";
- Chờ đợi quá trình cài đặt hoàn tất.

# 1-Cài đặt môi trường Python

## **Bước 3:** Kiểm tra cài đặt

- Mở Command Prompt (Windows);
- Gõ lệnh **`python --version`** để kiểm tra phiên bản Python;

## **Bước 4:** Cài đặt Trình quản lý gói (pip)

- Mở Command Prompt (Windows);
- Gõ lệnh **`python -m ensurepip --upgrade`** để cài đặt hoặc cập nhật Trình quản lý gói (pip);

## **Bước 5:** Kiểm tra cài đặt pip

- Gõ lệnh **`pip --version`** để kiểm tra phiên bản pip.

# 1-Cài đặt môi trường Python

Cài đặt PyCharm - môi trường phát triển tích hợp (IDE) cho Python:

**Bước 1:** Tải PyCharm <https://www.jetbrains.com/pycharm/download/>

- Chọn bản PyCharm phù hợp (*Community* hoặc *Professional*);
- Tải trình cài đặt PyCharm và chạy tệp cài đặt;

**Bước 2:** Cài đặt PyCharm

- Chạy trình cài đặt Python đã tải xuống;
- Chọn các tùy chọn cài đặt (đường dẫn, tạo biểu tượng trên desktop);

**Bước 3:** Khởi động PyCharm

- PyCharm yêu cầu nhập đường dẫn đến Python Interpreter (máy ảo).



## 2-Một số thư viện dùng trong phân tích dữ liệu

Cú pháp cài đặt thư viện bằng pip:

**pip install** <tên-thư-viện-1> <tên-thư-viện-2> <tên-thư-viện-3>...

**pip install** <tên-thư-viện>==<phiên-bản>

- 1. Pandas:** Cung cấp các cấu trúc dữ liệu linh hoạt như DataFrame và Series, và có thể thực hiện nhiều thao tác như lọc, nhóm, tính toán thống kê, và nhiều hơn nữa;
- 2. NumPy:** Cung cấp các cấu trúc dữ liệu và chức năng toán học cao cấp để làm việc với mảng và ma trận - thư viện quan trọng nhất cho tính toán khoa học và tính toán số;

## 2-Một số thư viện dùng trong PTDL

3. **Matplotlib**: Trực quan hóa dữ liệu mạnh mẽ, cho phép tạo ra các biểu đồ, đồ thị và hình ảnh chất lượng cao;
4. **Seaborn**: Trực quan hóa dữ liệu dựa trên Matplotlib, Seaborn cung cấp các chủ đề màu sắc và phong cách trực quan hóa tốt hơn;
5. **SciPy**: Mở rộng cho tính toán khoa học và tính toán số - các chức năng cho tối ưu hóa, tích phân, đại số tuyến tính, xử lý tín hiệu, và nhiều hơn nữa. Đây là thư viện quan trọng nhất cho PTDL.

### 3-Websites chứa dữ liệu dùng trong môn PTDL

#### 1. UCI Machine Learning Repository:

<http://archive.ics.uci.edu/datasets>

Kho dữ liệu lớn với nhiều bộ dữ liệu từ nhiều lĩnh vực khác nhau;

#### 2. Kaggle:

<https://www.kaggle.com/datasets>

Kaggle là một cộng đồng dữ liệu và thử thách khoa học dữ liệu. Trang web này cung cấp các bộ dữ liệu từ nhiều lĩnh vực, bao gồm tài chính, y tế, giao thông, thể thao và nhiều hơn nữa.



## **3-Websites chứa dữ liệu dùng trong môn PTDL**

### **3. Data.gov:**

<https://www.data.gov/>

Đây là trang web chính phủ Hoa Kỳ cung cấp các bộ dữ liệu từ nhiều cơ quan chính phủ khác nhau (kinh tế, môi trường, giáo dục,...).

### **4. Google Dataset Search:**

<https://datasetsearch.research.google.com/>

Google dành riêng cho việc tìm kiếm các bộ dữ liệu trực tuyến.

### **5. World Bank Open Data:**

<https://data.worldbank.org/>

Cung cấp các bộ dữ liệu về kinh tế, dân số, môi trường và các chủ đề xã hội khác từ khắp nơi trên thế giới.

## 4-Cấu trúc tập dữ liệu CSV (Comma-Separated Values)

- File CSV (**Comma-Separated Values**): định dạng file dữ liệu được sử dụng phổ biến để lưu trữ và truyền tải dữ liệu dưới dạng bảng;
- Dữ liệu được phân tách và định cấu trúc bằng dấu phẩy (,) hoặc (;) hoặc tab (**\t**).
- *Mỗi dòng trong file CSV đại diện cho một hàng trong bảng dữ liệu và các giá trị của các cột được phân tách ký tự phân tách;*
- Dữ liệu có thể được mở và xử lý bằng các chương trình và ngôn ngữ lập trình như Python, R, Excel và nhiều ngôn ngữ khác.

## 4-Cấu trúc tập dữ liệu CSV (Comma-Separated Values)

### Cách tạo file CSV:

- **Bước 1:** Mở trình soạn thảo văn bản Notepad hoặc Ms Excel;
- **Bước 2:** Tạo tiêu đề cột - Dòng đầu tiên của file, ghi tên các cột tương ứng với dữ liệu bạn muốn lưu trữ. Mỗi tên cột được phân tách bằng ký tự phân tách;
- **Bước 3:** Thêm dữ liệu vào các dòng tiếp theo - Trên các dòng tiếp theo, ghi thông tin tương ứng cho mỗi cột;
- **Bước 4:** Lưu file với định dạng CSV (MS-DOS).

## 4-Cấu trúc tập dữ liệu CSV (Comma-Separated Values)

	A	B	C	D	E	F	G	H
1	I1	I2	I3	I4	I5	I6	I7	I8
2	A	?	C	?	E	F	?	?
3	A	?	C	?	?	?	G	?
4	?	?	?	?	E	?	?	H
5	A	?	C	D	?	F	G	?
6	A	?	C	?	E	?	G	?
7	?	?	?	?	E	?	?	?
8	A	B	C	?	E	?	?	?
9	A	?	C	D	?	?	?	?
10	A	B	C	?	E	?	G	?
11	A	?	C	?	E	F	G	?

Mở file **CSV (MS-DOS)** trên **NotePad**

i1, i2, i3, i4, i5, i6, i7, i8|

A,?,C,?,E,F,?,?

A,?,C,?,?,?,G,?

?,?,?,?,E,?,?,H

A,?,C,D,?,F,G,?

A,?,C,?,E,?,G,?

?,?,?,?,E,?,?,?

A,B,C,?,E,?,?,?

A,?,C,D,?,?,?,?

A,B,C,?,E,?,G,?

A,?,C,?,E,F,G,?



## 5-Thực hành tiền xử lý dữ liệu

### Nội dung:

1. **Xử lý dữ liệu thiếu:** Kiểm tra và xử lý các giá trị thiếu trong tập dữ liệu, có thể bằng cách điền giá trị thiếu bằng một giá trị mặc định hoặc sử dụng các phương pháp như *điền giá trị trung bình, trung vị, hoặc phân phối dữ liệu*;
2. **Xử lý dữ liệu không hợp lệ:** Kiểm tra và xử lý các giá trị không hợp lệ hoặc ngoại lệ trong tập dữ liệu, có thể bằng cách loại bỏ các dòng chứa giá trị không hợp lệ hoặc thay thế chúng bằng các giá trị hợp lệ;

## 5-Thực hành tiền xử lý dữ liệu

3. **Chuyển đổi dữ liệu:** Chuyển đổi các kiểu dữ liệu không phù hợp sang kiểu dữ liệu phù hợp - chuyển đổi chuỗi thành số/ngược lại;
4. **Chuẩn hóa dữ liệu:** Chuẩn hóa dữ liệu để đảm bảo rằng các đặc trưng có cùng phạm vi hoặc phân phối tương tự - chuẩn hóa dữ liệu số thành dạng z-score hoặc min-max scaling;
5. **Xử lý dữ liệu dư thừa:** Loại bỏ các cột dữ liệu không cần thiết hoặc dư thừa trong tập dữ liệu, giúp giảm kích thước của tập dữ liệu và cải thiện hiệu suất phân tích;

## 5-Thực hành tiền xử lý dữ liệu

6. **Xử lý biến đổi:** Tạo ra các biến đổi mới từ các biến đầu vào, chẳng hạn như tạo biến tương tác, biến đổi logarit, hoặc biến đổi đa thức để tăng tính linh hoạt và khả năng biểu diễn của mô hình;
7. **Xử lý và mã hóa biến phân loại:** Chuyển đổi các biến phân loại thành dạng số để các mô hình máy học có thể xử lý được, chẳng hạn như mã hóa one-hot, mã hóa nhãn, hoặc mã hóa tần số;

## 5-Thực hành tiền xử lý dữ liệu

8. **Xử lý và loại bỏ nhiễu:** Xử lý và loại bỏ nhiễu trong dữ liệu, chẳng hạn như loại bỏ các giá trị ngoại lai hoặc các điểm dữ liệu không phù hợp;
9. **Tạo đặc trưng mới:** Tạo ra các đặc trưng mới từ dữ liệu ban đầu, chẳng hạn như tạo đặc trưng thời gian, đặc trưng phân loại dựa trên quan hệ giữa các biến, hoặc đặc trưng thống kê từ dữ liệu.



## 5-Thực hành tiền xử lý dữ liệu

### Ví dụ: Thống kê mô tả dữ liệu

```
import pandas as pd

# Đọc tập dữ liệu từ file CSV
data = pd.read_csv('chess.csv')

# Hiển thị 5 dòng đầu tiên của tập dữ liệu
print(data.head())

# Số lượng dòng và cột trong tập dữ liệu
num_rows, num_cols = data.shape
print("Số lượng dòng:", num_rows)
print("Số lượng cột:", num_cols)

# Số lượng giá trị duy nhất trong từng cột
unique_values = data.nunique()
print("Số lượng giá trị duy nhất trong từng cột:")
print(unique_values)

summary = data.describe()

print(summary)
```

```

  A  B  C  D  E  F  G
0  a  1  b  3  c  2  draw
1  a  1  c  1  c  2  draw
2  a  1  c  1  d  1  draw
3  a  1  c  1  d  2  draw
4  a  1  c  2  c  1  draw
Số lượng dòng: 28056
Số lượng cột: 7
Số lượng giá trị duy nhất trong từng cột:
A      4
B      4
C      8
D      8
E      8
F      8
G     18
dtype: int64

              B              D              F
count  28056.000000  28056.000000  28056.000000
mean         1.854006         4.512404         4.451811
std          0.926414         2.282723         2.248387
min          1.000000         1.000000         1.000000
25%          1.000000         3.000000         3.000000
50%          2.000000         5.000000         4.000000
75%          2.000000         6.000000         6.000000
max          4.000000         8.000000         8.000000

```

## 5-Thực hành tiền xử lý dữ liệu

**Ví dụ:** Thống kê mô tả dữ liệu – sử dụng biểu đồ Histogram

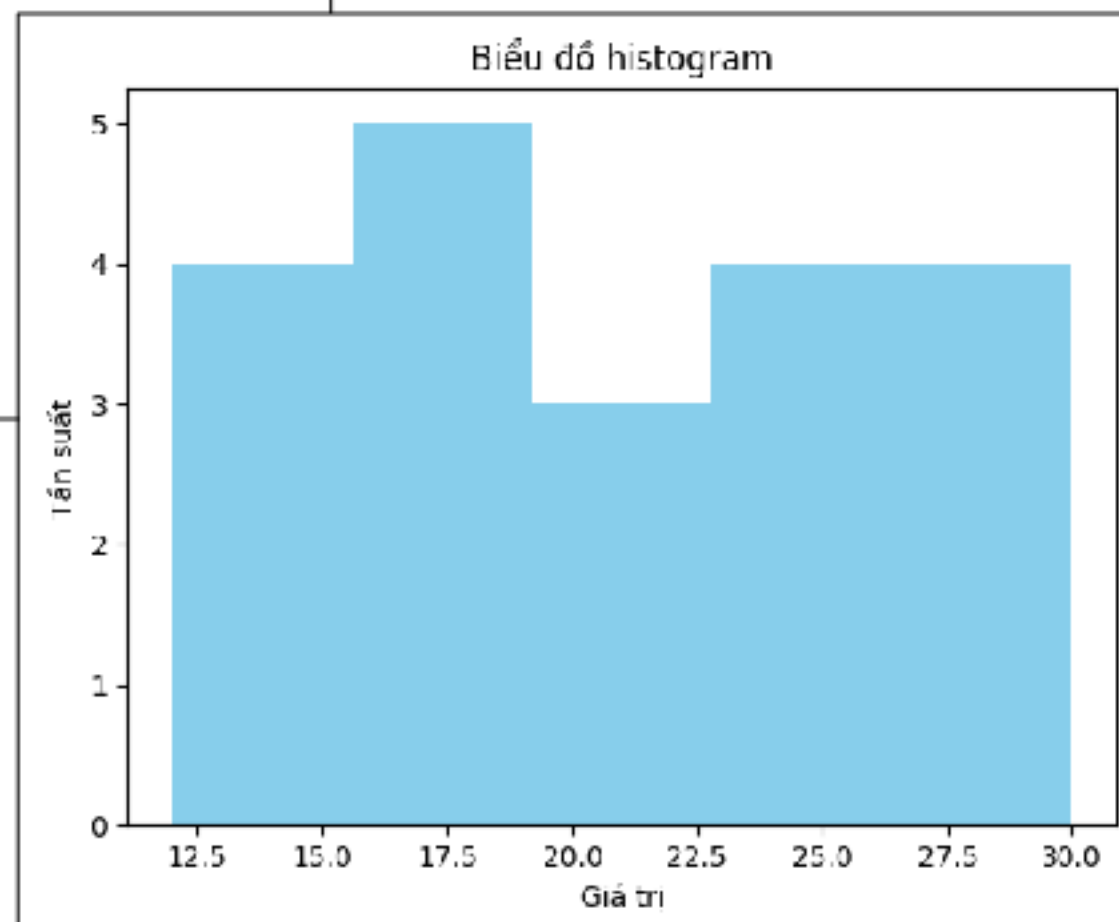
```
import matplotlib.pyplot as plt

# Dữ liệu mẫu
data = [22, 13, 18, 25, 30, 12, 15, 28, 20, 18, 16, 25, 27, 19, 23, 17, 21, 24, 29, 14]

# Vẽ biểu đồ histogram
plt.hist(data, bins=5, color='skyblue')

# Thiết lập các thông tin cho biểu đồ
plt.title('Biểu đồ histogram')
plt.xlabel('Giá trị')
plt.ylabel('Tần suất')

# Hiển thị biểu đồ
plt.show()
```



## 2-Tóm tắt mô tả về dữ liệu

### (Data Summarization & Data Description)

Ví dụ: Thống kê mô tả dữ liệu


```
import numpy as np
import pandas as pd

# Tạo dữ liệu mẫu
data = np.random.randint(0, 100, size=(100,))

# Tạo DataFrame từ dữ liệu
df = pd.DataFrame(data, columns=['Value'])

# Tóm tắt dữ liệu
summary = df.describe()

# In ra kết quả
print(summary)
```



	Value
count	100.000000
mean	51.210000
std	28.592679
min	1.000000
25%	31.000000
50%	47.000000
75%	74.500000
max	99.000000

## 2-Tóm tắt mô tả về dữ liệu

### (Data Summarization & Data Description)

Ví dụ: Tóm tắt dữ liệu (Bar Chart)

```
diem_toan = [9, 8, 7, 6, 9, 10, 5, 7, 8, 9]
from statistics import mean

diem_trung_binh = mean(diem_toan)
print("Điểm trung bình là:", diem_trung_binh)

diem_cao_nhat = max(diem_toan)
diem_thap_nhat = min(diem_toan)

print("Điểm cao nhất là:", diem_cao_nhat)
print("Điểm thấp nhất là:", diem_thap_nhat)
```

Điểm trung bình là: 7.8  
Điểm cao nhất là: 10  
Điểm thấp nhất là: 5



## 2-Tóm tắt mô tả về dữ liệu

### (Data Summarization & Data Description)

Ví dụ: Trực quan hóa dữ liệu

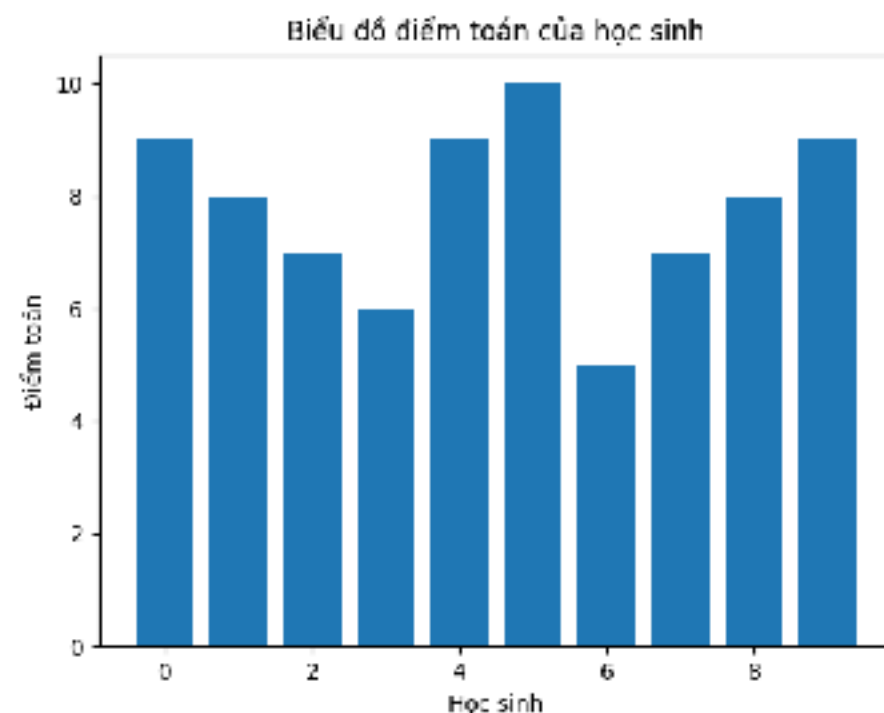
```
import matplotlib.pyplot as plt

diem_toan = [9, 8, 7, 6, 9, 10, 5, 7, 8, 9]

#Vẽ biểu đồ cột
plt.bar(range(len(diem_toan)), diem_toan)

plt.xlabel("Học sinh")
plt.ylabel("Điểm toán")

plt.title("Biểu đồ điểm toán của học sinh")
plt.show()
```



## 2-Tóm tắt mô tả về dữ liệu

### (Data Summarization & Data Description)

Ví dụ: Trực quan hóa dữ liệu (Box Plot)

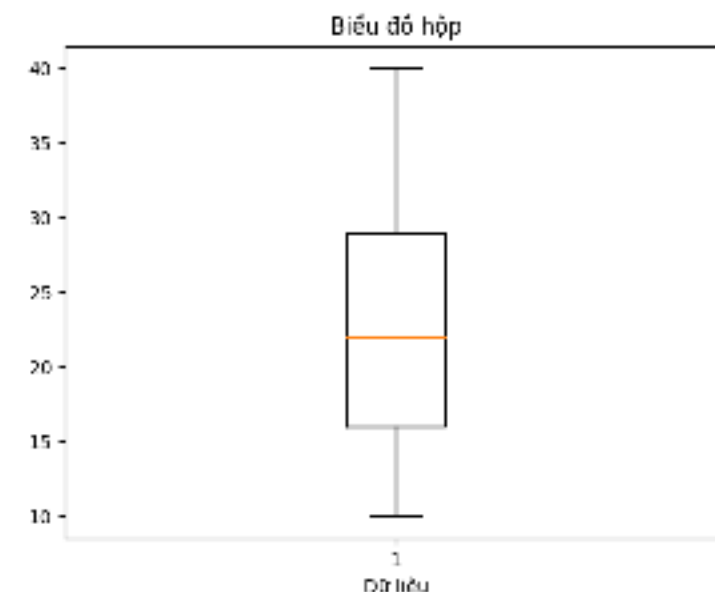
```
import matplotlib.pyplot as plt

# Dữ liệu mẫu
data = [10, 12, 15, 17, 20, 22, 25, 28, 30, 35, 40]

# Vẽ biểu đồ hộp
plt.boxplot(data)

# Đặt tiêu đề và nhãn cho biểu đồ
plt.title("Biểu đồ hộp")
plt.xlabel("Dữ liệu")

# Hiển thị biểu đồ
plt.show()
```



## 2-Tóm tắt mô tả về dữ liệu

### (Data Summarization & Data Description)

Ví dụ: Trực quan hóa dữ liệu (Scatter Plot)

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

# Đọc dữ liệu từ tệp CSV vào DataFrame
df = pd.read_csv('Vidu-phantan.csv')

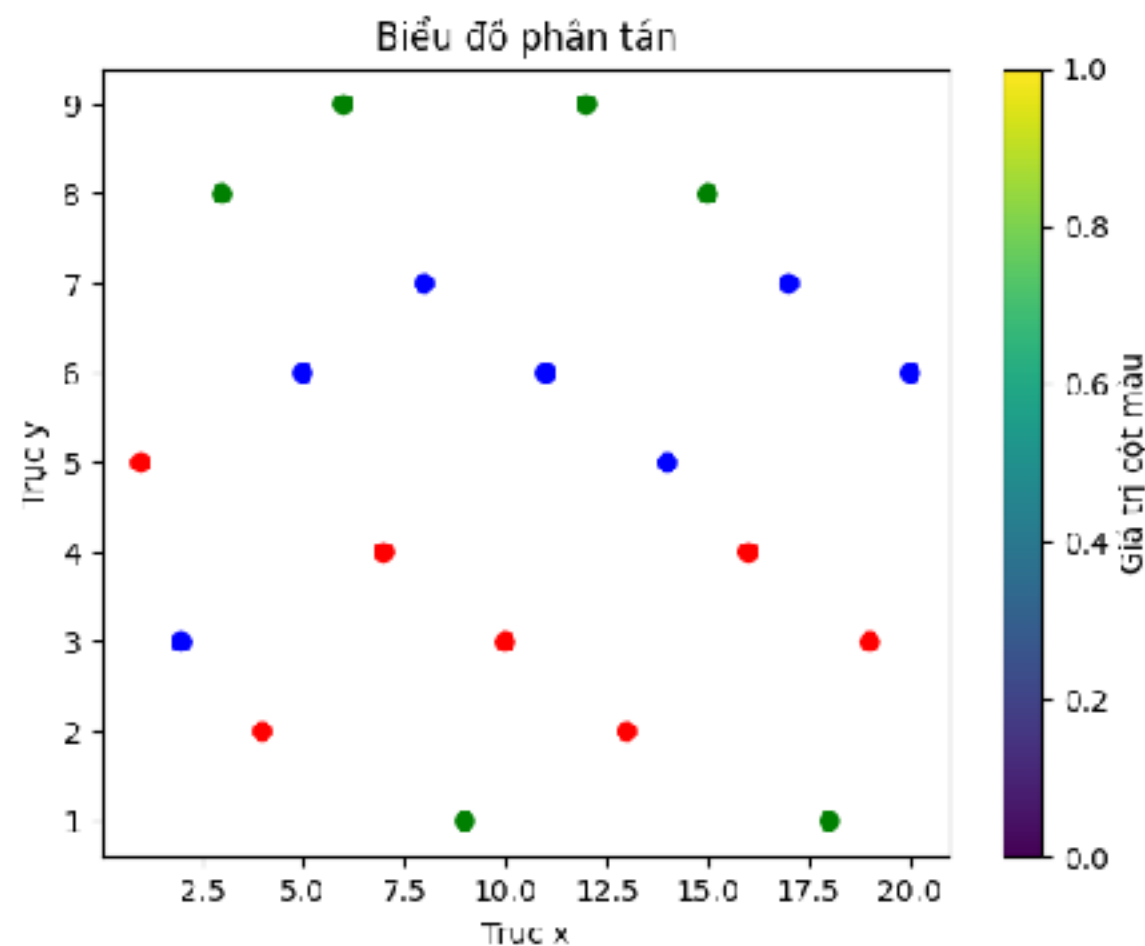
# Lấy các cột dữ liệu cần trực quan hóa
x = df['cot_x']
y = df['cot_y']
colors = df['cot_mau']

# Vẽ biểu đồ phân tán
plt.scatter(x, y, c=colors, cmap='viridis')

# Đặt tiêu đề và nhãn trục
plt.title('Biểu đồ phân tán')
plt.xlabel('Trục x')
plt.ylabel('Trục y')

# Thêm colorbar để hiển thị giá trị của cột màu
cbar = plt.colorbar()
cbar.set_label('Giá trị cột màu')

# Hiển thị biểu đồ
plt.show()
```



## 2-Tóm tắt mô tả về dữ liệu

### (Data Summarization & Data Description)

**Ví dụ:** Trực quan hóa dữ liệu (Network Graph)

```
import networkx as nx
import matplotlib.pyplot as plt

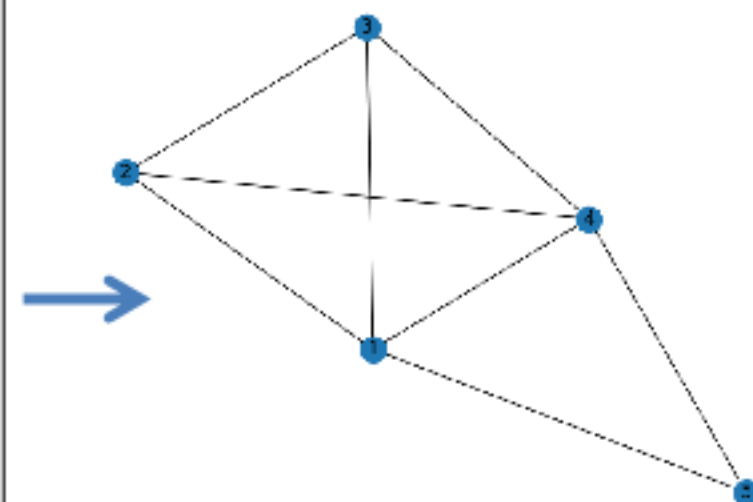
# Tạo đồ thị mạng
G = nx.Graph()

# Thêm các đỉnh vào đồ thị
G.add_nodes_from([1, 2, 3, 4, 5])

# Thêm các cạnh vào đồ thị
G.add_edges_from([(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4), (4, 5), (5, 1)])

# Vẽ đồ thị mạng
nx.draw(G, with_labels=True)

# Hiển thị đồ thị
plt.show()
```





## 2-Tóm tắt mô tả về dữ liệu

### (Data Summarization & Data Description)

**Ví dụ:** Trực quan hóa dữ liệu thiếu (Missing)

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

# Dữ liệu mẫu có dữ liệu thiếu
data = {'Tên': ['John', 'Mike', 'Sarah', 'Emily', 'David'],
        'Tuổi': [30, 25, None, 35, 28],
        'Lương': [5000, 4000, None, None, 4500]}

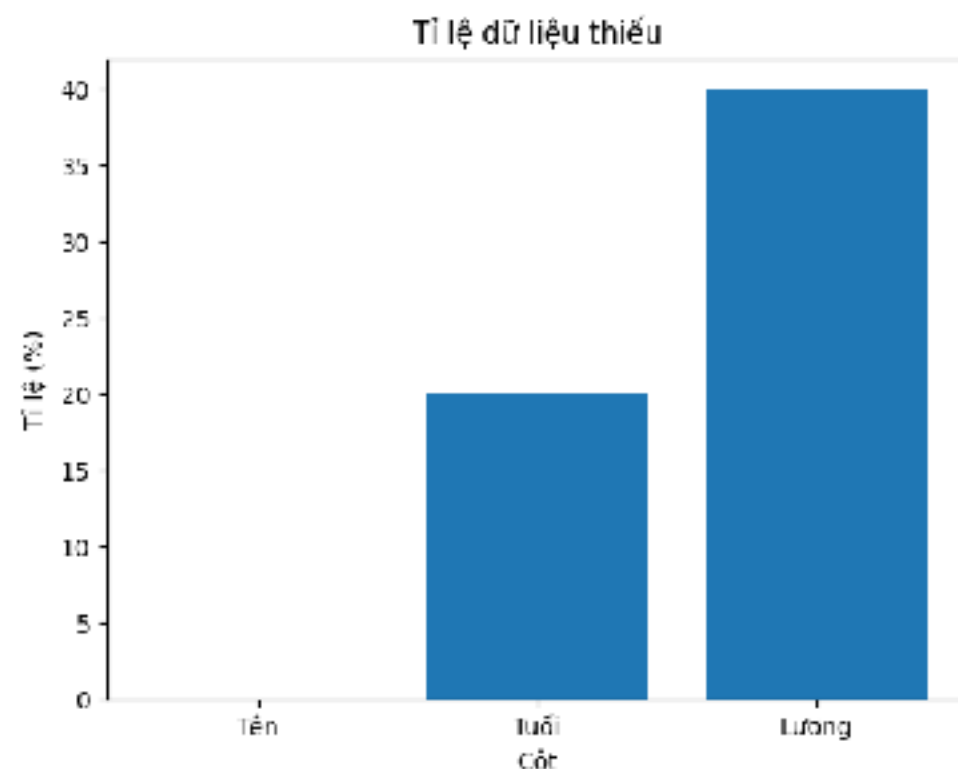
# Tạo DataFrame từ dữ liệu
df = pd.DataFrame(data)

# Tính tỷ lệ dữ liệu thiếu trong từng cột
missing_data = df.isnull().sum() / len(df) * 100

# Vẽ biểu đồ cột tỷ lệ dữ liệu thiếu
plt.bar(missing_data.index, missing_data.values)

# Đặt tiêu đề và nhãn cho biểu đồ
plt.title("Tỷ lệ dữ liệu thiếu")
plt.xlabel("Cột")
plt.ylabel("Tỷ lệ (%)")

# Hiển thị biểu đồ
plt.show()
```



## 2-Tóm tắt mô tả về dữ liệu

### (Data Summarization & Data Description)

**Ví dụ:** Loại bỏ dữ liệu thiếu (Missing)

```
import pandas as pd
import numpy as np

# Tạo một DataFrame chứa dữ liệu có nhiều
data = {'A': [1, 2, np.nan, 4, 5],
        'B': [6, np.nan, 8, np.nan, 10],
        'C': [11, 12, 13, 14, 15]}
df = pd.DataFrame(data)

summary = df.describe()
print(summary)

# Xác định và loại bỏ các giá trị NaN (nhieu)
df_cleaned = df.dropna()

# Tóm tắt và mô tả dữ liệu sau khi loại bỏ nhieu
summary = df_cleaned.describe()
print(summary)
```



	A	B	C
count	4.000000	3.0	5.000000
mean	3.000000	8.0	13.000000
std	1.825742	2.0	1.581139
min	1.000000	6.0	11.000000
25%	1.750000	7.0	12.000000
50%	3.000000	8.0	13.000000
75%	4.250000	9.0	14.000000
max	5.000000	10.0	15.000000

	A	B	C
count	2.000000	2.000000	2.000000
mean	3.000000	8.000000	13.000000
std	2.828427	2.828427	2.828427
min	1.000000	6.000000	11.000000
25%	2.000000	7.000000	12.000000
50%	3.000000	8.000000	13.000000
75%	4.000000	9.000000	14.000000
max	5.000000	10.000000	15.000000

## 3-Làm sạch dữ liệu (Data Cleaning)

Ví dụ: Loại bỏ dữ liệu trùng lặp

```
import numpy as np
# Tạo một mảng 2 chiều với dữ liệu trùng lặp
data = np.array([[1, 2, 3],
                 [4, 2, 3],
                 [5, 6, 1],
                 [4, 7, 8],
                 [1, 2, 3],
                 [9, 5, 6]])
# Chuyển mảng thành một tập hợp 2D duy nhất
unique_data = np.unique(data, axis=0)
print(unique_data)
```



```
[[1 2 3]
 [4 2 3]
 [4 7 8]
 [5 6 1]
 [9 5 6]]
```

Hàm `np.unique()` để tìm các hàng duy nhất trong mảng `data`. Tham số `axis=0` tìm các hàng duy nhất; `axis=1` tìm các cột duy nhất.

### 3-Làm sạch dữ liệu (Data Cleaning)

**Ví dụ:** Xử lý dữ liệu thiếu – thay thế bằng giá trị trung bình

```
import pandas as pd
import numpy as np
# Tạo một DataFrame với dữ liệu thiếu
data = {'A': [1, 2, np.nan, 4, 5],
        'B': [6, np.nan, 8, np.nan, 10],
        'C': [11, 12, 13, np.nan, 15]}
df = pd.DataFrame(data)
# Điền giá trị trung bình vào các ô thiếu
df_filled = df.fillna(df.mean())

print(df_filled)
```



	A	B	C
0	1.0	6.0	11.0
1	2.0	NaN	12.0
2	NaN	8.0	13.0
3	4.0	NaN	NaN
4	5.0	10.0	15.0

	A	B	C
0	1.0	6.0	11.00
1	2.0	8.0	12.00
2	3.0	8.0	13.00
3	4.0	8.0	12.75
4	5.0	10.0	15.00

Hàm **'fillna()'** điền giá trị trung bình vào các ô thiếu trong DataFrame;

Hàm **'mean()'** tính giá trị trung bình của từng cột.