2274802010480_Trần Khánh Linh

# #LAB 1: CÁC LỆNH CƠ BẢN TRONG NUMPY, PANDAS VÀ MATPLOTLIB

## 1.1 NUMPY

```python
import numpy as np

#Tạo mảng 1 chiều
mang_1D = np.array([1,2,3,4,5,6,7,8])
print(mang_1D)
#Tạo mảng 2 chiều
mang_2D = np.array([[1,3,5,7],[2,4,6,8]])
print(mang_2D)
#Tạo mảng ngẫu nhiên
mang_ND = np.random.rand(4,4)
print(mang_ND)
```

```
[1 2 3 4 5 6 7 8]
[[1 3 5 7]
 [2 4 6 8]]
[[0.06510498 0.5423376  0.66280598 0.35011642]
 [0.93297153 0.04727589 0.7971943  0.47451809]
 [0.64584772 0.89056892 0.85568721 0.78990233]
 [0.99720083 0.82065398 0.89932105 0.04202929]]
```

# 1.2 CAC THAO TAC CƠ BAN TRONG NUMPY

```
# Tạo một mảng
mang_1D = np.array([1,2,3,4,5,6,7,8])
print(mang_1D)
# Cộng và nhân mảng với một số
print(mang_1D + 5)
print(mang_1D * 5)
# Tính tổng trung bình, lớn nhất, nhỏ nhất
print( sum(mang_1D) )
print( mang_1D.mean() )
print( mang_1D.max() )
print( mang_1D.min() )

# Lấy phần tử
print(mang_1D[2])
```

[2]

```
[1 2 3 4 5 6 7 8]
[ 6  7  8  9 10 11 12 13]
[ 5 10 15 20 25 30 35 40]
36
4.5
8
1
3
```

# PANDAS

## 2.1 TẠO DATAFRAME

```python
import pandas as pd
#tạo dataframe và dictionary
data = {
    "tên": ["Anh", "Yêu", "Em"],
    "tuổi": [18, 20, 26],
    "điểm": [3, 7, 10]
}
df = pd.DataFrame(data)
print(df)
```

```
   tên  tuổi  điểm
0  Anh    18     3
1  Yêu    20     7
2   Em    26    10
```

# 2.2 THAO TÁC CƠ BẢN

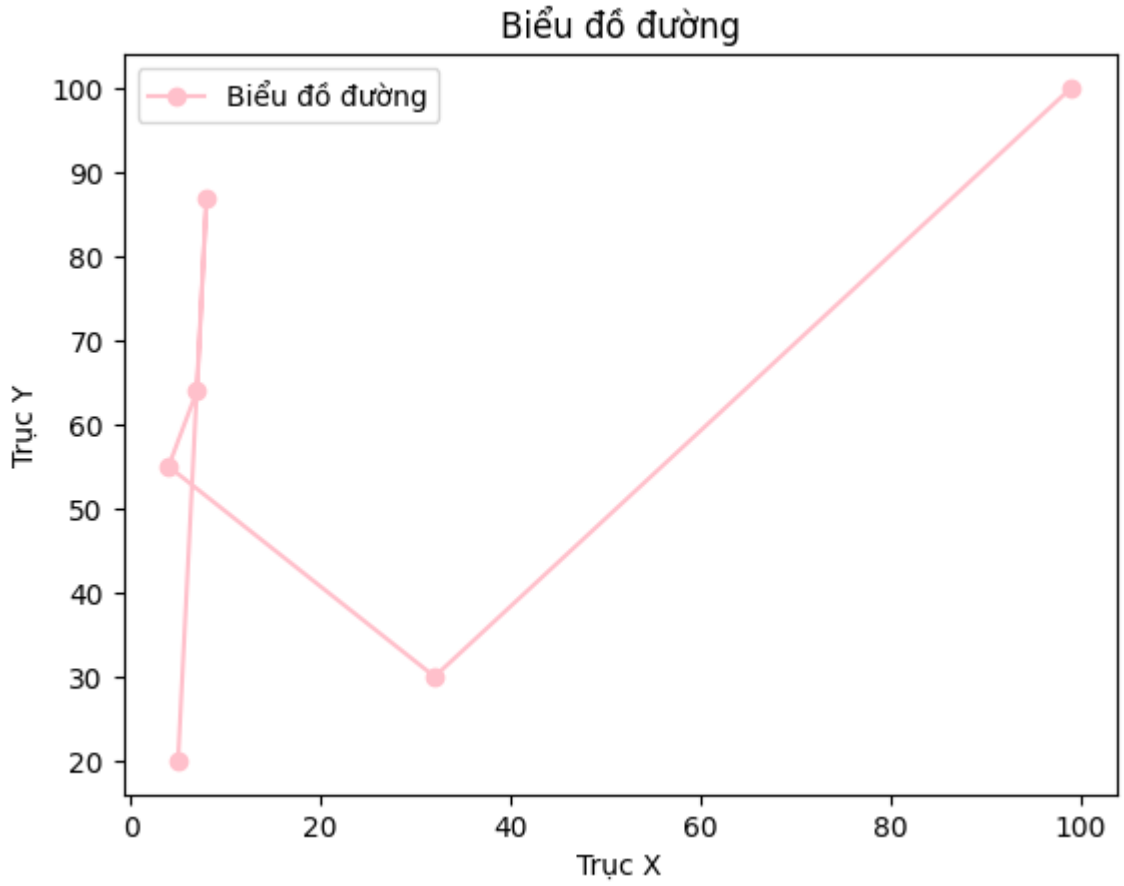
```
print(df["tuổi"])
#Lấy hàng thứ 2
print(df.iloc[1])
#Lấy ra người có tuổi lớn hơn 18
print(df[df["tuổi"] > 18])
#Tính điểm trung bình của người đó
print(df["điểm"].mean())
```

```
0    18
1    20
2    26
Name: tuổi, dtype: int64
tên     Yêu
tuổi     20
điểm      7
Name: 1, dtype: object
   tên  tuổi  điểm
1  Yêu    20     7
2   Em    26    10
6.666666666666667
```

# MATPLOTLIB

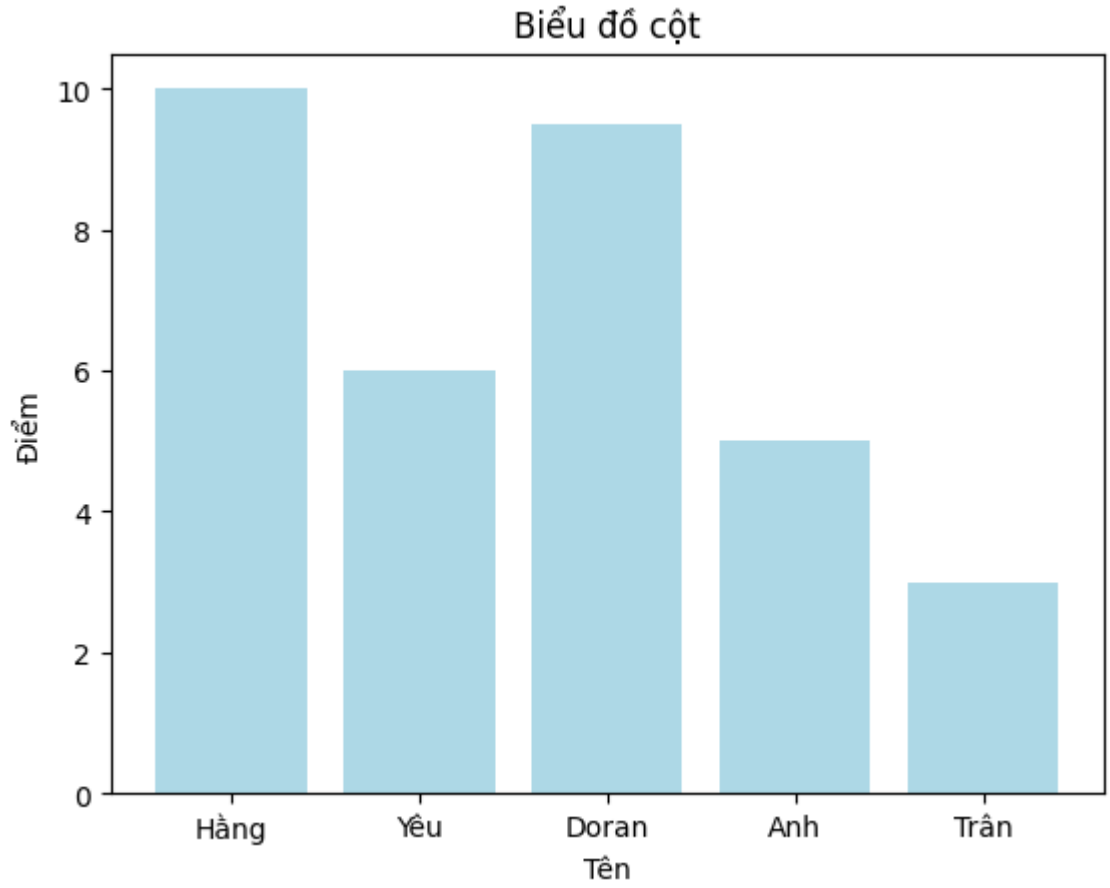
## 3.1 BIỂU ĐỒ ĐƯỜNG

```
import matplotlib.pyplot as plt

x = [5,8,7,4,32,99]
y = [20,87,64,55,30,100]

plt.plot(x,y, label = "Biểu đồ đường", color = "pink",marker = "o")
plt.title("Biểu đồ đường")
plt.xlabel("Trục X")
plt.ylabel("Trục Y")
plt.legend()
plt.show()
```



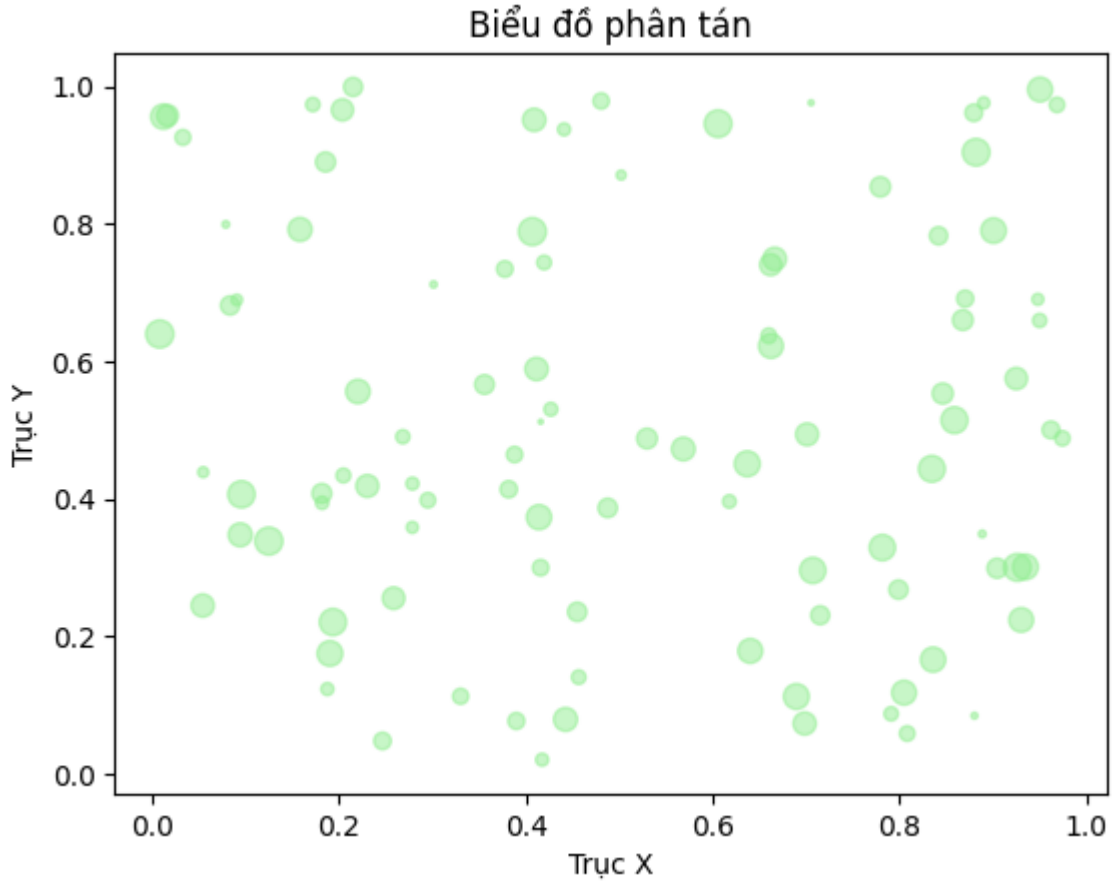
# 3.2 BIỂU ĐỒ CỘT

```python
import matplotlib.pyplot as plt
tên = ["Hằng", "Yêu", "Doran", "Anh", "Trân" ]
điểm = [10, 6, 9.5, 5, 3]
plt.bar(tên, điểm, color = "lightblue")
plt.title("Biểu đồ cột")
plt.xlabel("Tên")
plt.ylabel("Điểm")
plt.show()
```



## 3.3 Biểu đồ phân tán

```
import matplotlib.pyplot as plt
import numpy as np
x = np.random.rand(100)
y = np.random.rand(100)
sizes = np.random.rand(100) * 100

plt.scatter(x, y, s = sizes, alpha=0.5, color = "lightgreen")
plt.title("Biểu đồ phân tán")
plt.xlabel("Trục X")
plt.ylabel("Trục Y")
plt.show()
```



# Bài tập 1: Tạo mảng và thao tác cơ bản

1 Tạo một mảng NumPy với các giá trị từ 1 đến 20.
2 Tìm tổng, giá trị lớn nhất, nhỏ nhất và trung bình của mảng.
3 Tạo một mảng 2D (3x5) chứa các số ngẫu nhiên từ 0 đến 100.
4 Lấy hàng thứ 2 và cột thứ 3 của mảng 2D.

```python
import numpy as np
# 1. Tạo một mảng NumPy với các giá trị từ 1 đến 20
array_1 = np.arange(1, 21)
print("Mảng từ 1 đến 20:", array_1)

# 2. Tìm tổng, giá trị lớn nhất, nhỏ nhất và trung bình của mảng
print("Tổng:", sum(array_1) )
print("Giá trị lớn nhất:", array_1.max() )
print("Giá trị nhỏ nhất:", array_1.min() )
print("Trung bình:", array_1.mean() )

# 3. Tạo một mảng 2D (3x5) chứa các số ngẫu nhiên từ 0 đến 100
array_2d = np.array(np.random.randint(0, 100, 15)).reshape(3, 5)
print("Mảng 2D:", array_2d)

# 4. Lấy hàng thứ 2 và cột thứ 3 của mảng 2D
print("Hàng thứ 2:", array_2d[1])
print("Cột thứ 3:", array_2d[:, 2])
```

```
Mảng từ 1 đến 20: [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20]
Tổng: 210
Giá trị lớn nhất: 20
Giá trị nhỏ nhất: 1
Trung bình: 10.5
Mảng 2D: [[15 49 27 89 74]
 [ 7 80 40 26 12]
 [54  6 69 17 66]]
Hàng thứ 2: [ 7 80 40 26 12]
Cột thứ 3: [27 40 69]
```

```python
# 1. Tạo một mảng NumPy chứa 20 giá trị ngẫu nhiên từ 0 đến 1
random_array = np.random.rand(20)
print("Mảng ngẫu nhiên từ 0 đến 1:", random_array)

# 2. Chuẩn hóa mảng này (đưa các giá trị về khoảng [0, 1])
normalized_array = (random_array - random_array.min()) / (random_array.max() - randc
print("Mảng sau khi chuẩn hóa:", normalized_array)

# 3. Tính tích vô hướng (dot product) của hai mảng 1D
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
dot_product = np.dot(a, b)
print("Tích vô hướng của a và b:", dot_product)

# 4. Tạo một ma trận 5x5 và tính định thức, nghịch đảo
matrix = np.random.randint(0, 10, (5, 5))
print("Ma trận:\n", matrix)
determinant = np.linalg.det(matrix)
print("Định thức của ma trận:", determinant)
if determinant != 0:
    inverse_matrix = np.linalg.inv(matrix)
    print("Ma trận nghịch đảo:\n", inverse_matrix)
else:
    print("Ma trận không khả nghịch (định thức = 0).")
```

```
Mảng ngẫu nhiên từ 0 đến 1: [0.395553   0.39803904 0.80299579 0.99491837 0.49620087 0.2
 0.94491044 0.87137997 0.69801308 0.94686499 0.07053117 0.1764373
 0.7303673  0.45589873 0.523043   0.38247968 0.41463987 0.91702268
 0.96613052 0.74746484]
Mảng sau khi chuẩn hóa: [0.35160789 0.35429728 0.79237858 1.         0.46048852 0.21362
 0.94590153 0.86635643 0.67880852 0.94801595 0.         0.11456901
 0.71380925 0.41688976 0.48952628 0.3374652  0.37225601 0.91573261
 0.96885736 0.73230532]
Tích vô hướng của a và b: 32
Ma trận:
 [[5 6 5 6 8]
 [2 5 3 4 9]
 [0 6 4 5 4]
 [3 5 9 6 5]
 [3 0 0 3 0]]
Định thức của ma trận: -1386.0000000000002
Ma trận nghịch đảo:
 [[ 0.42424242 -0.27272727 -0.12121212 -0.09090909 -0.1010101 ]
 [ 0.48917749 -0.3961039   0.1969697  -0.22727273 -0.32395382]
 [-0.03896104 -0.02597403 -0.09090909  0.18181818 -0.0995671 ]
 [-0.42424242  0.27272727  0.12121212  0.09090909  0.43434343]
 [-0.16450216  0.27922078 -0.10606061  0.04545455  0.04256854]]
```

# Phần 2: Pandas cơ bản

## Bài tập 3: Làm quen với DataFrame

1 Tạo một DataFrame chứa thông tin sau:

| Name | Age | Score |
|---|---|---|
| Alice | 23 | 85 |
| Bob | 25 | 90 |
| Charlie | 22 | 78 |
| David | 24 | 92 |
| Eva | 21 | 88 |

2 Tính giá trị trung bình của cột "Score".
3 Lọc các hàng có "Score" lớn hơn 85.

```python
import pandas as pd

# 1. Tạo DataFrame
data = {
    "Name": ["Alice", "Bob", "Charlie", "David", "Eve"],
    "Age": [23, 25, 22, 24, 21],
    "Score": [85, 90, 78, 92, 88]
}
df = pd.DataFrame(data)
print("DataFrame:", df)

# 2. Tính giá trị trung bình của cột "Score"
print("Trung bình của cột Score:", df["Score"].mean())

# 3. Lọc các hàng có "Score" lớn hơn 85
filtered_df = df[df["Score"] > 85]
print("Các hàng có Score > 85:", filtered_df)
```

```
DataFrame:       Name  Age  Score
0    Alice   23     85
1      Bob   25     90
2  Charlie   22     78
3    David   24     92
4      Eve   21     88
Trung bình của cột Score: 86.6
Các hàng có Score > 85:     Name  Age  Score
1    Bob   25     90
3  David   24     92
4    Eve   21     88
```

## Bài tập 4: Đọc và phân tích dữ liệu từ file

1 Tải file Iris.csv từ Kaggle Iris Dataset.
2 Đọc dữ liệu từ file CSV vào DataFrame.
3 Hiển thị thông tin cơ bản (tổng quan, kiểu dữ liệu, số lượng null).
4 Tính trung bình, lớn nhất, nhỏ nhất của cột sepal_length.

```python
import pandas as pd
# Đọc file CSV
iris_df = pd.read_csv("Iris.csv")
print("Thông tin tổng quan về dữ liệu:", iris_df.info())
print("Mô tả dữ liệu:", iris_df.describe())

# Tính toán cơ bản
print("Trung bình sepal_length:", iris_df["SepalLengthCm"].mean() )
print("Giá trị lớn nhất sepal_length:", iris_df["SepalLengthCm"].max() )
print("Giá trị nhỏ nhất sepal_length:", iris_df["SepalLengthCm"].min() )
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             150 non-null    int64
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
Thông tin tổng quan về dữ liệu: None
Mô tả dữ liệu:                Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidt
count  150.000000     150.000000    150.000000     150.000000    150.000000
mean    75.500000       5.843333      3.054000       3.758667      1.198667
std     43.445368       0.828066      0.433594       1.764420      0.763161
min      1.000000       4.300000      2.000000       1.000000      0.100000
25%     38.250000       5.100000      2.800000       1.600000      0.300000
50%     75.500000       5.800000      3.000000       4.350000      1.300000
75%    112.750000       6.400000      3.300000       5.100000      1.800000
max    150.000000       7.900000      4.400000       6.900000      2.500000
Trung bình sepal_length: 5.843333333333334
Giá trị lớn nhất sepal_length: 7.9
Giá trị nhỏ nhất sepal_length: 4.3
```

# Phần 3: Làm sạch dữ liệu

## Bài tập 5: Xử lý dữ liệu thiếu

1 Tạo một DataFrame chứa các giá trị sau:

| Name | Age | City | Salary |
|---|---|---|---|
| Alice | 23 | New York | 60000 |
| Bob | NaN | Boston | 52000 |
| Charlie | 25 | NaN | NaN |
| David | 24 | Chicago | 58000 |
| Eva | 22 | Boston | NaN |

2 Điền giá trị thiếu trong cột Age bằng giá trị trung bình.
3 Xóa các hàng có nhiều hơn 1 giá trị thiếu.
4 Điền giá trị thiếu trong cột Salary bằng 50000.

```python
import pandas as pd

# Tạo DataFrame chứa dữ liệu thiếu
data_with_missing = {
    "Name": ["Alice", "Bob", "Charlie", "David", "Eva"],
    "Age": [23, None, 25, 24, 22],
    "City": ["New York", "Boston", None, "Chicago", "Boston"],
    "Salary": [60000, 52000, None, 58000, None]
}

df_missing = pd.DataFrame(data_with_missing)
print("Dữ liệu ban đầu:")
print(df_missing)

# 1. Điền giá trị thiếu trong cột Age bằng giá trị trung bình
df_missing["Age"] = df_missing["Age"].fillna(df_missing["Age"].mean())

# 2. Xóa các hàng có nhiều hơn 1 giá trị thiếu
df_missing = df_missing.dropna(thresh=df_missing.shape[1]-1)

# 3. Điền giá trị thiếu trong cột Salary bằng 50000
df_missing["Salary"] = df_missing["Salary"].fillna(50000)

print("\nDữ liệu sau khi xử lý:")
print(df_missing)
```

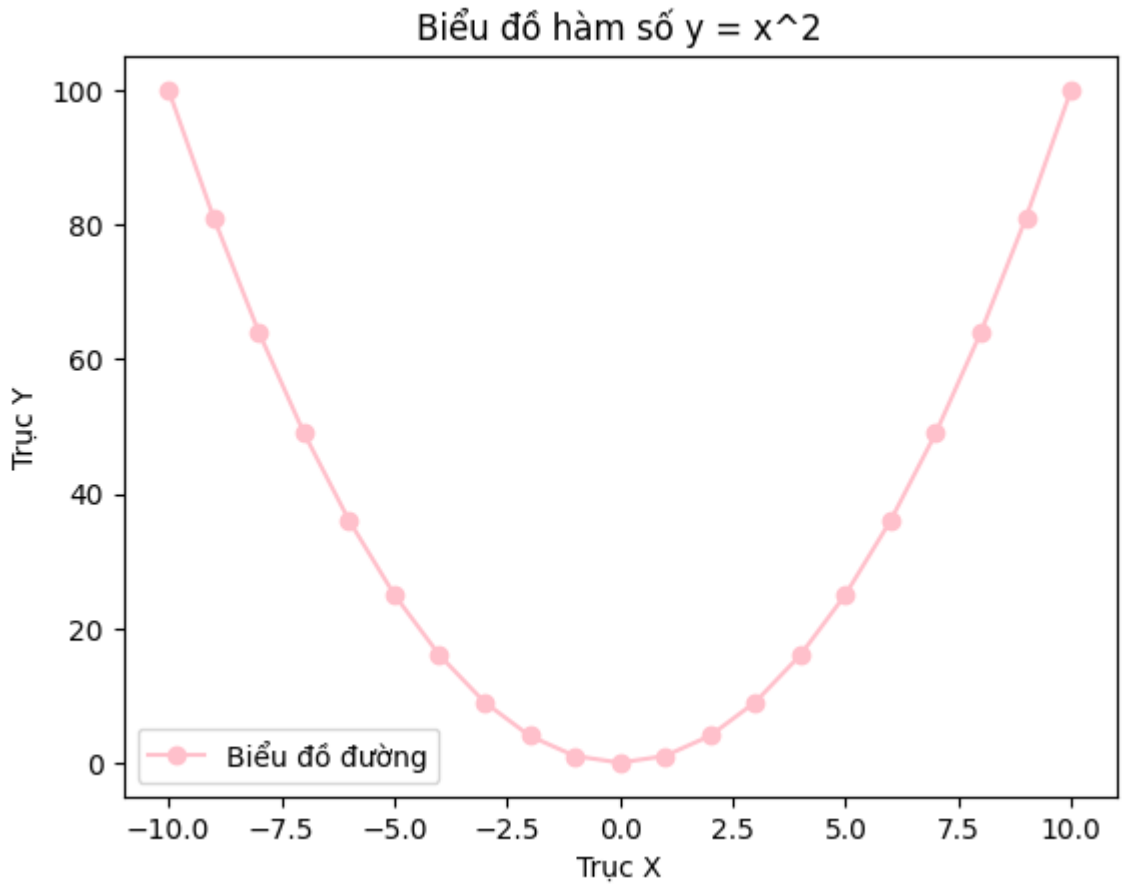
```
Dữ liệu ban đầu:
      Name   Age      City   Salary
0    Alice  23.0  New York  60000.0
1      Bob   NaN    Boston  52000.0
2  Charlie  25.0      None      NaN
3    David  24.0   Chicago  58000.0
4      Eva  22.0    Boston      NaN

Dữ liệu sau khi xử lý:
    Name   Age      City   Salary
0  Alice  23.0  New York  60000.0
1    Bob  23.5    Boston  52000.0
3  David  24.0   Chicago  58000.0
4    Eva  22.0    Boston  50000.0
```

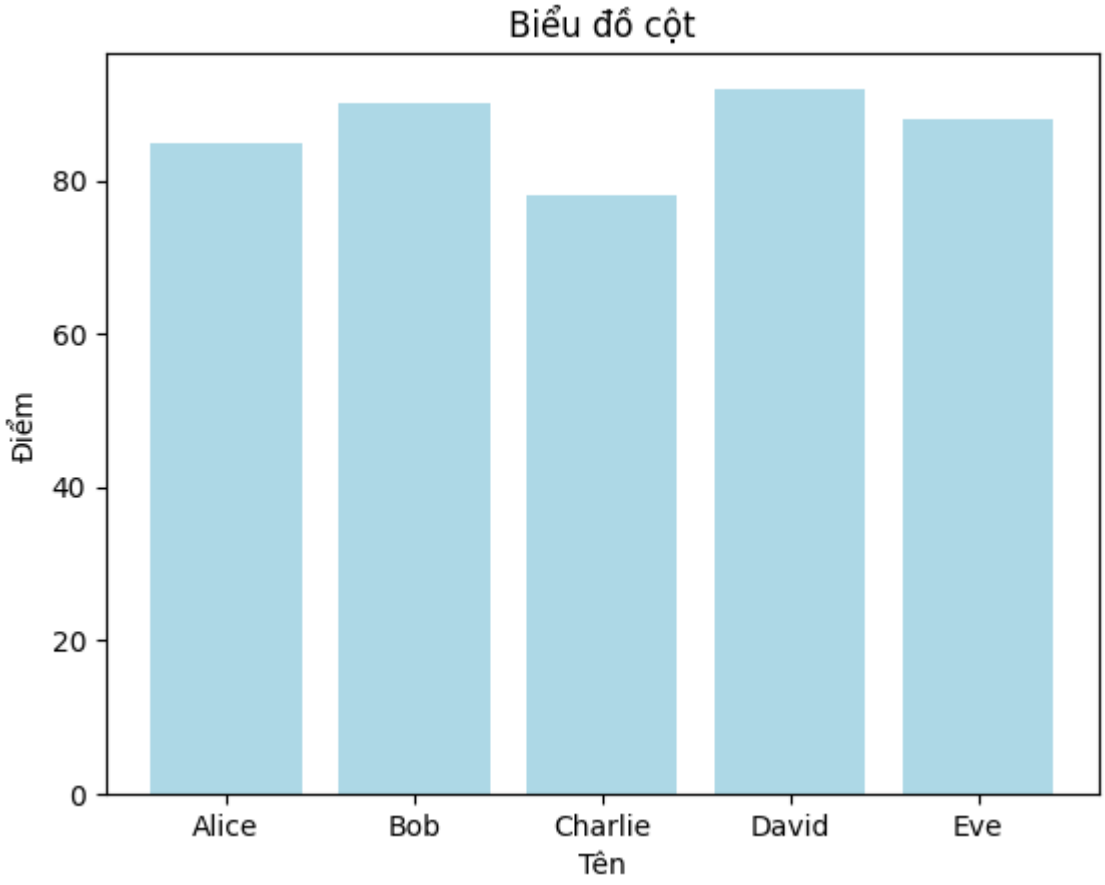
# Phần 4: Trực quan hóa dữ liệu với Matplotlib

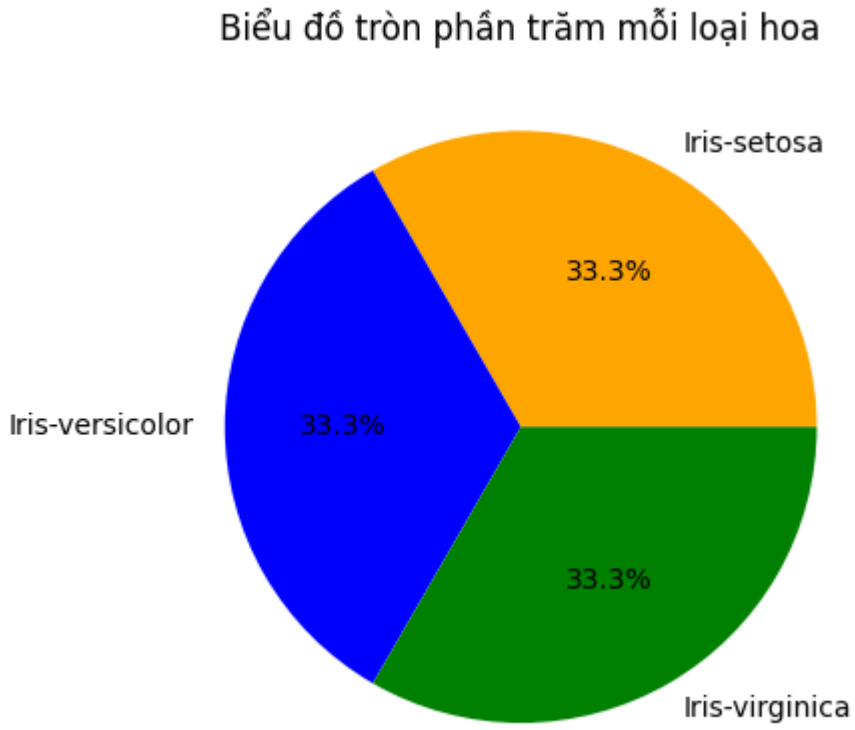
## Bài tập 6: Biểu đồ cơ bản

1 Tạo một biểu đồ đường biểu diễn hàm số y = x ** 2 trên khoảng [-10,10]
2 Vẽ biểu đồ cột thể hiện điểm số (Score) của các sinh viên từ Bài tập 3.
3 Tạo một biểu đồ tròn (pie chart) thể hiện phần trăm mỗi loại hoa trong tập dữ liệu Iris.

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
# 1. Biểu đồ đường hàm số y = x^2
#Code here
# Tạo một biểu đồ đường biểu diễn hàm số y = x ** 2 trên khoảng [-10,10]
x = np.array(range(-10, 11))
y = x**2
plt.plot(x,y, label = "Biểu đồ đường", color = "pink",marker = "o")
plt.title("Biểu đồ hàm số y = x^2")
plt.xlabel("Trục X")
plt.ylabel("Trục Y")
plt.legend()
plt.show()
# 2. Biểu đồ cột điểm số
#Code here
x = ["Alice", "Bob", "Charlie", "David", "Eve"]
y = [85, 90, 78, 92, 88]
plt.bar(x, y, color = "lightblue")
plt.title("Biểu đồ cột")
plt.xlabel("Tên")
plt.ylabel("Điểm")
plt.show()
# 3. Biểu đồ tròn phần trăm mỗi loại hoa
iris_df = pd.read_csv("Iris.csv")
species_counts = iris_df["Species"].value_counts()
plt.pie(species_counts, labels=species_counts.index, autopct='%1.1f%%', colors=["ora
plt.title("Biểu đồ tròn phần trăm mỗi loại hoa")
plt.show()
```


Biểu đồ hàm số y = x^2
Trục Y
Trục X
Biểu đồ đường
100
80
60
40
20
0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

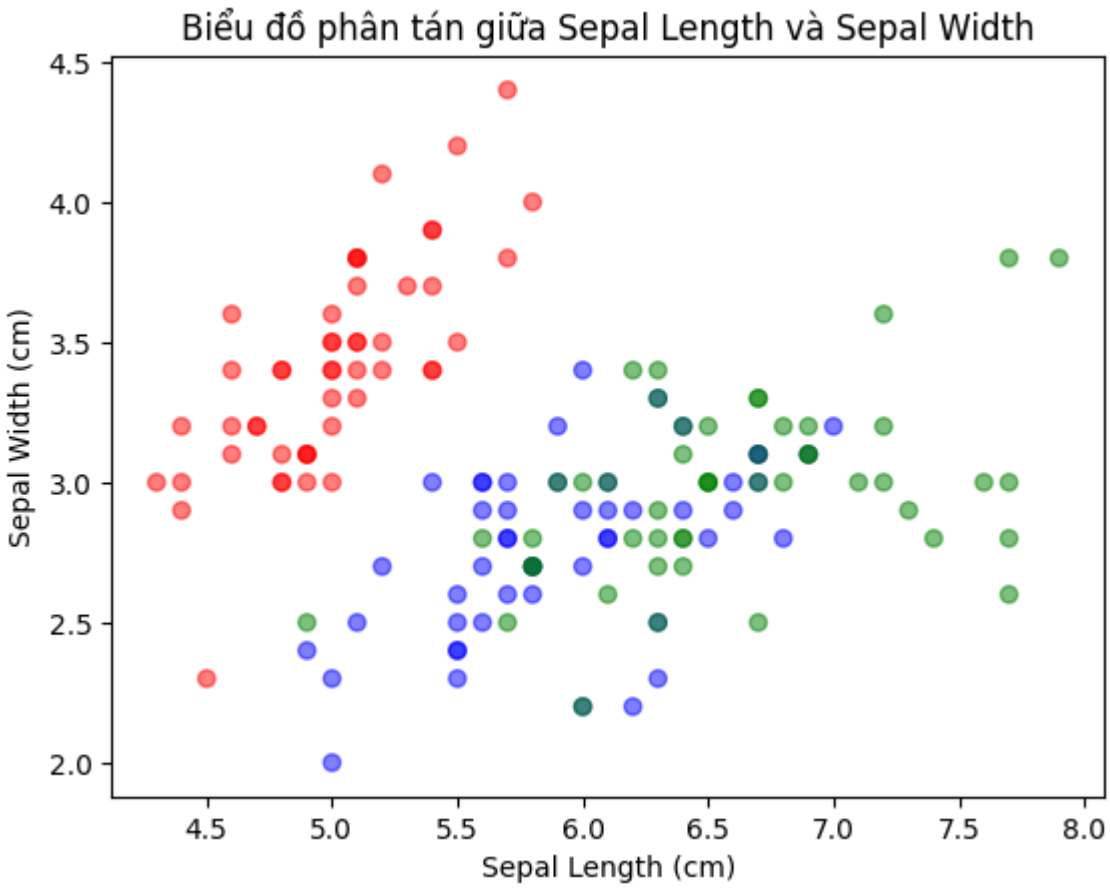

Biểu đồ cột
Điểm
Tên
80
60
40
20
0
Alice
Bob
Charlie
David
Eve




## Bài tập 7: Biểu đồ nâng cao

1 Vẽ biểu đồ phân tán (scatter plot) giữa sepal_length và sepal_width của tập dữ liệu Iris. Dùng màu sắc để phân biệt các loại hoa (species).
2 Thêm tiêu đề, nhãn trục và chú thích cho biểu đồ.

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

# Đọc file CSV
iris_df = pd.read_csv("Iris.csv")

# Lấy dữ liệu cho biểu đồ phân tán
x = iris_df["SepalLengthCm"]
y = iris_df["SepalWidthCm"]
colors = {"Iris-setosa": "red", "Iris-versicolor": "blue", "Iris-virginica": "green"
species = iris_df["Species"].map(colors)

# Vẽ biểu đồ phân tán
plt.scatter(x, y, c=species, alpha=0.5)

# Thêm tiêu đề, nhãn trục và chú thích cho biểu đồ
plt.title("Biểu đồ phân tán giữa Sepal Length và Sepal Width")
plt.xlabel("Sepal Length (cm)")
plt.ylabel("Sepal Width (cm)")
plt.show()
```


Biểu đồ phân tán giữa Sepal Length và Sepal Width
Sepal Width (cm)
Sepal Length (cm)
4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0