LAB: Dữ liệu với NumPy, Pandas và Matplotlib

Mục tiêu

- Làm quen với NumPy: Tạo mảng, thao tác cơ bản và nâng cao với mảng.
- Thực hành xử lý dữ liệu với Pandas: Đọc, phân tích và làm sạch dữ liệu.
- Trực quan hóa dữ liệu bằng Matplotlib: Biểu đồ cơ bản và nâng cao.

Phần 1: NumPy cơ bản

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.

In [ ]:
```
import numpy as np

# 1. Tạo một mảng NumPy với các giá trị từ 1 đến 20
array_1 = np.array(range(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:", array_1.sum())
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.random.randint(0, 101, (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: [[83 82 33 77  6]
 [16 39 41 23  9]
 [83 98 43 82 19]]
Hàng thứ 2: [16 39 41 23  9]
Cột thứ 3: [33 41 43]
```

Bài tập 2: Các thao tác nâng cao

1 Tạo một mảng NumPy chứa 20 giá trị ngẫu nhiên từ 0 đến 1.

2 Chuẩn hóa mảng này (đưa các giá trị về khoảng [0, 1]).

3 Tính tích vô hướng (dot product) của hai mảng 1D: [1, 2, 3] và [4, 5,6].

4 Tạo một ma trận 5x5 và tính định thức (determinant) và nghịch đảo của ma trận.

In [ ]:
```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 - np.min(random_array)) / (np.max(random_array) - np.min(random_array))
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.rand(5, 5)
print("Ma trận:", 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:", inverse_matrix)
else:
    print("Ma trận không khả nghịch (định thức = 0).")
```

```
Mảng ngẫu nhiên từ 0 đến 1: [0.62164722 0.60992175 0.63028242 0.32880919 0.70995204 0.90375438
 0.2043119  0.70153679 0.92004243 0.76792936 0.69753979 0.59781975
 0.02404331 0.18784227 0.31065684 0.68106858 0.63646669 0.43925139
 0.99596836 0.26831667]
Mảng sau khi chuẩn hóa: [0.61486625 0.60280208 0.62375089 0.31356933 0.70572184 0.90512233
 0.18547582 0.6970635  0.92188088 0.76537389 0.69295105 0.5903505
 0.         0.16853044 0.29489263 0.67600405 0.63011379 0.42720174
 1.         0.25132942]
Tích vô hướng của a và b: 32
Ma trận: [[0.94952627 0.6581323  0.18284102 0.11017888 0.16755453]
 [0.87425319 0.2468805  0.43306513 0.3538349  0.28919346]
 [0.07325185 0.02193019 0.18490415 0.38913363 0.71691777]
 [0.8557294  0.1326096  0.38131699 0.59838375 0.89407155]
 [0.7881544  0.94031743 0.18341757 0.28548011 0.79658292]]
Định thức của ma trận: -0.0012073802421505562
Ma trận nghịch đảo: [[ -6.15469795   0.64852858  -8.76926862   4.69703122   3.67954132]
 [ 16.63797172  -3.64876572  17.49165434  -8.03566958  -8.89823562]
 [-40.60057577  15.76359151 -35.91470758  10.21510796  23.67479492]
 [ 80.22217462 -21.99574088  79.82682898 -29.03890113 -48.13934597]
 [-32.95212275   7.91867849 -32.31019902  12.89318696  19.91954761]]
```

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.

In [ ]:
```
import pandas as pd
```

```python
# 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"


# 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
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.

In [ ]:
```python
# 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  PetalWidthCm
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.

In [ ]:

```python
import pandas as pd
import numpy as np

# Tạo DataFrame chứa dữ liệu thiếu
data_with_missing = {
    "Name": ["Alice", "Bob", "Charlie", "David", "Eve"],
    "Age": [23, None, 25, 24, 21],
    "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:\n", 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())
print("\nSau khi điền giá trị thiếu trong cột Age:\n", df_missing)

# 2. Xóa các hàng có nhiều hơn 1 giá trị thiếu
df_missing = df_missing.dropna(thresh=df_missing.shape[1] - 1)
print("\nSau khi xóa các hàng có nhiều hơn 1 giá trị thiếu:\n", df_missing)

# 3. Điền giá trị thiếu trong cột Salary bằng 50000
df_missing['Salary'] = df_missing['Salary'].fillna(50000)
print("\nSau khi điền giá trị thiếu trong cột Salary bằng 50000:\n", 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      Eve  21.0    Boston      NaN

Sau khi điền giá trị thiếu trong cột Age:
      Name    Age      City   Salary
0    Alice  23.00  New York  60000.0
1      Bob  23.25    Boston  52000.0
2  Charlie  25.00      None      NaN
3    David  24.00   Chicago  58000.0
4      Eve  21.00    Boston      NaN

Sau khi xóa các hàng có nhiều hơn 1 giá trị thiếu:
     Name    Age      City   Salary
0  Alice  23.00  New York  60000.0
1    Bob  23.25    Boston  52000.0
3  David  24.00   Chicago  58000.0
4    Eve  21.00    Boston      NaN

Sau khi điền giá trị thiếu trong cột Salary bằng 50000:
     Name    Age      City   Salary
0  Alice  23.00  New York  60000.0
1    Bob  23.25    Boston  52000.0
3  David  24.00   Chicago  58000.0
4    Eve  21.00    Boston  50000.0
```

Phần 4: Trực quan hóa dữ liệu với Matplotlib

Bài tập 6: Biểu đồ cơ bản

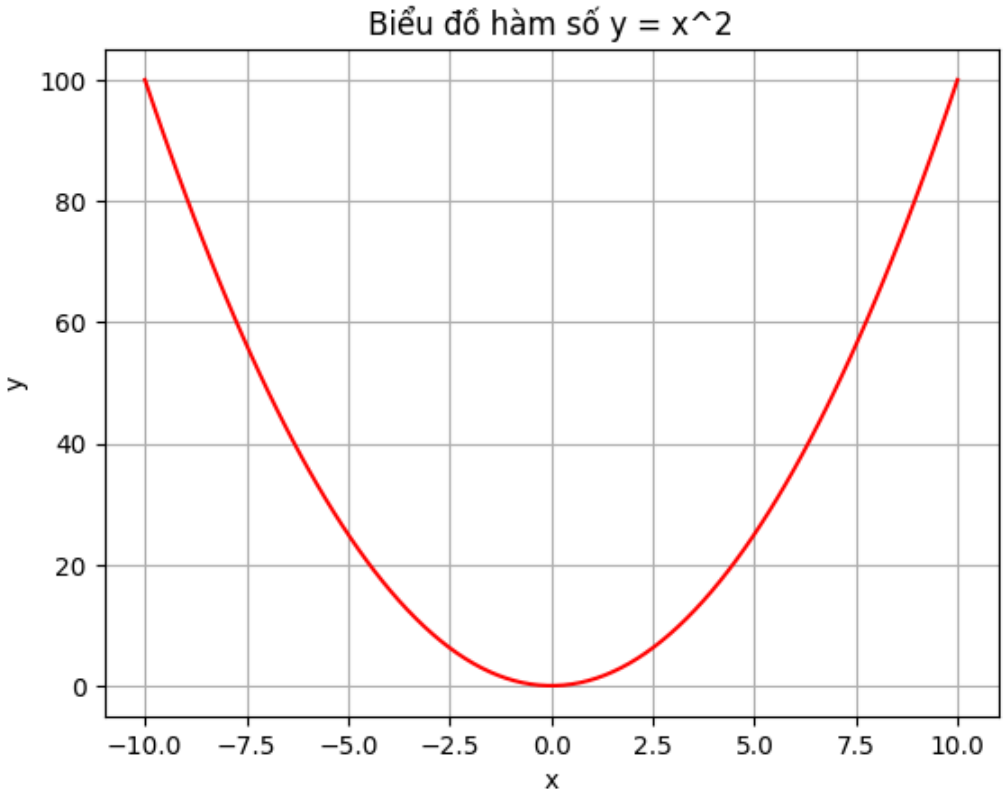
1 Tạo một biểu đồ đường biểu diễn hàm số y = x ** 2 trên khoảng [-10,10]

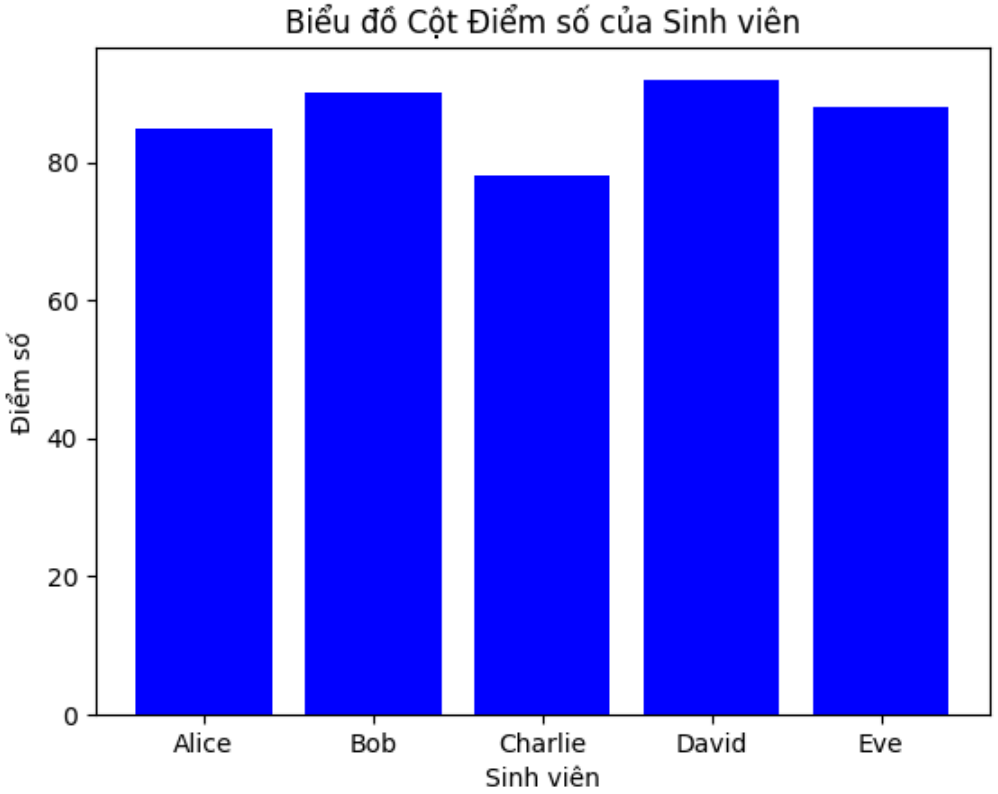
2 Vẽ biểu đồ cột thể hiện điểm số (Score) của các sinh viên từ Bài tập 3.

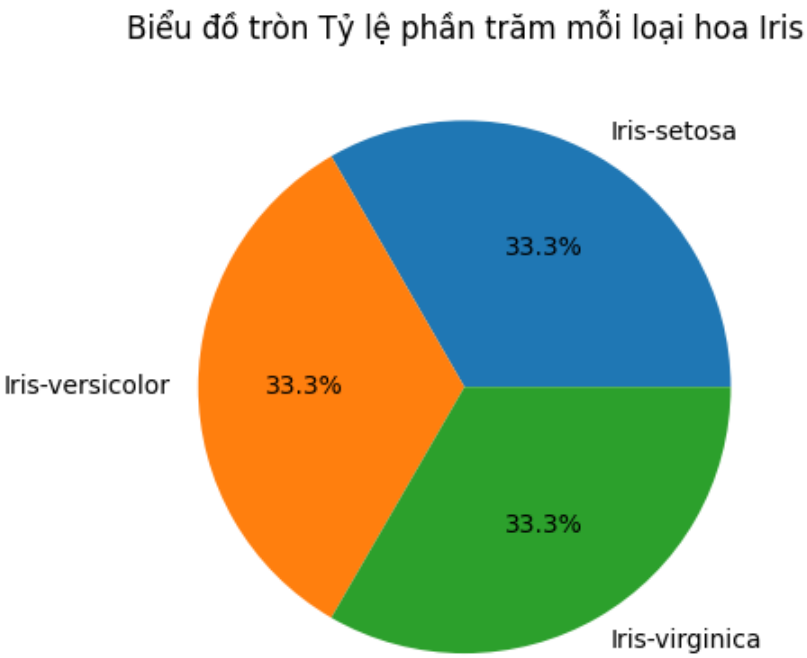
3 Tạo một biểu đồ tròn (pie chart) thể hiện phần trăm mỗi loại hoa trong tập dữ liệu Iris.

In [ ]:
```python
import matplotlib.pyplot as plt
import numpy as np
# 1. Biểu đồ đường hàm số y = x^2
x = np.linspace(-10, 10, 100)
y = x**2
```

```
# Vẽ biểu đồ đường
plt.plot(x, y, label='Biểu đồ đường', color='red')
plt.title('Biểu đồ hàm số y = x^2')
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.show()

# 2. Biểu đồ cột điểm số
Name = ["Alice", "Bob", "Charlie", "David", "Eve"]
Score = [85, 90, 78, 92, 88]
plt.bar(Name, Score, color='blue')
plt.title('Biểu đồ Cột Điểm số của Sinh viên')
plt.xlabel('Sinh viên')
plt.ylabel('Điểm số')
plt.show()


# Đọc dữ liệu Iris
iris_df = pd.read_csv('Iris.csv')

# Tính số lượng mỗi loại hoa
flower_counts = iris_df['Species'].value_counts()

# Vẽ biểu đồ tròn
plt.pie(flower_counts, labels=flower_counts.index, autopct='%1.1f%%')
plt.title('Biểu đồ tròn Tỷ lệ phần trăm mỗi loại hoa Iris')
plt.show()
```


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


Biểu đồ Cột Điểm số của Sinh viên
Điểm số
0
20
40
60
80
Alice
Bob
Charlie
David
Eve
Sinh viên

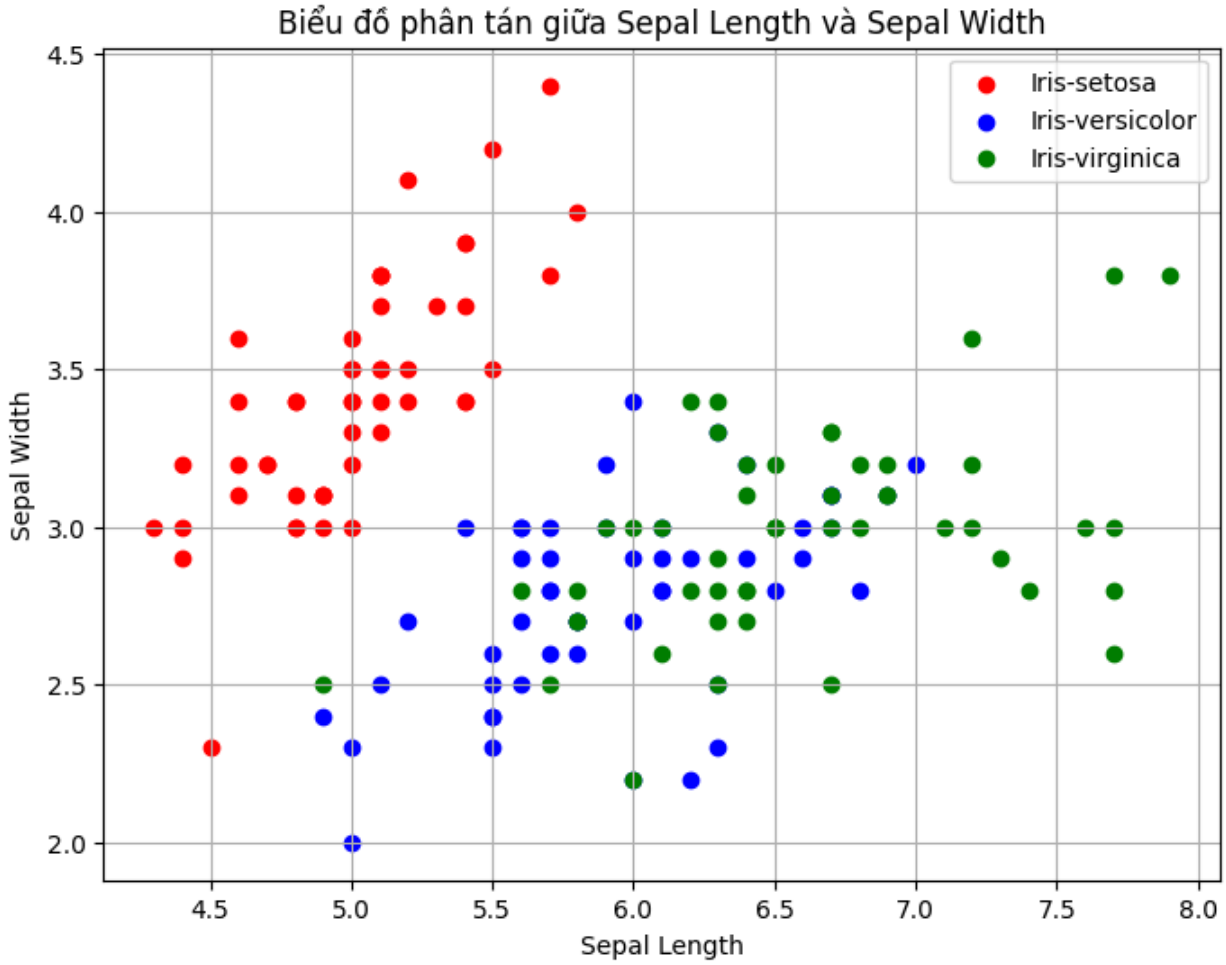


Bài tập 7: Biểu đồ nâng cao

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.

In [ ]:
```
# 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())
# x = iris_df['SepalLengthCm']
# y = iris_df['SepalWidthCm']
# sizes= np.random.rand(100)*100

# plt.scatter(x, y, s=sizes, alpha=0.5, c='green')
# plt.title("Biểu đồ phân tán")
# plt.xlabel("Trục x")
# plt.ylabel("Trục y")
# plt.show()
# 1. Biểu đồ phân tán với màu sắc theo loại hoa
#Code here
import pandas as pd
```

```python
import matplotlib.pyplot as plt

# Đọc dữ liệu Iris
iris_df = pd.read_csv('Iris.csv')

# Vẽ biểu đồ phân tán (scatter plot) giữa sepal_length và sepal_width
colors = {'Iris-setosa': 'red', 'Iris-versicolor': 'blue', 'Iris-virginica': 'green'}
species = iris_df['Species']

plt.figure(figsize=(8, 6))
for species_name in colors.keys():
    subset = iris_df[iris_df['Species'] == species_name]
    plt.scatter(subset['SepalLengthCm'], subset['SepalWidthCm'], label=species_name, color=colors[species_name])

plt.title('Biểu đồ phân tán giữa Sepal Length và Sepal Width')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.legend()
plt.grid(True)
plt.show()
```


Biểu đồ phân tán giữa Sepal Length và Sepal Width
Iris-setosa
Iris-versicolor
Iris-virginica
Sepal Width
Sepal Length
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