Phần 1

## 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.

```
import numpy as np

# 1. Tạo một mảng NumPy với các giá trị từ 1 đến 20
array_1 = np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20])
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:", max(array_1))
print("Giá trị nhỏ nhất:", min(array_1))
print("Trung bình:", np.mean(array_1))

# 3. Tạo một mảng 2D (3x5) chứa các số ngẫu nhiên từ 0 đến 100
array_2d = np.array([[ 8, 19, 44,  2, 81],
 [53  ,7 ,75 ,71, 75],
 [93 ,48 ,24 ,51 ,35]])
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: [[ 8 19 44  2 81]
 [53  7 75 71 75]
 [93 48 24 51 35]]
Hàng thứ 2: [53  7 75 71 75]
Cột thứ 3: [44 75 24]
```

## 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.

```
# 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
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: [0.59059052 0.64893076 0.48041253 0.63382547 0.32204471 0.66432244
 0.85019763 0.45527353 0.0780647  0.63623389 0.14221135 0.35266629
 0.75753178 0.5087066  0.4556592  0.98506639 0.80915564 0.6901029
 0.74227692 0.41740433]
Mảng sau khi chuẩn hóa: [0.59059052 0.64893076 0.48041253 0.63382547 0.32204471 0.66432244
 0.85019763 0.45527353 0.0780647  0.63623389 0.14221135 0.35266629
 0.75753178 0.5087066  0.4556592  0.98506639 0.80915564 0.6901029
 0.74227692 0.41740433]
Tích vô hướng của a và b: 32
Ma trận: [[0.71493648 0.99426125 0.11534178 0.73934036 0.52450174]
 [0.17467094 0.25986625 0.87836928 0.5497288  0.268465  ]
 [0.55014021 0.77788829 0.58763615 0.51975882 0.38926046]
 [0.50115119 0.97304396 0.39185575 0.33668906 0.41413107]
 [0.57237082 0.03005644 0.30513613 0.69647775 0.02929062]]
Định thức của ma trận: 0.004794861372118357
Ma trận nghịch đảo: [[-7.72442111e-01 -5.43603357e+00  1.66593690e+01 -1.10153953e+01
  -1.99668831e+00]
 [-9.08019283e-01  3.79916109e+00 -1.70936583e+01  1.44914248e+01
   3.71627444e+00]
 [-1.31214367e+00  1.67808991e-01  1.83781136e+00 -1.72421367e-01
  -2.77035901e-02]
 [ 1.10539623e+00  4.48881355e+00 -1.50406659e+01  9.59365498e+00
   3.30598635e+00]
 [ 3.41111620e+00 -6.15642638e+00  3.04924944e+01 -2.59408922e+01
  -8.97708387e+00]]
```

Phần 2

## 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', 'Eva'],
    '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"
average_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      Eva   21     88
Các hàng có Score > 85:     Name  Age  Score
1    Bob   25     90
3  David   24     92
4    Eva   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
# Đọc file CSV
iris_df = pd.read_csv(r'D:\dhvl\hocky_3\3.2\sohoa\lab1\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("\nTrung 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

# 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
# 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:", df_missing)

# 1. Điền giá trị thiếu trong cột Age bằng giá trị trung bình
avr = df_missing['Age'].mean()
df_missing['Age'].fillna(avr,inplace=True)
print("1")
print(df_missing)
# 2. Xóa các hàng có nhiều hơn 1 giá trị thiếu
df_delete=df_missing.dropna(thresh=3)
print("2")
print(df_delete)
# 3. Điền giá trị thiếu trong cột Salary bằng 50000

df_delete['Salary'].fillna(5000,inplace=True)
print("3")
print(df_delete)
```

```
     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   5000.0
```

Phần 4

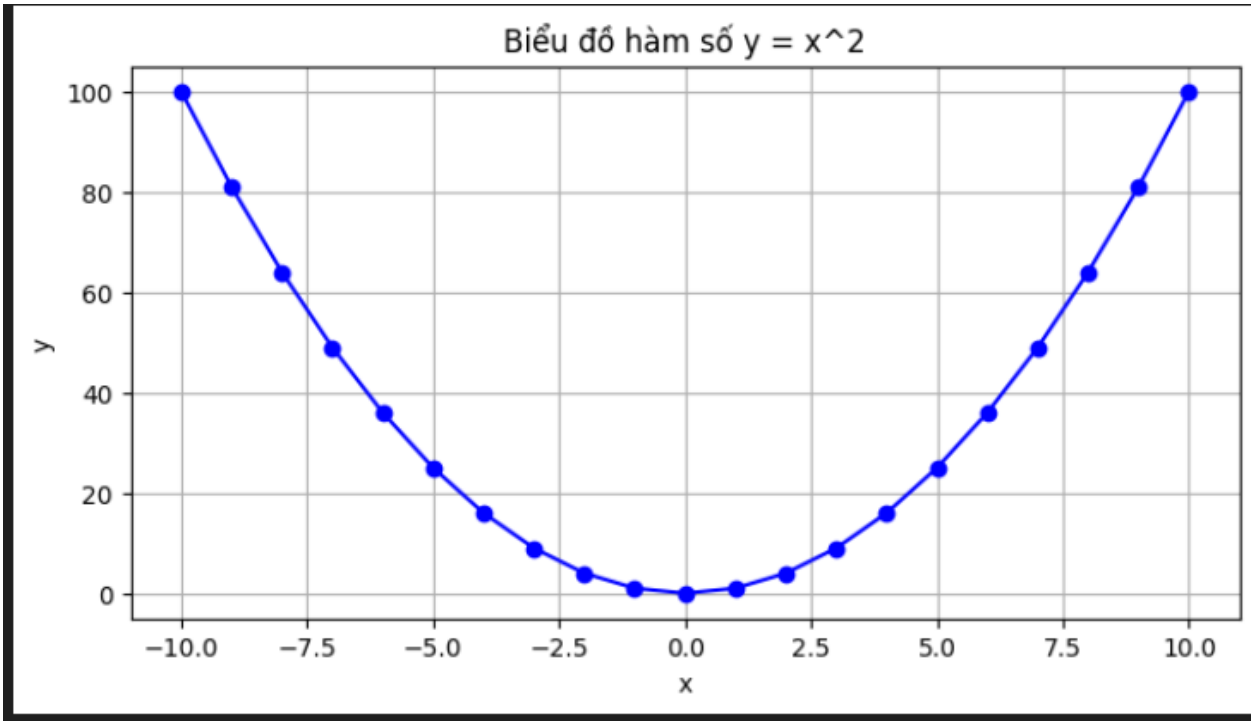
## Bài tập 6: Biểu đồ cơ bản

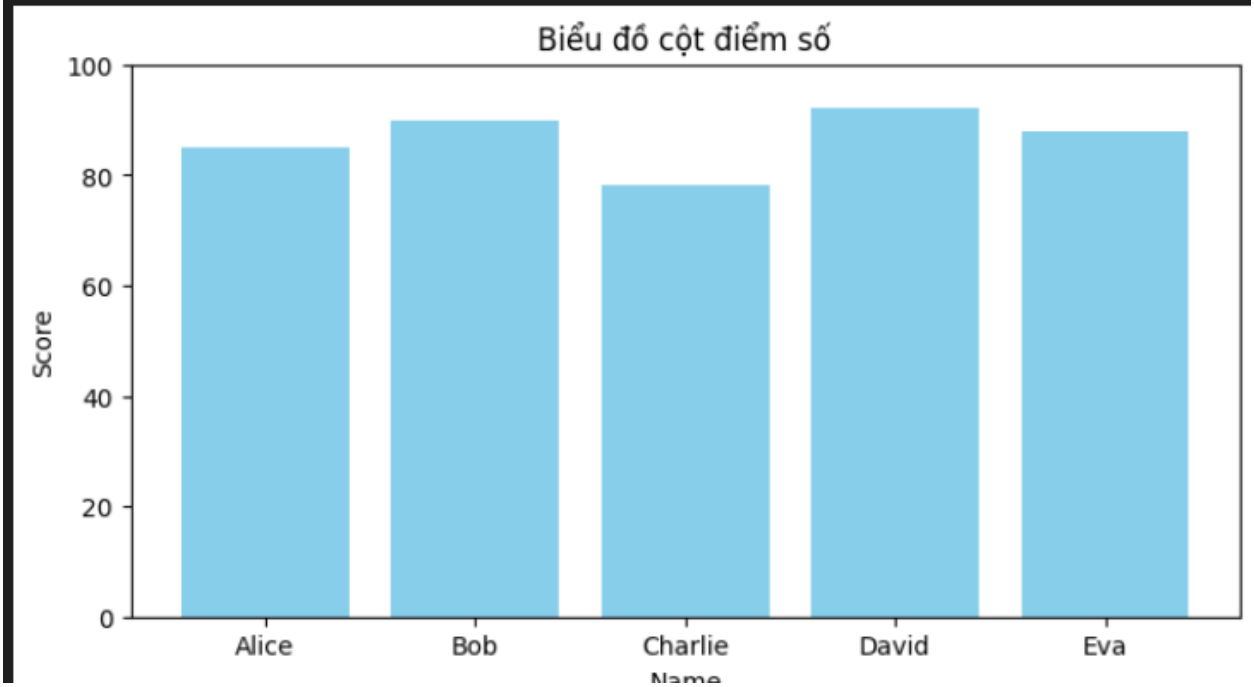
1 Tạo một biểu đồ đường biểu diễn hàm số y = x ** 2 trên khoảng [-10,10]
2 Vẽ biểu đồ cột thể hiện điểm số (Score) của các sinh viên từ Bài tập 3.
3 Tạo một biểu đồ tròn (pie chart) thể hiện phần trăm mỗi loại hoa trong tập dữ liệu Iris.

```python
import matplotlib.pyplot as plt

# 1. Biểu đồ đường hàm số y = x^2
x = range(-10, 11)
y = [i**2 for i in x]

plt.figure(figsize=(8, 4))
plt.plot(x, y, marker='o', linestyle='-', color='b')
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
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva'],
    'Score': [85, 90, 78, 92, 88]
}
df = pd.DataFrame(data)

plt.figure(figsize=(8, 4))
plt.bar(df['Name'], df['Score'], color='skyblue')
plt.title('Biểu đồ cột điểm số')
plt.xlabel('Name')
plt.ylabel('Score')
plt.ylim(0, 100)
plt.show()
```

Git-hub

https://github.com/Huysdfghf/sohoa_hk2/tree/main/lab1


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
100
80
60
40
20
0
Score
Alice
Bob
Charlie
David
Eva
Name

## 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.

```
# 1. Biểu đồ phân tán với màu sắc theo loại hoa
# Đọc dữ liệu từ file CSV
iris_df = pd.read_csv(r'D:\dhvl\hocky_3\3.2\sohoa\lab1\iris.csv')

# 1. Vẽ biểu đồ phân tán (scatter plot) giữa sepal_length và sepal_width
plt.figure(figsize=(10, 6))

# Dùng màu sắc để phân biệt các loài hoa (species)
colors = {'Iris-setosa': 'red', 'Iris-versicolor': 'green', 'Iris-virginica': 'blue'}

# Vẽ các điểm dữ liệu cho từng loài hoa
for species, color in colors.items():
    subset = iris_df[iris_df['Species'] == species]
    plt.scatter(subset['SepalLengthCm'], subset['SepalWidthCm'], c=color, label=species, alpha=0.6)

# 2. 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.legend(title='Species')
plt.grid(True)
plt.show()
```

Link git

https://github.com/Huysdfghf/sohoa_hk2/tree/main/lab1