

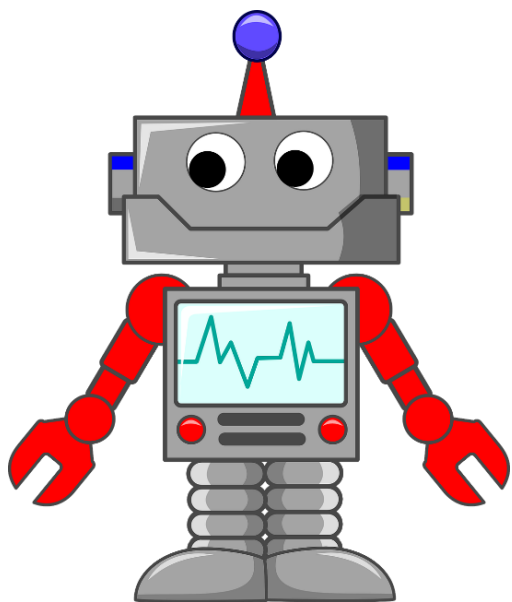


CS116 – LẬP TRÌNH PYTHON CHO MÁY HỌC

BÀI 02

LẬP TRÌNH VỚI NUMPY – MATPLOTLIB - PANDAS

TS. Nguyễn Vinh Tiệp





NỘI DUNG

1. LẬP TRÌNH VỚI NUMPY

2. TRỰC QUAN HÓA VỚI MATPLOTLIB

3. XỬ LÝ DỮ LIỆU BẢNG VỚI PANDAS



Thư viện Numpy

- **Giới thiệu:** là thư viện tính toán khoa học, hiệu năng cao và phổ biến trong Python, thực hiện trên vector, ma trận, tensor...
- Một số chủ đề chính:
 - Array – Indexing – slicing
 - Kiểu dữ liệu
 - Copy và View dữ liệu
 - Array shape và Reshape
 - Lặp trên Array
 - Nhập – tách Array
 - Tìm kiếm trên Array

[Tham khảo file Colab](#)



NỘI DUNG

1. LẬP TRÌNH VỚI NUMPY

2. TRỰC QUAN HÓA VỚI MATPLOTLIB

3. XỬ LÝ DỮ LIỆU BẢNG VỚI PANDAS



Thư viện Matplotlib

- **Giới thiệu:** là thư viện trực quan hóa phổ biến trên ngôn ngữ Python
- Một số chủ đề chính:
 - Hàm Plot
 - Vẽ với Subplot
 - Các loại biểu đồ
 - Load và hiển thị ảnh

[Tham khảo file Colab](#)



NỘI DUNG

1. LẬP TRÌNH VỚI NUMPY

2. TRỰC QUAN HÓA VỚI MATPLOTLIB

3. XỬ LÝ DỮ LIỆU BẢNG VỚI PANDAS



Thư viện Pandas

- Giới thiệu: là thư viện xử lý dữ liệu bảng phổ biến trên Python
- Một số chủ đề chính:
 - Khởi tạo bảng
 - Gom nhóm dữ liệu
 - Nối dữ liệu
 - Thay đổi giá trị
 - Trích xuất dữ liệu
 - Mở rộng dữ liệu
 - Vẽ biểu đồ cơ bản



Khởi tạo

- Tạo DataFrame
 - Khai báo dữ liệu theo **cột**

```
import pandas as pd

df = pd.DataFrame({
    "X" : [13, 30, 'A'],
    "Y" : [15, 32, 'B'],
    "Z" : [10, 29, 'O'],
    "T" : [12, 28, 'AB']},
    index = [1, 2, 3]
)
```

	X	Y	Z	T
1	13	15	10	12
2	30	32	29	28
3	A	B	O	AB



Khởi tạo

- Tạo DataFrame
 - Load dữ liệu từ file csv (bảng)

```
stocks = pd.read_csv('stocks.csv')
```

	date	symbol	open	high	low	close	volume
0	2019-03-01	AMZN	1655.13	1674.26	1651.00	1671.73	4974877
1	2019-03-04	AMZN	1685.00	1709.43	1674.36	1696.17	6167358
2	2019-03-05	AMZN	1702.95	1707.80	1689.01	1692.43	3681522
3	2019-03-06	AMZN	1695.97	1697.75	1668.28	1668.95	3996001



Quy ước

- Quy ước:

Chỉ mục (index)

Cột (column)

	date	symbol	open	high	low	close	volume
0	2019-03-01	AMZN	1655.13	1674.26	1651.00	1671.73	4974877
1	2019-03-04	AMZN	1685.00	1709.43	1674.36	1696.17	6167358
2	2019-03-05	AMZN	1702.95	1707.80	1689.01	1692.43	3681522
3	2019-03-06	AMZN	1695.97	1697.75	1668.28	1668.95	3996001

Mẫu quan sát (observation)

Cột dữ liệu (Variable)



Gom nhóm dữ liệu

- Gom nhóm dữ liệu với phương thức *pivot*

```
stocks.pivot(index='date', columns='symbol', values='close')
```

symbol	AAPL	AMZN	GOOG
date			
2019-03-01	174.97	1671.73	1140.99
2019-03-04	175.85	1696.17	1147.80
2019-03-05	175.53	1692.43	1162.03
2019-03-06	174.52	1668.95	1157.86
2019-03-07	172.50	1625.95	1143.30



Gom nhóm dữ liệu

- Gom nhóm dữ liệu với phương thức *pivot*

```
stocks.pivot(index='date', columns='symbol', values=['close', 'volume'])
```

	close			volume		
symbol	AAPL	AMZN	GOOG	AAPL	AMZN	GOOG
date						
2019-03-01	174.97	1671.73	1140.99	25886167.0	4974877.0	1450316.0
2019-03-04	175.85	1696.17	1147.80	27436203.0	6167358.0	1446047.0
2019-03-05	175.53	1692.43	1162.03	19737419.0	3681522.0	1443174.0
2019-03-06	174.52	1668.95	1157.86	20810384.0	3996001.0	1099289.0
2019-03-07	172.50	1625.95	1143.30	24796374.0	4957017.0	1166559.0



Gom nhóm dữ liệu

- Gom nhóm dữ liệu với phương thức *pivot_table*

```
import numpy as np
stocks.pivot_table(index='symbol', values=['close', 'volume'],
                    aggfunc=np.mean)
```

	close	volume
symbol		
AAPL	174.674	23733309.4
AMZN	1671.046	4755355.0
GOOG	1150.396	1321077.0



Nối dữ liệu

- Nối dữ liệu theo chiều **dọc** với *concat* (mặc định axis=0)

```
# Nối hai data frame theo chiều dọc
df1 = pd.DataFrame({
    "X" : ['A', 'B', 'O', 'AB'],
    "Y" : [15, 12, 10, 12],
    "Z" : [30, 28, 23, 29]},
    index = [1, 2, 3, 4])
df2 = pd.DataFrame({
    "X" : ['O', 'A', 'B'],
    "Y" : [20, 21, 22],
    "Z" : [32, 30, 20],
    "T" : [1, 0, 1]},
    index = [1, 2, 3])

df_new = pd.concat([df1, df2])
```

	X	Y	Z	T
1	A	15	30	NaN
2	B	12	28	NaN
3	O	10	23	NaN
4	AB	12	29	NaN
1	O	20	32	1.0
2	A	21	30	0.0
3	B	22	20	1.0



Nối dữ liệu

- Nối dữ liệu theo chiều **ngang** với *concat* (*axis=1*)

```
# Nối hai data frame theo chiều ngang
df1 = pd.DataFrame({
    "X" : ['A', 'B', 'O', 'AB'],
    "Y" : [15, 12, 10, 12],
    "Z" : [30, 28, 23, 29]},
    index = [1, 2, 3, 4])

df2 = pd.DataFrame({
    "U" : [0, 1, 0],
    "V" : [20, 1, 6]},
    index = [1, 2, 3])

pd.concat([df1, df2], axis=1)
```

	X	Y	Z	U	V
1	A	15	30	0.0	20.0
2	B	12	28	1.0	1.0
3	O	10	23	0.0	6.0
4	AB	12	29	NaN	NaN



Điền giá trị

- Điền giá trị khuyết với `fillna(value)`

```
df_new.fillna(-1)
```

	X	Y	Z	T
1	A	15	30	NaN
2	B	12	28	NaN
3	O	10	23	NaN
4	AB	12	29	NaN
1	O	20	32	1.0
2	A	21	30	0.0
3	B	22	20	1.0



	X	Y	Z	T
1	A	15	30	-1.0
2	B	12	28	-1.0
3	O	10	23	-1.0
4	AB	12	29	-1.0
1	O	20	32	1.0
2	A	21	30	0.0
3	B	22	20	1.0



Trích xuất dữ liệu

- Lấy tập con theo dòng

```
# Lấy tập con theo dòng  
sub_df = df1[df1.Y > 10]
```

	X	Y	Z
1	A	15	30
2	B	12	28
3	O	10	23
4	AB	12	29



	X	Y	Z
1	A	15	30
2	B	12	28
4	AB	12	29



Trích xuất dữ liệu

- Lấy tập con theo dòng

```
sub_df = df1[df1.X.isin(['AB', 'A'])]
```

	X	Y	Z
1	A	15	30
2	B	12	28
3	O	10	23
4	AB	12	29



	X	Y	Z
1	A	15	30
4	AB	12	29



Trích xuất dữ liệu

- Lấy tập con theo cột

```
# Lấy tập con gồm nhiều cột  
columns = df1[['X', 'Z']]
```

	X	Y	Z
1	A	15	30
2	B	12	28
3	O	10	23
4	AB	12	29



	X	Z
1	A	30
2	B	28
3	O	23
4	AB	29



Trích xuất dữ liệu

- Lấy tập con theo cột

```
# Lấy tập con của một cột  
colX = df1.X  
# hoặc  
colX = df1['X']
```

	X	Y	Z
1	A	15	30
2	B	12	28
3	O	10	23
4	AB	12	29




```
1    A  
2    B  
3    O  
4   AB  
Name: X, dtype: object
```



Mở rộng dữ liệu

- Tạo thêm cột mới

```
stocks['value'] = stocks.close*stocks.volume
```



	date	symbol	open	high	low	close	volume	value
0	2019-03-01	AMZN	1655.13	1674.26	1651.00	1671.73	4974877	8.316651e+09
1	2019-03-04	AMZN	1685.00	1709.43	1674.36	1696.17	6167358	1.046089e+10
2	2019-03-05	AMZN	1702.95	1707.80	1689.01	1692.43	3681522	6.230718e+09
3	2019-03-06	AMZN	1695.97	1697.75	1668.28	1668.95	3996001	6.669126e+09
4	2019-03-07	AMZN	1667.37	1669.75	1620.51	1625.95	4957017	8.059862e+09

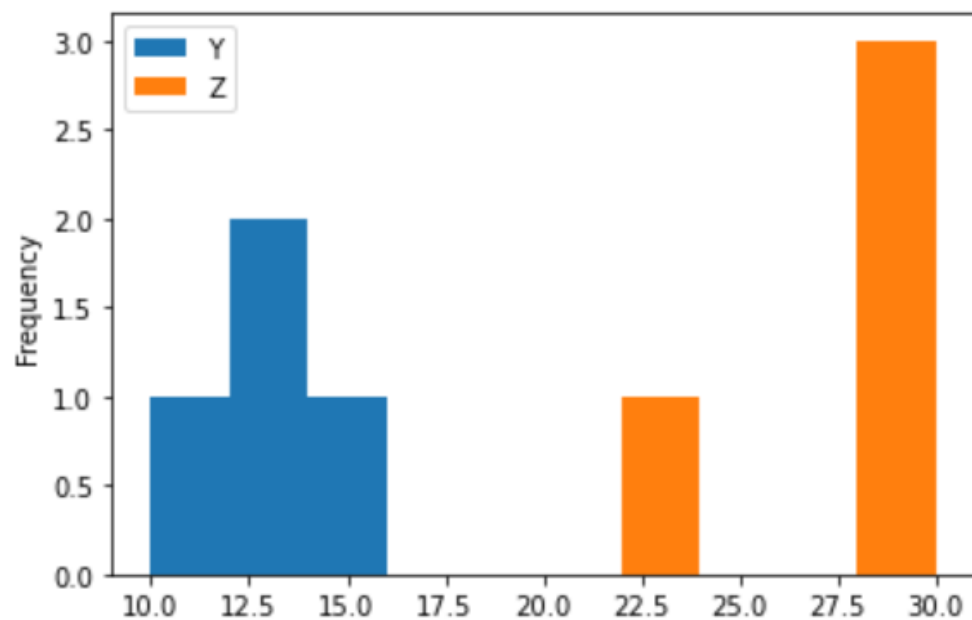


Vẽ biểu đồ cơ bản

- Hàm plot và scatter

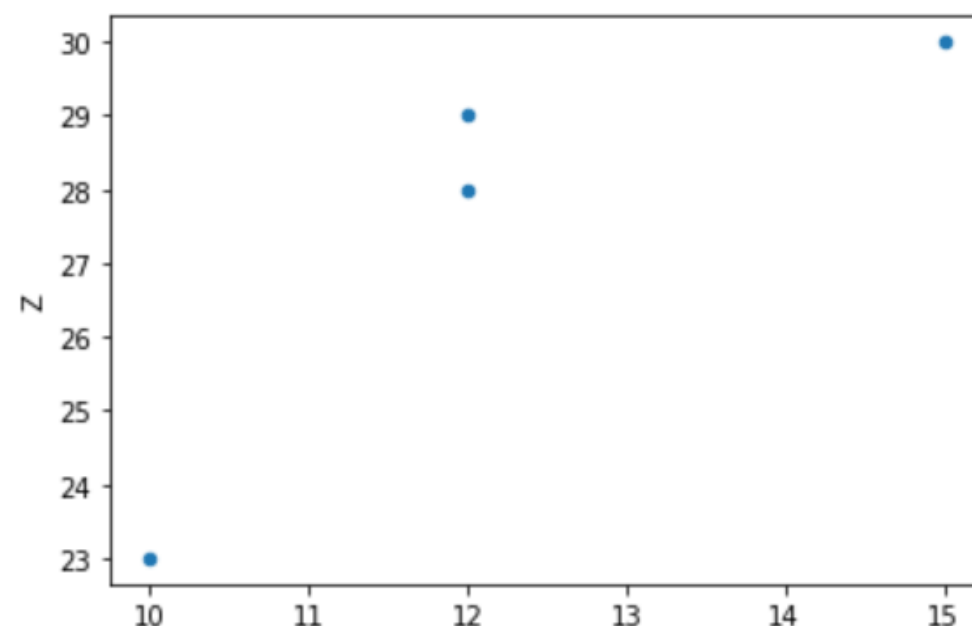
```
df.plot.hist()
```

<matplotlib.axes._subplots.AxesSubplot at 0x2545b11a7c0>



```
df.plot.scatter(x='Y', y='Z')
```

<matplotlib.axes._subplots.AxesSubplot at 0x2545b1bc730>





BÀI QUIZ VÀ HỎI ĐÁP