



CS116 – LẬP TRÌNH PYTHON CHO MÁY HỌC

Bài 02

PYTHON CHO KHOA HỌC DỮ LIỆU

Exploratory Data Analysis

TS. Nguyễn Vinh Tiệp



NỘI DUNG

1. Cài đặt jupyter notebook
2. Đọc ghi file cơ bản
3. Trực quan hóa dữ liệu với Matplotlib
4. Thao tác trên dữ liệu với Pandas



Cài đặt Jupyter notebook

- Cài đặt Jupyter Notebook trên Anaconda


```
conda install -c anaconda jupyter
```

- Khởi động Jupyter Notebook
 - cd <Thư mục làm việc>
 - jupyter notebook



Cài đặt Jupyter notebook

← → ↻ 🏠 ⓘ localhost:8888/tree 🔍 ☆ 🔴 🟢 🟢 ⋮

 jupyter Quit Logout

Files Running Clusters

Select items to perform actions on them.

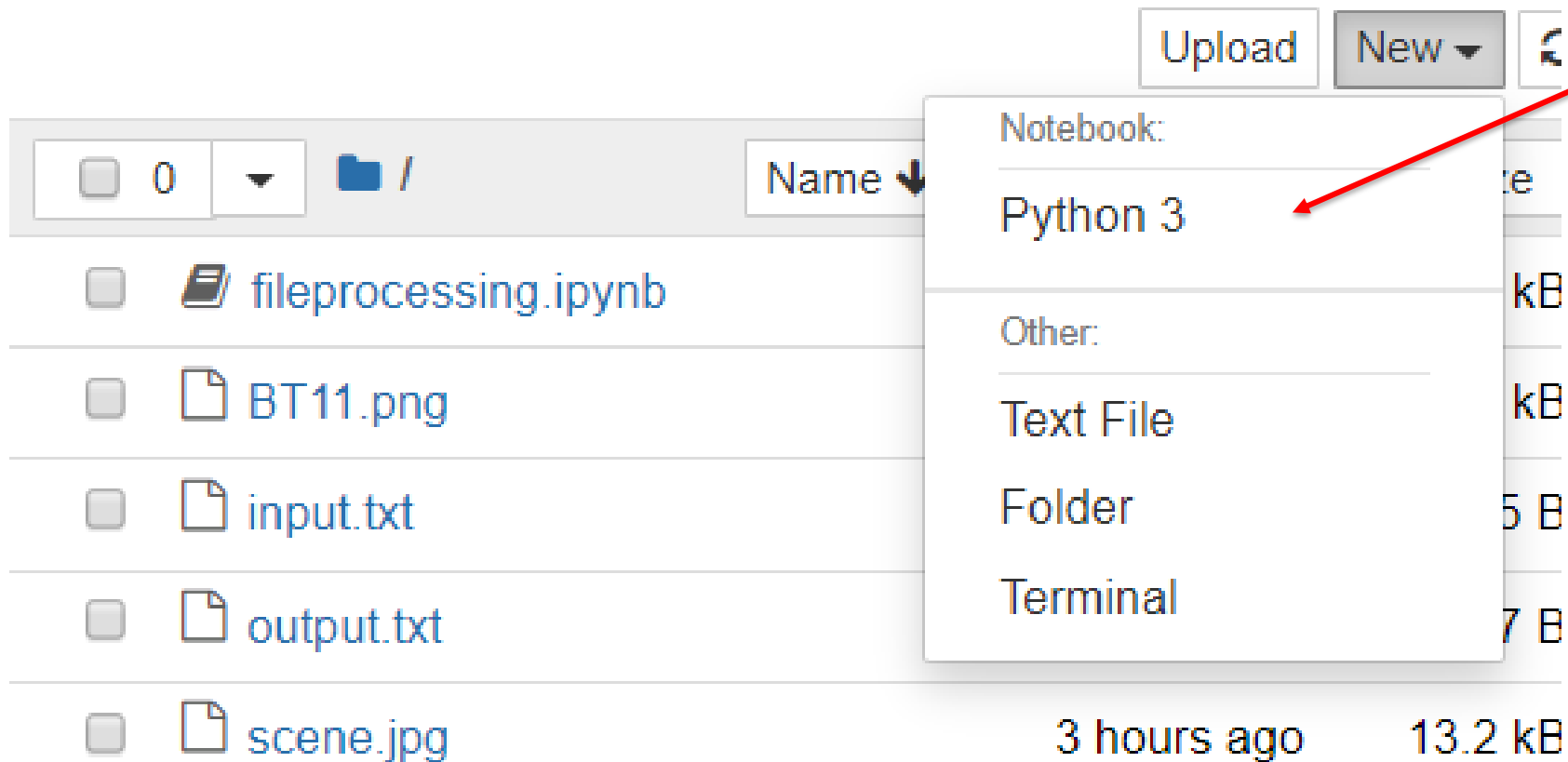
Upload New ▼ ↻

<input type="checkbox"/> 0 ▼ 📁 /	Name ▼	Last Modified	File size
<input type="checkbox"/> 📄 fileprocessing.ipynb		7 minutes ago	119 kB
<input type="checkbox"/> 📄 BT11.png		5 days ago	5.1 kB
<input type="checkbox"/> 📄 input.txt		14 hours ago	35 B
<input type="checkbox"/> 📄 output.txt		13 hours ago	27 B
<input type="checkbox"/> 📄 scene.jpg		3 hours ago	13.2 kB



Cài đặt Jupyter notebook

- Tạo mới file jupyter notebook



Tạo mới
notebook
Python 3



Không gian làm việc

jupyter Untitled Last Checkpoint: 2 minutes ago (unsaved changes)

```
File Edit View Insert Cell Kernel Widgets Help
[Save] [New] [Cut] [Copy] [Paste] [Undo] [Redo] [Run] [Stop] [Refresh] [Next] [Code] [Help]

In [1]: import os
import numpy as np

x = np.arange(1, 10, 1)

In [2]: print(x)

[1 2 3 4 5 6 7 8 9]

In [3]: y = [i**2 for i in x]
print(y)

[1, 4, 9, 16, 25, 36, 49, 64, 81]

In [ ]:
```



Đọc ghi file cơ bản

- Đọc file văn bản

```
# Đọc toàn bộ file văn bản  
# Lưu ý: 'mở' đi đôi với 'đóng'  
file = open("input.txt", "r")  
print(file.read())  
file.close()
```

```
Hello world!  
This is another line.
```



Đọc ghi file cơ bản

- Ghi file văn bản

```
# Ghi file văn bản
# Lưu ý: 'mở' đi đôi với 'đóng'
file = open("output.txt", "w")
a = [1,3,5,7,9,10]
file.write("Write array:\n")
for x in a:
    file.write('%d ' % x)
file.close()
```




Đọc ghi file cơ bản

- Đọc file ảnh với thư viện Pillow
 - Sử dụng khi không cần các thao tác xử lý nâng cao
- Cài đặt Pillow (nếu chưa có sẵn):

```
conda install -c anaconda pillow
```

```
from PIL import Image
import numpy as np
im = Image.open("scene.jpg")
np_im = np.array(im)
print("Kích thước file ảnh: ", np_im.shape)
```

Kích thước file ảnh: (183, 275, 3)



Đọc ghi file cơ bản

- Ghi file ảnh với thư viện Pillow

```
from PIL import Image

# Đọc ảnh từ file
im = Image.open("scene.jpg")

# Thao tác xử lý ảnh khác nếu cần
# .....

# Ghi ảnh với tên và định dạng khác
im.save("scene-copy.png")
```



Đọc ghi file cơ bản

- Đọc file ảnh với thư viện Opencv
 - Sử dụng khi muốn xử lý nâng cao: lọc ảnh, rút trích đặc trưng, phân lớp
 - Không có mặc định trong Anaconda nên cần cài thêm:

```
conda install -c conda-forge opencv
```



Đọc ghi file cơ bản

- Đọc file ảnh với thư viện Opencv

```
import cv2
import numpy as np
bgr_im = cv2.imread("scene.jpg")
gray_im = cv2.imread("scene.jpg", 0)

print("Kích thước ảnh màu: ", bgr_im.shape)
print("Kích thước mức xám: ", gray_im.shape)
```

```
Kích thước ảnh màu:  (183, 275, 3)
Kích thước mức xám:  (183, 275)
```



Đọc ghi file cơ bản

- Thao tác + hiển thị + ghi file ảnh với Opencv

```
# Thao tác trên ảnh như trên ma trận
gray_im[50:100, 50:100] = 0 # xóa vùng ảnh

# Hiển thị trên cửa sổ khác
window_name = 'image'
cv2.imshow(window_name, gray_im)
cv2.waitKey(0)
cv2.destroyAllWindows()

# Ghi file ảnh xuống file
cv2.imwrite('scene-cut.png', gray_im)
```





Trực quan hóa với Matplotlib

- Cài thư viện Matplotlib (nếu chưa có):

```
conda install -c conda-forge matplotlib
```

- Khởi tạo dữ liệu dùng numpy:

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

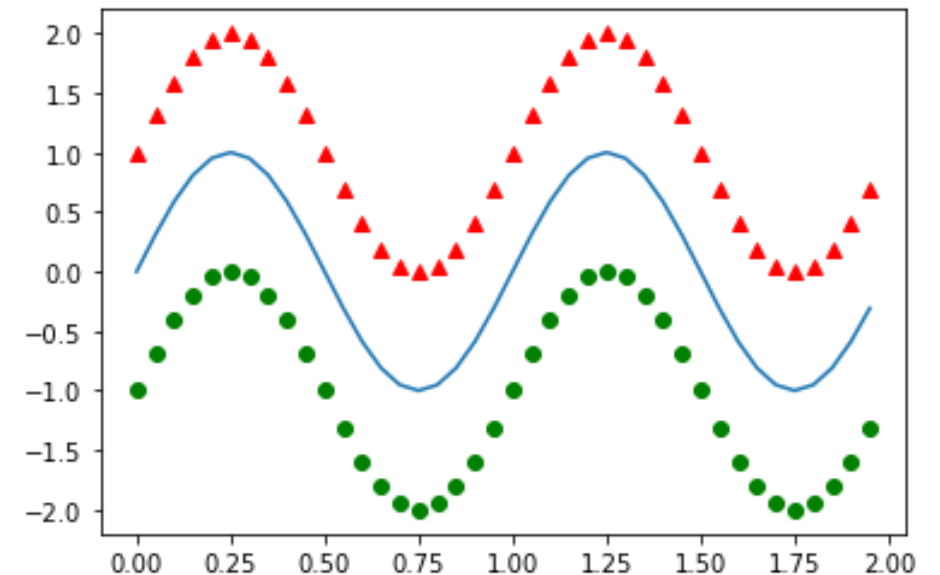
# Tạo dữ liệu hình sin
t = np.arange(0.0, 2.0, 0.05) # t lấy mẫu từ 0 đến 2, bước nhảy 0.05
s = np.sin(2 * np.pi * t)     # s tính theo t: s = sin(2*pi*t)
```



Thực quan hóa với Matplotlib

- Vẽ dạng đường và điểm
 - Mặc định là dạng đường (line)
 - Tham số để vẽ điểm 'r^' □ tam giác đỏ, 'go' □ tròn xanh lá

```
# Vẽ dạng đường và điểm  
plt.plot(t, s)  
plt.plot(t, s+1, 'r^')  
plt.plot(t, s-1, 'go')  
plt.show()
```

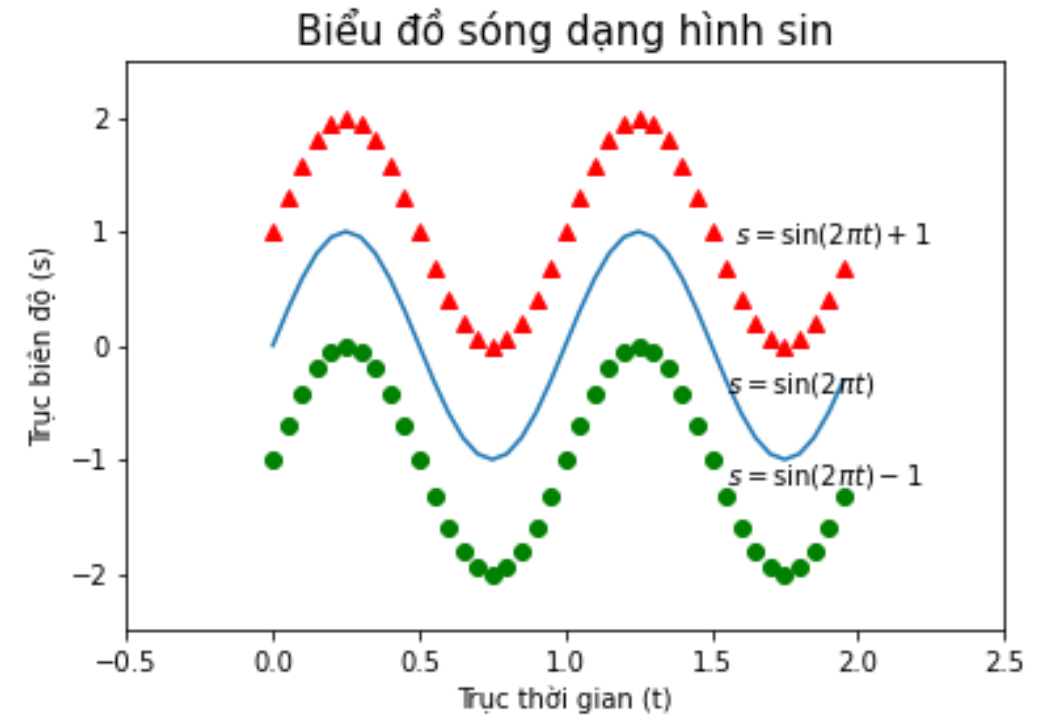




Trực quan hóa với Matplotlib

- Cấu hình biểu đồ:

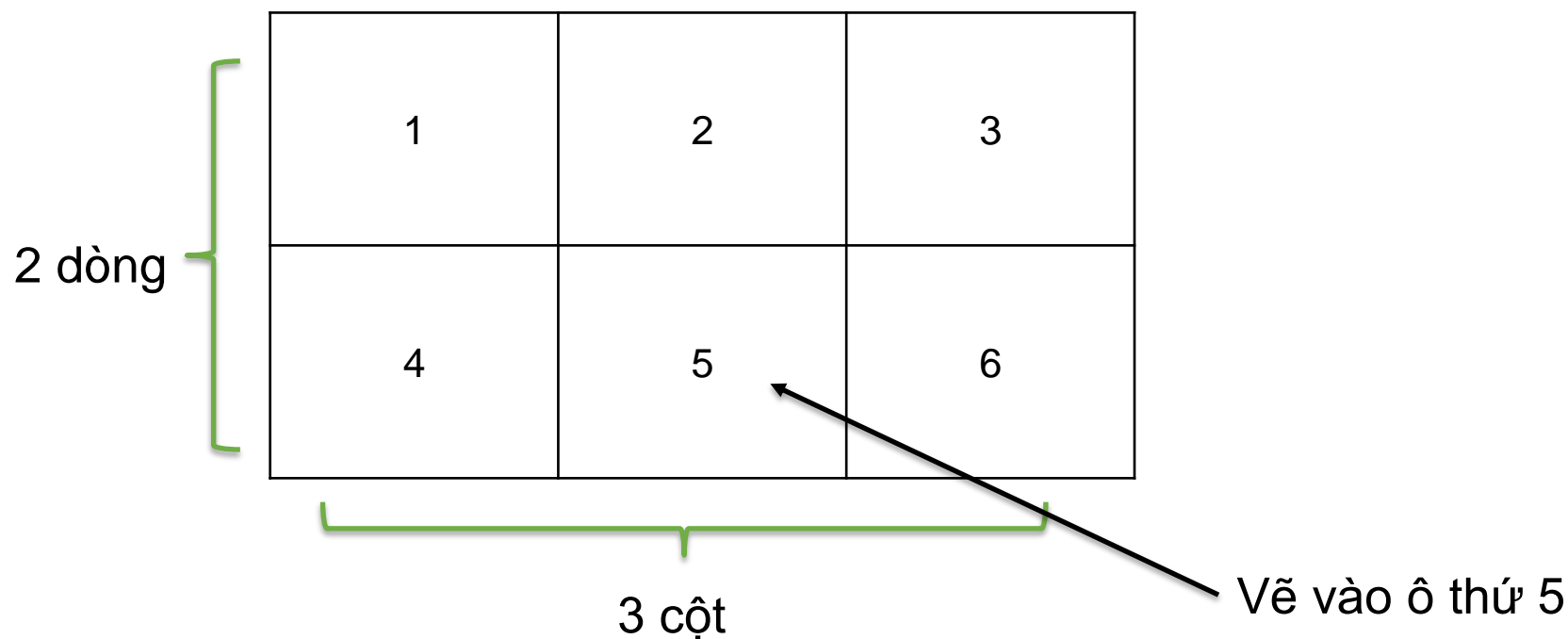
```
plt.plot(t, s)
plt.plot(t, s+1, 'r^')
plt.plot(t, s-1, 'go')
plt.title('Biểu đồ sóng dạng hình sin', fontsize=15)
plt.xlabel('Trục thời gian (t)')
plt.ylabel('Trục biên độ (s)')
plt.text(1.55, -0.4, r'$s=\mathrm{sin}(2 \pi t)$')
plt.text(1.58, 0.9, r'$s=\mathrm{sin}(2 \pi t) + 1$')
plt.text(1.55, -1.2, r'$s=\mathrm{sin}(2 \pi t) - 1$')
plt.xlim(-0.5, 2.5)
plt.ylim(-2.5, 2.5)
plt.show()
```





Trực quan hóa với Matplotlib

- Vẽ nhiều biểu đồ với hàm: `subplot(<nrow><ncol><index>)`
- Ví dụ: `subplot(235)`

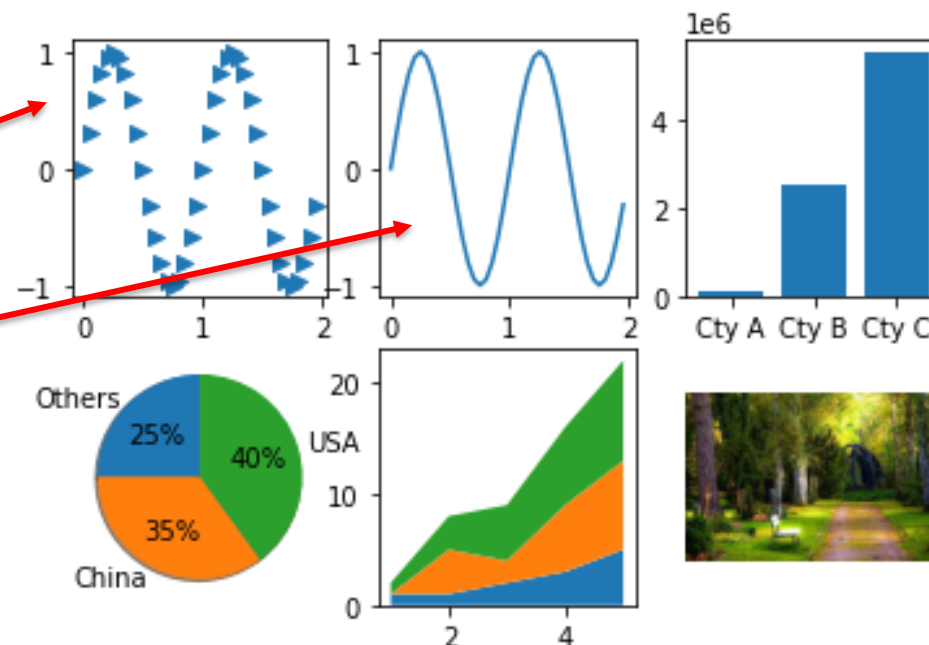




Trực quan hóa với Matplotlib

```
# Vẽ ô thứ 1 trong bảng 2x3  
# Biểu đồ dạng điểm  
plt.subplot(231)  
plt.scatter(t, s, marker=">")
```

```
# Vẽ ô thứ 2 trong bảng 2x3  
# Biểu đồ dạng đường  
plt.subplot(232)  
plt.plot(t, s)
```

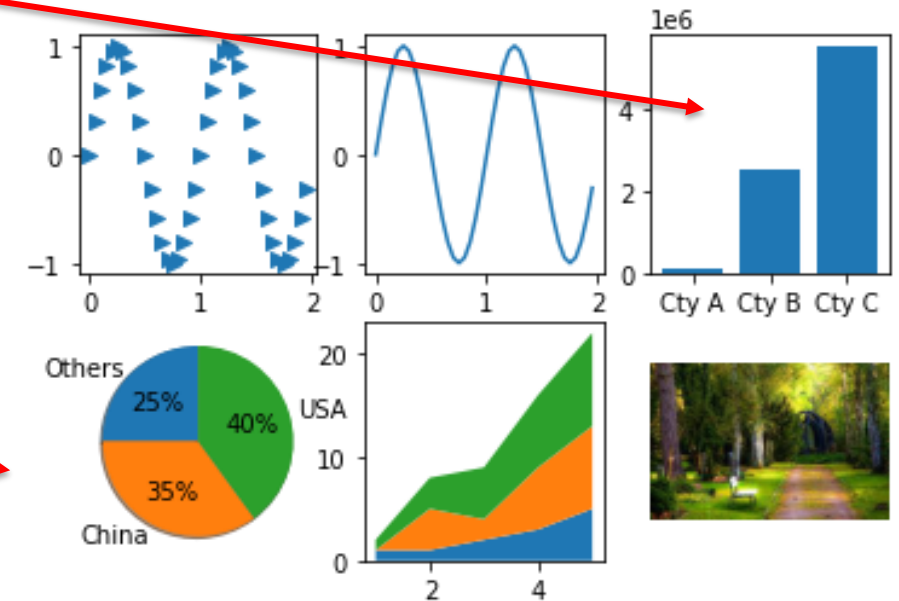




Trực quan hóa với Matplotlib

```
# Vẽ ô thứ 3 trong bảng 2x3
# Biểu đồ cột
plt.subplot(233)
x = np.arange(3)
money = [1.5e5, 2.5e6, 5.5e6]
plt.bar(x, money)
plt.xticks(x, ('Cty A', 'Cty B', 'Cty C'))

# Vẽ ô thứ 4 trong bảng 2x3
# Biểu đồ tròn
plt.subplot(234)
labels = 'Others', 'China', 'USA'
sizes = [25, 35, 40]
plt.pie(sizes, labels=labels, autopct='%1.0f%%',
        shadow=True, startangle=90)
```



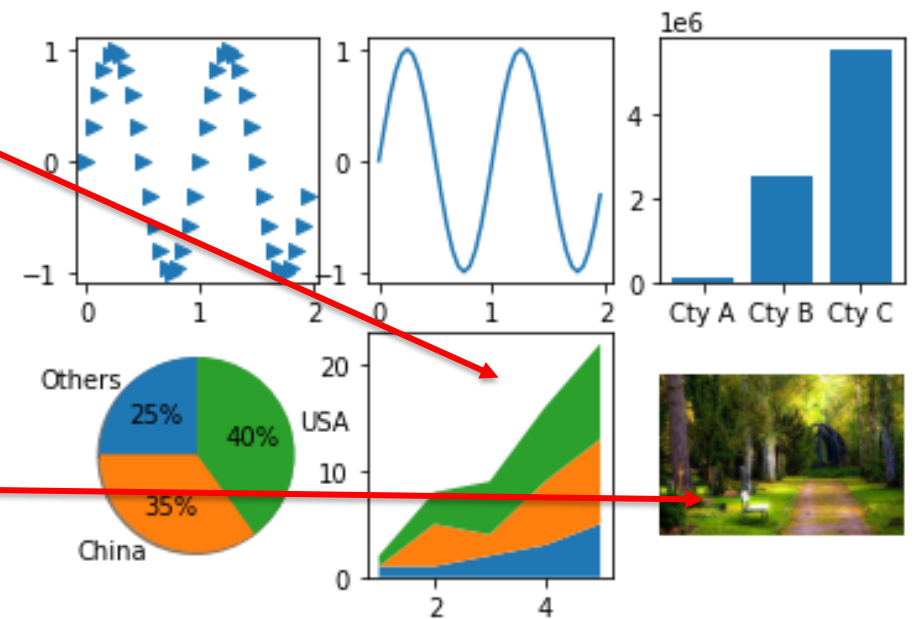


Trực quan hóa với Matplotlib

```
# Vẽ ô thứ 5 trong bảng 2x3
# Biểu đồ xếp chồng
plt.subplot(235)
x = [1, 2, 3, 4, 5]
y1 = [1, 1, 2, 3, 5]
y2 = [0, 4, 2, 6, 8]
y3 = [1, 3, 5, 7, 9]
y = np.vstack([y1, y2, y3])
plt.stackplot(x, y1, y2, y3, labels=labels)

# Vẽ ô thứ 6 trong bảng 2x3
# Vẽ ảnh
plt.subplot(236)
image = plt.imread('scene.jpg')
plt.imshow(image)
plt.axis('off')

plt.show()
```





Xử lý dữ liệu với Pandas

- Cài thư viện Pandas (nếu chưa có):

```
conda install -c anaconda pandas
```



Xử lý dữ liệu với Pandas

- 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



Xử lý dữ liệu với Pandas

- 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



Xử lý dữ liệu với Pandas

- 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



Xử lý dữ liệu với Pandas

- 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



Xử lý dữ liệu với Pandas

- 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



Xử lý dữ liệu với Pandas

- 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



Xử lý dữ liệu với Pandas

- 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



Xử lý dữ liệu với Pandas

- 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
```



Xử lý dữ liệu với Pandas

- Tạo mới cột

```
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

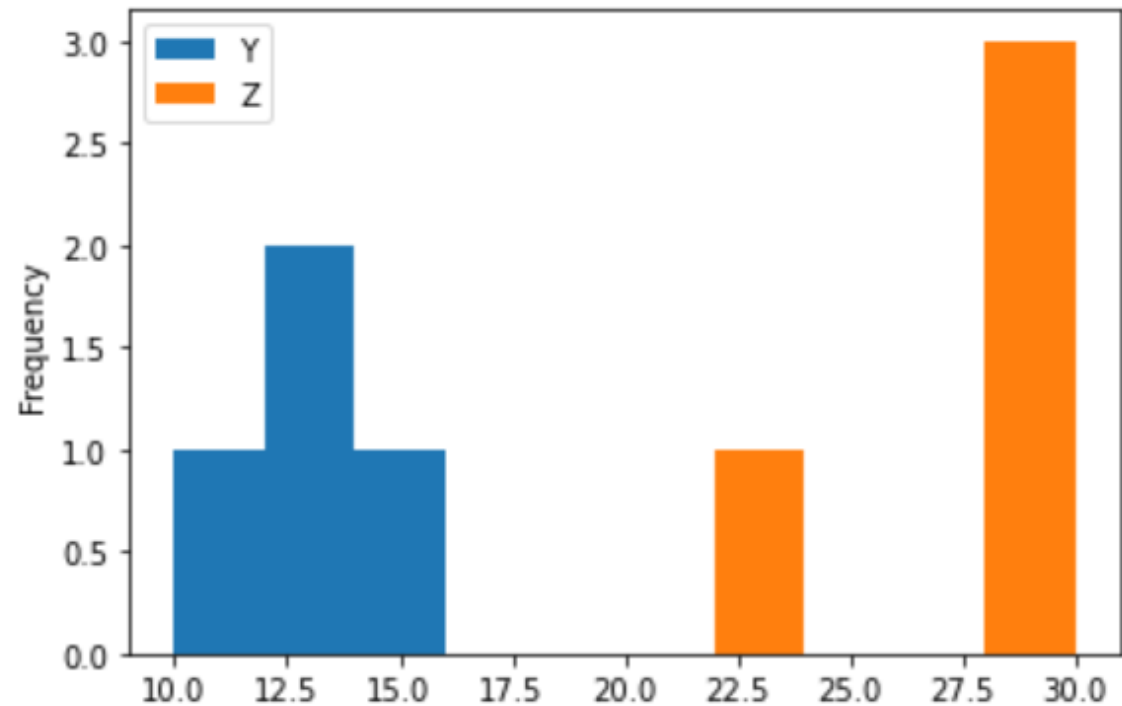


Xử lý dữ liệu với Pandas

- Vẽ biểu đồ với plot

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

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



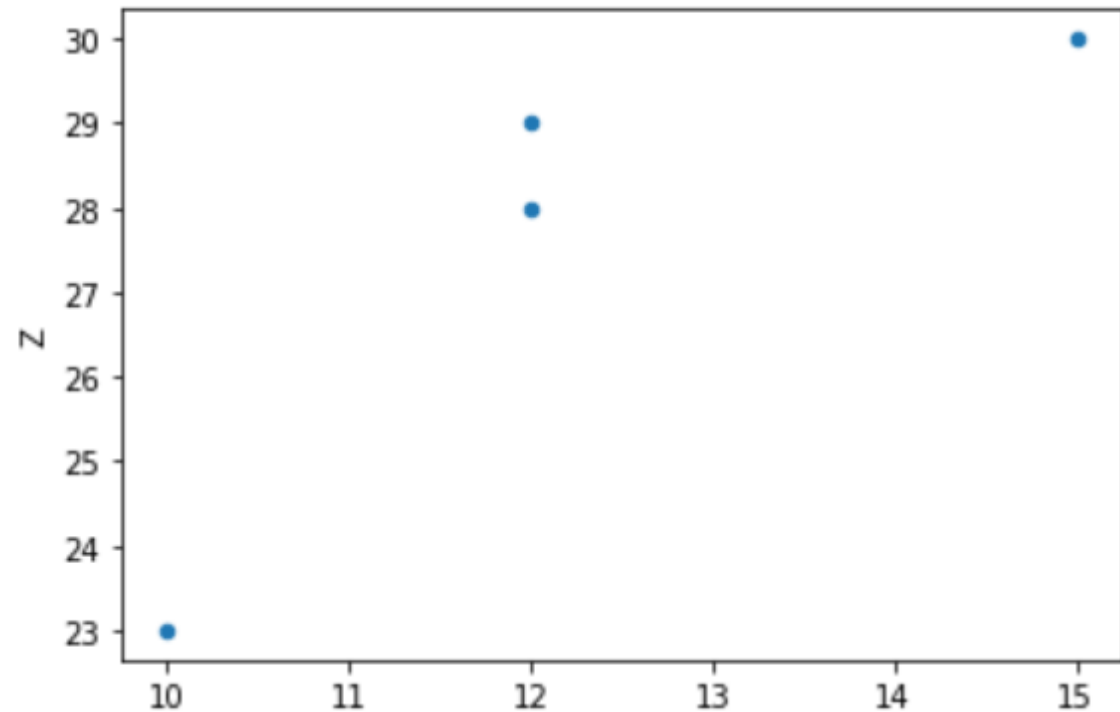


Xử lý dữ liệu với Pandas

- Vẽ biểu đồ với plot

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

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





Xử lý dữ liệu với Pandas

- 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



Xử lý dữ liệu với Pandas

- 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)



Xử lý dữ liệu với Pandas

- 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



Xử lý dữ liệu với Pandas

- 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



Xử lý dữ liệu với Pandas

- 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



QUIZ & CÂU HỎI