

Bài 9: Giới thiệu Matplotlib

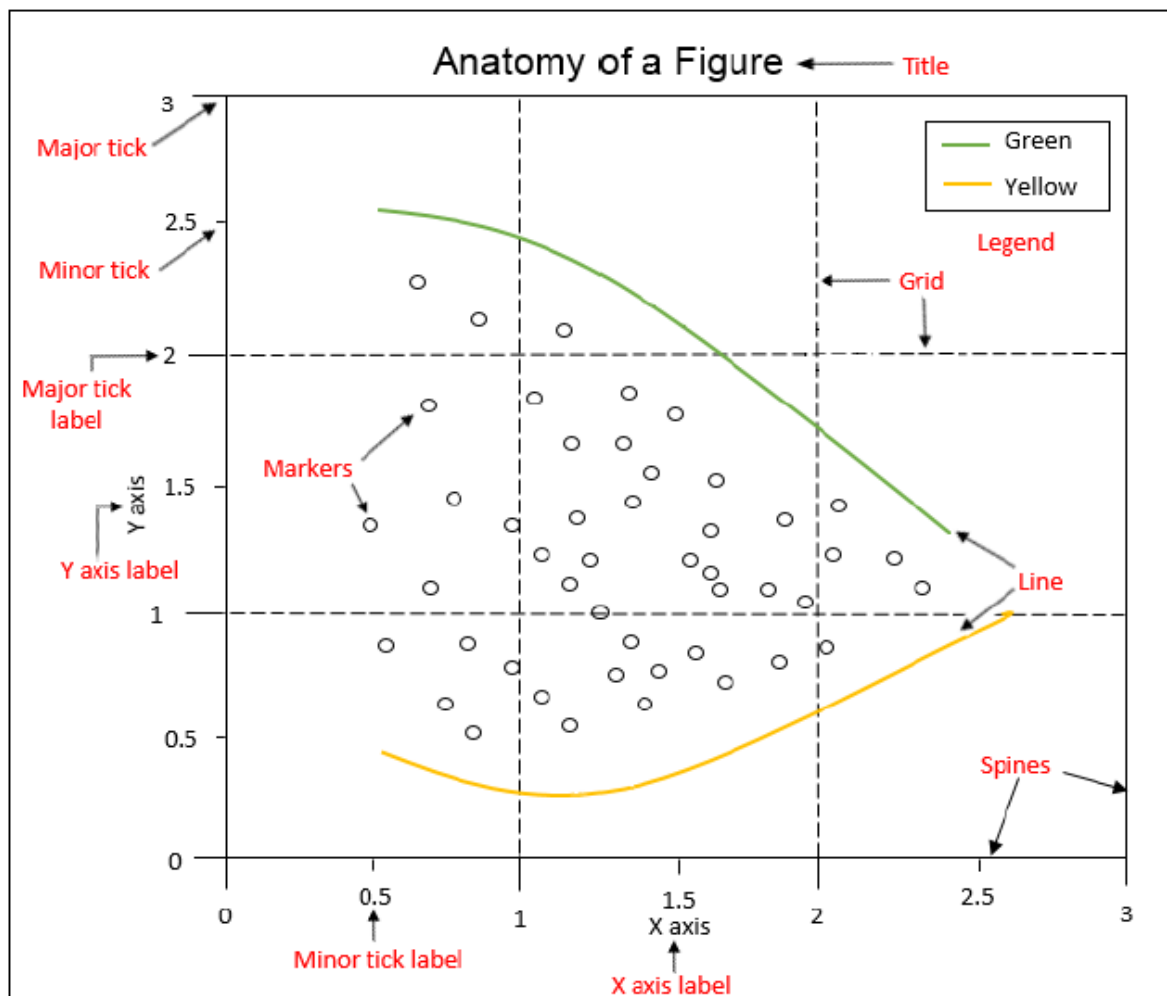
Trong ngày đầu tiên bạn đã làm quen với các loại biểu đồ và đã biết thư viện `matplotlib`. Tuy nhiên bài này sẽ dành riêng để tìm hiểu về thư viện vẽ biểu đồ phổ biến trong Python này. Matplotlib rất phổ biến trong giới data science (khoa học dữ liệu) và machine learning (máy học). Matplotlib được John Hunter phát triển từ năm 2003, lấy ý tưởng từ phần mềm nổi tiếng MATLAB.

Cốt lõi của Matplotlib

Matplotlib xem biểu đồ gồm có 2 thành phần chính:

- **Figure**: Figure được xem như là khung vải mà họa sĩ chuẩn bị để vẽ một bức tranh
- **Axes**: Axes là đối tượng cần vẽ, giống như nội dung bức tranh. Trong bức tranh này có **trục x**, **trục y**, các giá trị cần thể hiện trong không gian x,y (**Markers, Lines, Grid**); các thành phần khác để trang trí như: **tên trục x** (x axis label), **tên trục y** (y axis label), **tiêu đề** bức tranh (title), **ghi chú** (Legend).

Trên hai trục xy thì có thêm các **kí hiệu chia đơn vị chính** (Major stick), **nhãn giá trị đơn vị chính** (Major stick label); **kí hiệu chia đơn vị phụ** (Minor stick), **nhãn giá trị đơn vị phụ** (Minor stick label)



Sub module pyplot của Matplotlib

Module `pyplot` sẽ giúp chúng ta vẽ các biểu đồ mà không cần tốn nhiều thì giờ cho việc trang trí (sử dụng `Figure` và `Axes`).

Để sử dụng sub module `pyplot` của `matplotlib` thì dùng cú pháp sau:

```
import matplotlib.pyplot as plt
```

Nạp thư viện `pyplot` với tên viết tắt (alias) `plt`.

Tạo figure - tạo khung tranh

Đầu tiên là gọi hàm `.figure()` để tạo ra đối tượng `Figure`:

```
fig = plt.figure()
```

```
<Figure size 432x288 with 0 Axes><Figure size 432x288 with 0 Axes>
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

Mặc định Python sẽ tạo ra bức tranh có kích thước 432 x 288 (tương ứng width x height). Kích thước này tương ứng với 6.4 inches chiều rộng và 4.8 inches chiều ngang với dpi là 100 ⁵).

Để thay đổi kích thước của biểu đồ thì truyền thêm tham số `figsize`:

```
# Thiết lập bề rộng và chiều cao
fig = plt.figure(figsize=(10, 5))
# Thiết lập dpi: Số điểm ảnh trên một đơn vị Inch
fig = plt.figure(dpi=300)
```

```
Out: <Figure size 1800x1200 with 0 Axes><Figure size 720x360 with 0
Axes>
<Figure size 1800x1200 with 0 Axes>
```

Đóng figure - đóng khung tranh

Đối tượng `figure` được tạo dùng để vẽ tiếp chi tiết bức tranh. Khi không dùng nữa thì gọi hàm `close()` để đóng đối tượng. Tức là hủy đối tượng `figure`.

Ví dụ bạn là họa sĩ trên Python, chuẩn bị bày tám vải ra chuẩn bị vẽ biểu đồ thì có ai đó rủ đi café, đá bóng, tám chuyện thì vội đóng lại. Code Python như sau:

```
import matplotlib.pyplot as plt

# Thiết lập bề rộng và chiều cao
plt.figure(figsize=(10, 5))
# Thiết lập dpi: Số điểm ảnh trên một đơn vị Inch
plt.figure(dpi=300)

plt.close()
```

Lệnh `plt.close()` không có tham số thì mặc định cái `figure` hiện tại sẽ bị hủy (đóng). Nếu có nhiều `figure` được tạo và muốn hủy tất cả thì truyền thêm tham số chuỗi `'all'`, hoặc `"all"` `"all"`)

```
plt.close('all')
```

Nếu muốn đóng một `figure` cụ thể thì chỉ rõ số thứ tự trong tham số `num`:

```
import matplotlib.pyplot as plt
```

⁵ Để chuyển đổi độ phân giải màn hình và dpi (Dots per Inch) sang kích thước thật thì không quá phức tạp. Tuy nhiên, bạn hãy tạm bỏ qua cái này.

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
# Tạo Figure với số 1. Thiết lập bề rộng và chiều cao
plt.figure(num=1, figsize=(10, 5))
# Thiếp lập dpi: Số điểm ảnh trên một đơn vị Inch
plt.figure(dpi=300)

# Đóng Figure với số 1
plt.close(1)
```

Cấu trúc format

Format ở đây là định dạng cho biểu đồ.

Format này gồm 3 phần [color][marker][line] được trình bày theo 2 dạng:

- Dạng gọn: 'bo--'
- Dạng đầy đủ: color='blue', marker='o', linestyle='dashed'

Ví dụ Format được sử dụng trong tham số thứ ba của hàm `plot([x], y, [format])` sẽ được giải thích ở mục tiếp theo.

Định dạng marker

Marker là kí hiệu để vẽ biểu đồ. Marker phổ biến là điểm (point) tại giá trị của (x_i, y_j) :

Tra cứu kí hiệu marker tại link:






https://matplotlib.org/stable/api/markers_api.html





Vài marker tiêu biểu:

marker	symbol	description
","	●	point
","	·	pixel
"o"	●	circle
"v"	▼	triangle_down
"^"	▲	triangle_up
"<"	◀	triangle_left
">"	▶	triangle_right
"1"	⋈	tri_down

marker	symbol	description
"2"	⋈	tri_up
"3"	◀	tri_left
"4"	▶	tri_right
"8"	●	octagon
"s"	■	square
"p"	⬠	pentagon
"P"	+	plus (filled)
"*"	★	star

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

marker	symbol	description
"h"		hexagon1
"H"		hexagon2
"+"		plus
"x"		x
"X"		x (filled)

marker	symbol	description
"D"		diamond
"d"		thin_diamond
" "		vline
"_"		hline

Định dạng màu sắc

Kí hiệu	Màu	Kí hiệu	Màu
b	blue	c	cyan
r	red	b	black
g	green	w	white
m	magenta	y	yellow

Định dạng loại đường kẻ (styleline) – nối các point

Kí hiệu	Mô tả	Ví dụ
'-'	solid line style	_____
'--'	dashed line style	-----
'-.'	dash-dot line style	-.-.-.-
':'	dotted line style

Các biểu đồ cơ bản

Vẽ biểu đồ với 2 dãy x, y

Hàm `plot([x], y, [format])` sẽ vẽ biểu đồ gồm các điểm theo tọa độ của x, y. Nếu không có x thì mặc định x sẽ là dãy số 0, 1, 2, ...

[format] là định dạng 3 thông tin: `color`, `marker` và `styleline`. Định dạng này có thể viết tắt gồm các kí hiệu đã mô tả trong phần trên:

```
plt.plot(x, y, 'bo--')
```

hoặc viết đầy đủ theo dạng truyền tham số thông thường:

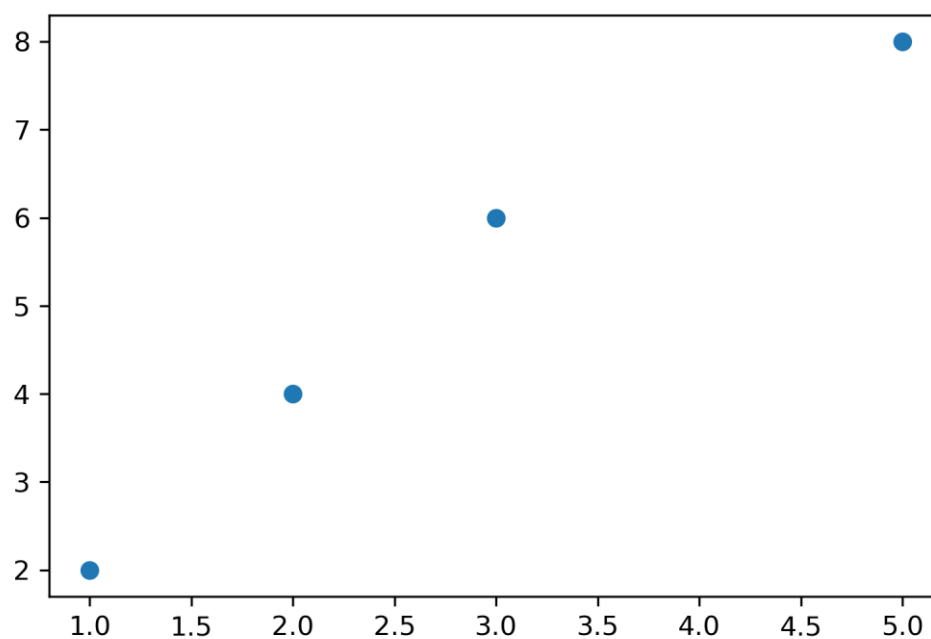
```
plt.plot(x, y, color='blue', marker='o', linestyle='dashed')
```

Nhắc lại: kí hiệu [] bao đóng tham số cho biết là không bắt buộc được chỉ định (sẽ sử dụng giá trị mặc định).

```
import matplotlib.pyplot as plt

# Tạo Figure với số 1. Thiết lập bề rộng và chiều cao
plt.figure(num=1, figsize=(10, 5))
# Thiếp lập dpi: Số điểm ảnh trên một đơn vị Inch
plt.figure(dpi=300)

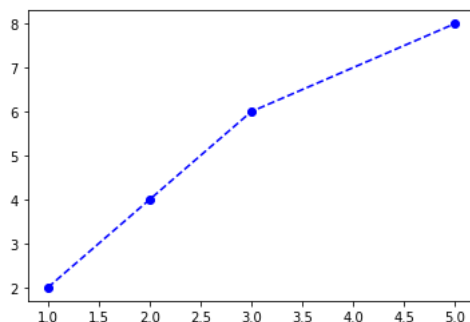
x = [1, 2, 3, 5]
y = [2, 4, 6, 8]
plt.plot(x, y, 'o')
```



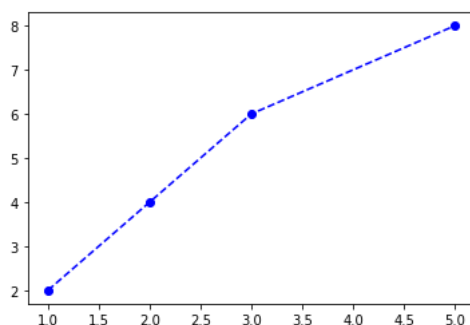
Thử plot với format khác nhau để tự khám phá ý nghĩa các kí hiệu:

```
plt.plot(x, y, 'bo--')
```

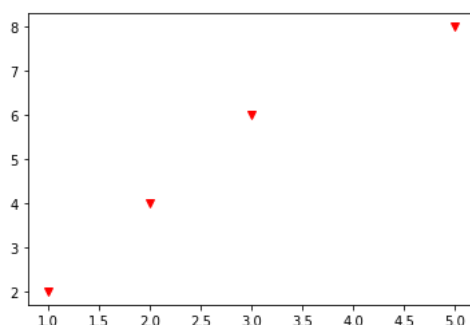
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



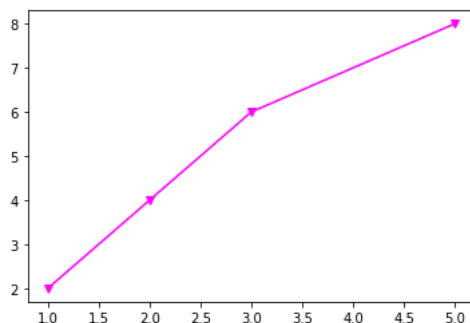
```
plt.plot(x, y, color='blue', marker='o', linestyle='dashed')
```



```
plt.plot(x, y, 'rv')
```



```
plt.plot(x, y, color='magenta', marker='v')
```



Lệnh này truyền 2 tham số `color` và `marker`, không truyền tham số `styleline` thì mặc định có kẻ đường nối giữa các point kề nhau.

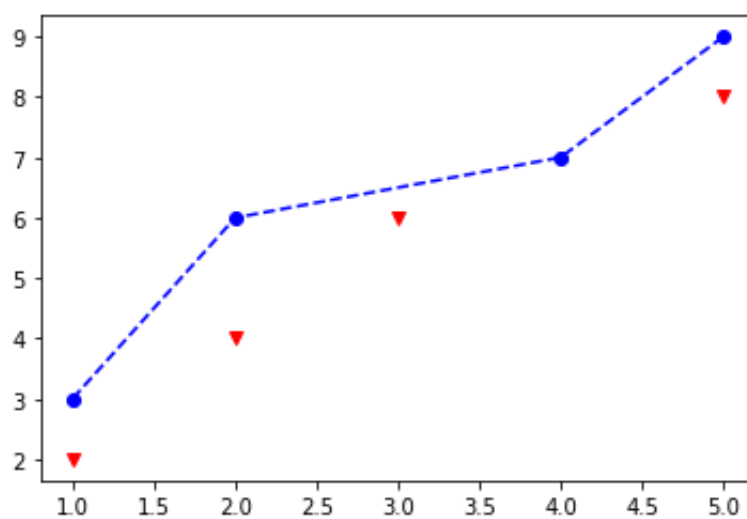
Vẽ biểu đồ với nhiều dãy x, y

Có thể mở rộng gồm 2 bộ tham số:

```
x1 = [1, 2, 3, 5]
y1 = [2, 4, 6, 8]

x2 = [1, 2, 4, 5]
y2 = [3, 6, 7, 9]

plt.plot(x1, y1, 'rv', x2, y2, 'bo--')
```



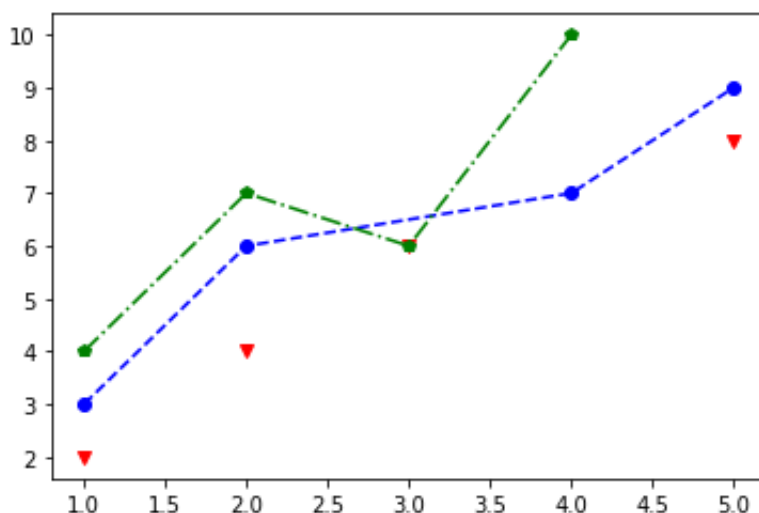
Với 3 bộ tham số:

```
x1 = [1, 2, 3, 5]
y1 = [2, 4, 6, 8]

x2 = [1, 2, 4, 5]
y2 = [3, 6, 7, 9]

x3 = [1, 2, 3, 4]
y3 = [4, 7, 7, 10]

plt.plot(x1, y1, 'rv', x2, y2, 'bo--', x3, y3, 'gp-.')
```



Vẽ biểu đồ line với data frame

Dùng thư viện `matplotlib.pyplot` kết hợp với thư viện `pandas.DataFrame` với cấu trúc sau:

```
plt.plot(x_key, y_key, data=df)
```

Quay lại ví dụ trong Bài 7, vẽ biểu đồ tăng trưởng GDP và CPI bằng cách tự tạo data frame như sau:

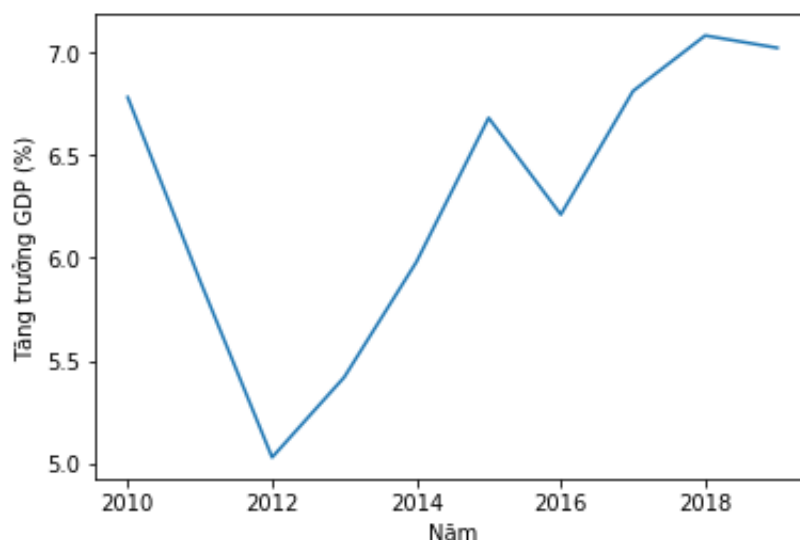
```
import pandas as pd
import matplotlib.pyplot as plt

gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08,
7.02]
year = [2010,2011,2012,2013,2014,2015,2016,2017,2018,2019]

# Tạo data frame
df = pd.DataFrame({'gdp': gdp, 'year': year})

plt.figure().suptitle('Biểu đồ tăng trưởng GDP của Việt Nam
từ năm 2010 đến 2019')
plt.xlabel('Năm')
plt.ylabel('Tăng trưởng GDP (%)')
plt.plot('year', 'gdp', data = df)
```

Biểu đồ tăng trưởng GDP của Việt Nam từ năm 2010 đến 2019



Bạn để ý so với Bài 10 thì khác ở dòng cuối cùng: vẽ với trục x là giá trị cột tên 'year', trục y là giá trị cột tên 'gdp', với data frame là **df**.

Vẽ nhiều biểu đồ

Để hỗ trợ vẽ nhiều biểu đồ trong một “bức tranh” thì Matplotlib cung cấp hàm `.subplots()`.

Khởi tạo Figure và Axe

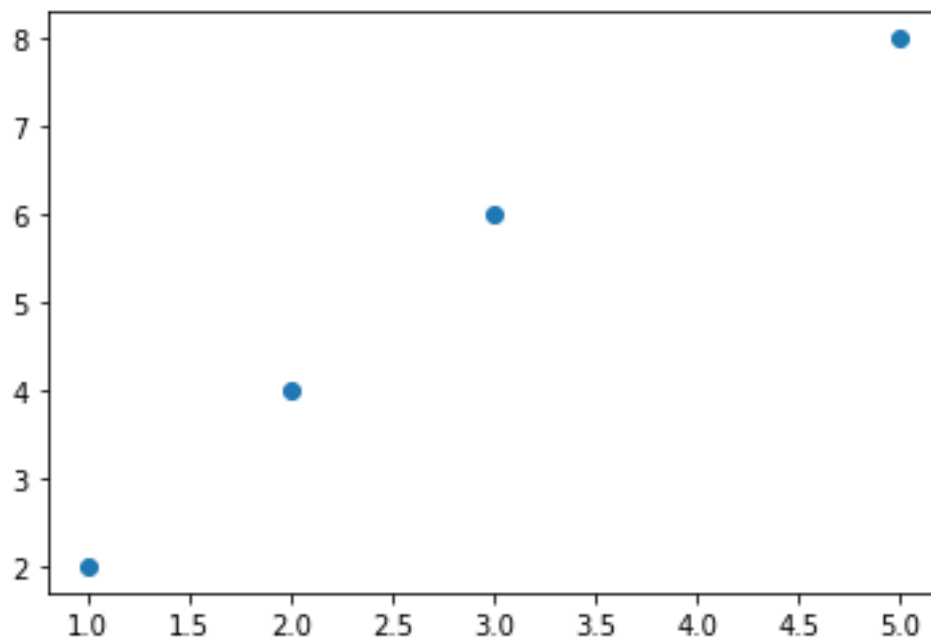
Lệnh Python để gọi tạo Figure và Axe từ hàm `.subplots()` như sau:

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
```

Dùng Axe để vẽ 1 biểu đồ

Vẽ biểu đồ từ 2 dãy số:

```
x = [1, 2, 3, 5]
y = [2, 4, 6, 8]
ax.plot(x, y, 'o')
```



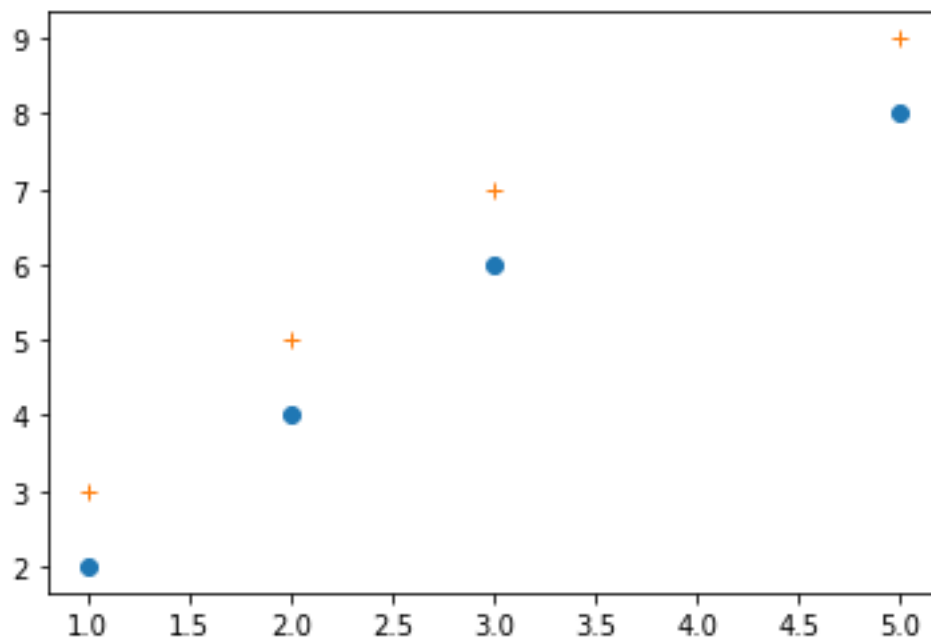
Dùng Axe để vẽ nhiều biểu đồ

Bạn hình dung có một bức tranh (Figure) mà trong đó có 2 biểu đồ được vẽ từ 2 bộ dãy số (x, y) và (x1, y1). Bạn có thể tự thêm các dãy số mới và gọi hàm plot với mark

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()

x = [1, 2, 3, 5]
y = [2, 4, 6, 8]
ax.plot(x, y, 'o')

x1 = x
y1 = [3, 5, 7, 9]
ax.plot(x1, y1, '+')
```



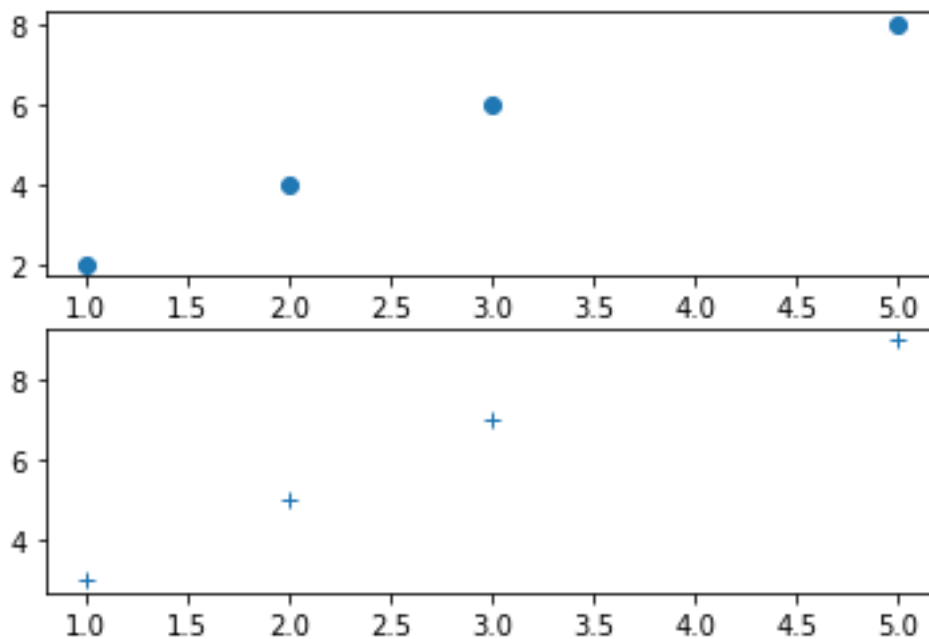
Vẽ biểu đồ trên nhiều Figure

Đoạn code sau tạo ra biểu đồ gồm 2 Figures bằng cách gọi hàm `.subplot` (số lượng figure) từ module `matplotlib.pyplot`:

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(2)

x = [1, 2, 3, 5]
y = [2, 4, 6, 8]
ax[0].plot(x, y, 'o')

x1 = x
y1 = [3, 5, 7, 9]
ax[1].plot(x1, y1, '+')
```



Thiết lập nhãn trục x, y cho biểu đồ con

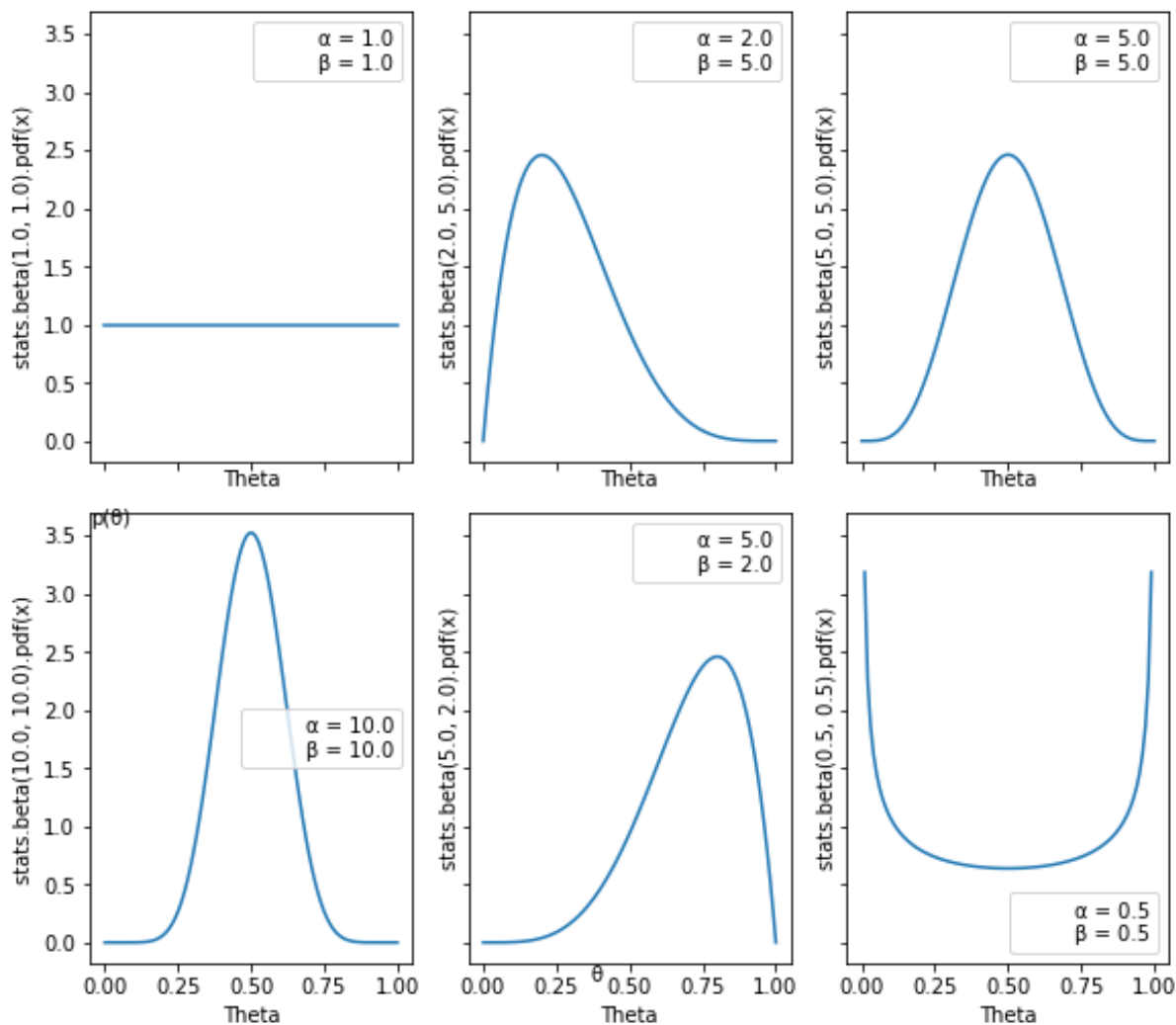
Tham khảo ví dụ sau:

```
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats

params = [[1, 1], [2,5], [5,5], [10,10], [5,2], [0.5, 0.5]];
x = np.linspace(0, 1, 100)
f, ax = plt.subplots(2, 3, sharex=True, sharey=True,
                    figsize=(8, 7), constrained_layout=True)
for index, p in enumerate(params):
    a = p[0];
    b = p[1];
    i = index // 3;
    j = index % 3;

    print('i=%d; j=%d; a=%f; b=%f' % (i, j, a, b))
    print(type)
    y = stats.beta(a, b).pdf(x)
    ax[i,j].plot(x, y)
    ax[i,j].plot(0, 0, label="α = {:.21f}\nβ =
    {:.21f}".format(a, b), alpha=0)
```

```
ax[i,j].legend()
ax[i,j].set_xlabel('Theta')
ax[i,j].set_ylabel('stats.beta({:2.1f},
{:2.1f}).pdf(x)'.format(a,b))
# ax[1,0].set_yticks([])
# ax[1,0].set_xticks(np.linspace(0, 1, 5))
f.text(0.5, 0.05, 'θ', ha='center')
f.text(0.07, 0.5, 'p(θ)', va='center', rotation=0)
```



Trang trí biểu đồ

Thêm tên biểu đồ

Có thể dùng một trong 2 lệnh sau:

- `plt.figure().suptitle('Tiêu đề')`
- `plt.title('Tiêu đề')`

Thêm tên trục x, trục y

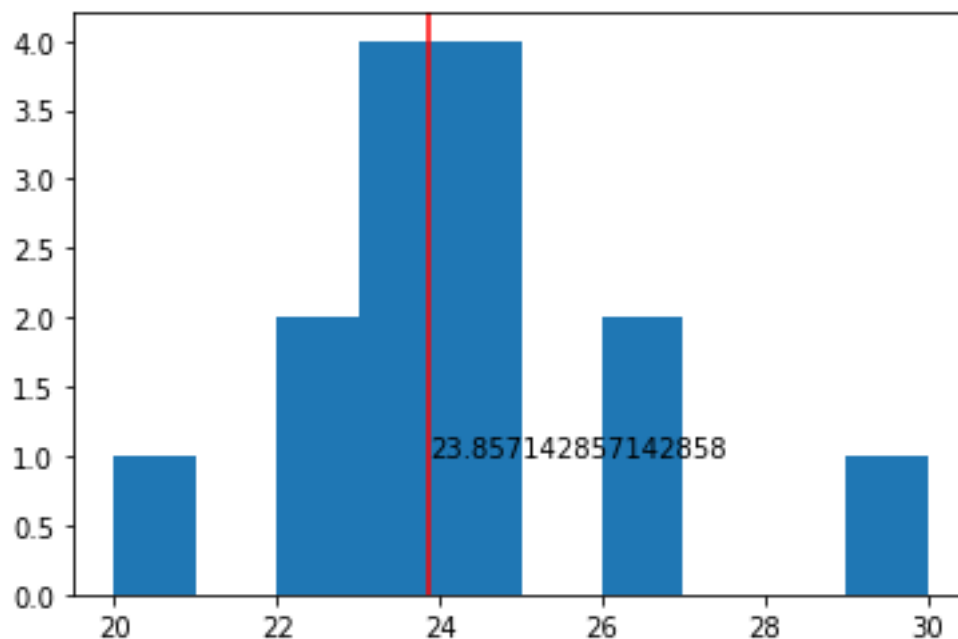
```
plt.xlabel('Nhân trục x')  
plt.ylabel('Nhân trục y')
```

Vẽ thêm text tại vị trí x, y

Sử dụng hàm `.text(x, y, value)` để hiển thị một giá trị tại vị trí x,y:

Ví dụ vẽ thêm giá trị trung bình

```
import pandas as pd  
import matplotlib.pyplot as plt  
  
df =  
pd.read_csv('https://thachln.github.io/datasets/TuyenVN.csv')  
plt.hist(df['age'])  
plt.axvline(df['age'].mean(), color='red')  
plt.text(df['age'].mean(), 1, df['age'].mean())
```



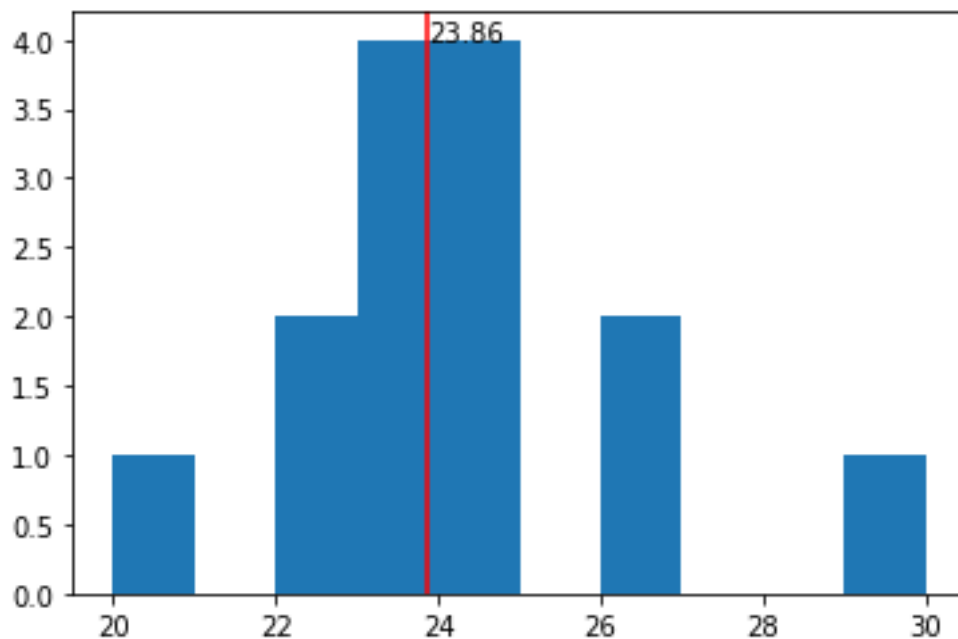
Cải tiến một chút cho biểu đồ:

```
import pandas as pd  
import matplotlib.pyplot as plt  
  
df =  
pd.read_csv('https://thachln.github.io/datasets/TuyenVN.csv')  
hist = plt.hist(df['age'])
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
age_mean = df['age'].mean()
plt.axvline(age_mean, color='red')

plt.text(age_mean, hist[0].max(), '%.2f' % age_mean)
```



Việc giải thích những chỗ thay đổi (bôi đậm) thì dành cho bạn nhé!

Thêm legend

Để giải thích thêm cho từng loại dữ liệu trong biểu đồ thì khi plot kèm theo tham số label. Tiếp theo gọi hàm legend(). Xem phần in đậm trong đoạn chương trình sau:

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

gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08,
7.02]

year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017,
2018,2019]

cpi = [9.19,18.58, 9.21, 6.60, 4.09, 0.63, 2.66, 3.53, 3.54,
3.89]

# Tạo data frame
df = pd.DataFrame({'gdp': gdp, 'year': year, 'cpi': cpi})

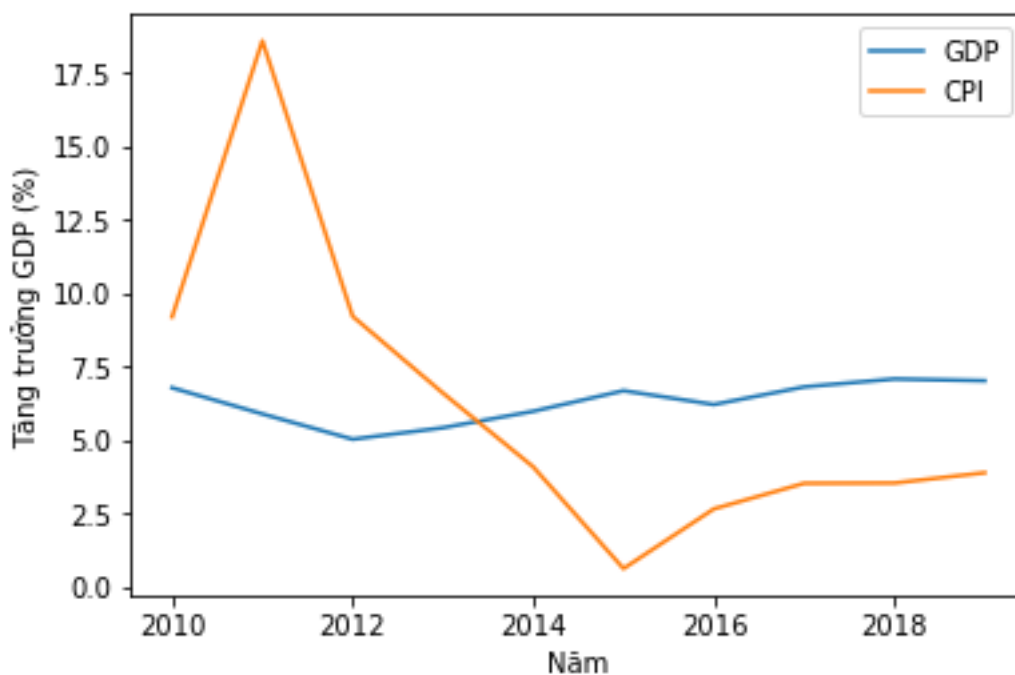
plt.figure()
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
plt.title('Biểu đồ tăng trưởng GDP của Việt Nam từ năm 2010 đến 2019')

plt.xlabel('Năm')
plt.ylabel('Tăng trưởng GDP (%)')
plt.plot(df['year'], df['gdp'], label = 'GDP')
plt.plot(df['year'], df['cpi'], label = 'CPI')
plt.legend()
```

Biểu đồ tăng trưởng GDP của Việt Nam từ năm 2010 đến 2019



Vẽ biểu đồ line với data frame từ file CSV

Trong trường hợp bạn có sẵn file CSV thì có thể đọc dữ liệu vào data frame với thư viện pandas và vẽ biểu đồ cho hai cột dữ liệu đơn giản như sau:

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

df =
pd.read_csv('https://thachln.github.io/datasets/TuyenVN.csv')
df.head()

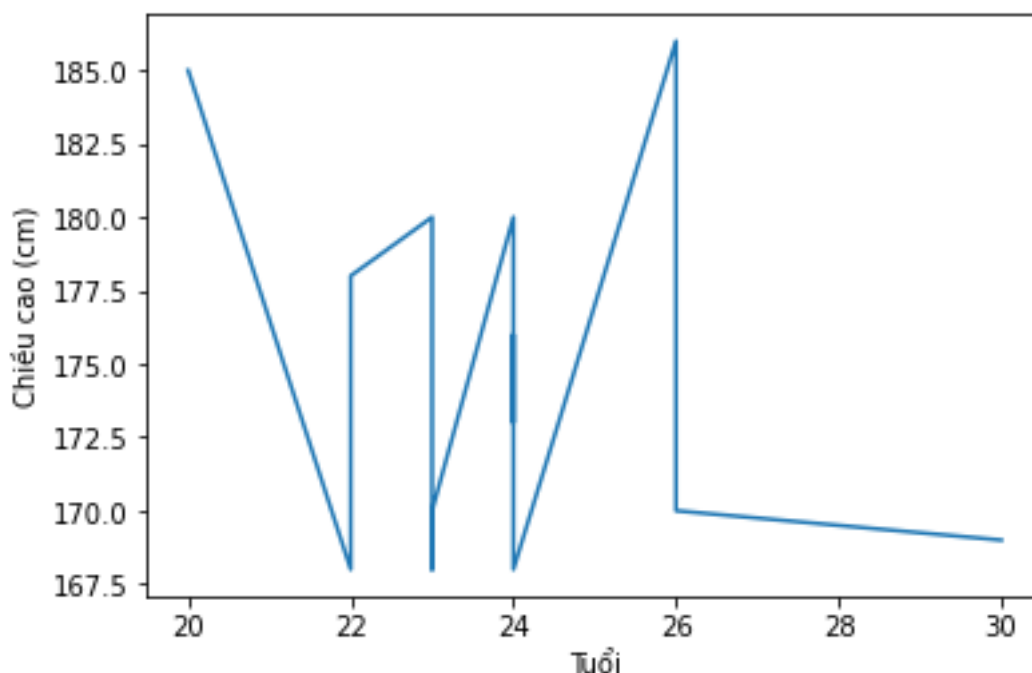
# Sắp xếp dữ liệu theo tuổi
df = df.sort_values(by = 'age')
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
plt.figure().suptitle('Biểu đồ tuổi và chiều cao của nam  
tuyển thủ bóng đá.')  
plt.xlabel('Tuổi')  
plt.ylabel('Chiều cao (cm)')  
  
plt.plot('age', 'height', data = df)
```

Trong đoạn code trên có dùng hàm `df.sort_values(by = 'age')` để sắp xếp lại dữ liệu theo cột `age`. Kết quả biểu đồ:

Biểu đồ tuổi và chiều cao của nam tuyển thủ bóng đá.



Vẽ biểu đồ theo nhu cầu quan sát dữ liệu

Đọc dữ liệu

Đọc dữ liệu từ nghiên cứu về sức khỏe, luyện tập lại một số lệnh, kỹ thuật:

- Thêm cột mới cho DataFrame.
- Dùng lệnh `print` để hiển thị thông tin về vài dòng dữ liệu, thông tin về tên cột.

```
import pandas as pd  
fp =  
'https://thachln.github.io/datasets/sample_health_vn.csv'  
df = pd.read_csv(fp)  
df['whtr'] = df['waist'] / df['height']
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

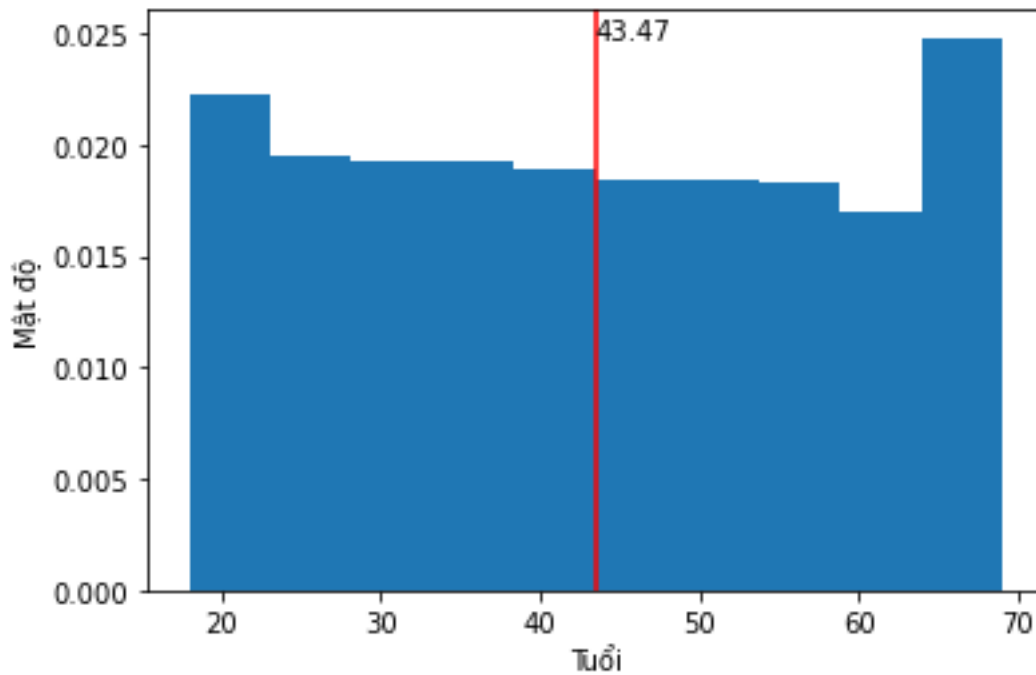
```
print(df.head())  
print(df.columns)
```

```
   id  age  sex  height  waist  risk  weight  hit  life  
0    1   23    0   148     69   0.0     38   77  77.50  
1    2   26    1   171     82   0.0     57   95  75.37  
2    3   66    1   164     86   0.0     77   89  66.09  
3    4   55    1   170     89   0.0     73   90  69.59  
4    5   30    0   154     76   0.0     59   83  70.01  
Index(['id', 'age', 'sex', 'height', 'waist', 'risk', 'weight', 'hit',  
       'life',  
       'whtr'],  
      dtype='object')
```

Xem phân bố của 1 biến với hàm .hist

```
import matplotlib.pyplot as plt  
hist = plt.hist(df['age'], density=True)  
plt.xlabel('Tuổi')  
plt.ylabel('Mật độ')  
  
age_mean = df['age'].mean()  
plt.axvline(age_mean, color='red')  
  
plt.text(age_mean, hist[0].max(), '%.2f' % age_mean)  
  
plt.show()
```

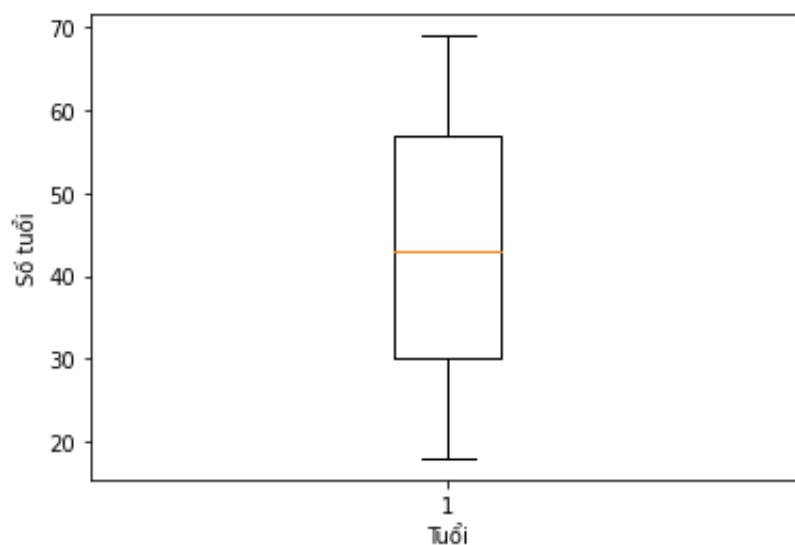
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



Xem dữ liệu tuổi với boxplot

```
import matplotlib.pyplot as plt
plt.boxplot(df['age'])
plt.xlabel('Tuổi')
plt.ylabel('Số tuổi')

plt.show()
```



Tham khảo thêm:

```
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.boxplot.html
```

Xem dữ liệu theo thời gian

Sử dụng hàm `.plot` với trục x là thời gian

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

fp =
'https://thachln.github.io/datasets/vnindex_20200424.txt'
df1 = pd.read_csv(fp)

df1['date'] = pd.to_datetime(df1['<DTYYYYMMDD>'],
format='%Y%m%d')

plt.plot(df1['date'], df1['<High>'])
plt.title('Chi số VNIndex.')
```



Sử dụng hàm `.plot_date`

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

gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08,
7.02]
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

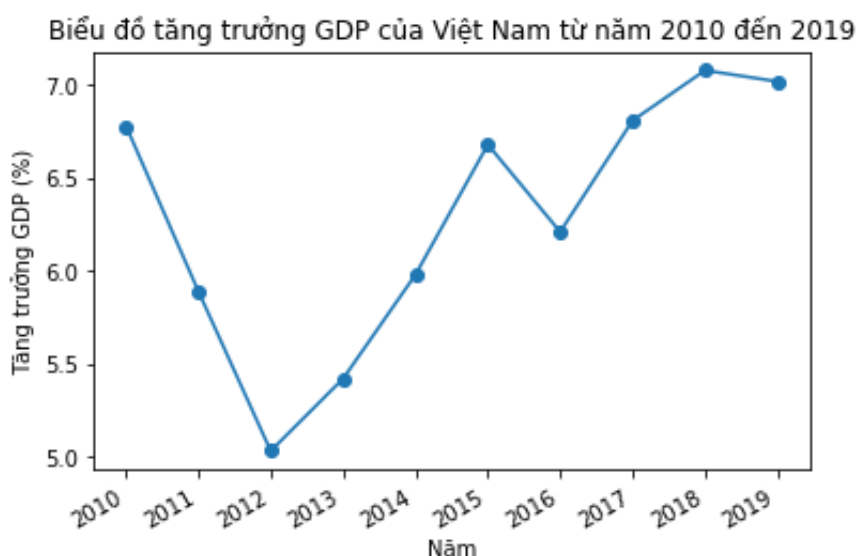
# Tạo data frame
df1 = pd.DataFrame({'gdp': gdp, 'year': year})
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
df1.index = df1['year']
plt.title('Biểu đồ tăng trưởng GDP của Việt Nam từ năm 2010
đến 2019')
plt.xlabel('Năm')
plt.ylabel('Tăng trưởng GDP (%)')

df1['date'] = pd.to_datetime(df1['year'], format='%Y')

plt.gcf().autofmt_xdate()
plt.plot_date(df1['date'], df1['gdp'], linestyle='solid')
```



Ghi chú:

- Lệnh `plt.gcf().autofmt_xdate()` sẽ tự động định dạng độ nghiêng cho nhãn trên trục x sao cho không chồng lên nhau.

So sánh 2 biến với biểu đồ bar

Đếm số lượng nam và nữ trong dữ liệu của một nghiên cứu và vẽ biểu đồ bar:

```
import pandas as pd
fp =
'https://thachln.github.io/datasets/sample_health_vn.csv'
df = pd.read_csv(fp)
df['whtr'] = df['waist'] / df['height']

df.loc[df['sex'] == 0, 'sex_label'] = 'Nữ'
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
df.loc[df['sex'] == 1, 'sex_label'] = 'Nam'

print(df.head())
print(df.columns)

group_sex_age_count =
df.groupby('sex_label')['age'].count().reset_index()
print(group_sex_age_count)

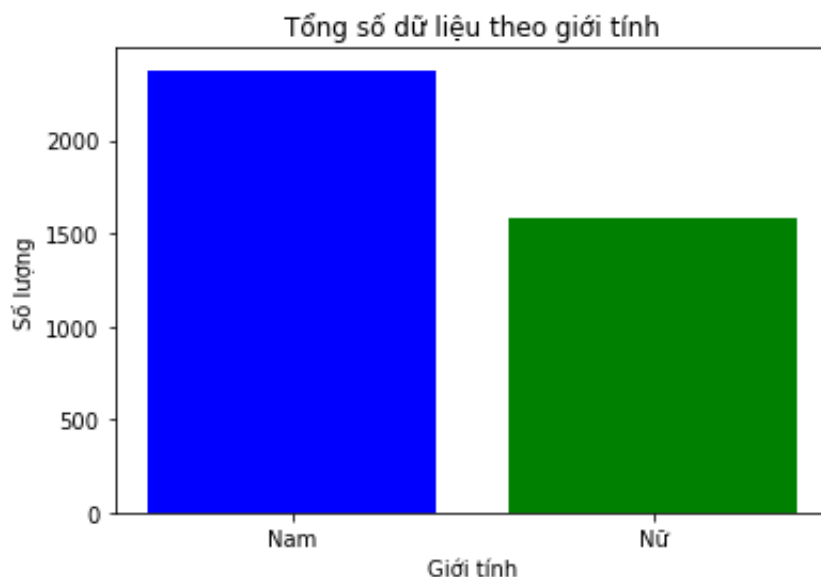
import matplotlib.pyplot as plt

plt.bar(group_sex_age_count['sex_label'],
group_sex_age_count['age'], color=['b', 'g'])
plt.title('Tổng số dữ liệu theo giới tính')
plt.xlabel('Giới tính')
plt.ylabel('Số lượng')

plt.show()
```

```
   id  age  sex  height  waist  risk  weight  hit  life  wht
r sex_label
0  1   23   0   148     69   0.0     38   77  77.50  0.466216
Nữ
1  2   26   1   171     82   0.0     57   95  75.37  0.479532
Nam
2  3   66   1   164     86   0.0     77   89  66.09  0.524390
Nam
3  4   55   1   170     89   0.0     73   90  69.59  0.523529
Nam
4  5   30   0   154     76   0.0     59   83  70.01  0.493506
Nữ
Index(['id', 'age', 'sex', 'height', 'waist', 'risk', 'weight', 'hit',
      'life',
      'whtr', 'sex_label'],
      dtype='object')
sex_label  age
0         Nam  2377
```

```
1         Nữ  1583
```



So sánh 1 biến theo thời gian

Đoạn chương trình bên dưới vẽ 2 bức tranh trong cùng một biểu đồ. Mỗi bức tranh là một biểu đồ line để theo dõi giá trị (GDP hoặc CPI) theo thời gian.

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

gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08,
7.02]
cpi = [9.19, 18.58, 9.21, 6.60, 4.09, 0.63, 2.66, 3.53, 3.54,
3.89]
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

# Tạo data frame
df = pd.DataFrame({'gdp': gdp, 'year': year, 'cpi': cpi})

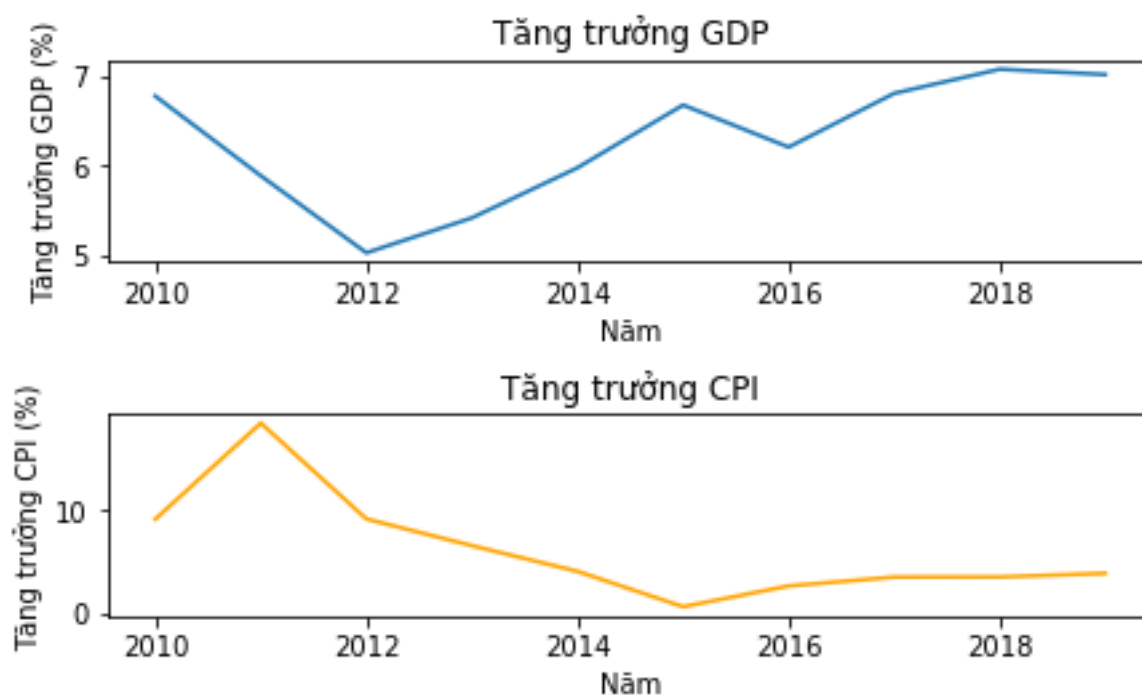
fig, (ax1, ax2) = plt.subplots(2, constrained_layout=True)
ax1.set_title('Tăng trưởng GDP')
ax1.set_xlabel('Năm')
ax1.set_ylabel('Tăng trưởng GDP (%)')
ax1.plot(df['year'], df['gdp'])

ax2.set_title('Tăng trưởng CPI')
ax2.set_xlabel('Năm')
ax2.set_ylabel('Tăng trưởng CPI (%)')
ax2.plot(df['year'], df['cpi'], color='orange')
```

```
fig.suptitle('Biểu đồ tăng trưởng GDP và CPI của Việt Nam từ năm 2010 đến 2019')

plt.show()
```

Biểu đồ tăng trưởng GDP và CPI của Việt Nam từ năm 2010 đến 2019



So sánh 2 biến theo thời gian

Cải tiến đoạn chương trình ở trên 1 chút để vẽ 2 đường tăng trưởng GDP, CPI trong cùng một biểu đồ.

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

gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08,
       7.02]
cpi = [9.19, 18.58, 9.21, 6.60, 4.09, 0.63, 2.66, 3.53, 3.54,
       3.89]
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

# Tạo data frame
df = pd.DataFrame({'gdp': gdp, 'year': year, 'cpi': cpi})

fig, ax = plt.subplots(1, constrained_layout=True)
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

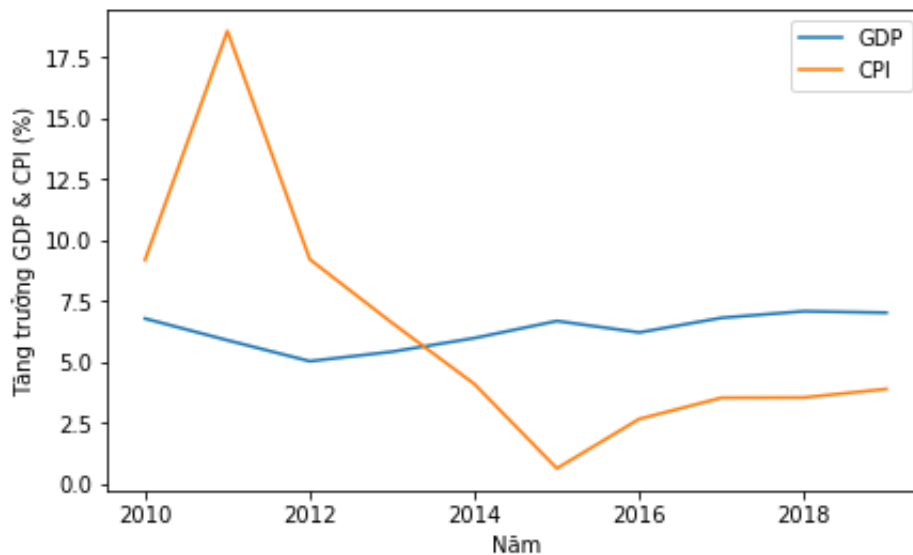
```
ax.plot(df['year'], df['gdp'], label = 'GDP')
ax.plot(df['year'], df['cpi'], label = 'CPI')

fig.suptitle('Biểu đồ tăng trưởng GDP và CPI của Việt Nam từ
năm 2010 đến 2019')

plt.xlabel('Năm')
plt.ylabel('Tăng trưởng GDP & CPI (%)')

plt.legend()
plt.show()
```

Biểu đồ tăng trưởng GDP và CPI của Việt Nam từ năm 2010 đến 2019

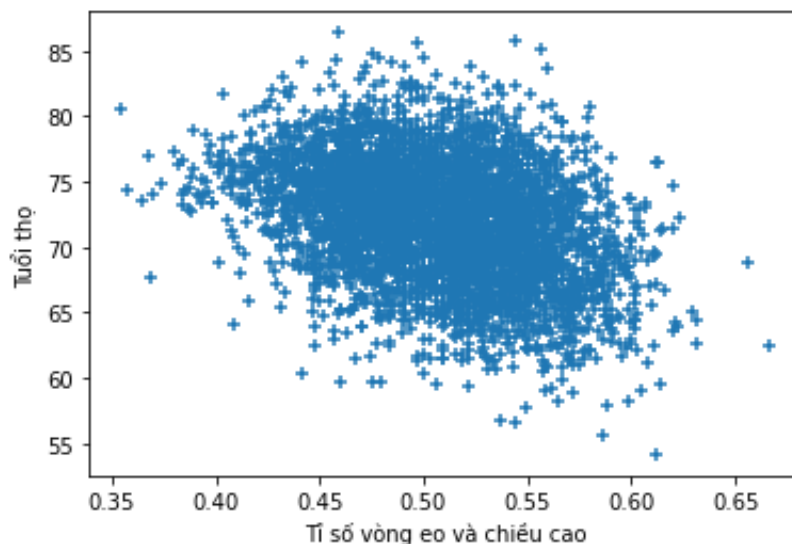


Xem tương quan giữa 2 biến với hàm `.scatter`

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

fp =
'https://thachln.github.io/datasets/sample_health_vn.csv'
df = pd.read_csv(fp)
df['whtr'] = df['waist'] / df['height']

plt.scatter(df['whtr'], df['life'], marker='+')
# Gọi scatter với DataFrame: x, y là tên cột
# plt.scatter('whtr', 'life', data=df, marker='+')
plt.xlabel('Ti số vòng eo và chiều cao')
plt.ylabel('Tuổi thọ')
plt.show()
```



Tham khảo thêm:

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.scatter.html

Lưu biểu đồ

Hàm `plt.savefig(fname)` sẽ lưu Figure hiện hành. Các tham số tùy chọn gồm `dpi`, `format`, `transparent`.

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

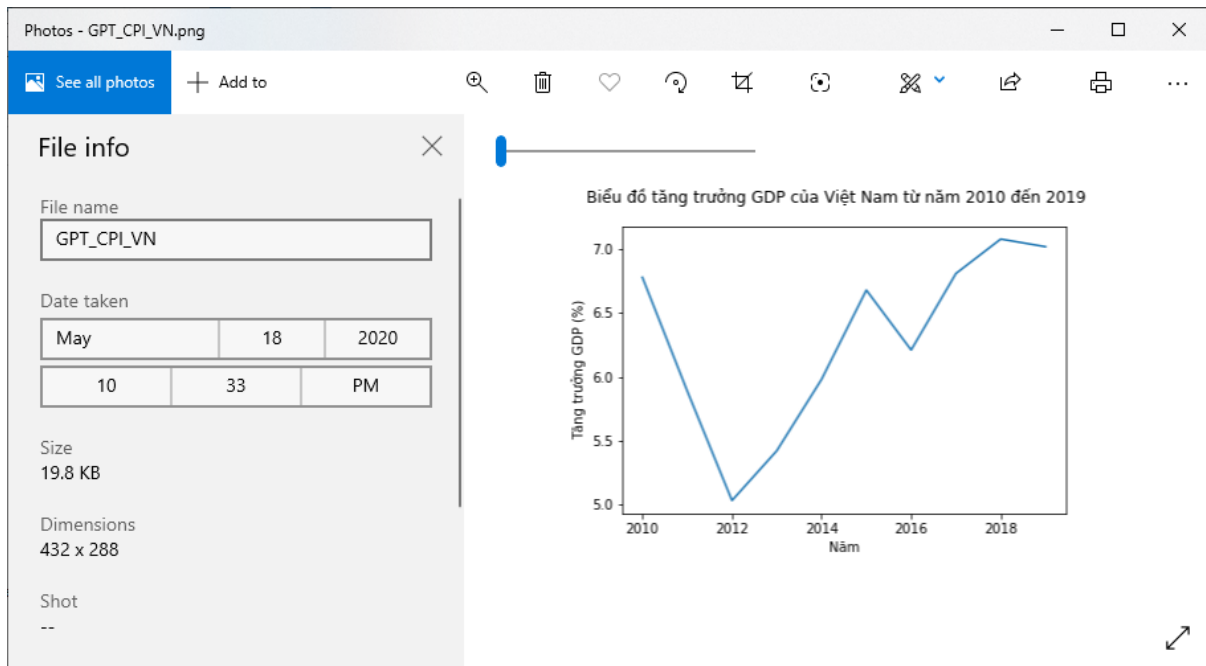
gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08,
       7.02]
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

# Tạo data frame
df = pd.DataFrame({'gdp': gdp, 'year': year})

plt.figure().suptitle('Biểu đồ tăng trưởng GDP của Việt Nam
từ năm 2010 đến 2019')
plt.xlabel('Năm')
plt.ylabel('Tăng trưởng GDP (%)')
plt.plot('year', 'gdp', data = df)
plt.savefig("D:/GPT_CPI_VN.png")
```

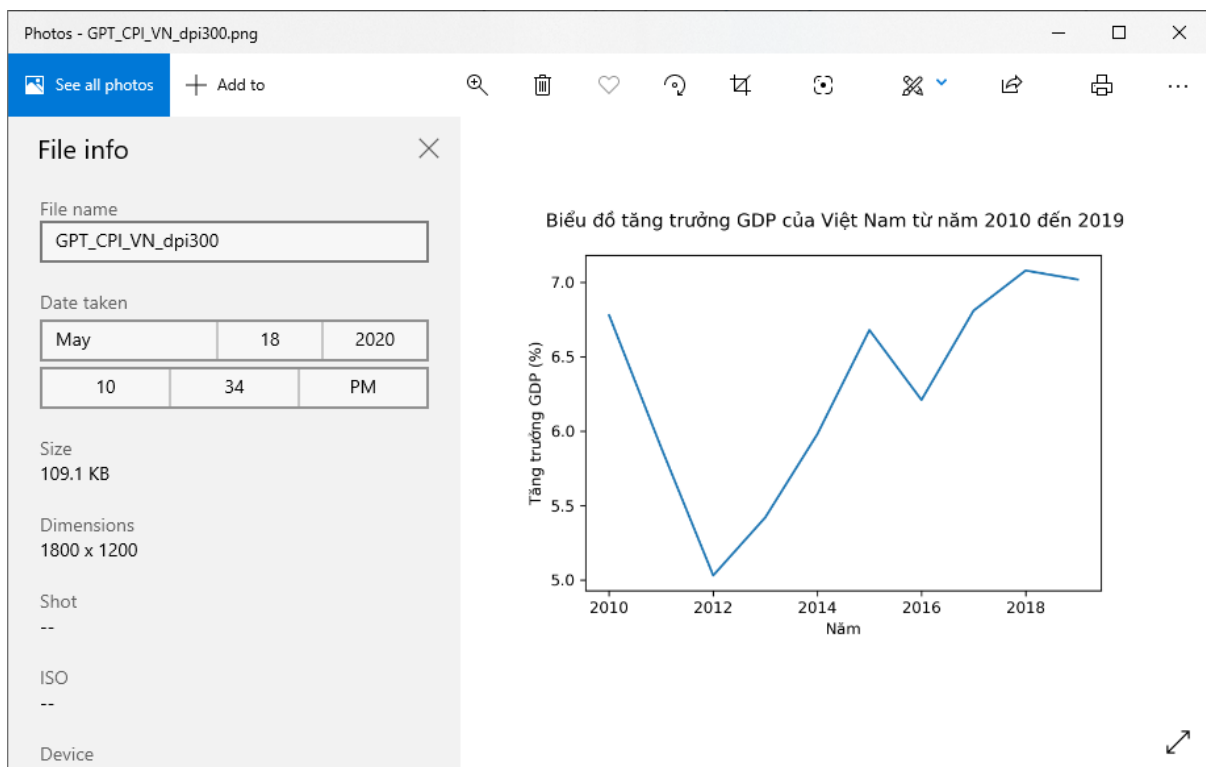
Kết quả:

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



Thử sửa lại dòng cuối cùng

```
plt.savefig("D:/GPT_CPI_VN_dpi300.png", dpi=300)
```



Hãy quan sát kích thước ảnh và Dimensions khác nhau giữa 2 lệnh trên.

Thử thêm tham số `transparent` và kiểm tra tính trong suốt (`transparent`) của ảnh:

```
plt.savefig("D:/GPT_CPI_VN_0.png", transparent = 0)
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
plt.savefig("D:/GPT_CPI_VN_1.png", transparent = 0.1)
plt.savefig("D:/GPT_CPI_VN_2.png", transparent = 0.5)
plt.savefig("D:/GPT_CPI_VN_3.png", transparent = 1)
```

Liên lạc | Tài trợ

Lê Ngọc Thạch	
  (+084) 0908 550 642	 facebook.com/ThachLN
Website: https://xmyworkspace.com/learn/sponsor	

