



Bài giảng môn học:
Khoa học dữ liệu (7080509)

Chương 4: Một số thư viện Python quan trọng trong khoa học dữ liệu – Phần 4

Đặng Văn Nam

dangvannam@hmg.edu.vn

Nội dung phần 4 – Thư viện Matplotlib:

1. Tầm quan trọng của trực quan hóa dữ liệu
2. Thư viện trực quan hóa dữ liệu với Python
3. Biểu đồ Line chart
4. Biểu đồ Bar chart
5. Biểu đồ Pie chart
6. Biểu đồ Scatter plot



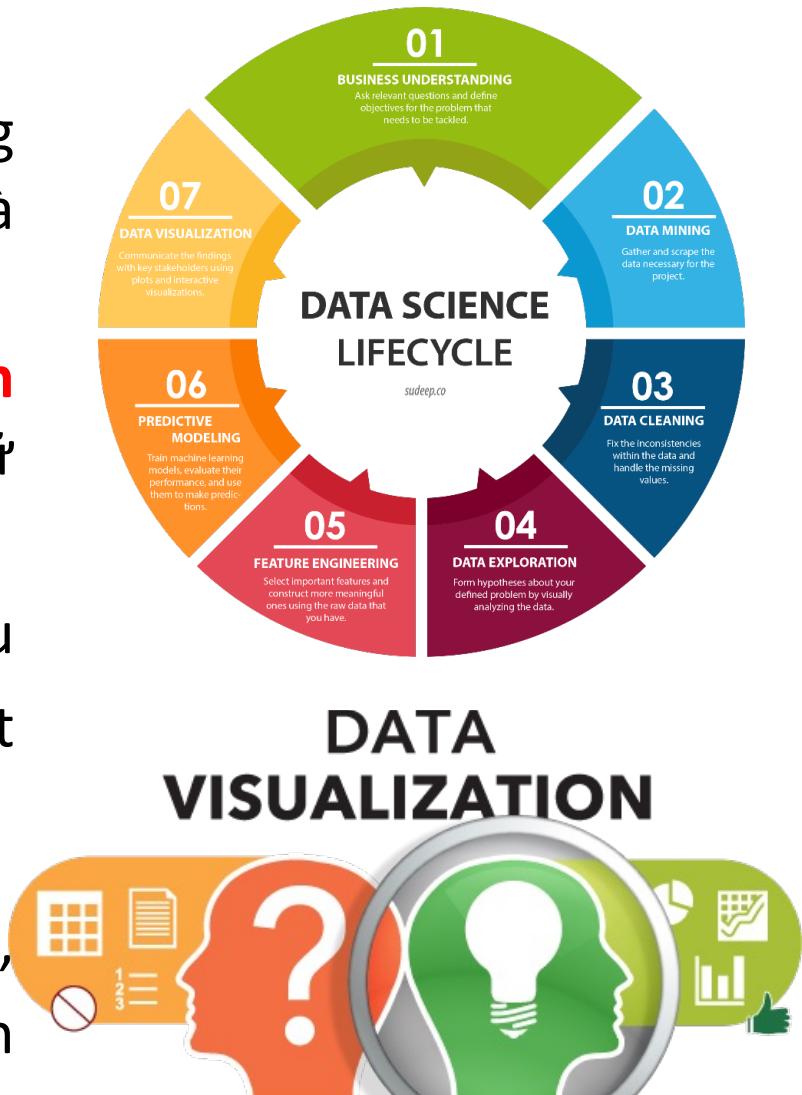


1. Tầm quan trọng của trực quan hóa dữ liệu

Tâm quan trọng

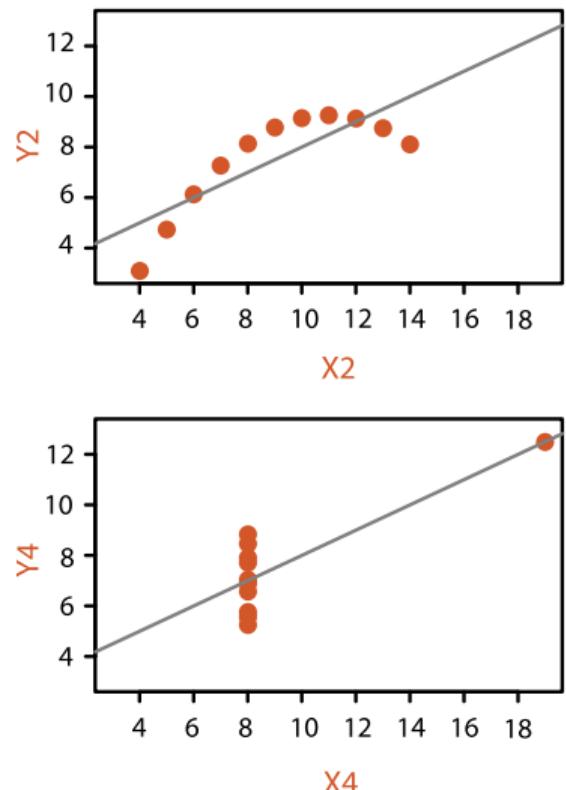
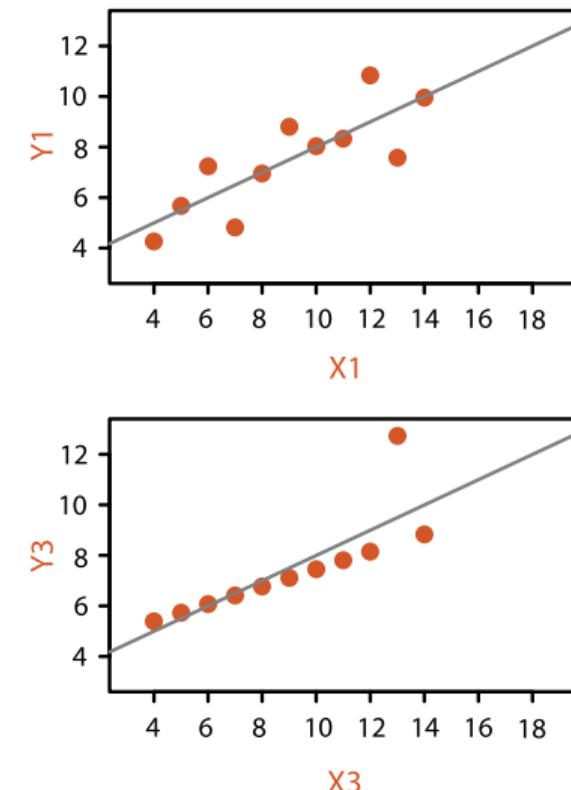
Trực quan hóa dữ liệu là gì?

- Trực quan hóa dữ liệu là việc **biểu diễn đồ họa** các thông tin trừu tượng nhằm 2 mục đích: Phân tích dữ liệu và truyền thông.
- Trực quan hóa dữ liệu là **một công cụ mạnh mẽ để khám phá và trích rút các thông tin có giá trị (insight)** từ tập dữ liệu.
- Bản chất của Trực quan hóa dữ liệu là sự trình bày dữ liệu theo định dạng hình ảnh hoặc đồ họa, từ đó truyền đạt thông tin rõ ràng và hiệu quả cho người dùng.
- Là yếu tố giao tiếp bằng hình ảnh của phân tích dữ liệu, giúp chuyển đổi dữ liệu thành thông tin và thông tin thành thông tin hữu ích.



Tâm quan trọng

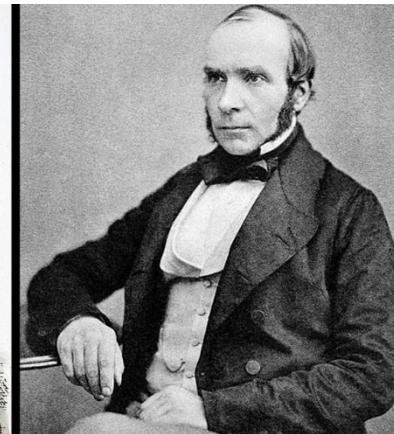
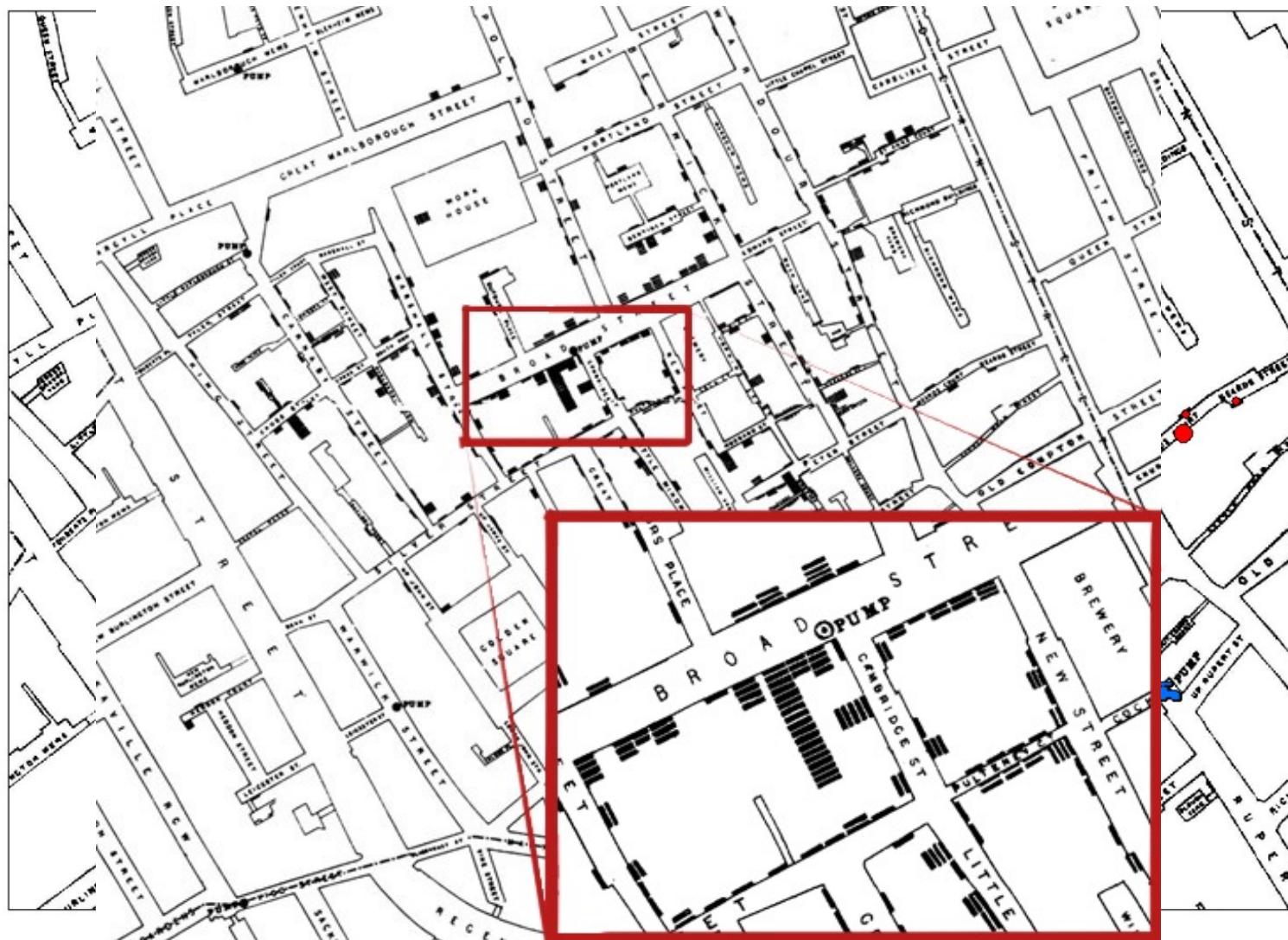
	1		2		3		4	
	X	Y	X	Y	X	Y	X	Y
	10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
	8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
	13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
	9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
	11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
	14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
	6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
	4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
	12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
	7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
	5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89
Mean	9.0	7.5	9.0	7.5	9.0	7.5	9.0	7.5
Variance	10.0	3.75	10.0	3.75	10.0	3.75	10.0	3.75
Correlation	0.816		0.816		0.816		0.816	



Tâm quan trọng

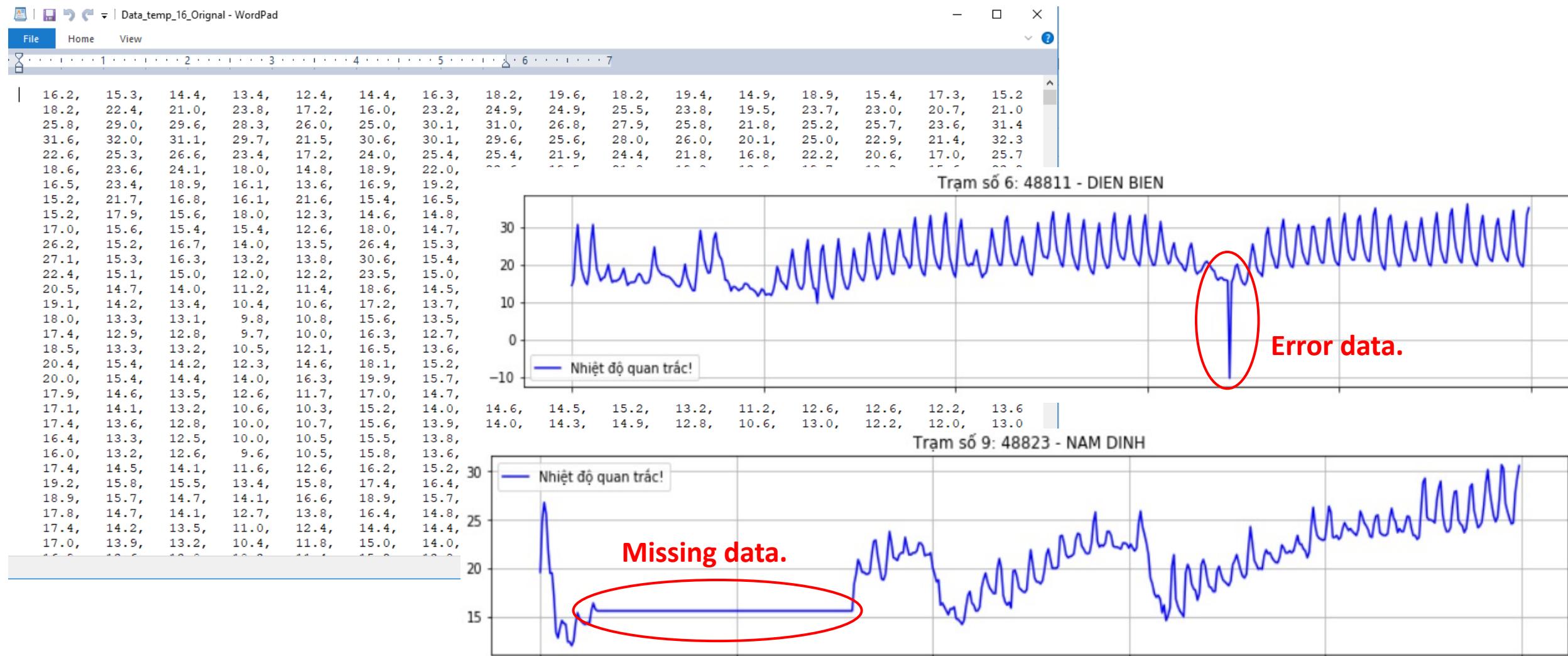


Tâm quan trọng



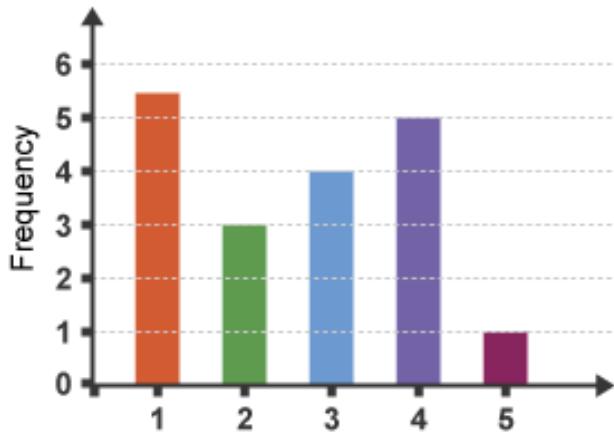
Dịch tả (London 1854), John Snow
600 người chết chỉ trong vài tuần...

Tâm quan trọng

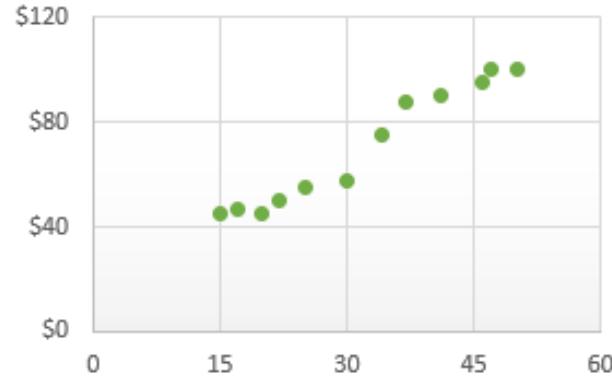


Các dạng biểu đồ quan trọng

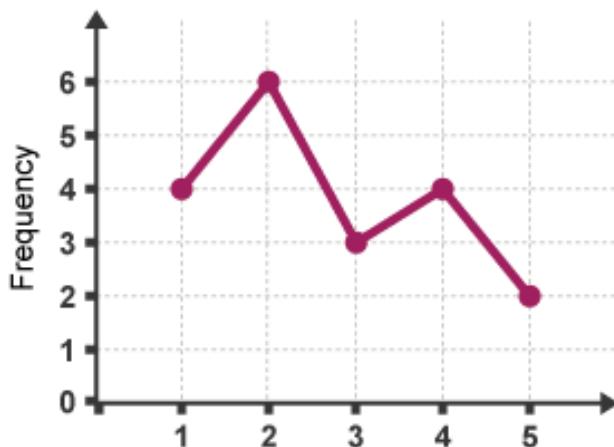
Bargraphs



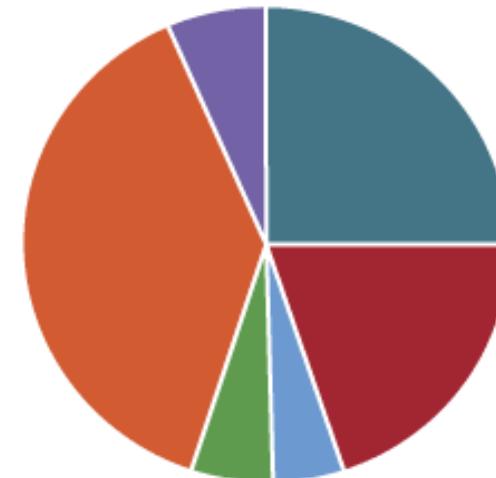
Scatter plot



Line graphs



Pie charts





2. Thư viện trực quan hoá với Python

Một số thư viện trực quan hóa

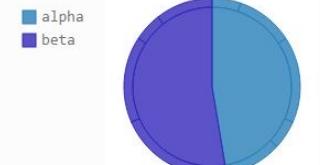
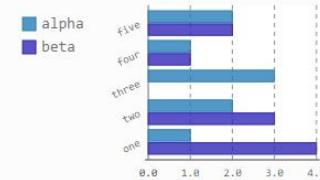
Có nhiều thư viện mạnh mẽ để trực quan hóa dữ liệu với ngôn ngữ lập trình Python.



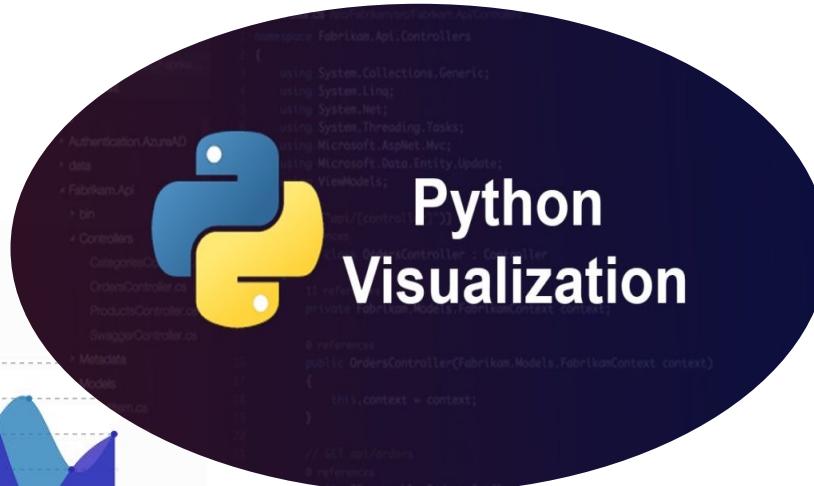
plotly

Pygal

Sexy python charting



matplotlib



seaborn

bokeh

Altair

Declarative Visualization in Python



Thư viện Matplotlib

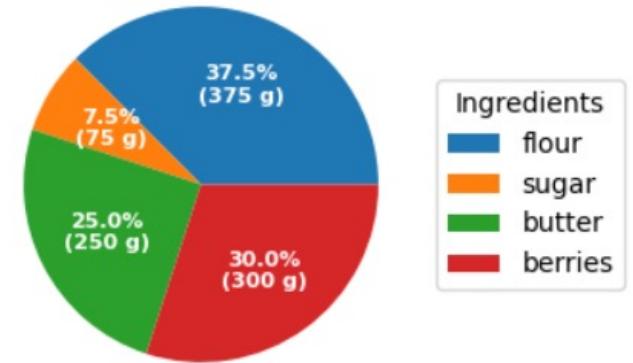
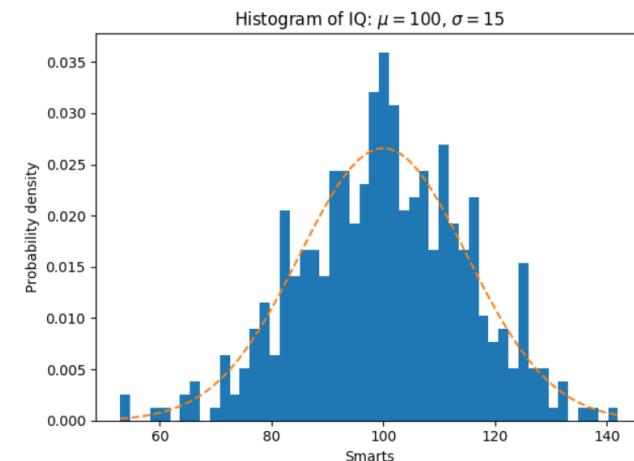
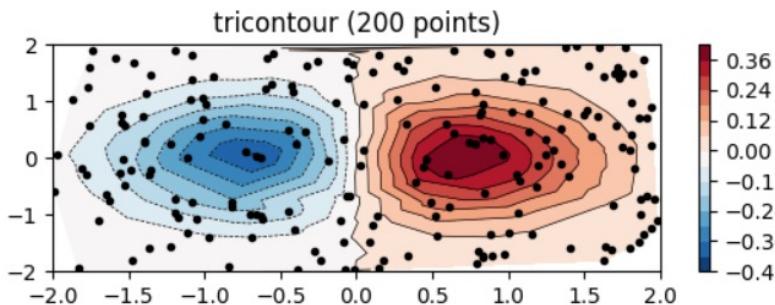
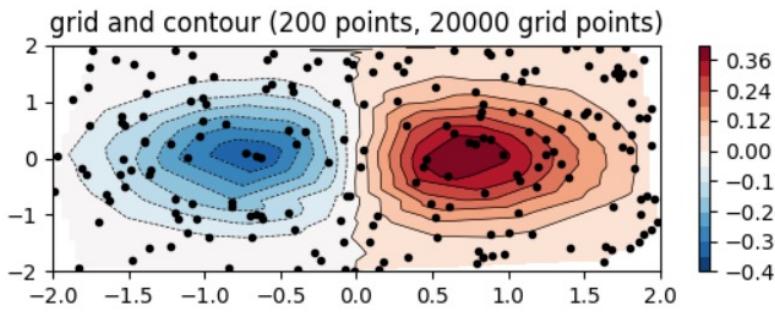
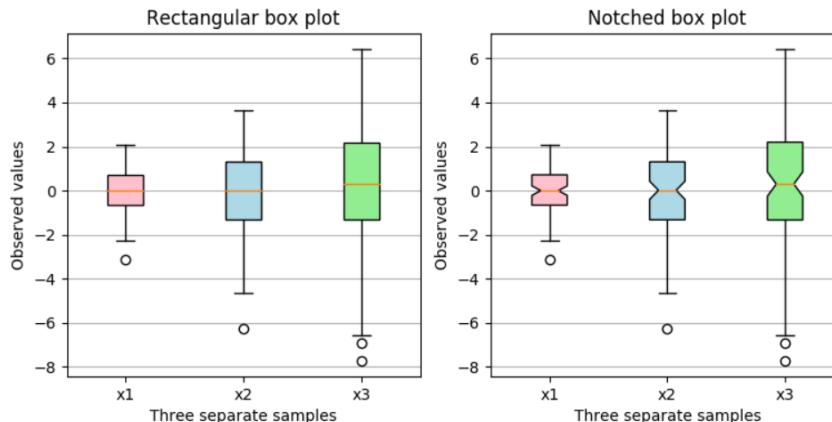
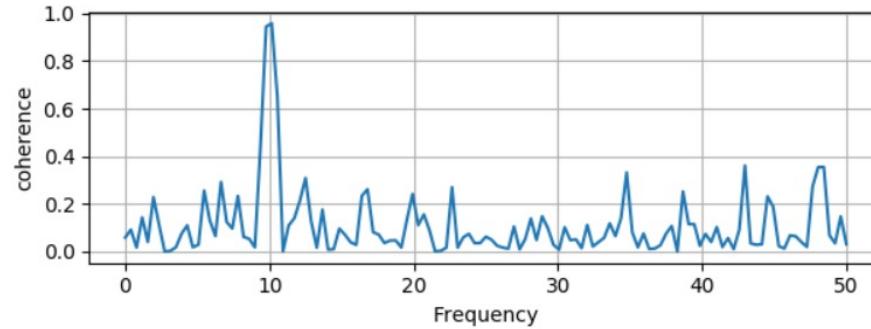
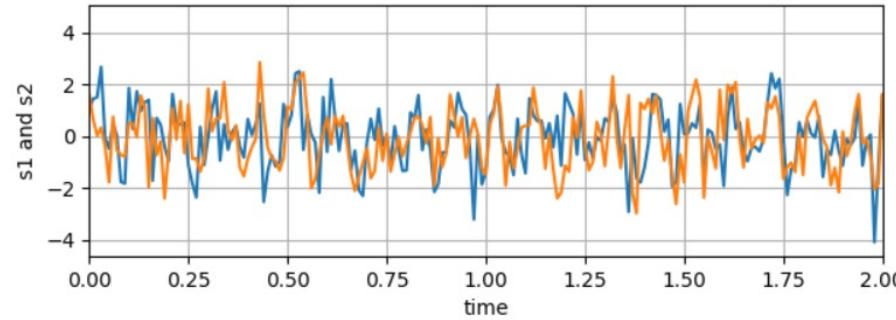
- **Matplotlib** là thư viện dùng để vẽ đồ thị rất mạnh mẽ, có cú pháp tương tự như Matlab. Thư viện này được phát triển sớm nhất, 2003.
- Hỗ trợ nhiều loại biểu đồ, đặc biệt là các loại được sử dụng trong nghiên cứu hoặc kinh tế như biểu đồng đường, cột, tần suất (histograms), tương quan, scatterplots...
- Cấu trúc của Matplotlib gồm nhiều phần, phục vụ cho các mục đích sử dụng khác nhau. Trong đó module pyplot được sử dụng nhiều nhất, có cú pháp tương tự như Matlab.
- Matplotlib miễn phí và mã nguồn mở.

Tham khảo:

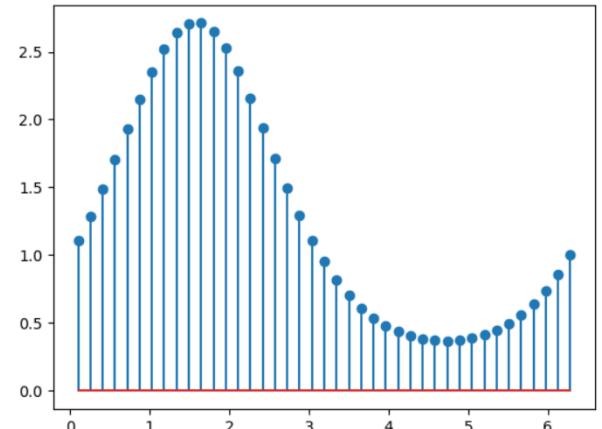
- + File: **CheatSheet-Matplotlib**
- + Link web: [Matplotlib package!](#)



Thư viện matplotlib

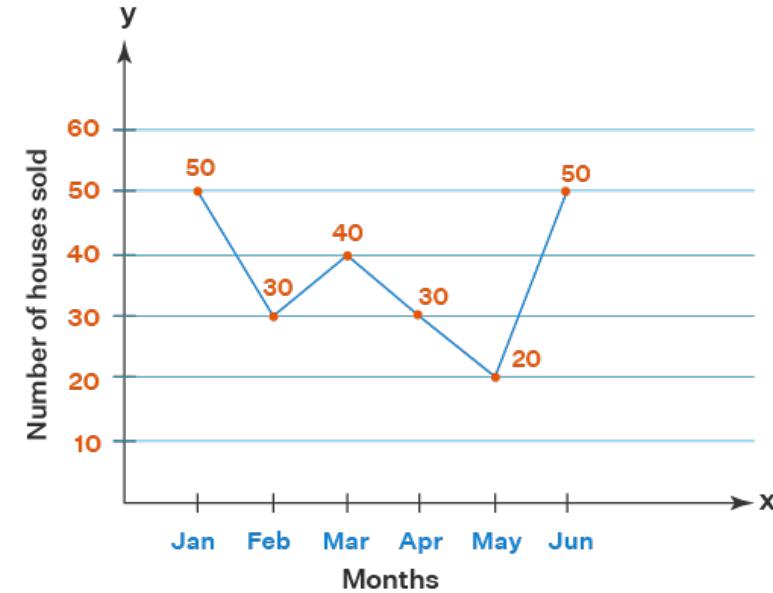
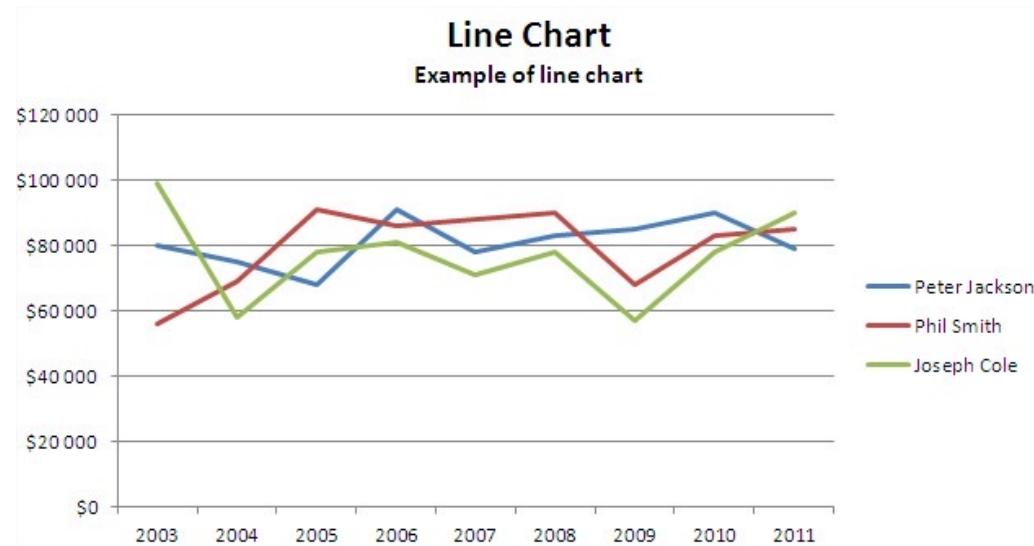


matplotlib



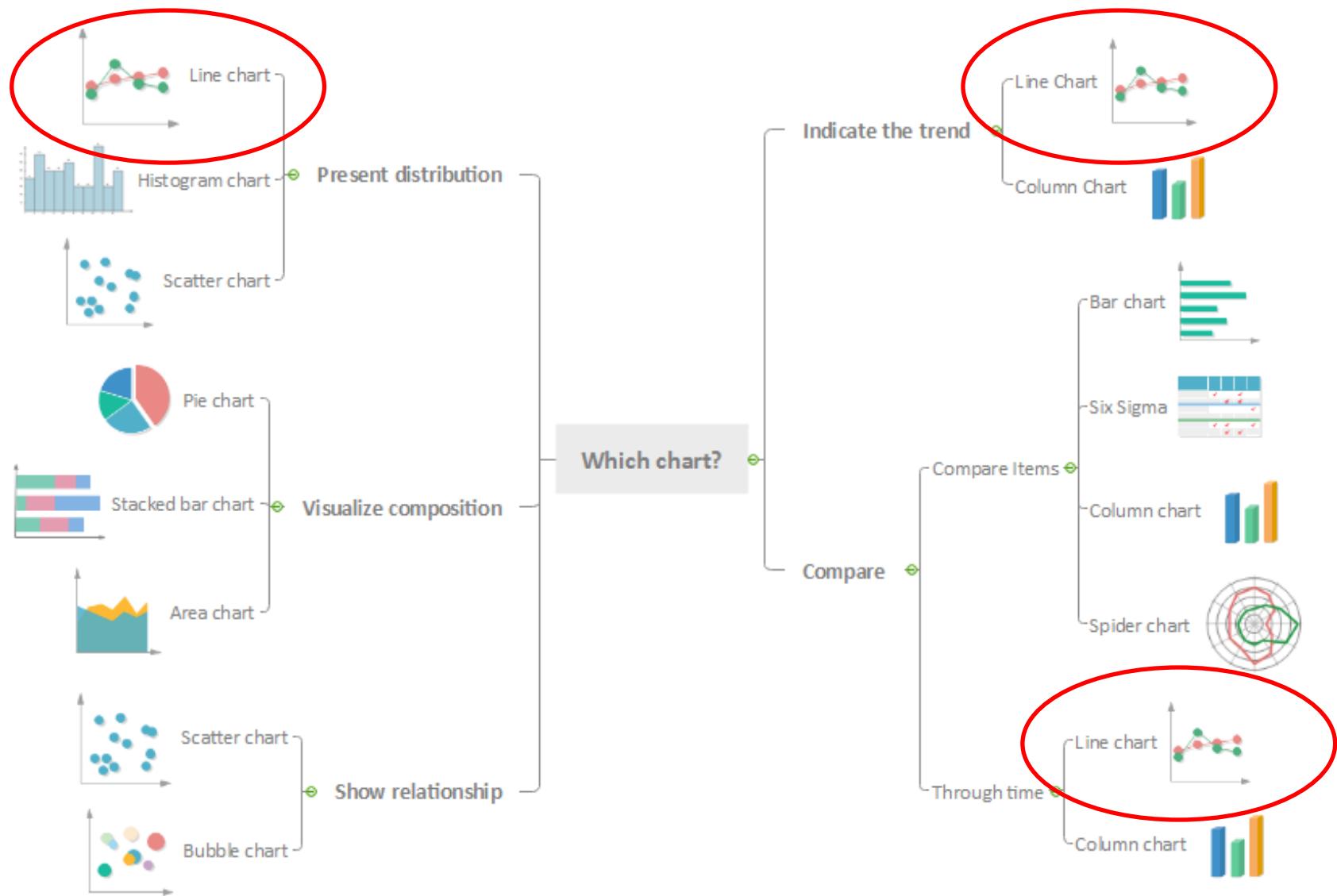
2. Biểu đồ đường (line chart)

Đồ thị dạng đường (line chart)



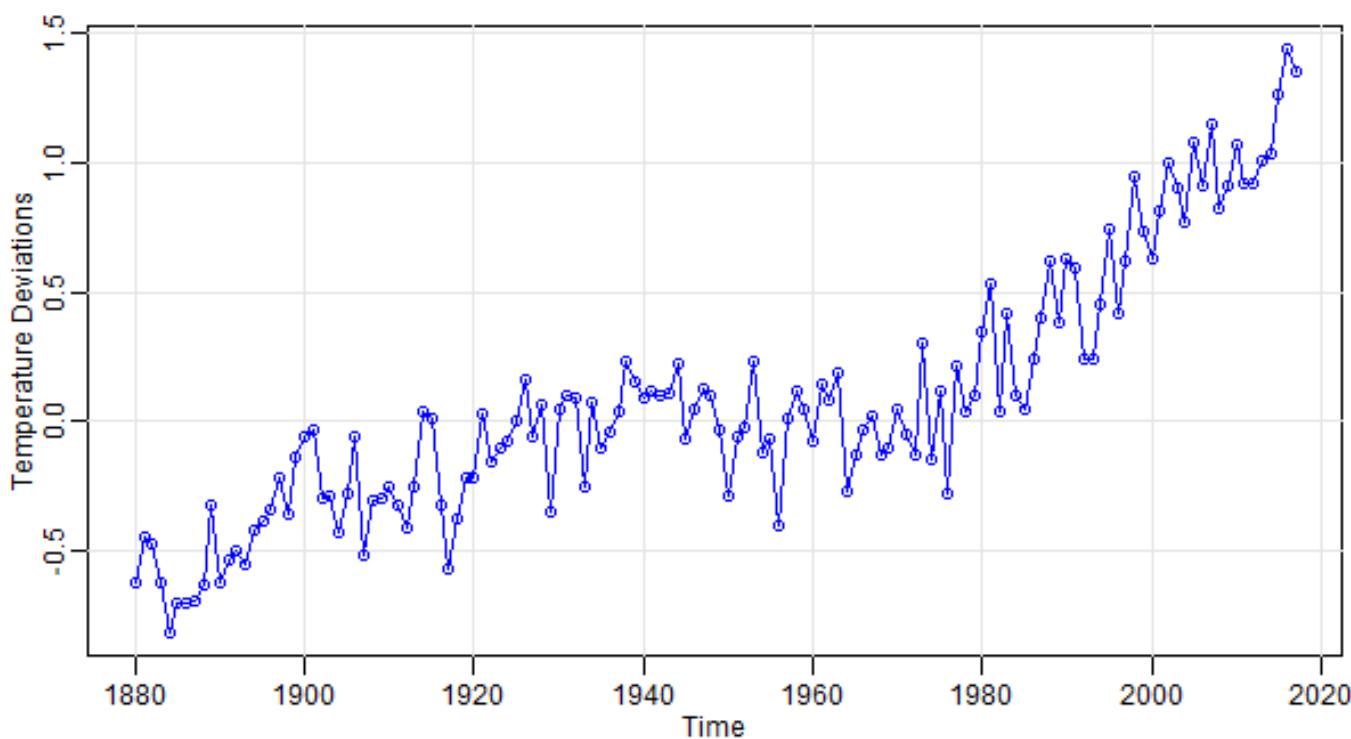
Line chart là một trong những dạng đồ thị phổ biến và hay được sử dụng trong thực tế.

- Khi muốn trình bày các dữ liệu liền mạch biểu đồ line chart là sự lựa chọn phù hợp. Các điểm trong biểu đồ đường được nối liền thành một đường, thể hiện mối quan hệ giữa các điểm đó. Thông thường các dòng dữ liệu sẽ liên quan đến các đơn vị đo lường thời gian như ngày, tháng, quý và năm.

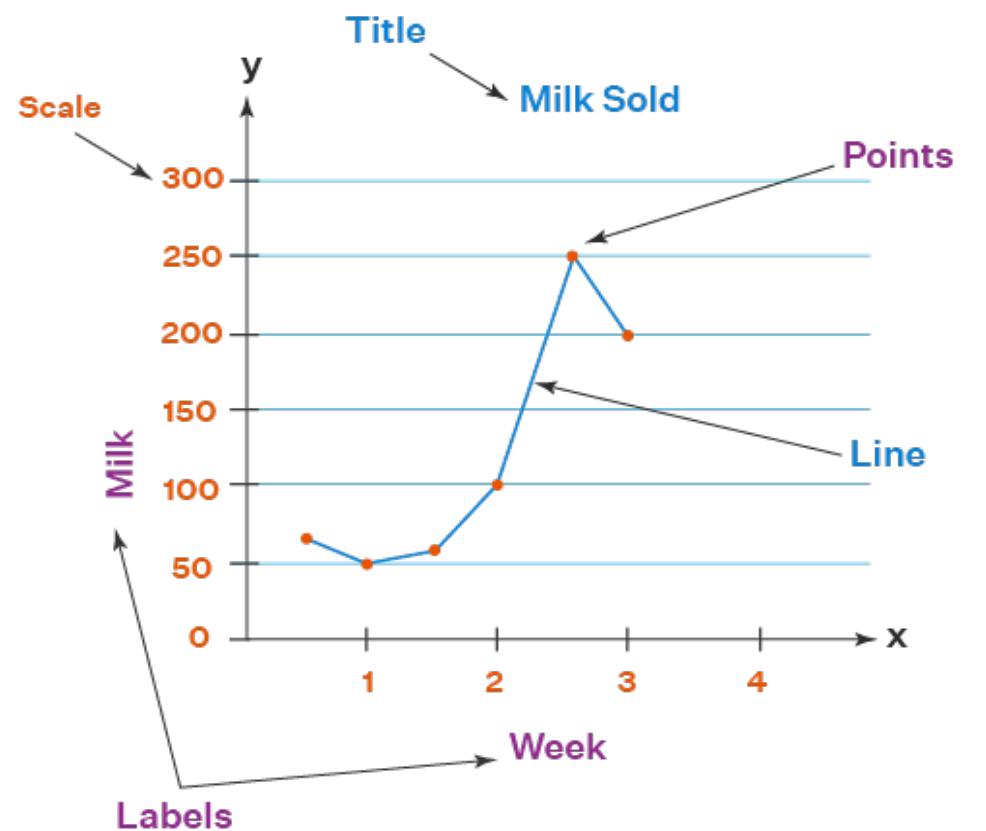


Đồ thị dạng đường (line chart)

Line chart giúp nhấn mạnh sự thay đổi trong dữ liệu của một biến vẽ trên trục x so với biến thứ 2 trên trục y.



Parts of Line Chart



Đồ thị dạng đường với Matplotlib

Tập dữ liệu `gas_prices.csv`: Lưu trữ giá Gas của 10 nước trên thế giới trong giai đoạn từ năm 1990 - 2008

```
1 data = pd.read_csv('Data_Visualize/gas_prices.csv')  
2 data
```

	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	UK	USA
0	1990	NaN	1.87	3.63	2.65	4.59	3.16	1.00	2.05	2.82	1.16
1	1991	1.96	1.92	3.45	2.90	4.50	3.46	1.30	2.49	3.01	1.14
2	1992	1.89	1.73	3.56	3.27	4.53	3.58	1.50	2.65	3.06	1.13
3	1993	1.73	1.57	3.41	3.07	3.68	4.16	1.56	2.88	2.84	1.11
4	1994	1.84	1.45	3.59	3.52	3.70	4.36	1.48	2.87	2.99	1.11
5	1995	1.95	1.53	4.26	3.96	4.00	4.43	1.11	2.94	3.21	1.15
6	1996	2.12	1.61	4.41	3.94	4.39	3.64	1.25	3.18	3.34	1.23
7	1997	2.05	1.62	4.00	3.53	4.07	3.26	1.47	3.34	3.83	1.23
8	1998	1.63	1.38	3.87	3.34	3.84	2.82	1.49	3.04	4.06	1.06
9	1999	1.72	1.52	3.85	3.42	3.87	3.27	1.79	3.80	4.29	1.17
10	2000	1.94	1.86	3.80	3.45	3.77	3.65	2.01	4.18	4.58	1.51

Đồ thị dạng đường với Matplotlib

Cú pháp:

`plt.plot(x, y, color, linestyle, linewidth, marker, markersize)`

Trong đó:

- * X, Y – dữ liệu trục X, Y

Hàm pyplot.plot() còn có các tham số cơ bản sau:

- * Color (c): Màu của đường line
- * Linewidth (lw): Số thực - Độ rộng của đường đồ thị
- * linestyle (ls): Kiểu đường đồ thị
- * marker: Kiểu của điểm
- * markersize (ms): Số thực - Kích thước của điểm dữ liệu

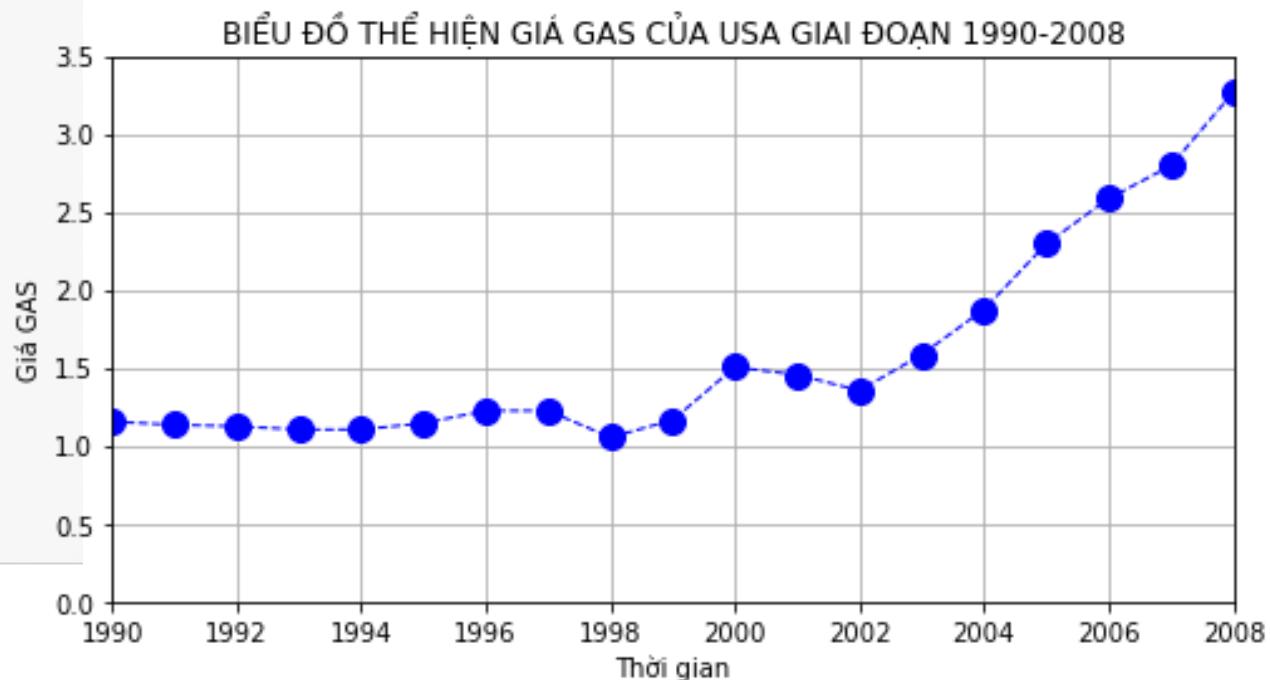
Các tham số color, marker, linestyle có thể được biểu diễn ở dạng '[color][marker][linestyle]', ví dụ: 'ro-' tương đương với color='r', marker='o', linestyle='-'.



a. Single line chart

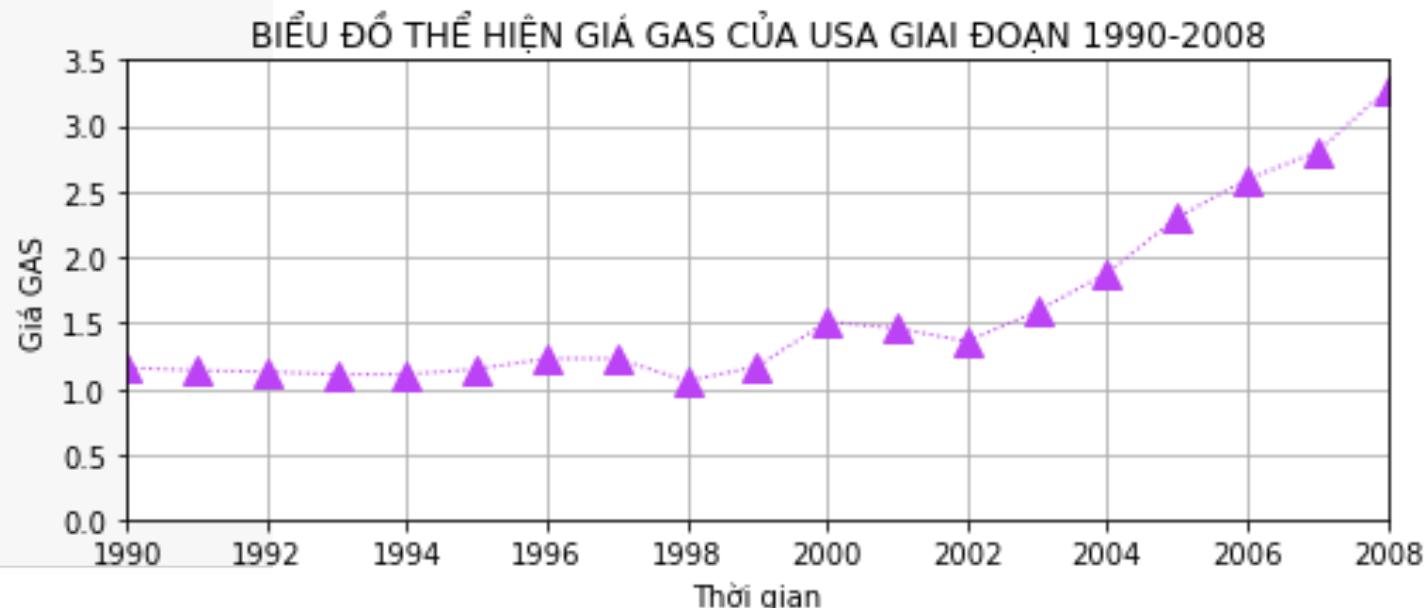
Single line chart

```
1 plt.figure(figsize = (8,4)) #Thiết lập kích thước biểu đồ
2
3 plt.plot(x,
4             y,
5             color='b',
6             linestyle='--',
7             linewidth=1.0,
8             marker='o',
9             markersize = 10) #Kích thước điểm
10
11 #Tiêu đề của đồ thị
12 plt.title('BIỂU ĐỒ THỂ HIỆN GIÁ GAS CỦA USA GIAI ĐOẠN 1990-2008')
13 #Nhãn cho trục X
14 plt.xlabel('Thời gian')
15 #Nhãn cho trục Y
16 plt.ylabel('Giá GAS')
17
18 #Setup giới hạn cho trục X:
19 plt.xlim(1990,2008)
20
21 #Setup giới hạn cho trục Y:
22 plt.ylim(0,3.5)
23
24 #Hiển thị lưới:
25 plt.grid()
26
27 plt.show()
```



Single line chart

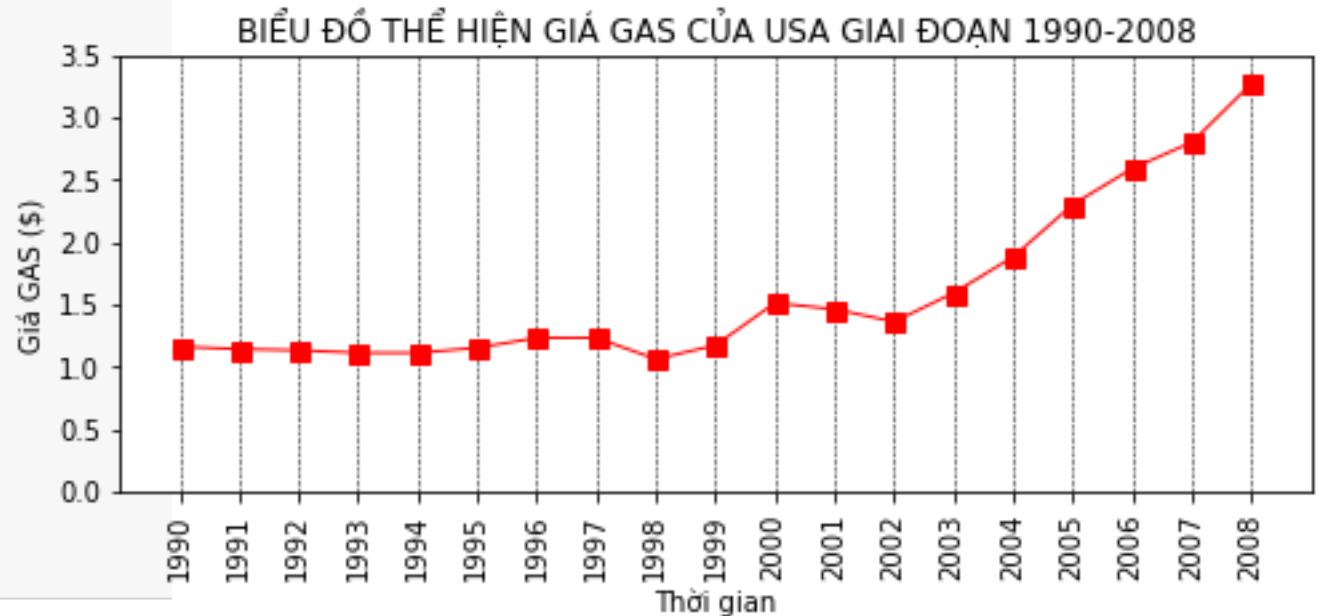
```
1 plt.figure(figsize = (8,3)) #Thiết lập kích thước biểu đồ
2
3 plt.plot(x,                      #Dữ liệu trục X
4           y,                      #Dữ liệu trục Y
5           c ='#bc42f5',          #Màu của đường
6           ls =':',            #Kiểu đường
7           lw=1.0,              #Độ rộng của đường line
8           marker='^',          #Kiểu điểm
9           ms = 10)             #Kích thước điểm
10
11 #Tiêu đề của đồ thị
12 plt.title('BIỂU ĐỒ THỂ HIỆN GIÁ GAS CỦA USA GIAI ĐOẠN 1990-2008')
13 #Nhãn cho trục X
14 plt.xlabel('Thời gian')
15 #Nhãn cho trục Y
16 plt.ylabel('Giá GAS')
17
18 #Setup giới hạn cho trục X:
19 plt.xlim(1990,2008)
20
21 #Setup giới hạn cho trục Y:
22 plt.ylim(0,3.5)
23
24 #Hiển thị lưới:
25 plt.grid()
26
27 plt.show()
```



Simple line chart

```
1 plt.figure(figsize = (8,3)) #Thiết lập kích thước biểu đồ
2
3 plt.plot(x,                      #Dữ liệu trục X
4           y,                      #Dữ liệu trục Y
5           'r-s',                  #Mã màu và kiểu điểm
6           linewidth=1.0,          #Độ rộng của đường line
7           markersize = 7)        #Kích thước điểm
8
9 #Tiêu đề của đồ thị
10 plt.title('BIỂU ĐỒ THỂ HIỆN GIÁ GAS CỦA USA GIAI ĐOẠN 1990-2008')
11 #Nhãn cho trục X
12 plt.xlabel('Thời gian')
13 #Nhãn cho trục Y
14 plt.ylabel('Giá GAS ($)')
15 #Setup giới hạn cho trục X:
16 plt.xlim(1989,2009)
17 #Setup giới hạn cho trục Y:
18 plt.ylim(0,3.5)
19 #Setup tick cho trục X:
20 plt.xticks(x,
21             rotation=90)
22 #Thiết lập lưới:
23 plt.grid(axis='x',
24           c='black',
25           ls='--',
26           lw=0.5)
27
28 plt.show()
```

Các tham số *color*, *marker*, *linestyle* có thể
được biểu diễn ở dạng
[color][marker][linestyle],
ví dụ: ‘ro-’ tương đương với *color*='r',
marker='o', *linestyle*='-'.

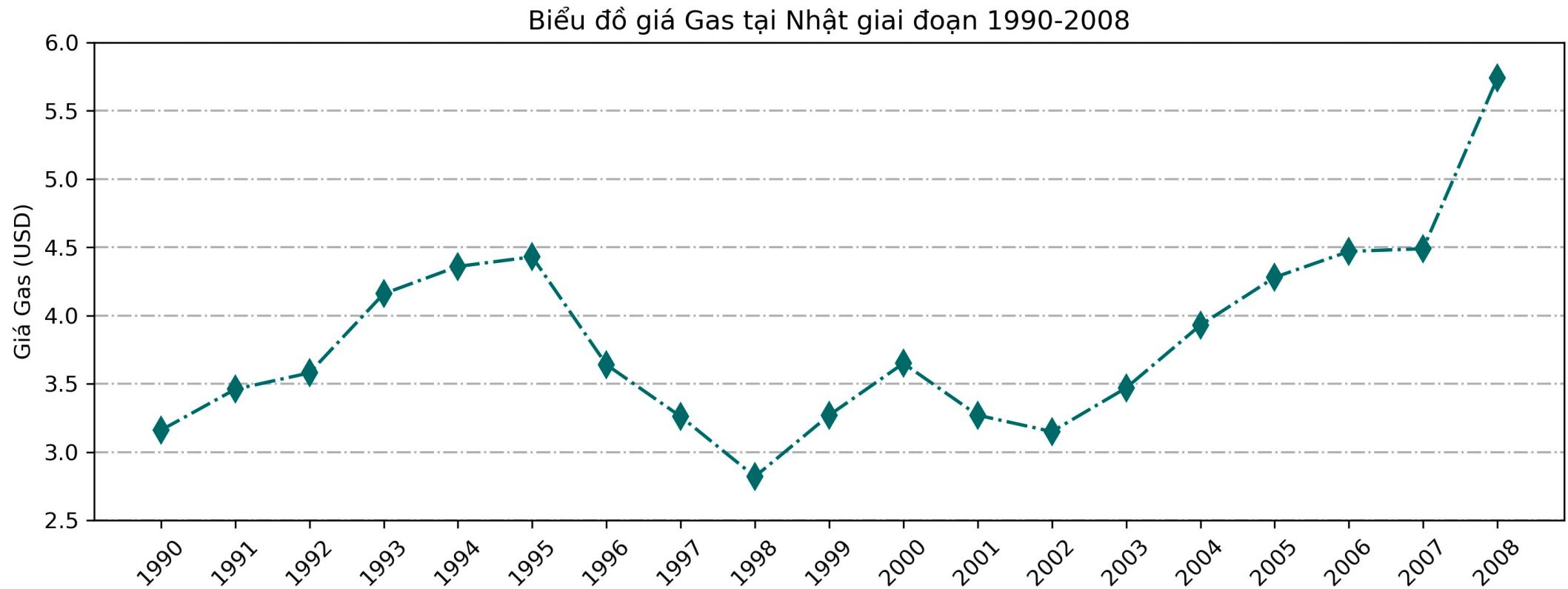


Thực hành 1



Thực hành

YÊU CẦU: Vẽ biểu đồ giá gas của Japan, thiết lập các tham số để thu được biểu đồ như sau:

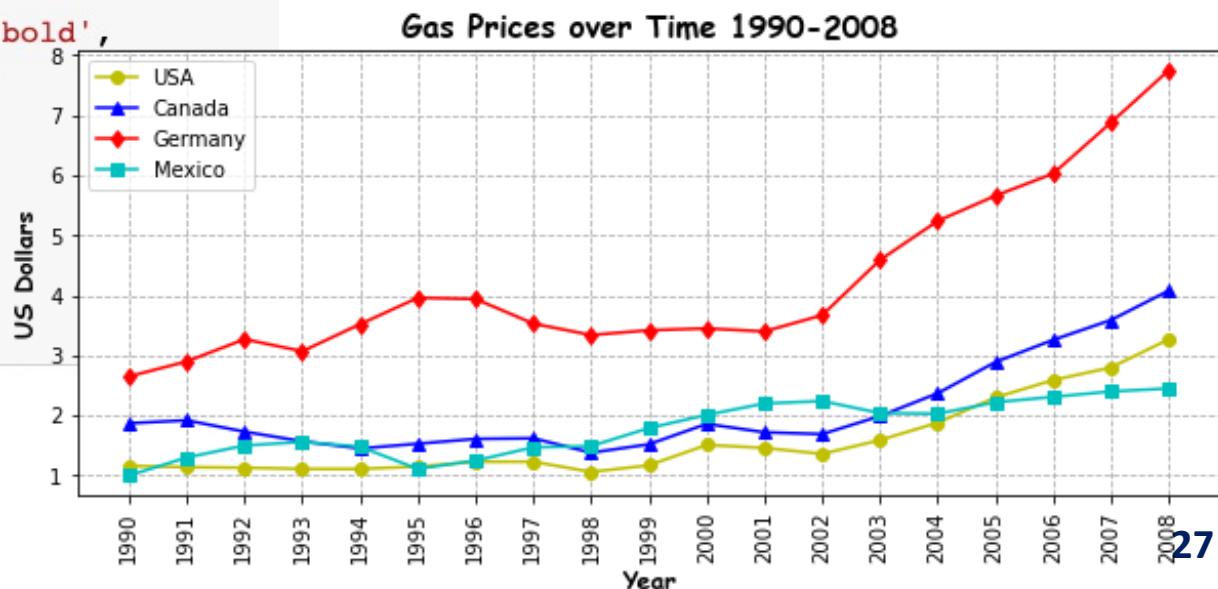




b. Multiple line chart

Multiple lines chart

```
1 plt.figure(figsize=(10,4))
2
3 #Vẽ multiple line:
4 plt.plot(x,y1,'y-o', label='USA')
5 plt.plot(x,y2,'b-^', label='Canada')
6 plt.plot(x,y3,'r-d', label='Germany')
7 plt.plot(x,y4,'c-s', label='Mexico')
8
9
10 plt.title('Gas Prices over Time 1990-2008',fontdict={'fontname':'Comic Sans MS',
11                               'fontweight':'bold',
12                               'fontsize':15})
13 plt.xlabel('Year',fontdict={'fontname':'Comic Sans MS',
14                               'fontweight':'bold',
15                               'fontsize':12})
16 plt.ylabel('US Dollars',fontdict={'fontname':'Comic Sans MS',
17                               'fontweight':'bold',
18                               'fontsize':12})
19 plt.xticks(x,rotation=90)
20 plt.grid(True,ls='--')
21
22 #Hiển thị chú thích trong biểu đồ:
23 plt.legend()
24
25 plt.show()
```

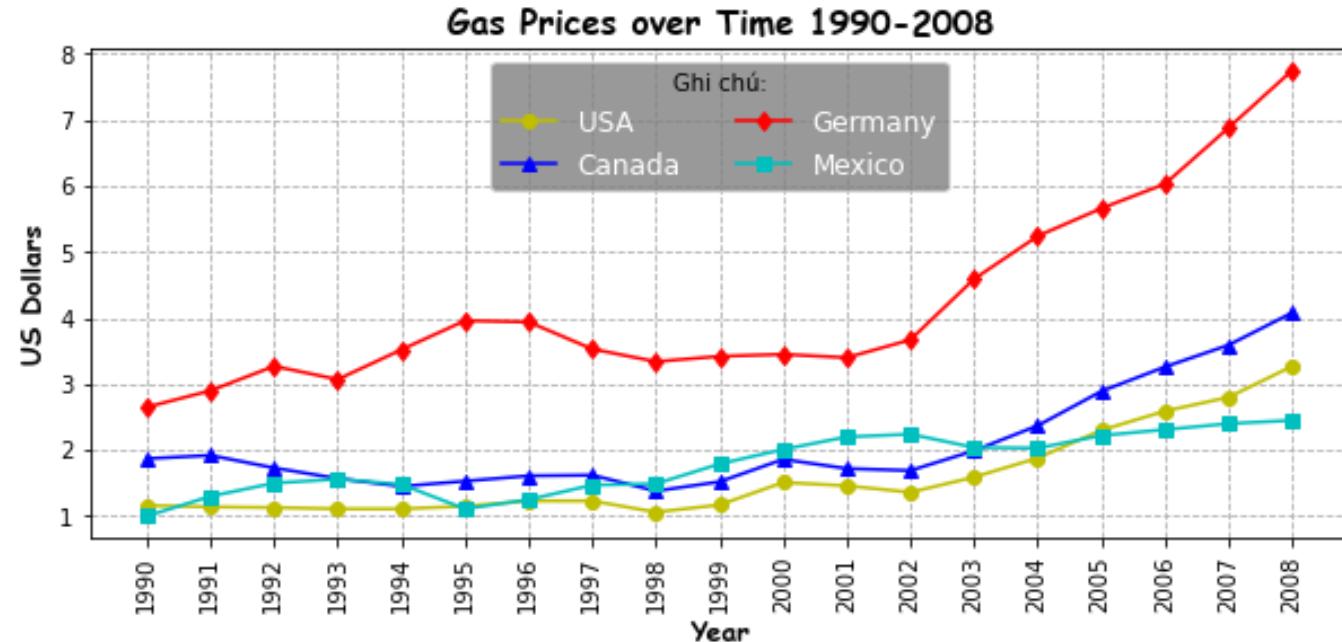


Multiple lines chart

Phương thức legend(): Hiển thị chú thích của biểu đồ Bao gồm các tham số chính:

1.loc: Xác định vị trí hiển thị của chú thích trong biểu đồ, gồm các tuỳ chọn sau:

- 'best' | 0
- 'upper right' | 1
- 'upper left' | 2
- 'lower left' | 3
- 'lower right' | 4
- 'right' | 5
- 'center left' | 6
- 'center right' | 7
- 'lower center' | 8
- 'upper center' | 9
- 'center' | 10



2.ncol: Số cột của chú thích (số nguyên, mặc định là 1)

3.fontsize: kích thước font chữ trong chú thích

4.labelcolor: Màu chữ trong chú thích (mặc định màu đen)

5.facecolor: Màu nền của ô chú thích (mặc định None)

6.title: Dòng tiêu đề trong chú thích

Multiple lines chart

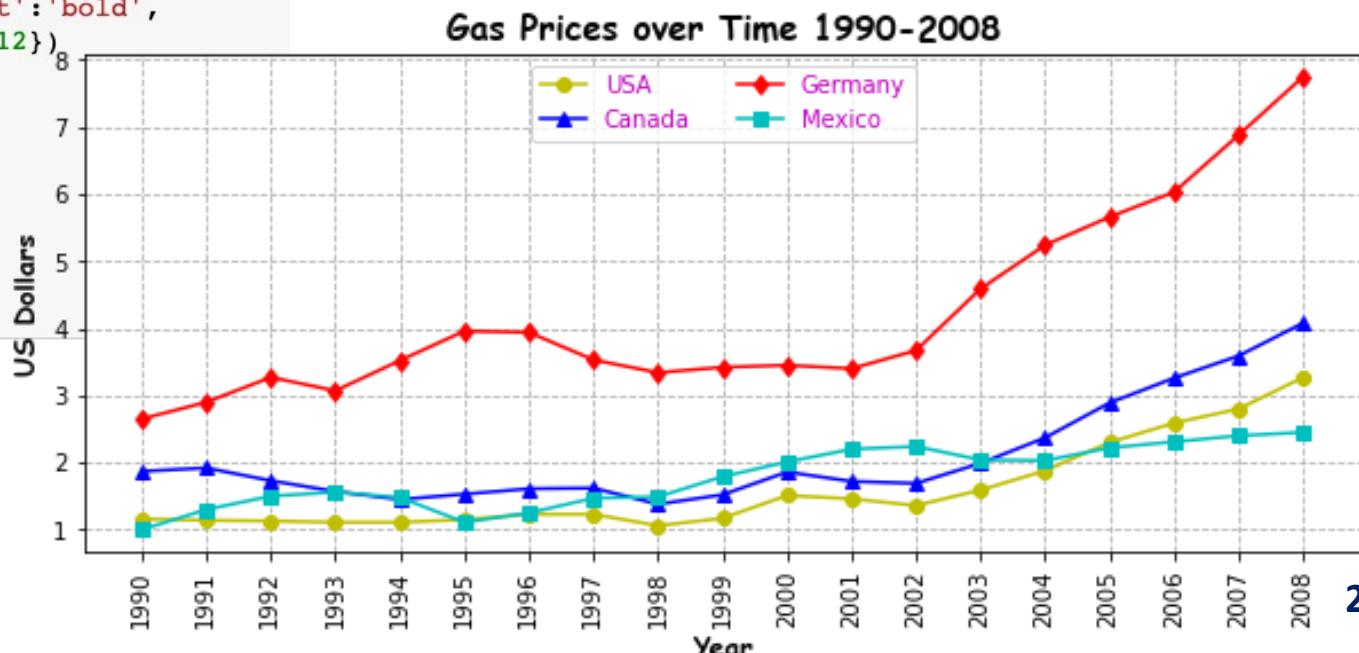
```
1 plt.figure(figsize=(10,4))
2
3 #Vẽ multiple line:
4 plt.plot(x,y1,'y-o', label='USA')
5 plt.plot(x,y2,'b^-', label='Canada')
6 plt.plot(x,y3,'r-d', label='Germany')
7 plt.plot(x,y4,'c-s', label='Mexico')
8
9 plt.title('Gas Prices over Time 1990-2008',fontdict={'fontname':'Comic Sans MS',
10                                'fontweight':'bold',
11                                'fontsize':15})
12 plt.xlabel('Year',fontdict={'fontname':'Comic Sans MS',
13                                'fontweight':'bold',
14                                'fontsize':12})
15 plt.ylabel('US Dollars',fontdict={'fontname':'Comic Sans MS',
16                                'fontweight':'bold',
17                                'fontsize':12})
18 plt.xticks(x,rotation=90)
19 plt.grid(True,ls='--')
20 plt.legend(loc = 9, ncol=2,labelcolor='m')
21 #Lưu đồ thị:
22 plt.savefig('Save_charts/Gas',dpi=300, format='png')
23 plt.savefig('Save_charts/Gas',dpi=500,format='pdf')
24
25 plt.show()
```

Lưu biểu đồ:

plt.savefig(fname, dpi, format)

Trong đó:

1. fname: đường dẫn lưu file
2. dpi: độ phân giải của đồ thị khi lưu (số pixel điểm ảnh trên mỗi inch)
3. format: định dạng file ('png', 'pdf',...)

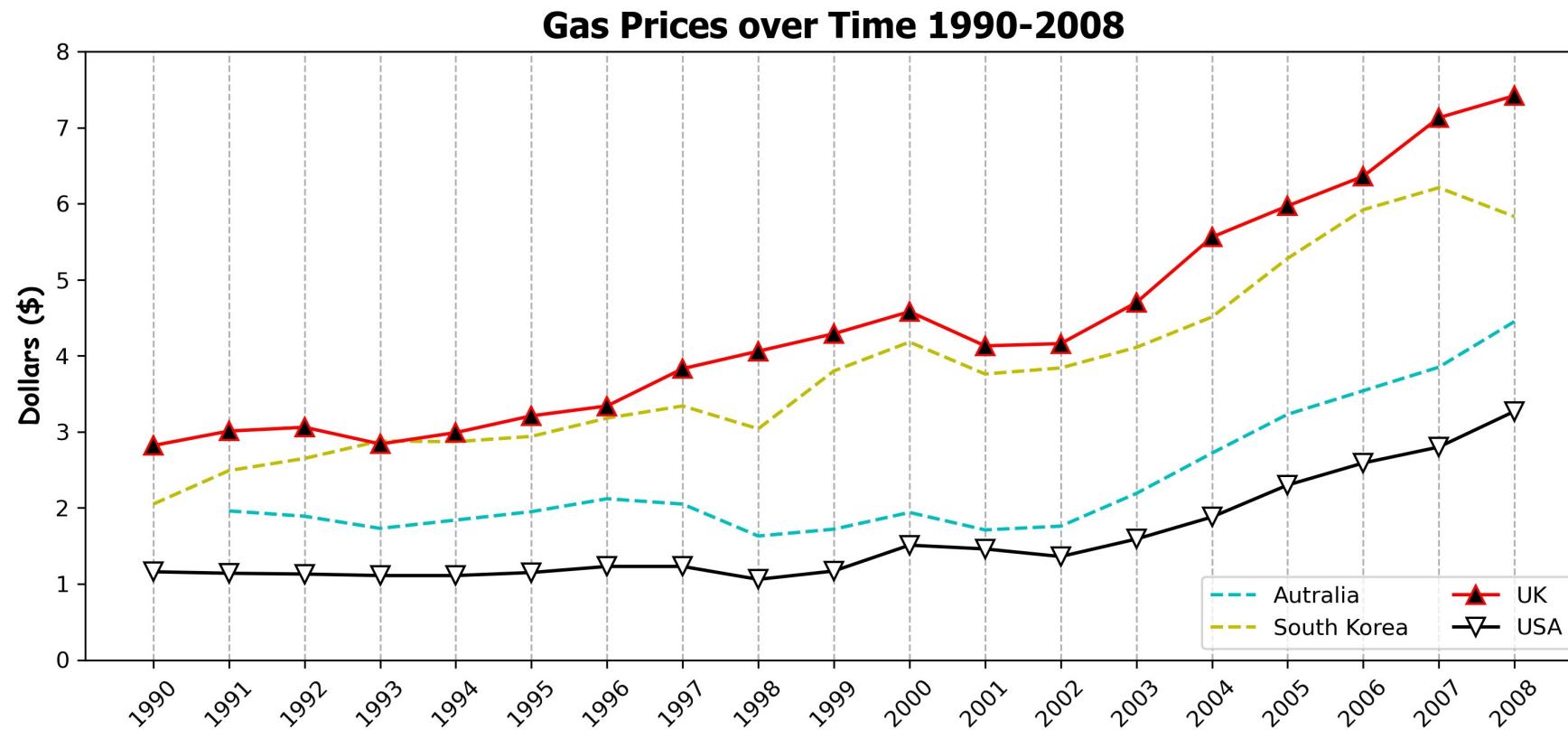


Thực hành 2



Thực hành

YÊU CẦU: Trực quan hoá dữ liệu giá Gas của 4 nước: UK, USA, Australia, South Korea. với mục tiêu nhấn mạnh quốc gia có giá Gas cao nhất và thấp nhất. (như minh họa)





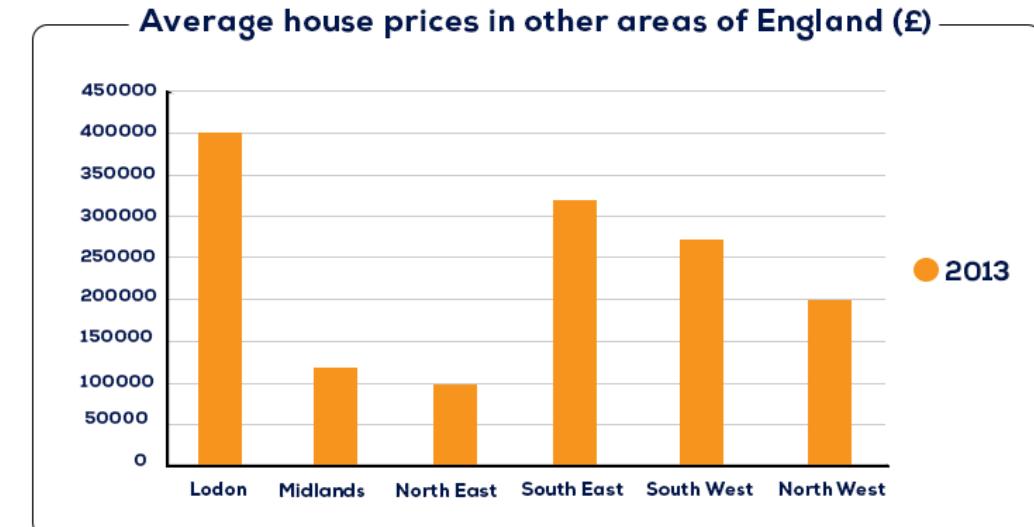
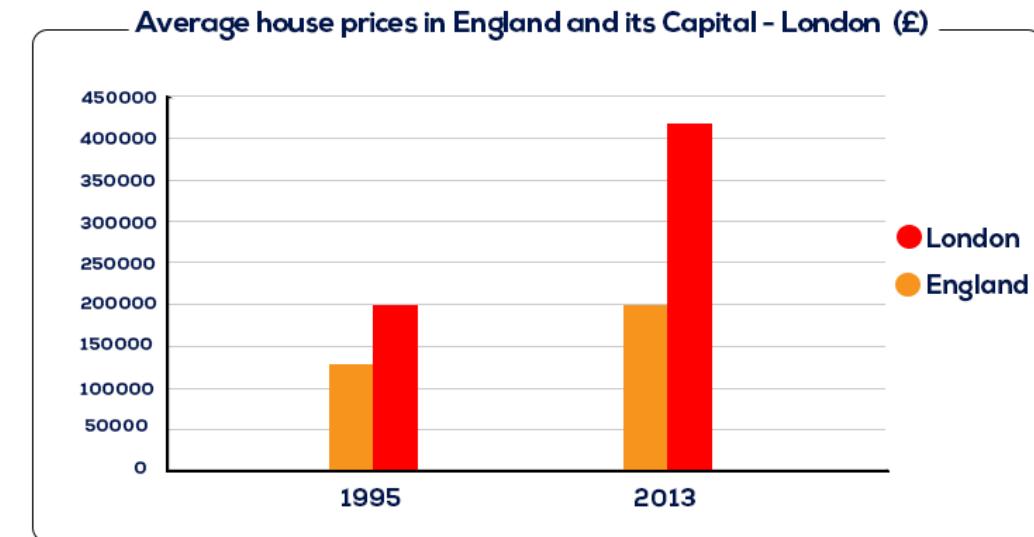
4. Biểu đồ thanh/cột (bar chart)

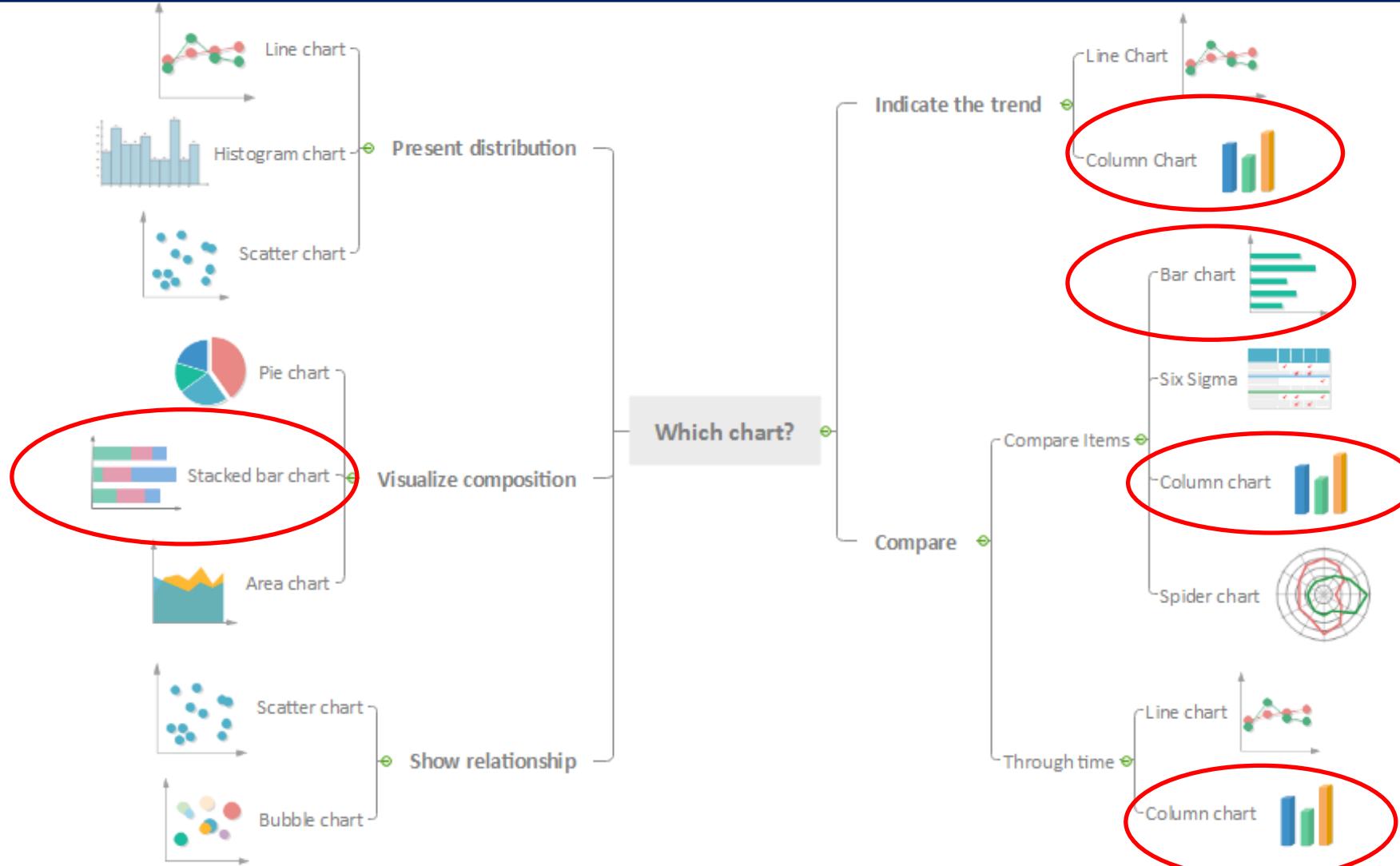
Biểu đồ Bar/Column chart

Bar chart cũng như line chart đây là một trong những dạng đồ thị phổ biến và hay được sử dụng trong thực tế.

Bar chart là một cách cụ thể để biểu diễn dữ liệu bằng cách sử dụng các thanh hình chữ nhật, trong đó chiều dài của mỗi thanh tỷ lệ với giá trị mà chúng đại diện. Nó là một cách biểu diễn đồ họa của dữ liệu bằng cách sử dụng các thanh có độ cao khác nhau.

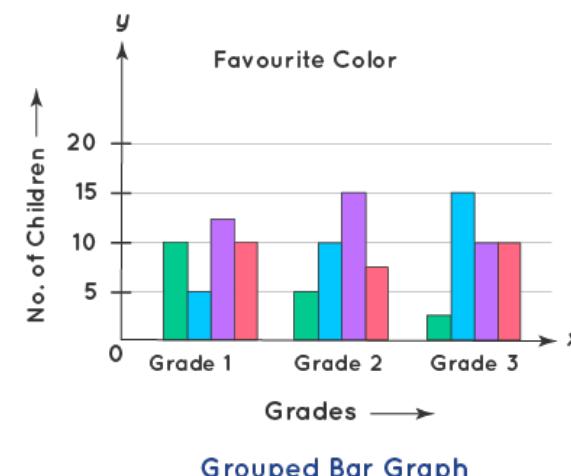
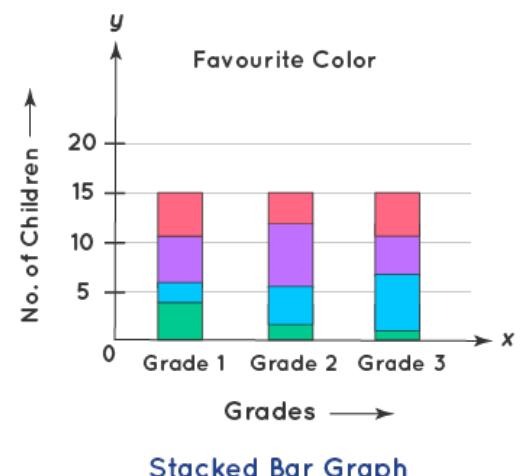
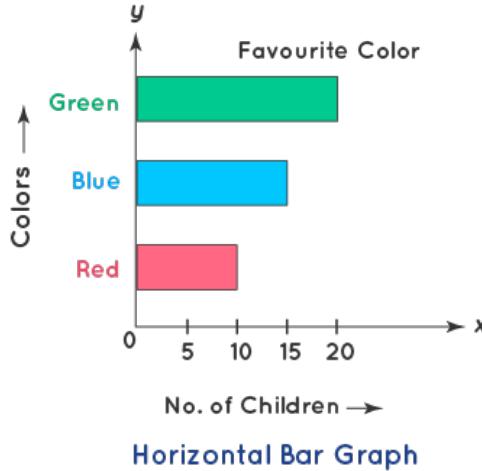
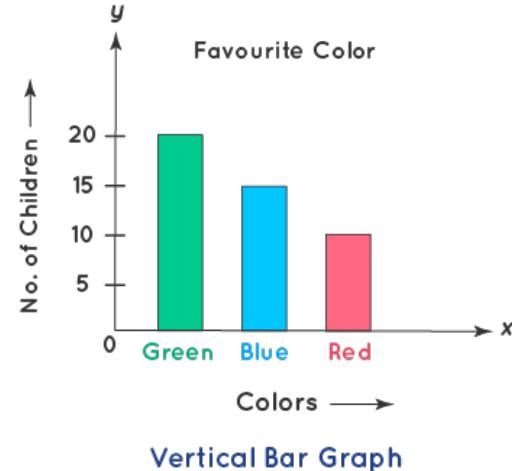
Bar chart là công cụ tuyệt vời để biểu diễn dữ liệu độc lập với nhau mà không cần theo bất kỳ thứ tự cụ thể nào khi biểu diễn.





Biểu đồ Bar/Column chart

Types of Bar Graph

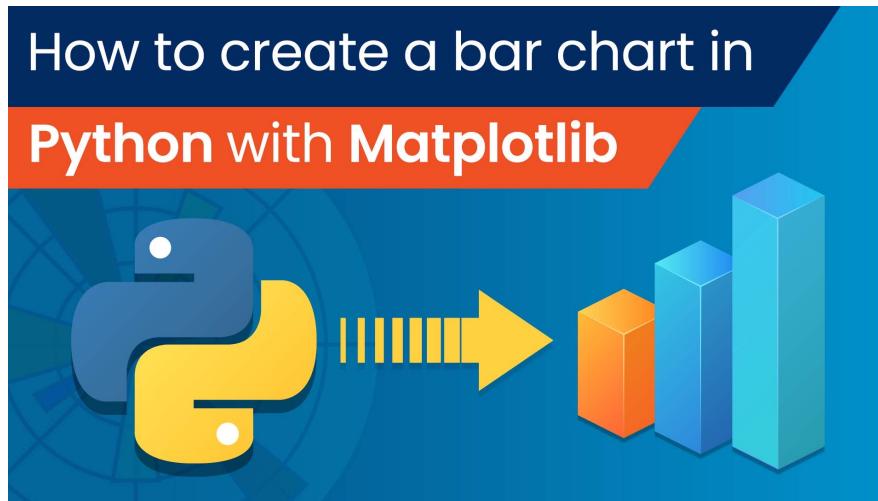


Đặc điểm của Bar chart:

- Tất cả các thanh chữ nhật phải có chiều rộng bằng nhau và phải có khoảng trống giữa chúng bằng nhau.
- Các thanh hình chữ nhật có thể vẽ theo chiều ngang hoặc chiều dọc.
- Chiều cao của hình chữ nhật tương đương với giá trị của dữ liệu mà chúng đại diện.
- Các thanh hình chữ nhật phải nằm trên cùng một trực cơ sở.

Biểu đồ Bar chart với Matplotlib

Tập dữ liệu **gas_prices.csv**: Lưu trữ giá Gas của 10 nước trên thế giới trong giai đoạn từ năm 1990 - 2008



	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	UK	USA
0	1990	NaN	1.87	3.63	2.65	4.59	3.16	1.00		2.05	2.82
1	1991	1.96	1.92	3.45	2.90	4.50	3.46	1.30		2.49	3.01
2	1992	1.89	1.73	3.56	3.27	4.53	3.58	1.50		2.65	3.06
3	1993	1.73	1.57	3.41	3.07	3.68	4.16	1.56		2.88	2.84
4	1994	1.84	1.45	3.59	3.52	3.70	4.36	1.48		2.87	2.99
5	1995	1.95	1.53	4.26	3.96	4.00	4.43	1.11		2.94	3.21
6	1996	2.12	1.61	4.41	3.94	4.39	3.64	1.25		3.18	3.34
7	1997	2.05	1.62	4.00	3.53	4.07	3.26	1.47		3.34	3.83
8	1998	1.63	1.38	3.87	3.34	3.84	2.82	1.49		3.04	4.06
9	1999	1.72	1.52	3.85	3.42	3.87	3.27	1.79		3.80	4.29
10	2000	1.94	1.86	3.80	3.45	3.77	3.65	2.01		4.18	4.58

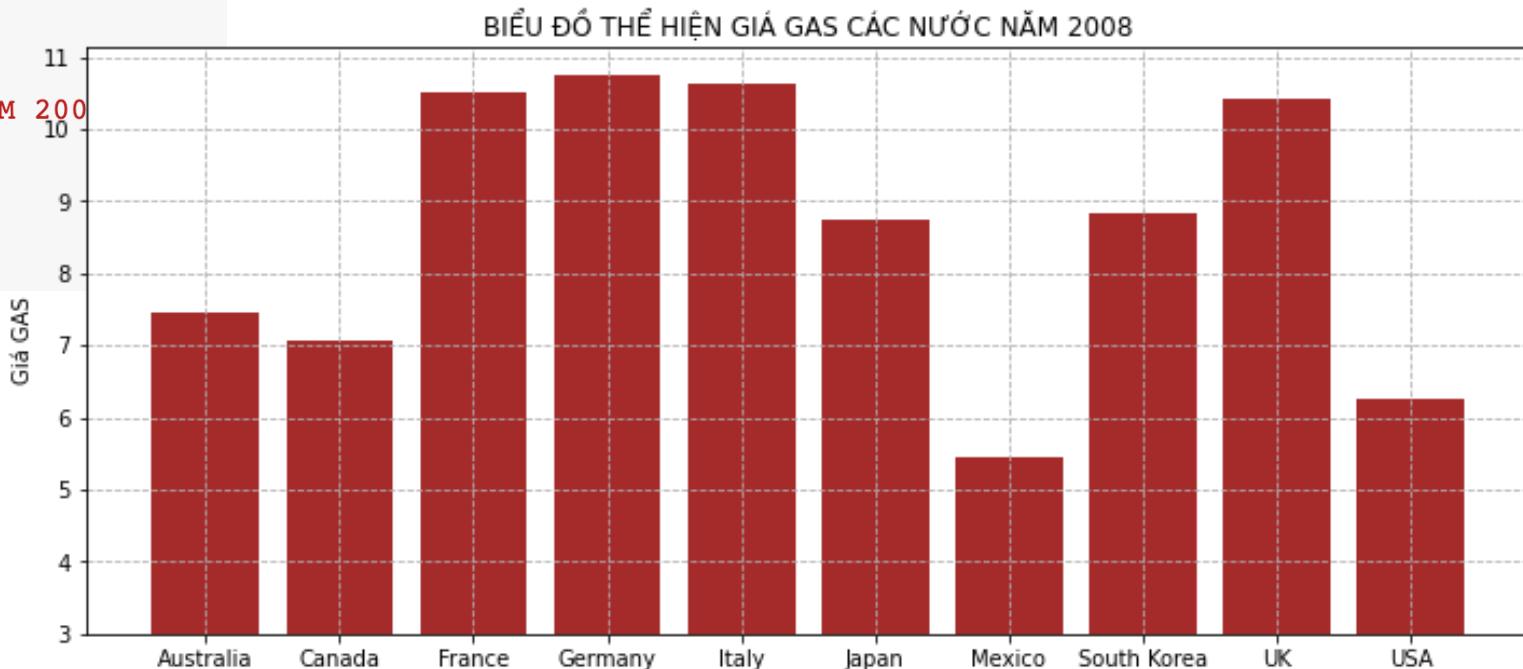


a. Vertical Bar chart

Biểu đồ cột dạng thẳng đứng

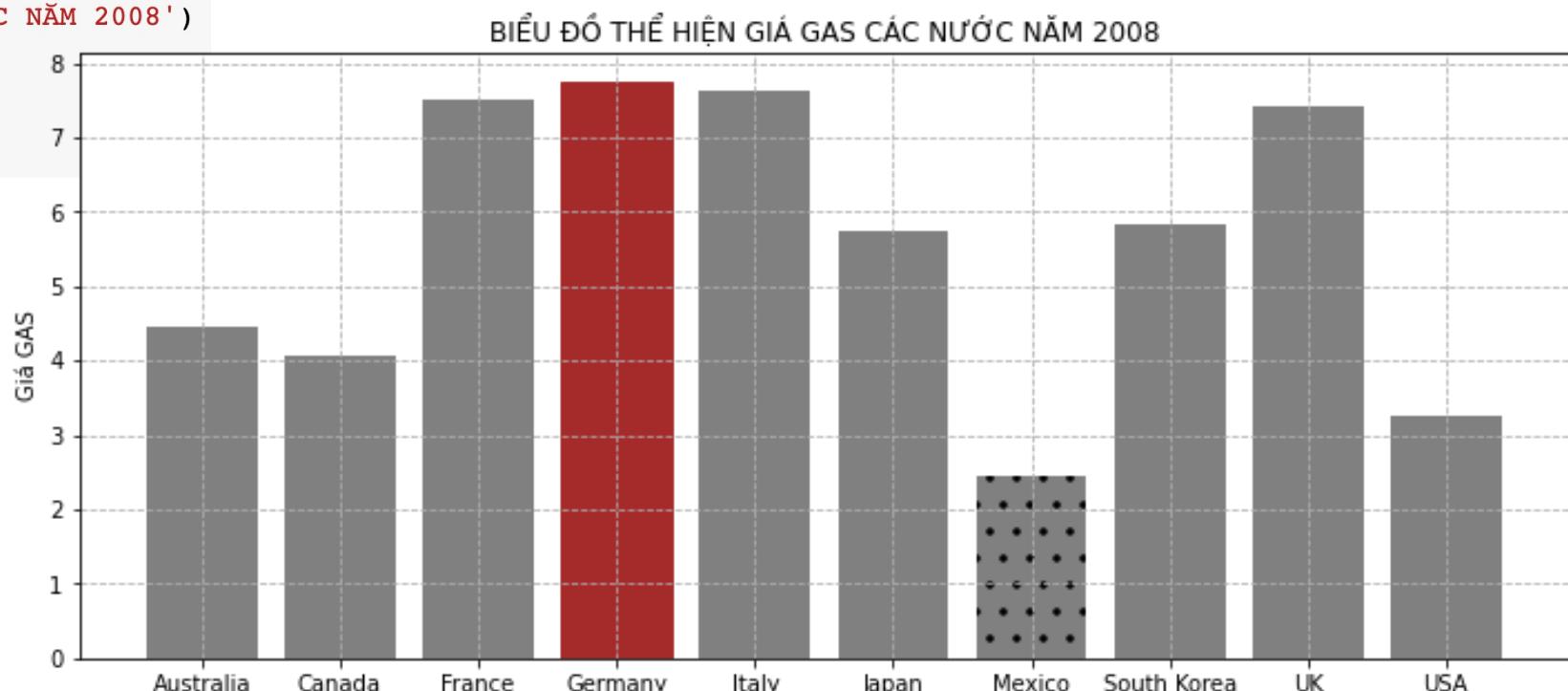
Cú pháp: plt.bar (labels, y)

```
1 plt.figure(figsize = (12,5)) #Thiết lập kích thước biểu đồ
2
3 #Vẽ biểu đồ cột:
4 plt.bar(labels,           #Nhãn của trục X
5         y_2008,          #Giá trị tương ứng với nhãn
6         color='brown',   #Màu của thanh
7         bottom=3,         #Giá trị bắt đầu của trục Y
8         width = 0.8)     #Chiều rộng của thanh
9
10 #Tiêu đề của đồ thị
11 plt.title('BIỂU ĐỒ THỂ HIỆN GIÁ GAS CÁC NƯỚC NĂM 2008')
12 plt.ylabel('Giá GAS')
13 plt.grid(ls='--')
14
15 plt.show()
```



Biểu đồ cột dạng thẳng đứng

```
1 #Làm nổi bật một thanh:  
2 plt.figure(figsize = (12,5))  
3  
4 bar = plt.bar(labels,y_2008,color='gray')  
5  
6 #Thay đổi màu sắc khác, tạo hatch cho thanh  
7 bar[3].set_color('brown')  
8 bar[6].set_hatch('.')  
9  
10 #Tiêu đề của đồ thị  
11 plt.title('BIỂU ĐỒ THỂ HIỆN GIÁ GAS CÁC NƯỚC NĂM 2008')  
12 plt.ylabel('Giá GAS')  
13 plt.grid(ls='--')  
14  
15 plt.show()
```





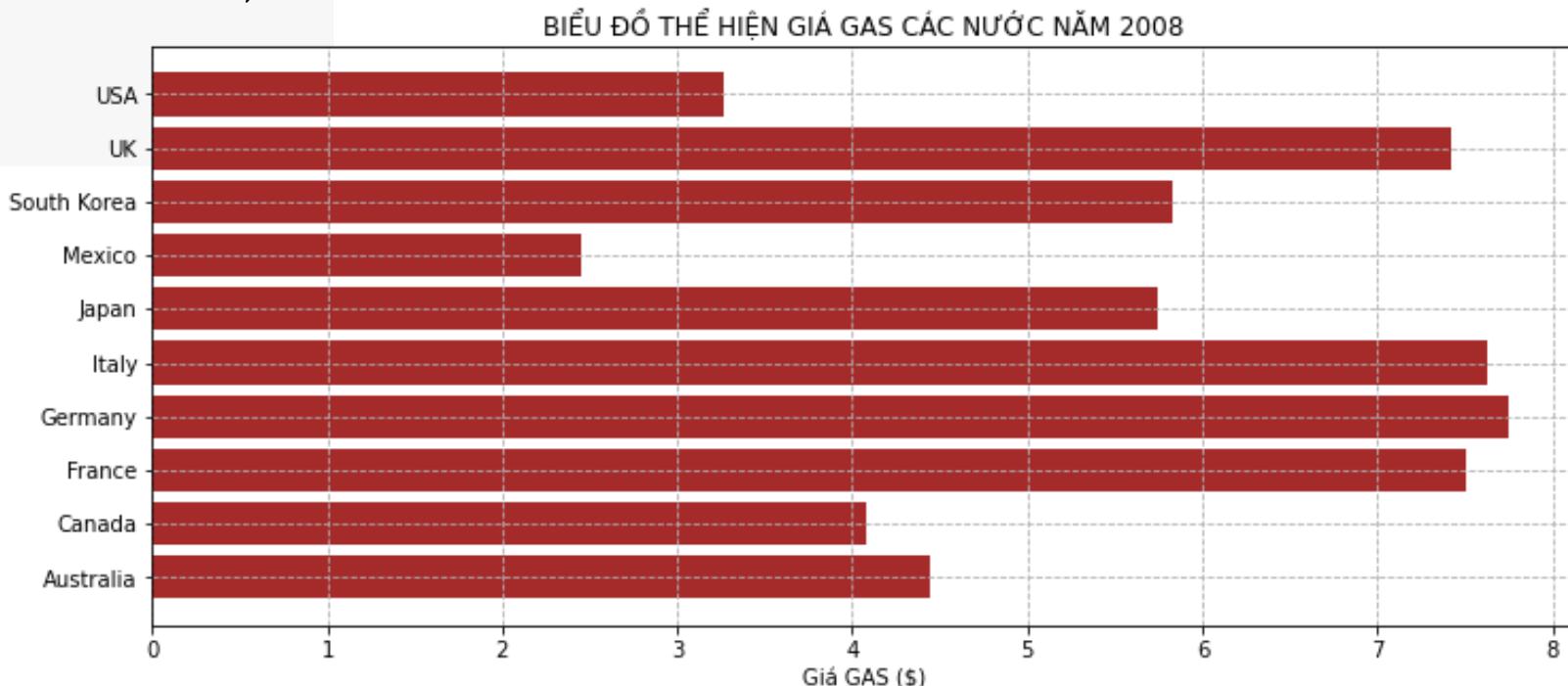
b. Horizontal Bar chart



Biểu đồ cột dạng nằm ngang

Cú pháp: plt.barh (labels, y)

```
1 plt.figure(figsize = (12,5)) #Thiết lập kích thước biểu đồ
2
3 plt.barh(labels,           #Nhãn của trục X
4          y_2008,          #Giá trị tương ứng
5          color='brown',   #Màu của thanh
6          height = 0.8)    #Chiều rộng của thanh
7
8 #Tiêu đề của đồ thị
9 plt.title('BIỂU ĐỒ THỂ HIỆN GIÁ GAS CÁC NƯỚC NĂM 2008')
10 plt.xlabel('Giá GAS ($)')
11 plt.grid(ls='--')
12
13 plt.show()
```

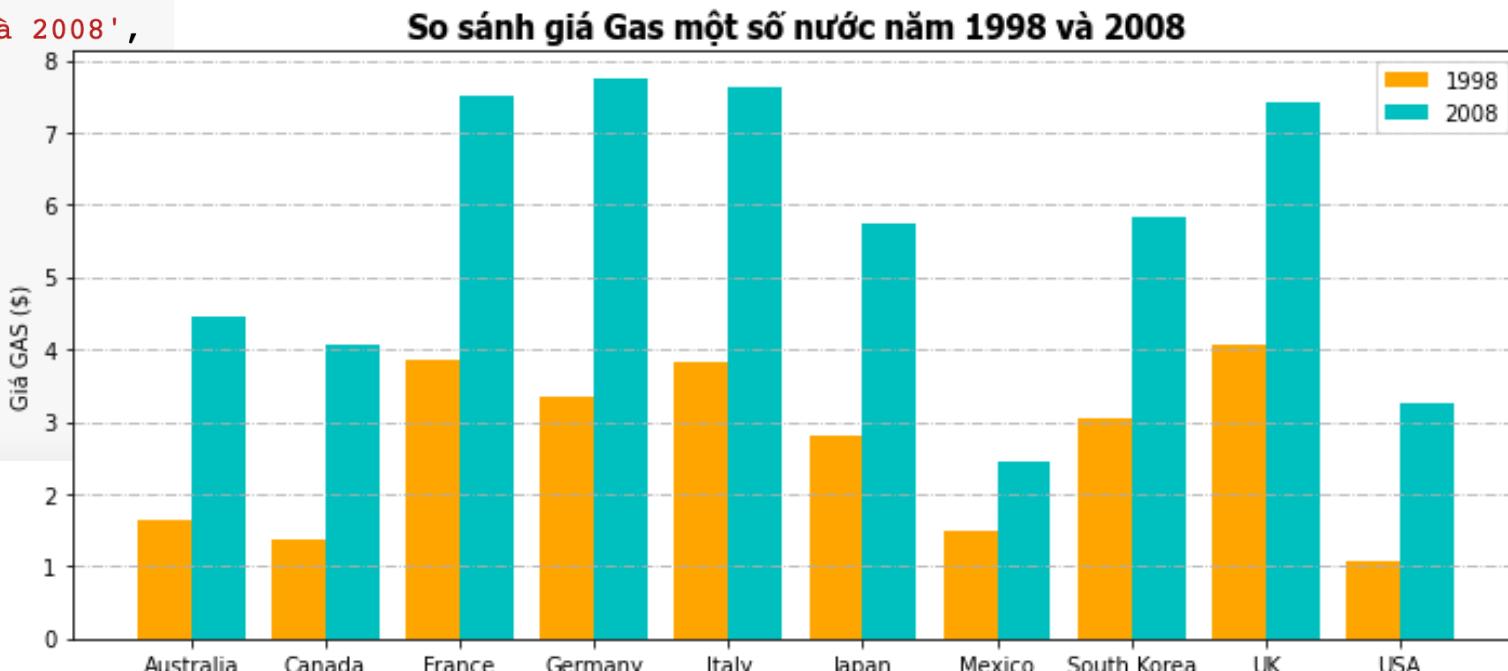




c. Grouped Bar chart

Nhiều cột trên biểu đồ

```
1 #Multiple bar:  
2 w=0.4 #Thiết lập độ rộng của thanh (Tổng < 1.0)  
3 bar1 = np.arange(len(labels))  
4 bar2 = [i+w for i in bar1]  
5  
6 plt.figure(figsize = (12,5))  
7 #Vẽ các biểu đồ cột cho từng bộ dữ liệu:  
8 plt.bar(bar1,y_1998,width=w,color='orange',label='1998')  
9 plt.bar(bar2,y_2008,width=w,color='c',label='2008')  
10  
11 plt.title('So sánh giá Gas một số nước năm 1998 và 2008',  
12         fontdict={'fontname':'Tahoma',  
13                     'fontweight':'bold',  
14                     'fontsize':15})  
15 plt.ylabel("Giá GAS ($)")  
16 plt.grid(axis='y',ls='-.')  
17 plt.legend()  
18  
19 #Hiển thị nhãn của trục x, căn vào giữa 2 biểu đồ  
20 plt.xticks(bar1+w/2,labels)  
21  
22 plt.show()
```

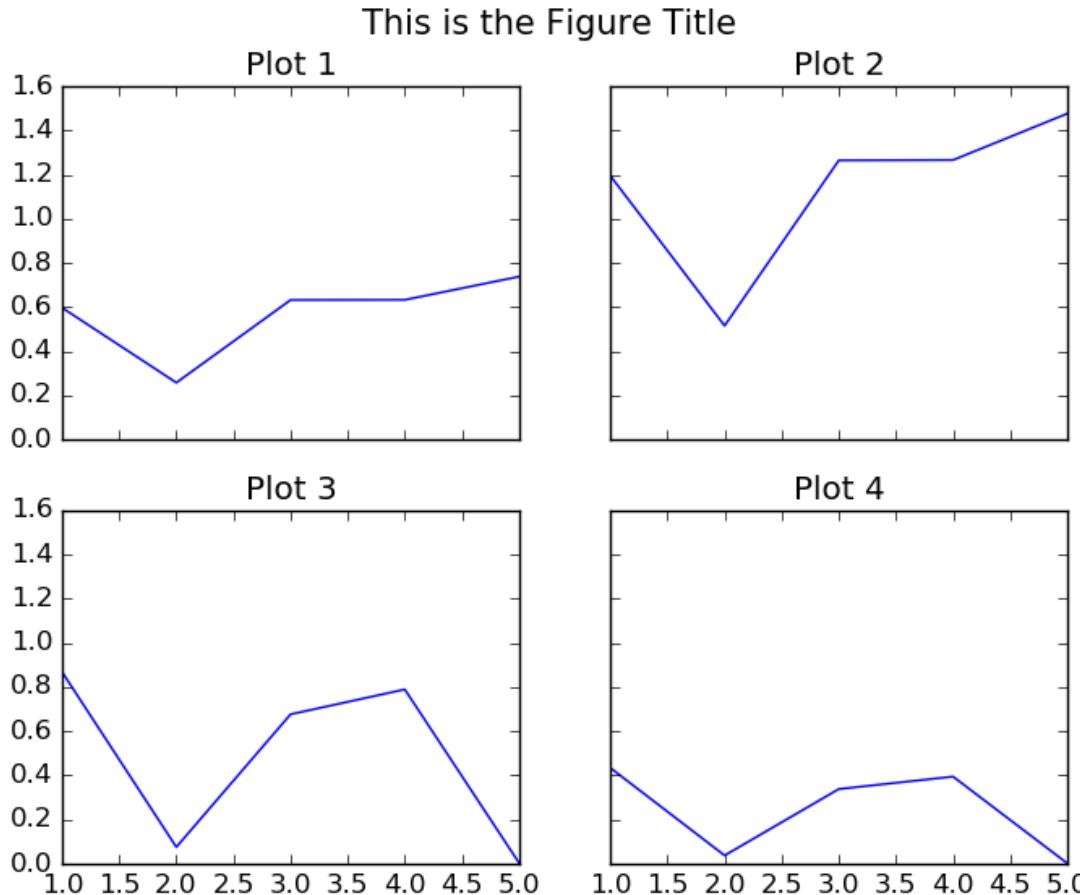




d. Hiển thị nhiều khung biểu đồ

Multiple plot

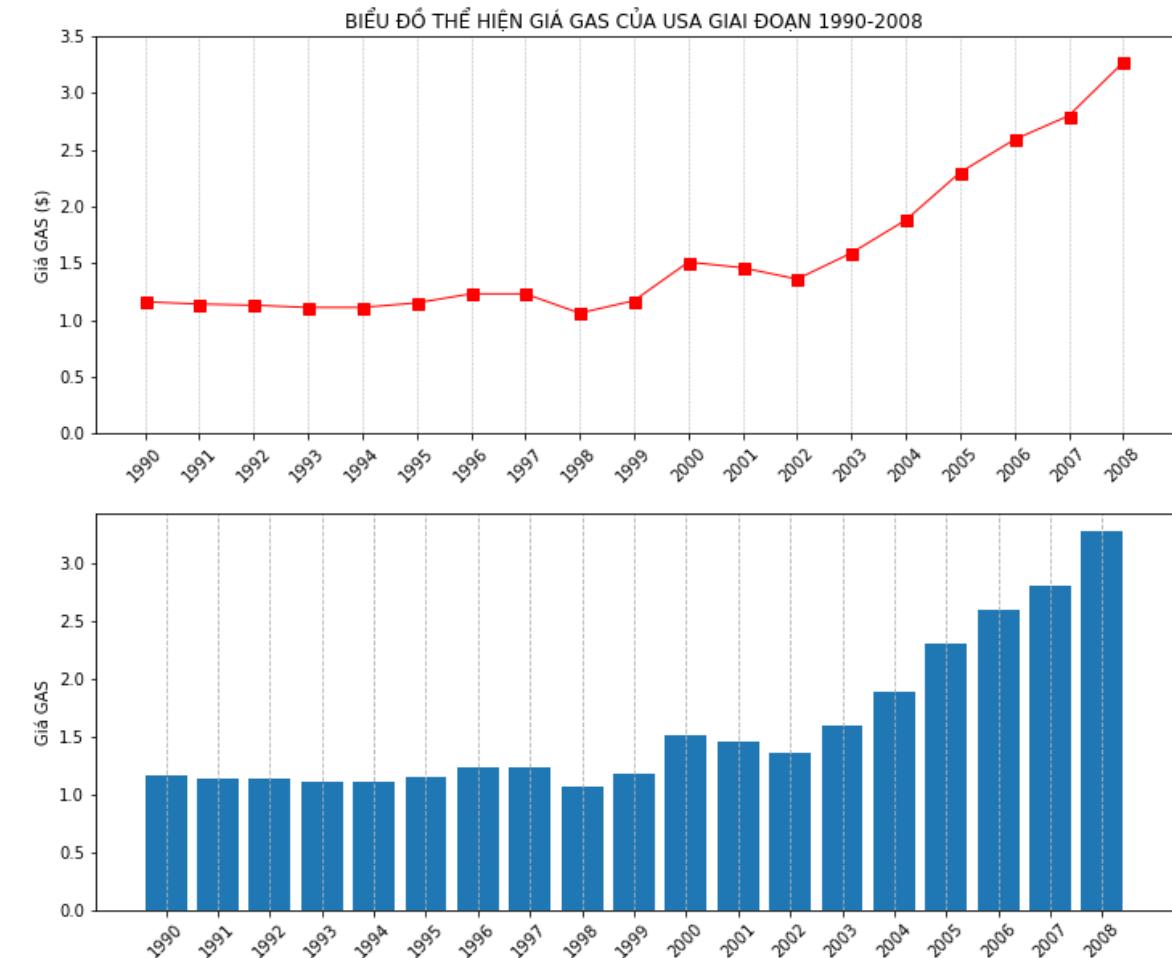
Cú pháp: **plt.subplot (nrows, ncols,Position)**



Multiple plot

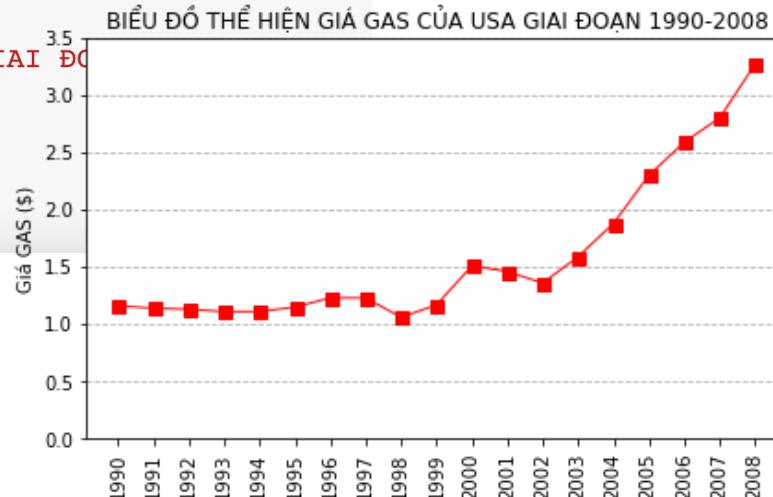
plt.subplot (2, 1, Position)

```
1 plt.figure(figsize = (12,10)) #Thiết lập kích thước biểu đồ
2 #Vẽ biểu đồ đường trên khung 1:
3 plt.subplot(2,1,1) #Thiết lập Khung biểu đồ gồm 2 hàng 1 cột
4
5 plt.plot(x, y, 'r-s',lw=1.0, ms=7) #Vẽ biểu đồ đường plot 1
6
7 plt.title('BIỂU ĐỒ THỂ HIỆN GIÁ GAS CỦA USA GIAI ĐOẠN 1990-2008')
8 plt.ylabel('Giá GAS ($)')
9 plt.ylim(0,3.5)
10 plt.xticks(x,rotation=45)
11 plt.grid(axis='x',ls='--')
12 #
13 #-----#
14 #Vẽ biểu đồ Bar trên khung 2:
15 plt.subplot(2,1,2)
16
17 plt.bar(x,y) #Vẽ biểu đồ cột plot 2
18
19 plt.ylabel('Giá GAS')
20 plt.xticks(x,rotation=45)
21 plt.grid(axis='x', ls='--')
22 plt.show()
```

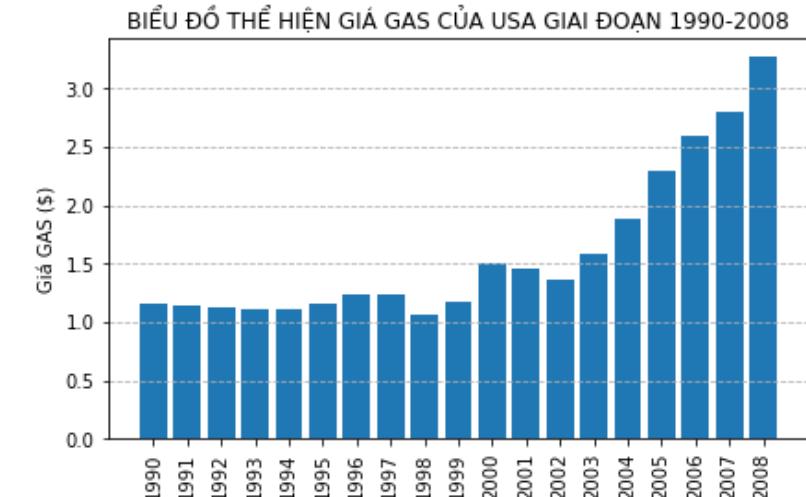


Multiple plot

```
1 plt.figure(figsize = (15,4))
2 #Vẽ biểu đồ đường trên khung 1:
3 plt.subplot(1,2,1) #Thiết lập Khung biểu đồ gồm 1 hàng 2 cột
4
5 plt.plot(x, y, 'r-s', lw=1.0, ms = 7) #Vẽ biểu đồ đường plot 1
6
7 plt.title('BIỂU ĐỒ THỂ HIỆN GIÁ GAS CỦA USA GIAI ĐOẠN 1990-2008')
8 plt.ylabel('Giá GAS ($)')
9 plt.ylim(0,3.5)
10 plt.xticks(x,rotation=90)
11 plt.grid(axis='y',ls='--')
12 #
13 #Vẽ biểu đồ Bar trên khung 2:
14 plt.subplot(1,2,2)
15
16 plt.bar(x,y) #Vẽ biểu đồ cột plot 2
17
18 plt.title('BIỂU ĐỒ THỂ HIỆN GIÁ GAS CỦA USA GIAI ĐOẠN 1990-2008')
19 plt.ylabel('Giá GAS ($)')
20 plt.xticks(x,rotation=90)
21 plt.grid(axis='y', ls='--')
22
23 plt.show()
```



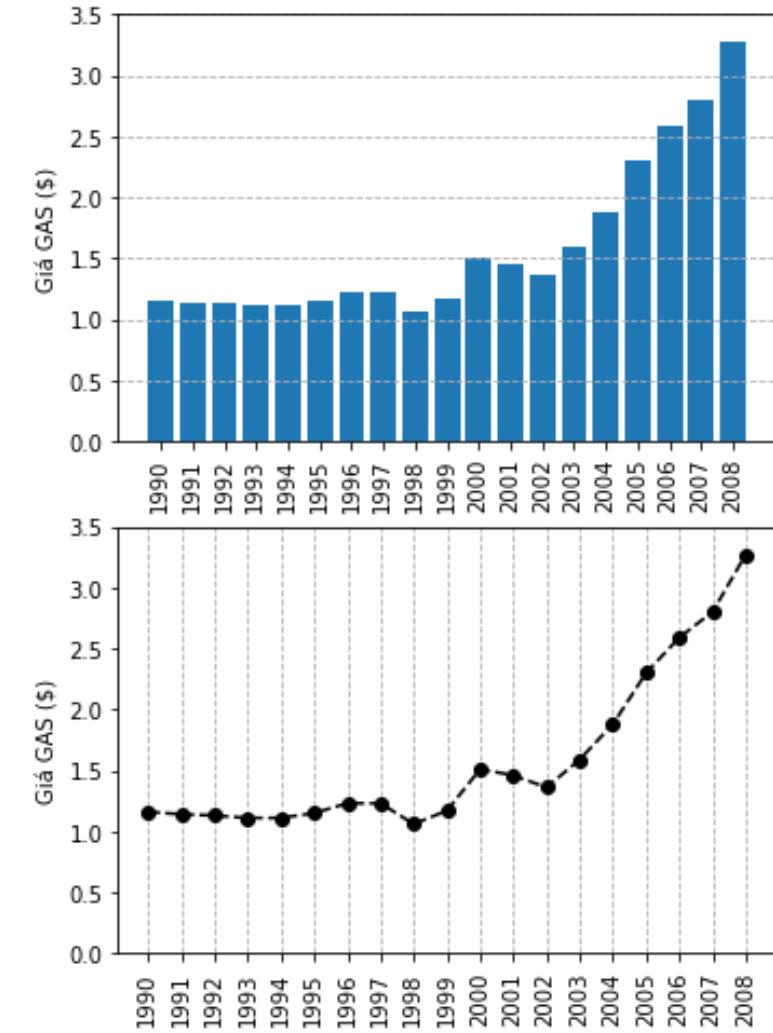
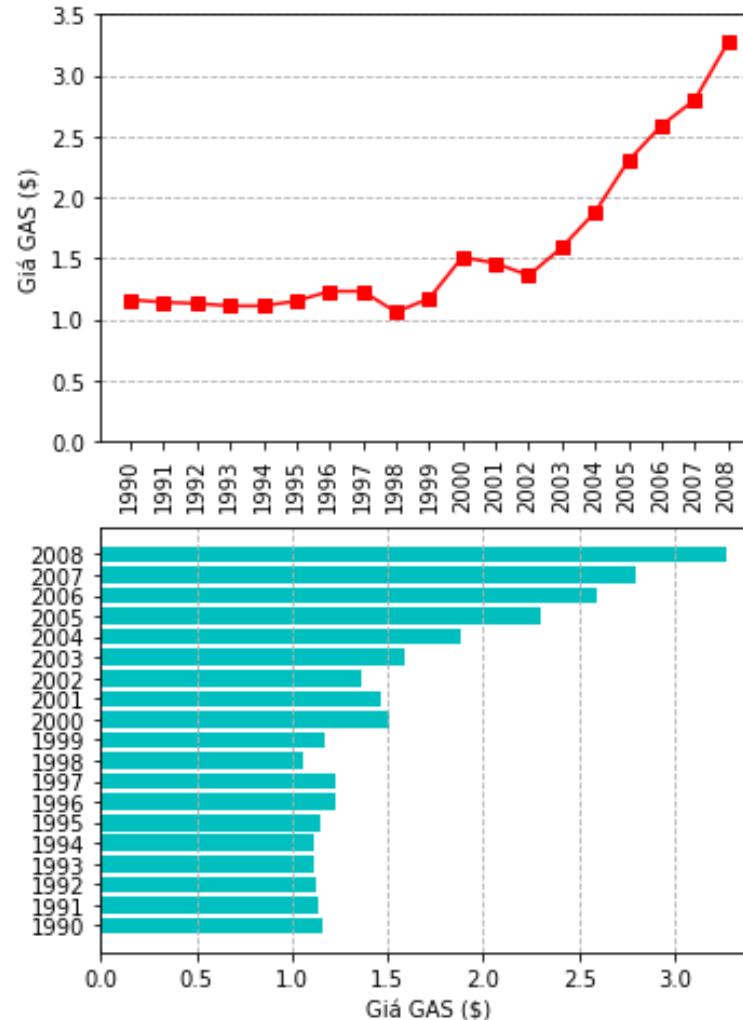
plt.subplot (1, 2, Position)



Multiple plot

```
1 plt.figure(figsize = (12,8))
2
3 #Thiết lập Khung biểu đồ gồm 2 hàng 2 cột
4 #####Vẽ biểu đồ đường trên khung 1#####
5 plt.subplot(2,2,1)
6 #Vẽ biểu đồ đường trên plot 1:
7 plt.plot(x, y,'r-s')
8
9 plt.ylabel('Giá GAS ($)')
10 plt.ylim(0,3.5)
11 plt.xticks(x,rotation=90)
12 plt.grid(axis='y',ls='--')
13 #####Vẽ biểu đồ đường trên khung 2#####
14 plt.subplot(2,2,2)
15 #Vẽ biểu đồ cột đứng trên plot 2:
16 plt.bar(x, y)
17 plt.ylabel('Giá GAS ($)')
18 plt.ylim(0,3.5)
19 plt.xticks(x,rotation=90)
20 plt.grid(axis='y',ls='--')
21 #####Vẽ biểu đồ đường trên khung 3#####
22 plt.subplot(2,2,3)
23 #Vẽ biểu đồ cột ngang trên plot 3:
24 plt.barh(x,y,color='c')
25 plt.xlabel('Giá GAS ($)')
26 plt.yticks(x)
27 plt.grid(axis='x',ls='--')
28 #####Vẽ biểu đồ đường trên khung 4#####
29 plt.subplot(2,2,4)
30 #Vẽ biểu đồ đường trên plot 4:
31 plt.plot(x, y,'k--o')
32 plt.ylabel('Giá GAS ($)')
33 plt.ylim(0,3.5)
34 plt.xticks(x,rotation=90)
35 plt.grid(axis='x',ls='--')
36
37 plt.show()
```

plt.subplot (2, 2, Position)



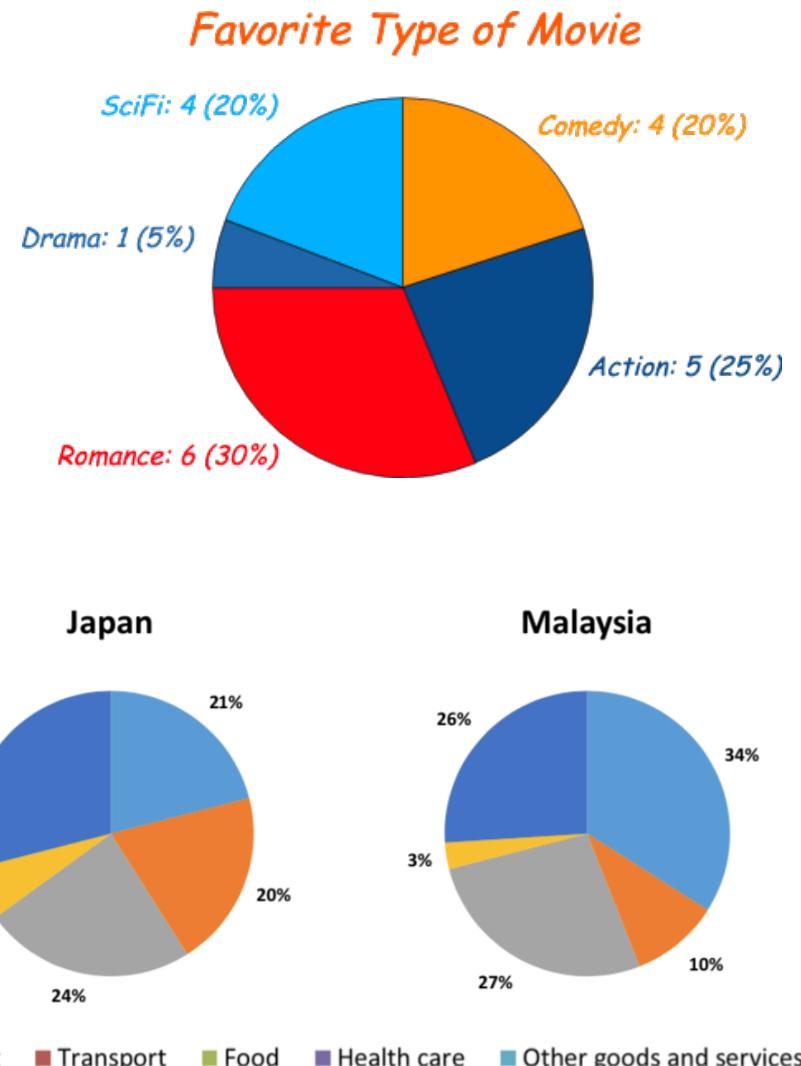


5. Biểu đồ tròn (hình bánh)



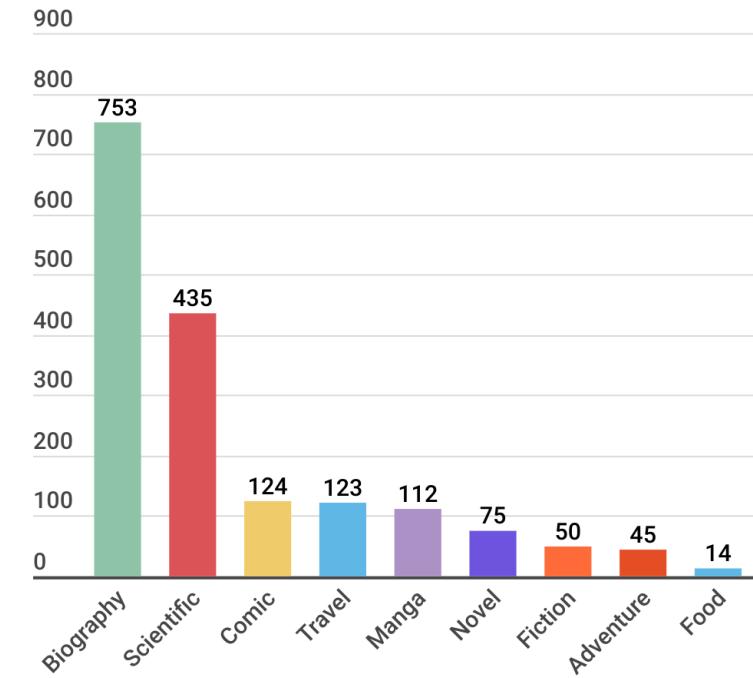
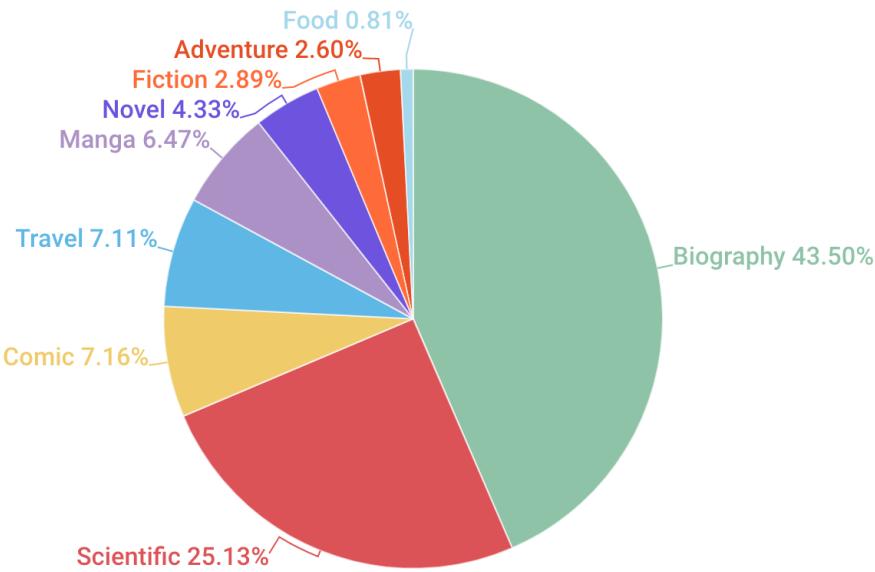
Biểu đồ tròn (Pie chart)

- **Biểu đồ tròn (Pie Chart)** là dạng biểu đồ hình tròn phẳng (cũng có tình huống được trình bày ở dạng 3D) dùng để so sánh giá trị phần trăm trong tổng thể.
- Các giá trị biểu diễn số liệu cho một đối tượng thông qua màu sắc riêng biệt. Đối tượng nào có màu sắc tương ứng đó và được liệt kê ở phần chú thích của biểu đồ. Phần màu càng lớn thì số liệu càng lớn và ngược lại.
- Pie Chart được sử dụng để biểu diễn tỉ lệ phần trăm của các thành phần so với tổng thể. Vì vậy, nó không được dùng để biểu diễn giá trị chính xác.



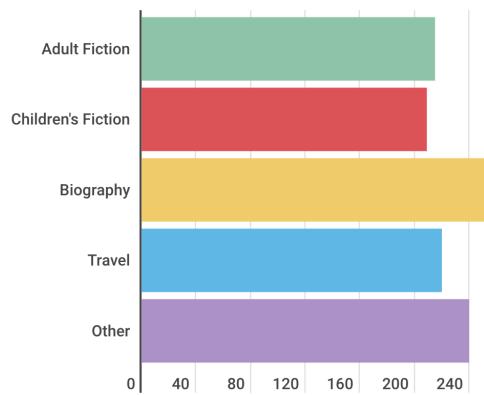
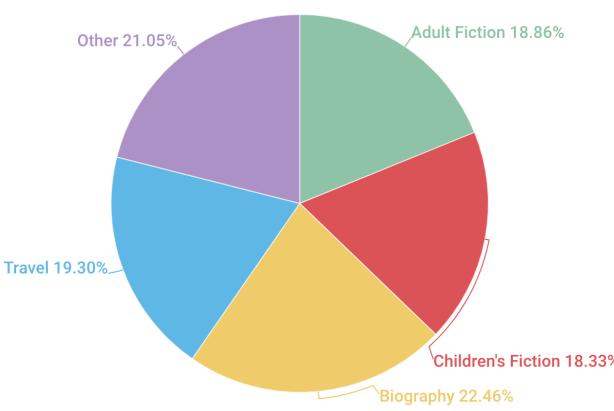
Lưu ý khi sử dụng Pie chart

- Đảm bảo tổng các thành phần là 100%:** Với các công cụ hỗ trợ thì không cần lo lắng về lỗi này vì các công cụ đã đảm bảo được sự chính xác của số liệu khi biểu diễn. Nếu vẽ Pie chart thủ công thì chúng ta cần kiểm tra lại tính đúng đắn một lần nữa.
- Chỉ nên dùng Pie chart khi số lượng thể loại ít hơn 6:** Việc sử dụng Pie Chart khi có quá nhiều thể loại sẽ khiến cho biểu đồ khá rối. Nếu có quá nhiều thể loại, nên xem xét một biểu đồ khác như Bar Chart.

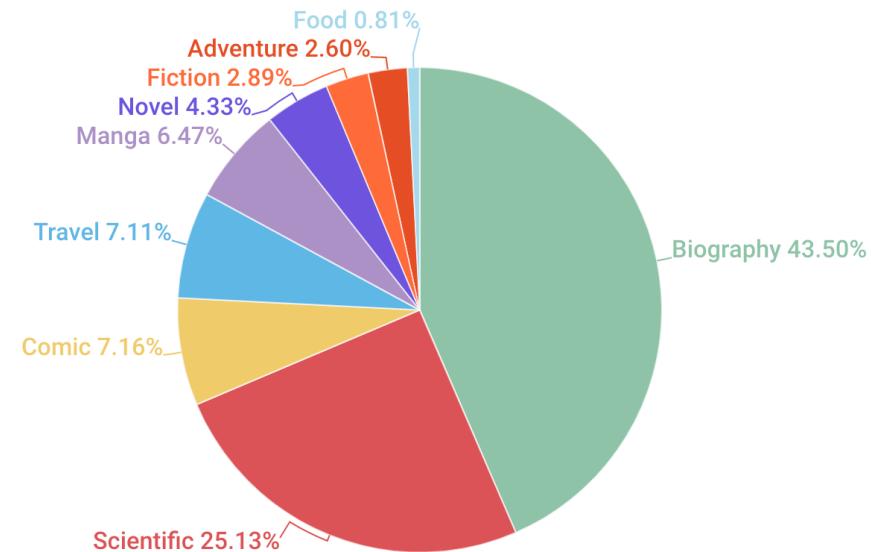


Lưu ý khi sử dụng Pie chart

3. Không dùng Pie Chart nếu tỉ lệ giữa các thể loại gần tương đương nhau: Nếu tỉ lệ giữa các thể loại là tương đương nhau thì dường như Pie Chart lúc này là vô dụng vì không thể hiện cụ thể một ý nghĩa gì. Giải pháp lúc này là xem xét một dạng biểu đồ khác như **Column Chart** hoặc **Bar Chart**.

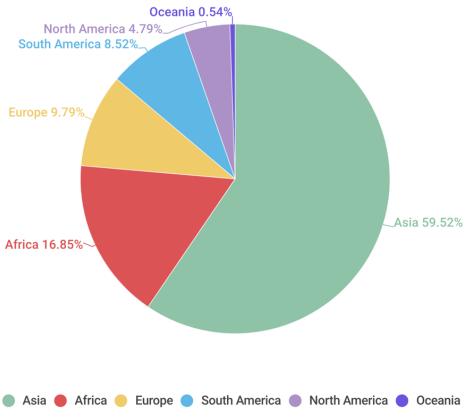


4. Nên sắp xếp giá trị các thể loại để dễ hiểu hơn: Sắp xếp lại dữ liệu giúp cho người xem nhận ra ngay thể loại có tỉ lệ cao nhất. Đồng thời với 2 thể loại gần nhau tương đương thì biết được thể loại nào có giá trị lớn hơn. Thông thường, giá trị trong Pie Chart được sắp xếp từ lớn đến nhỏ theo chiều kim đồng hồ như ví dụ bên dưới.

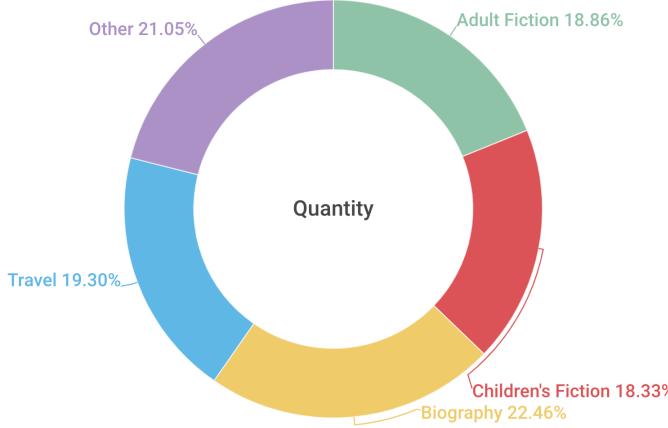


Một số dạng Pie chart

Global population by continent as of mid-2018



1. Pie chart

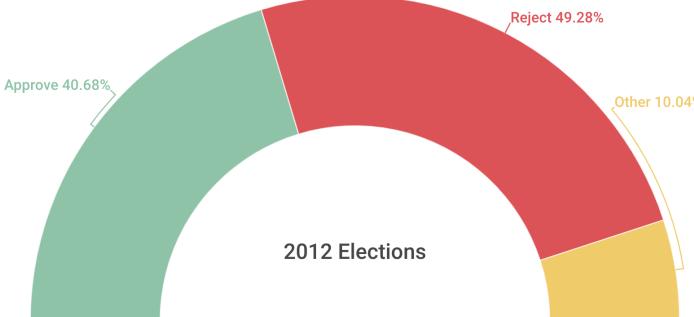


2. Donut chart

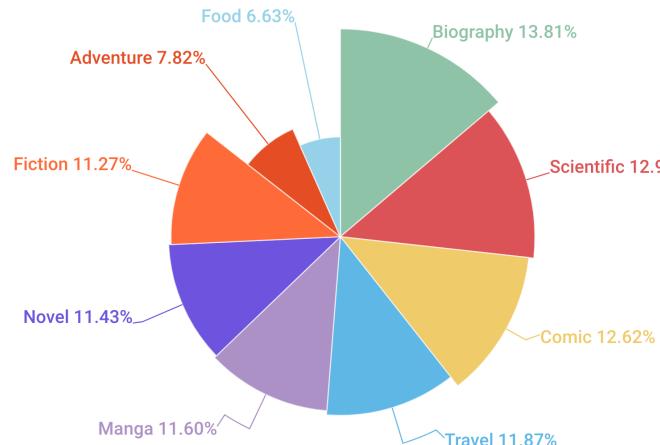
Total Revenue
Q1 1997 Q2 1997 Q3 1997 Q4 1997



3. Stacked Donut chart



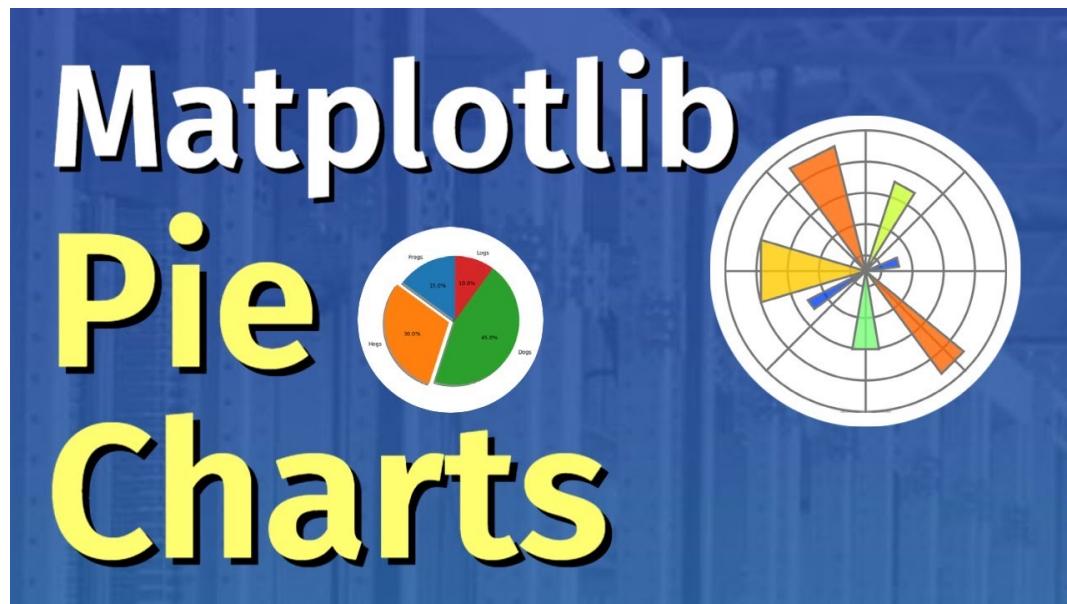
4. Semi-Circle Pie chart



5. Irregular Pie chart

Biểu đồ Pie chart với Matplotlib

Số liệu: Số lượng SV của các Khoa theo Giới tính.



Khoa	K60	K61	K62	K63	K64	K65
Nam	200	340	260	440	300	180
Nữ	30	60	90	160	180	220

```
1 #Tạo dữ liệu: Số lượng SV Nam - Nữ theo từng khoá.  
2 labels = ['K60', 'K61', 'K62', 'K63', 'K64', 'K65']  
3 boys = [200, 340, 260, 440, 300, 180]  
4 girls = [30, 60, 90, 160, 180, 220]  
5 total = list(np.array(boys) + np.array(girls))  
6  
7 sex =[ 'Nam', 'Nữ']  
8 sum_boy = sum(boys)  
9 sum_girl = sum(girls)
```

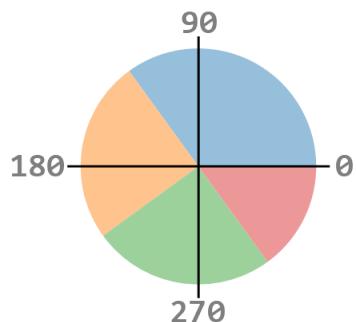


a. Pie chart

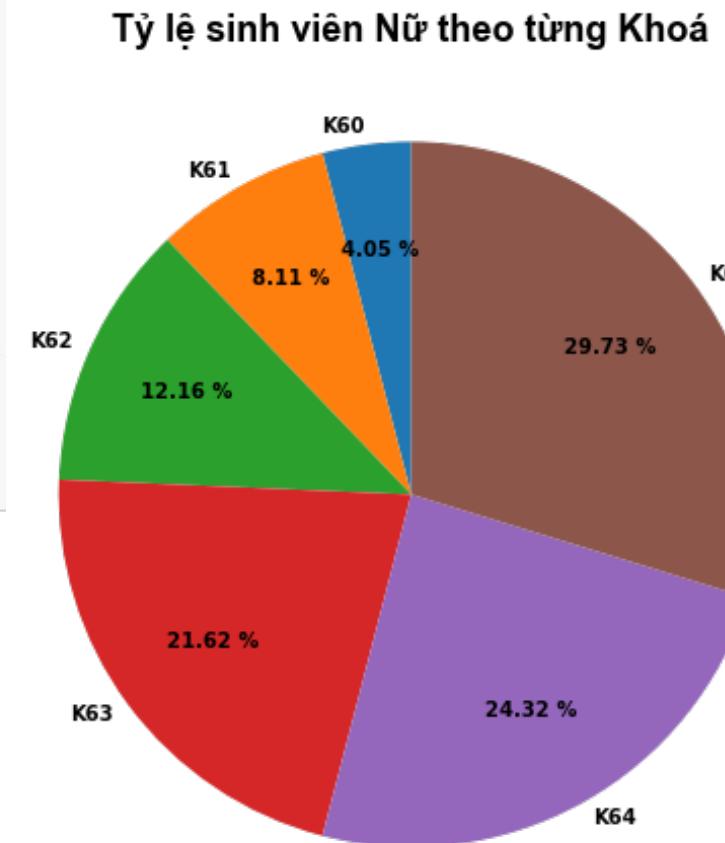
Biểu đồ tròn (Pie chart)

Cú pháp: plt.pie (values, labels)

```
1 plt.figure(figsize = (12,8))
2 #Vẽ biểu đồ tròn:
3 plt.pie(girls,           #Giá trị thể hiện
4          labels=labels,    #Nhãn tương ứng
5          autopct='%.2f %%', #Tính toán và hiển thị % tương ứng
6          pctdistance=0.7,   #Khoảng cách hiển thị giá trị % tới tâm.
7          startangle=90,     #Góc bắt đầu của biểu đồ
8          textprops={'color':'k','fontweight':'bold'},#thiết lập label
9          labeldistance=1.05, #Khoảng cách từ label tới biểu đồ
10         rotatelabels=False) #Label có xoay không?
11
12 plt.title('Tỷ lệ sinh viên Nữ theo từng Khoa',
13            fontdict={'fontname':'Arial',
14                      'fontweight':'bold',
15                      'fontsize':18})
16 plt.show()
```



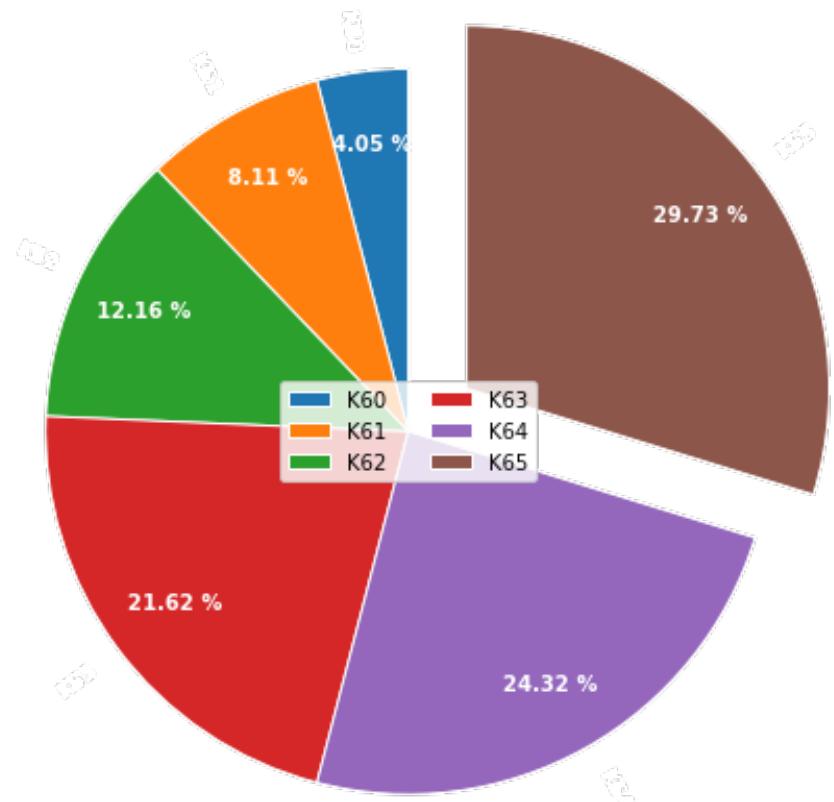
startangle:



Biểu đồ tròn (Pie chart)

```
1 plt.figure(figsize = (12,8))
2 #Làm nổi bật một phần:
3 e = [0,0,0,0,0,0.2]
4 plt.pie(girls,
5         labels=labels,
6         autopct='%.2f %%',
7         pctdistance=0.8,
8         startangle=90,
9         labeldistance=1.05,
10        textprops={'color':'w','fontweight':'bold'},
11        rotatelabels=True,
12        wedgeprops=dict(edgecolor='w'),#Đường viền màu trắng
13        explode=e)#Làm nổi bật một phần
14
15 plt.title('Tỷ lệ sinh viên Nữ theo từng Khoa', fontdict={'fontname':'Arial',
16                                         'fontweight':'bold',
17                                         'fontsize':18})
18 plt.legend(ncol=2, loc='center')
19 plt.show()
```

Tỷ lệ sinh viên Nữ theo từng Khoa



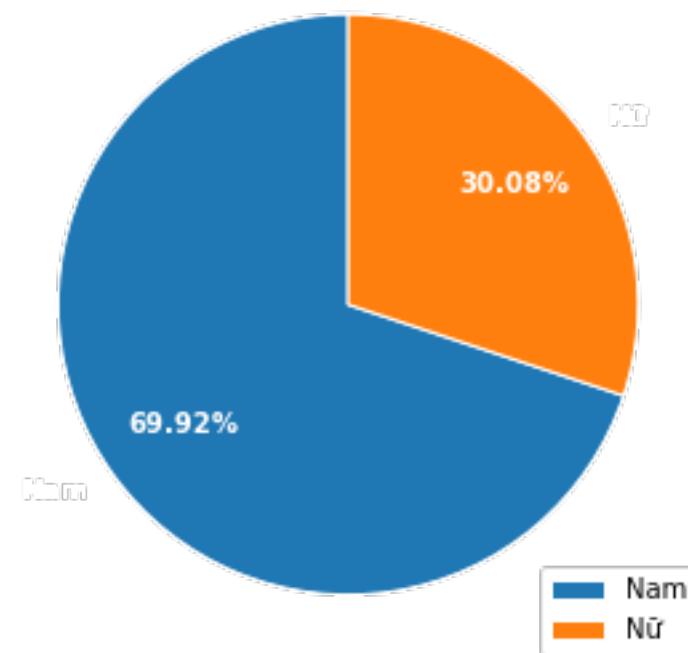
Thực hành 3



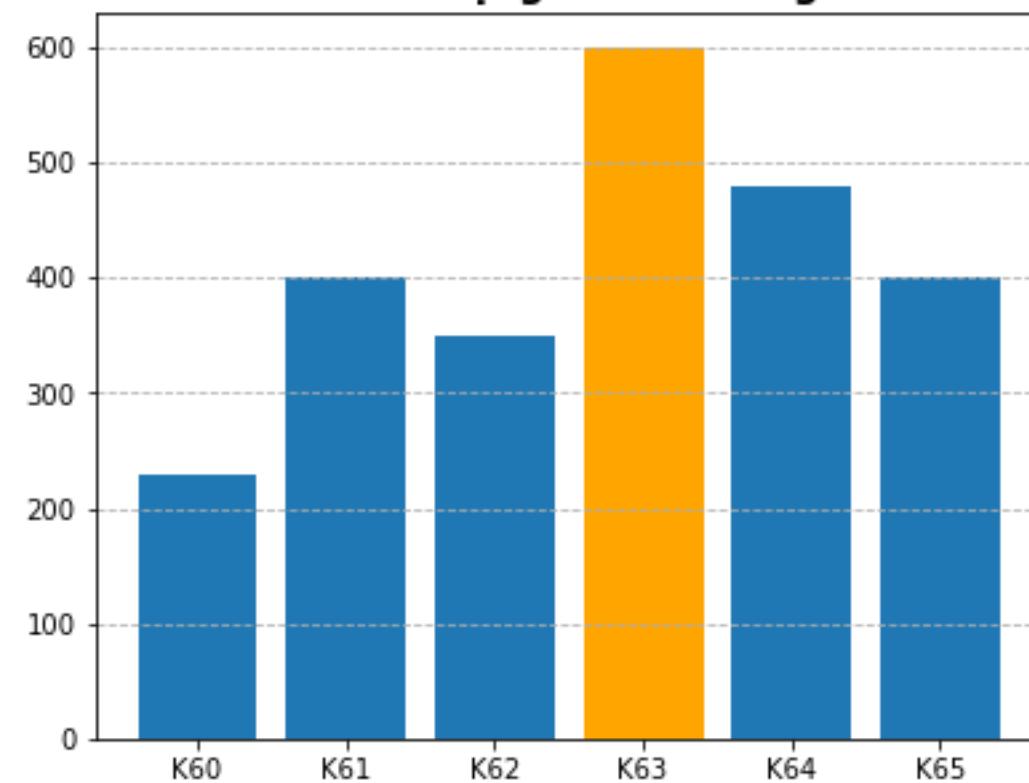
Thực hành

Yêu cầu: Sử dụng số liệu đã có, thực hiện vẽ biểu đồ như minh họa dưới đây:

Biểu đồ tỷ lệ SV Nam - Nữ của Khoa CNTT



Biểu đồ số lượng SV theo từng khoá





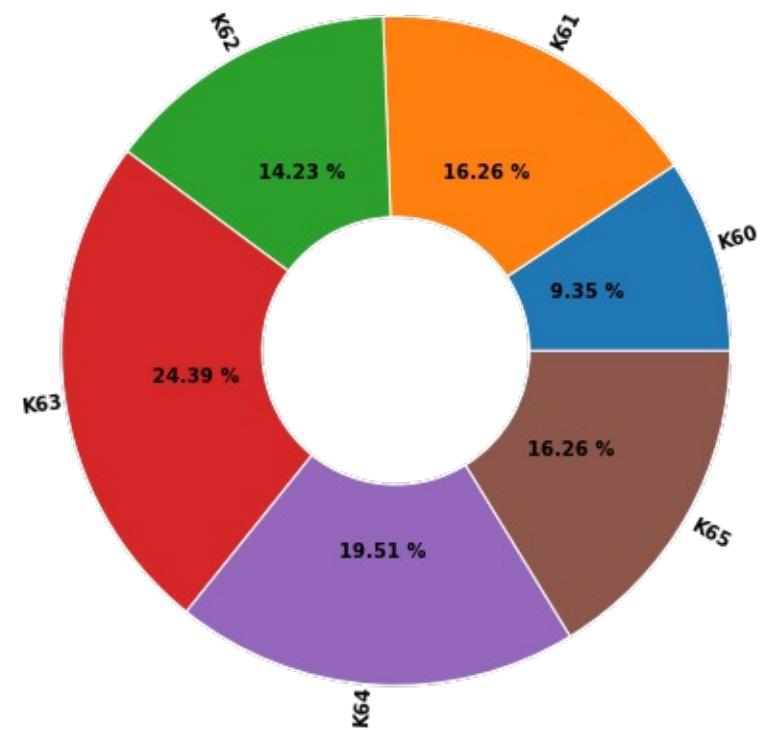
b. Donus chart

Donut chart

Cú pháp: plt.pie (values, labels)

```
1 plt.figure(figsize = (12,8))
2 #Vẽ biểu đồ:
3 plt.pie(total,
4         labels=labels,
5         textprops={'color':'k','fontweight':'bold'},
6         rotatelabels=True,
7         labeldistance=1.0,
8         wedgeprops=dict(width=0.6,edgecolor='w'), #Xác định độ rộng của Pie
9         autopct='%.2f %%',
10        pctdistance=0.6)
11
12 plt.title('TỶ LỆ SINH VIÊN CỦA TỪNG KHOÁ', fontdict={'fontname':'Tahoma',
13                                         'fontweight':'bold',
14                                         'fontsize':18})
15 plt.show()
```

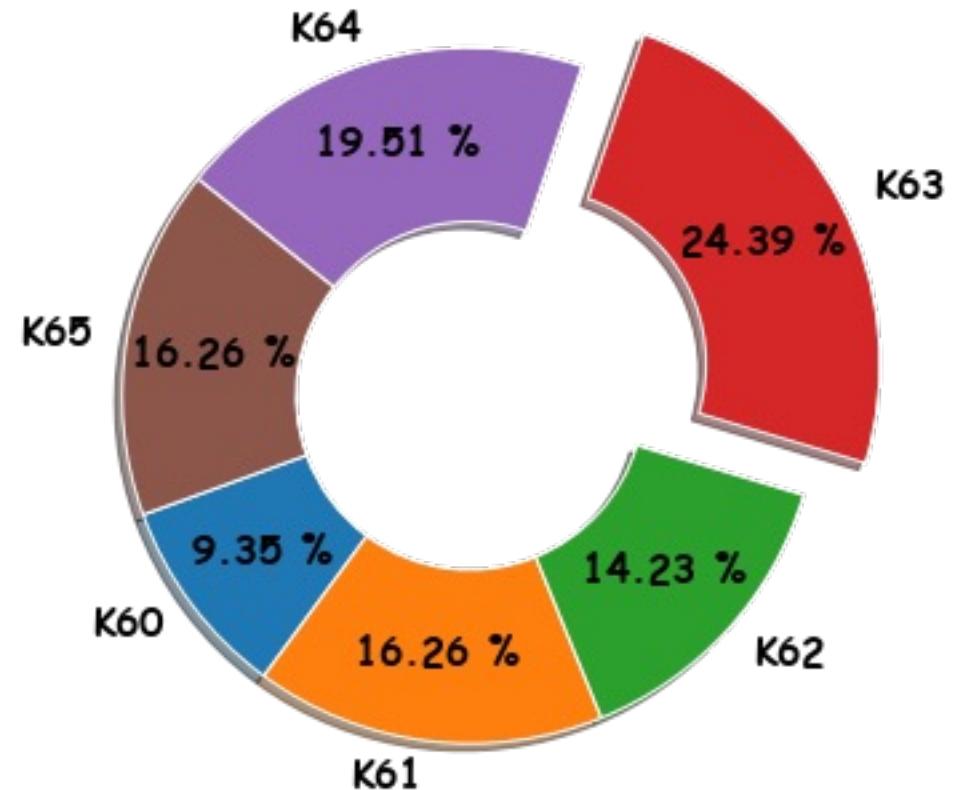
TỶ LỆ SINH VIÊN CỦA TỪNG KHOÁ



Donut chart

```
1 plt.figure(figsize = (12,6))
2
3 #Làm nổi bật một phần
4 e = [0,0,0,0.2,0,0]
5 plt.pie(total,
6         labels=labels,
7         explode=e, #Làm nổi bật biểu đồ
8         startangle=200,
9         textprops={'color':'k','fontweight':'bold',
10                 'fontname':'Comic Sans MS',
11                 'fontsize':15},
12         wedgeprops=dict(width=0.5,edgecolor='w'),
13         shadow=True,      #Tạo bóng cho biểu đồ
14         autopct='%.2f %%',
15         pctdistance=0.75)
16
17 plt.title('TỶ LỆ SINH VIÊN CỦA TỪNG KHOÁ',
18             fontdict={'fontname':'Tahoma',
19                         'fontweight':'bold',
20                         'fontsize':18})
21 plt.show()
```

TỶ LỆ SINH VIÊN CỦA TỪNG KHOÁ

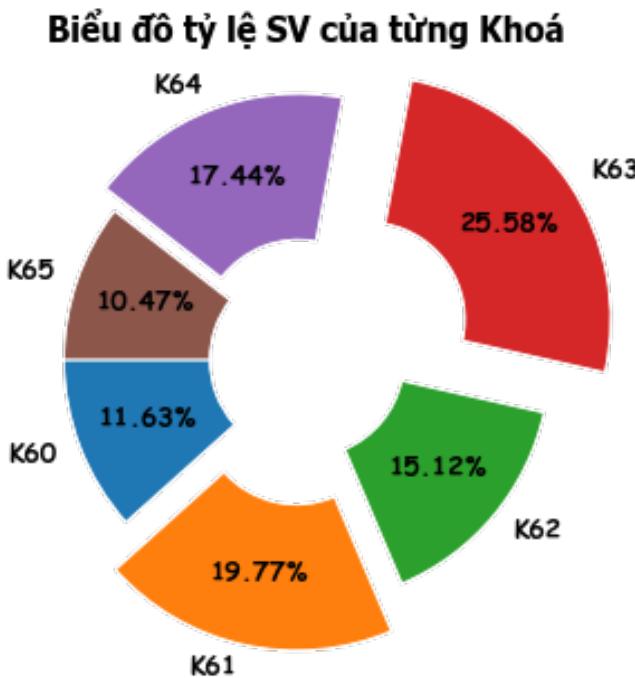
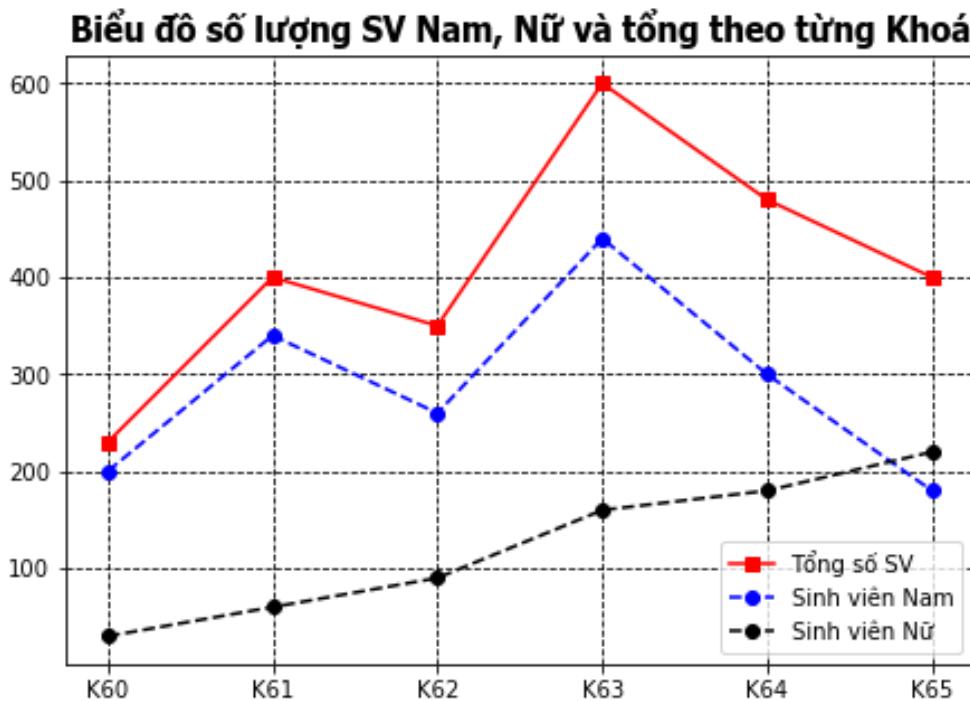


Thực hành 4



Thực hành

Yêu cầu: Sử dụng số liệu đã có, thực hiện vẽ biểu đồ line chart và pie chart như minh họa dưới đây:

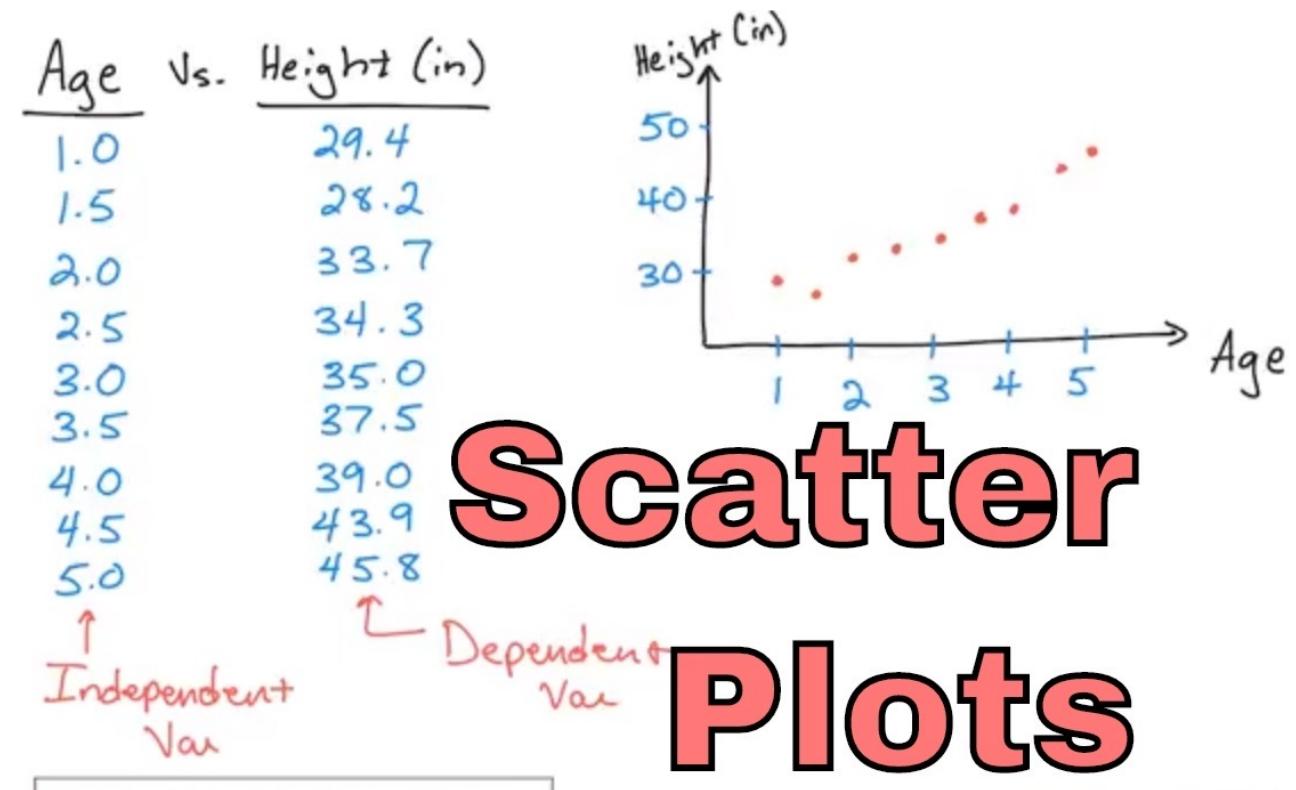


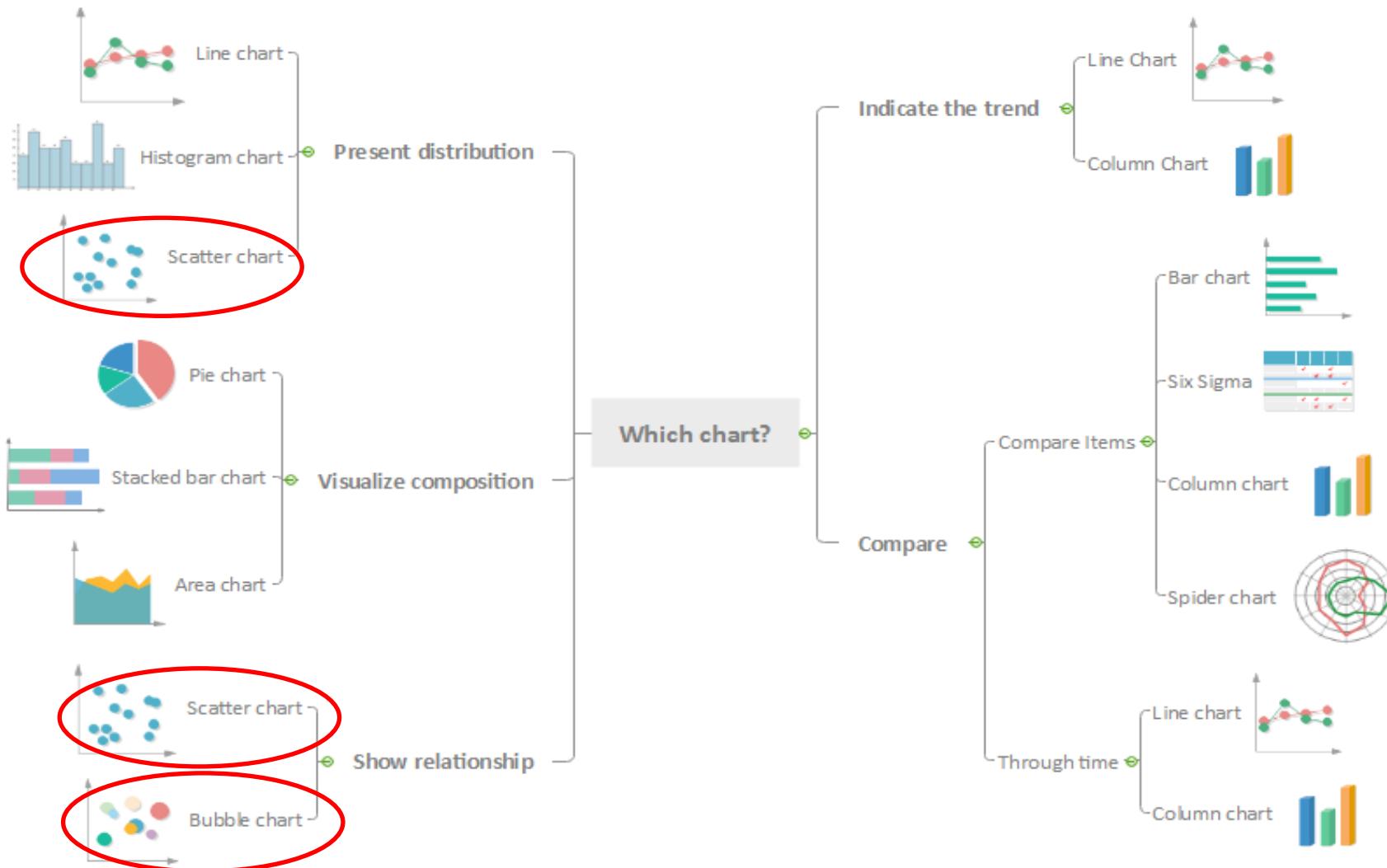


6. Biểu đồ phân tán (Scatter chart)

Biểu đồ phân tán (Scatter chart)

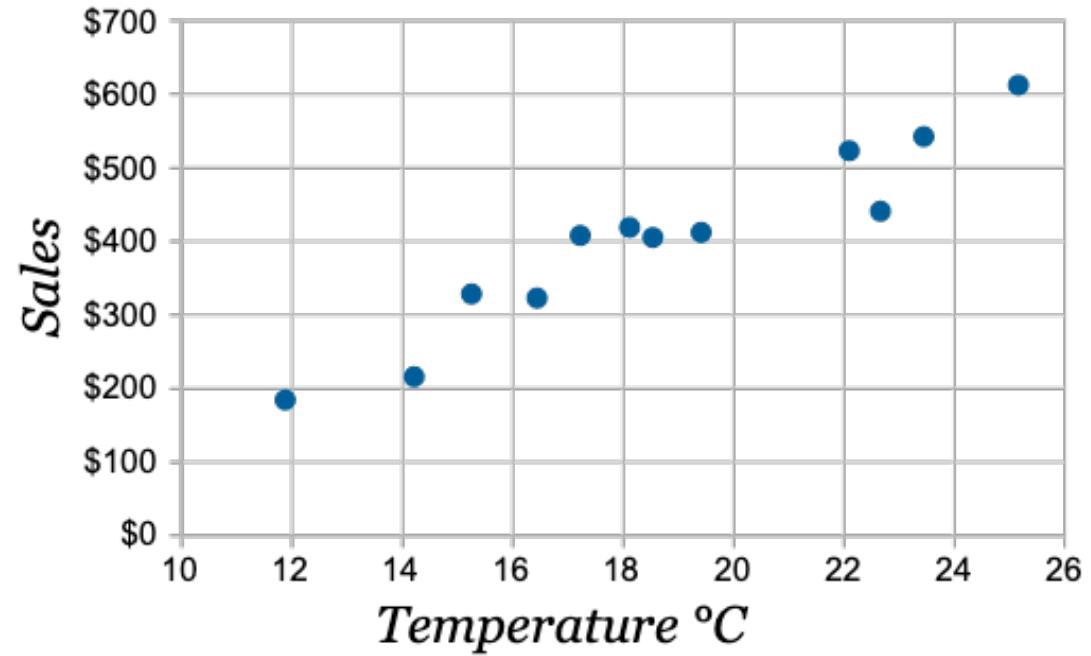
- **Biểu đồ phân tán (Scatter chart, Scatterplot, scatter graph)** là loại biểu đồ được dựng bởi các điểm theo tọa độ toán học để xác định mối tương quan giữa 2 biến.
- Đồ thị thể hiện 2 bộ dữ liệu, trục tung Y được sử dụng cho biến được dự đoán (biến phụ thuộc), trục hoành X được sử dụng cho biến dùng để dự đoán (biến độc lập).
- Scatter plot được sử dụng khi có 2 cặp dữ liệu (biến) và muốn xác định 2 biến có liên quan với nhau hay không? Liên quan nhiều hay ít và như thế nào?





Biểu đồ phân tán (Scatter chart)

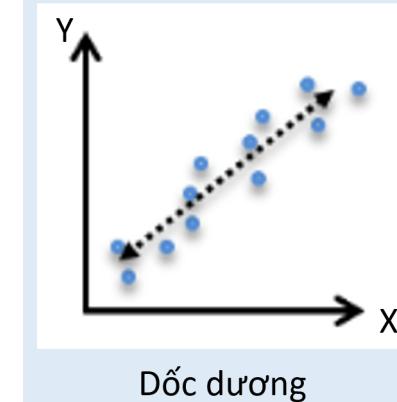
X và Y
có
tương
quan...



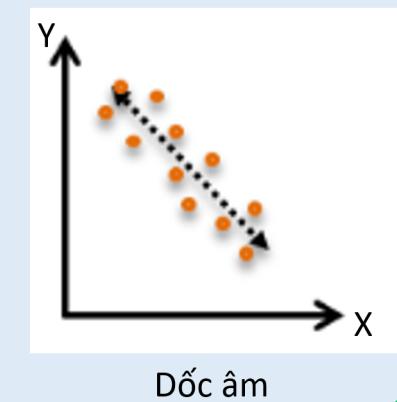
- **Biểu đồ phân tán** sẽ cho chúng ta thấy được mức độ tương quan giữa 2 biến

Biểu đồ phân tán (Scatter chart)

- Dựa vào hình dạng, bờ dốc và độ tập trung điểm của biểu đồ để xác định mối tương quan giữa 2 biến.



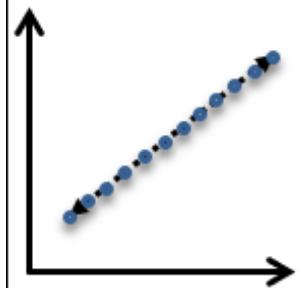
Dốc dương



Dốc âm

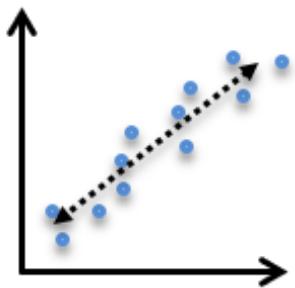
Scatter Plots & Correlation Examples

Perfect
Positive
Correlation



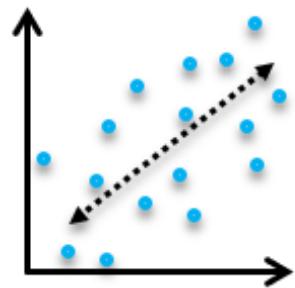
$r = 1$

Highly
Positive
Correlation



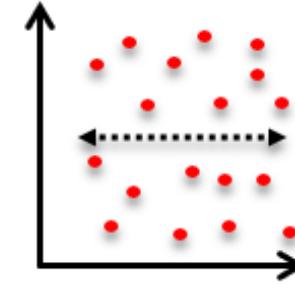
$r = 0.8$

Low
Positive
Correlation



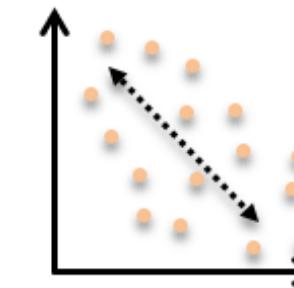
$r = 0.3$

No
Correlation



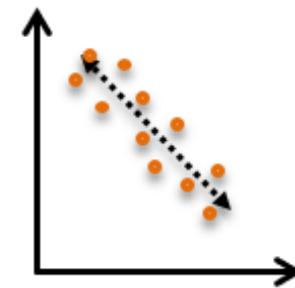
$r = 0$

Low
Negative
Correlation



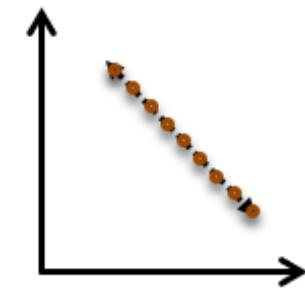
$r = -0.3$

Highly
Negative
Correlation



$r = -0.8$

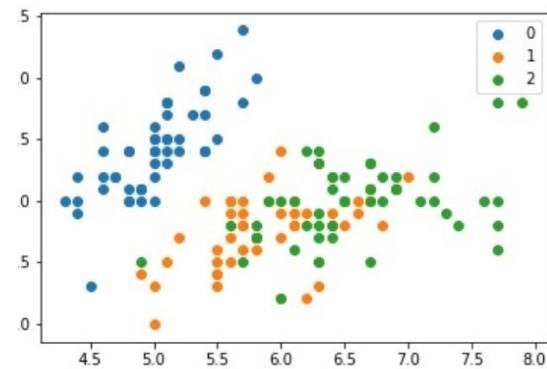
Perfect
Negative
Correlation



$r = -1$

Scatter chart với Matplotlib

Tập dữ liệu **Diamonds.txt** lưu trữ trọng lượng (cara) và giá (\$) tương ứng của 50 viên kim cương.



MATPLOTLIB
SCATTER PLOTS

DATA VISUALIZATION PYTHON

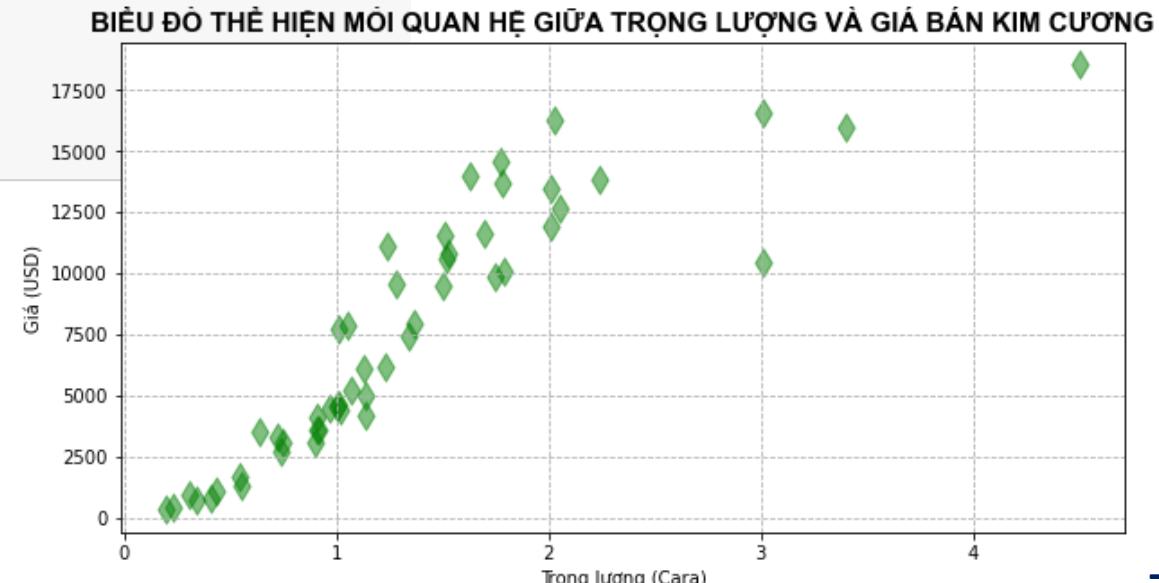


Dia...	
0.23	484
0.31	942
0.2	345
1.02	4459
1.63	14022
1.14	4212
2.01	11925
1.28	9548
1.7	11605
1.01	4642
0.64	3541
0.97	4504
1.78	13691
3.4	15964
3.01	10453
1.51	11560
1.37	7979
1.5	9533
0.54	1723
0.72	3344
1.13	6133
2.24	13827
3.01	16538
4.5	18531
0.92	3625
1.05	7879
0.55	1319
0.74	2761
0.91	3620
1.23	6165

Biểu đồ phân tán (Scatter chart)

Cú pháp: plt.scatter (x, y)

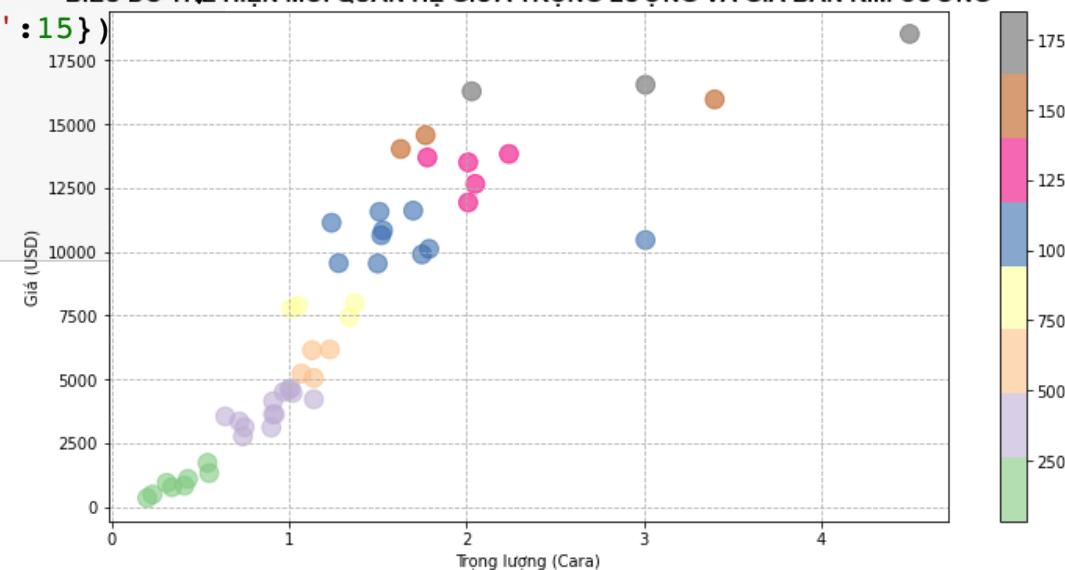
```
1 plt.figure(figsize = (10,5))
2
3 plt.scatter(weight,           #Dữ liệu trục X
4               price,          #Dữ liệu trục Y
5               c='g',            #Màu của Point
6               marker='d',       #Kiểu Point
7               s=120,             #Kích thước của Point
8               alpha=0.5)        #Độ trong suốt của Point
9
10 plt.title('BIỂU ĐỒ THỂ HIỆN MỐI QUAN HỆ GIỮA TRỌNG LƯỢNG VÀ GIÁ BÁN KIM CƯƠNG',
11            fontdict={'fontname':'Arial','fontweight':'bold','fontsize':15})
12 plt.grid(ls='--')
13 plt.xlabel('Trọng lượng (Cara)')
14 plt.ylabel('Giá (USD)')
15 plt.show()
```



Biểu đồ phân tán (Scatter chart)

Sử dụng colorbar:

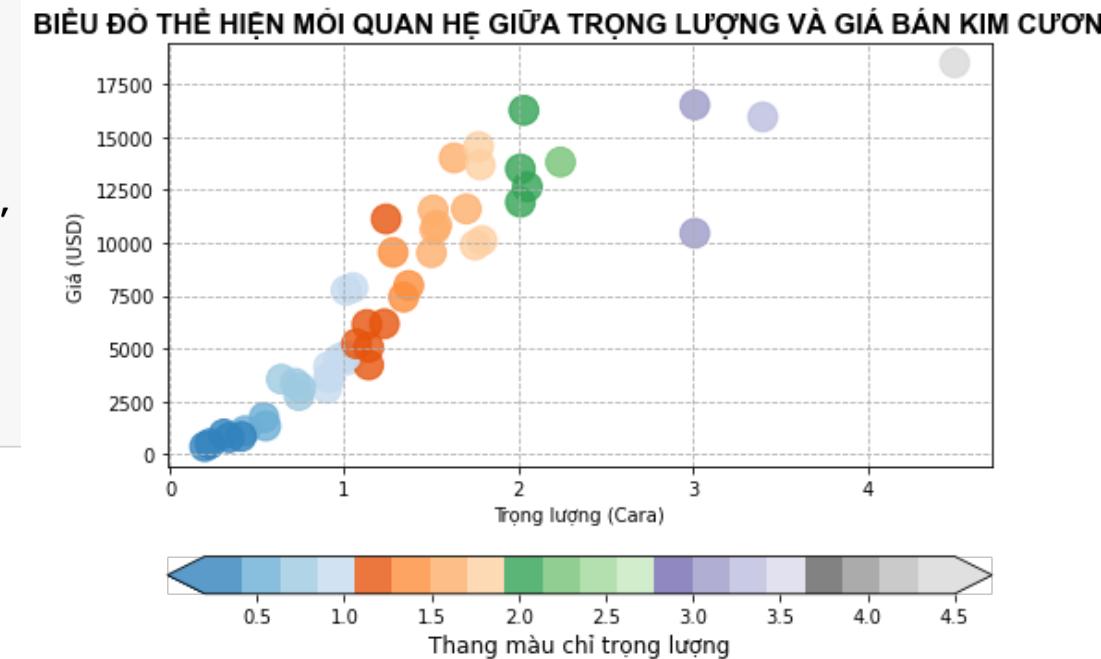
```
1 plt.figure(figsize = (12,6))
2 #Vẽ biểu đồ scatter:
3 plt.scatter(weight, price,
4             s=140,
5             alpha=0.6,
6             c=price,
7             cmap='Accent') #Sử dụng color map
8 plt.colorbar() #Hiển thị thanh color bar:
9
10
11 plt.title('BIỂU ĐỒ THỂ HIỆN MỐI QUAN HỆ GIỮA TRỌNG LƯỢNG VÀ GIÁ BÁN KIM CƯƠNG')
12 fontdict={'fontname':'Arial','fontweight':'bold','fontsize':15})
13 plt.grid(ls='--')
14 plt.xlabel('Trọng lượng (Cara)')
15 plt.ylabel('Giá (USD)')
16
17 plt.show()
```



Biểu đồ phân tán (Scatter chart)

Sử dụng colorbar:

```
1 plt.figure(figsize = (8,6))
2 #Vẽ biểu đồ scatter:
3 plt.scatter(weight, price,
4             s=250,
5             alpha=0.8,
6             c=weight,
7             cmap='tab20c') #Sử dụng color map
8
9 #Hiển thị và setup thanh color bar:
10 cbar = plt.colorbar(location ='bottom', #Vị trí của colorbar
11                      extend='both', #Đầu của colorbar
12                      pad=0.15) #Khoảng cách giữa thang màu và biểu đồ
13 cbar.set_label(label='Thang màu chỉ trọng lượng',size=12)
14
15 plt.title('BIỂU ĐỒ THỂ HIỆN MỐI QUAN HỆ GIỮA TRỌNG LƯỢNG VÀ GIÁ BÁN KIM CƯƠNG',
16            fontdict={'fontname':'Arial','fontweight':'bold','fontsize':15})
17 plt.grid(ls='--')
18 plt.xlabel('Trọng lượng (Cara)')
19 plt.ylabel('Giá (USD)')
20
21 plt.show()
```



Bài thực hành tổng hợp



Thực hành

Mô tả tập dữ liệu:

Tập dữ liệu chứa thông tin của 500 người, bao gồm:

- **Gender** : Male / Female
- **Age** : Number (Age)
- **Height** : Number (cm)
- **Weight** : Number (Kg)
- **Index** :
 - 0 - Extremely Weak
 - 1 - Weak
 - 2 - Normal
 - 3 - Overweight
 - 4 - Obesity
 - 5 - Extreme Obesity

Data_500

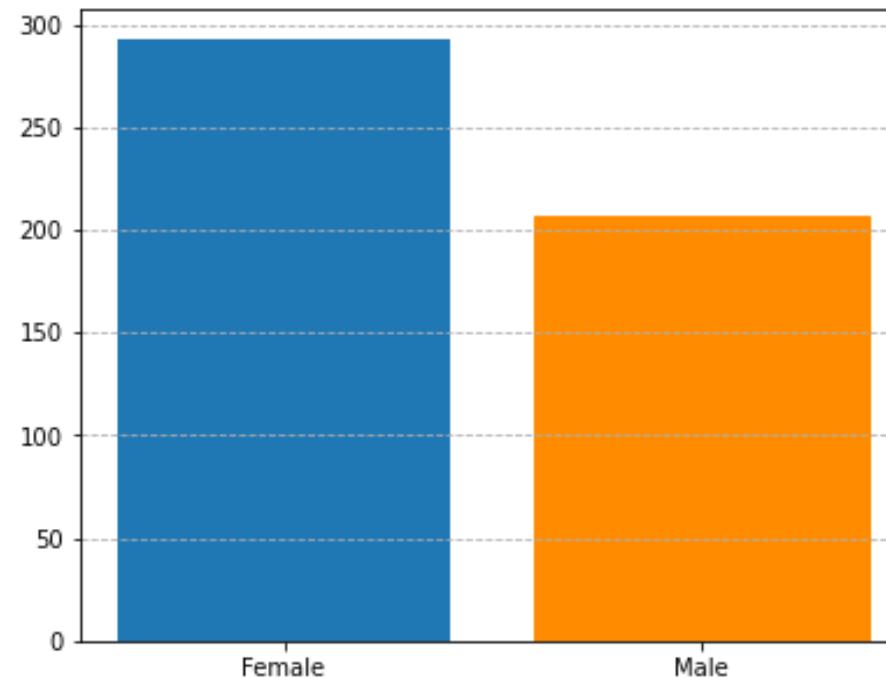
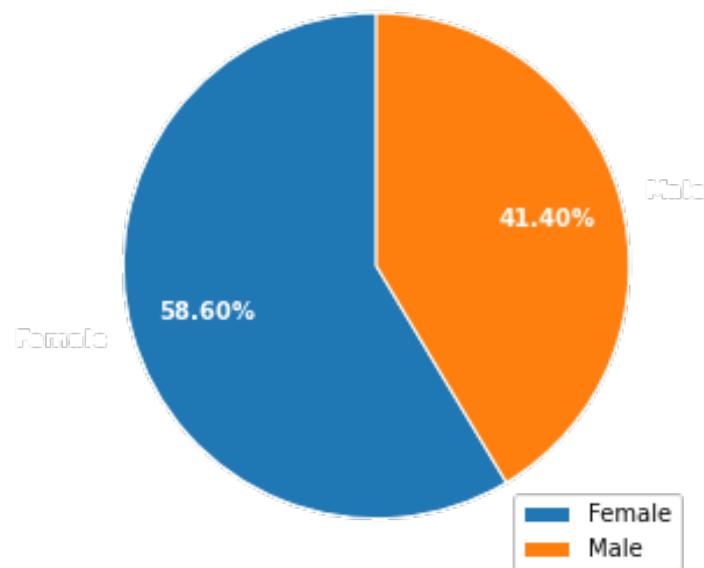
Gender	Age	Height	Weight	Index
Male	28	174	96	4
Male	24	189	87	2
Female	39	185	110	4
Female	49	195	104	3
Male	26	149	61	3
Male	26	189	104	3
Male	39	147	92	5

Thực hành

Yêu cầu 1:

Thực hiện thống kê số lượng Nam – Nữ trong tập dữ liệu và trực quan hóa kết quả:

THỐNG KÊ DỮ LIỆU THEO GIỚI TÍNH TRONG TẬP DỮ LIỆU

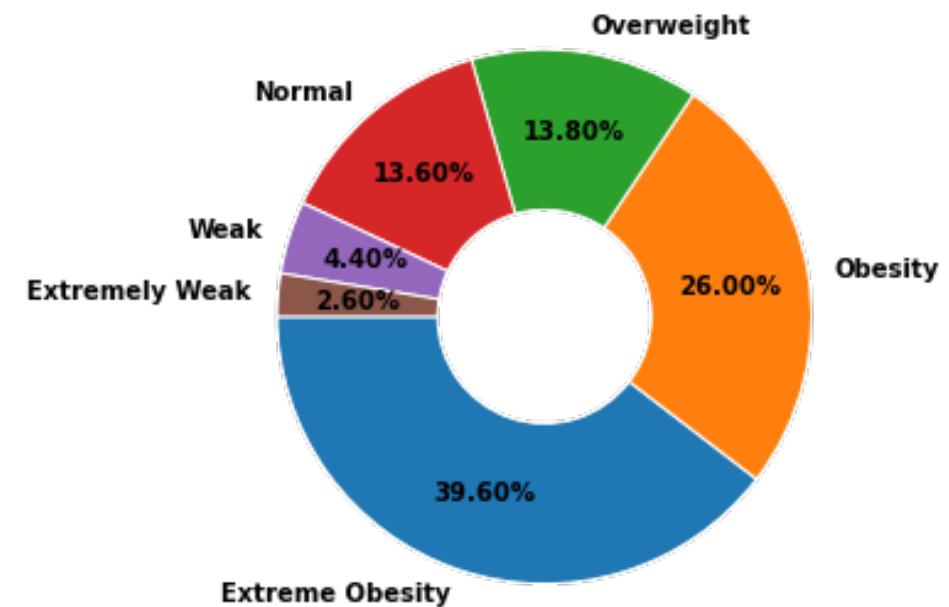
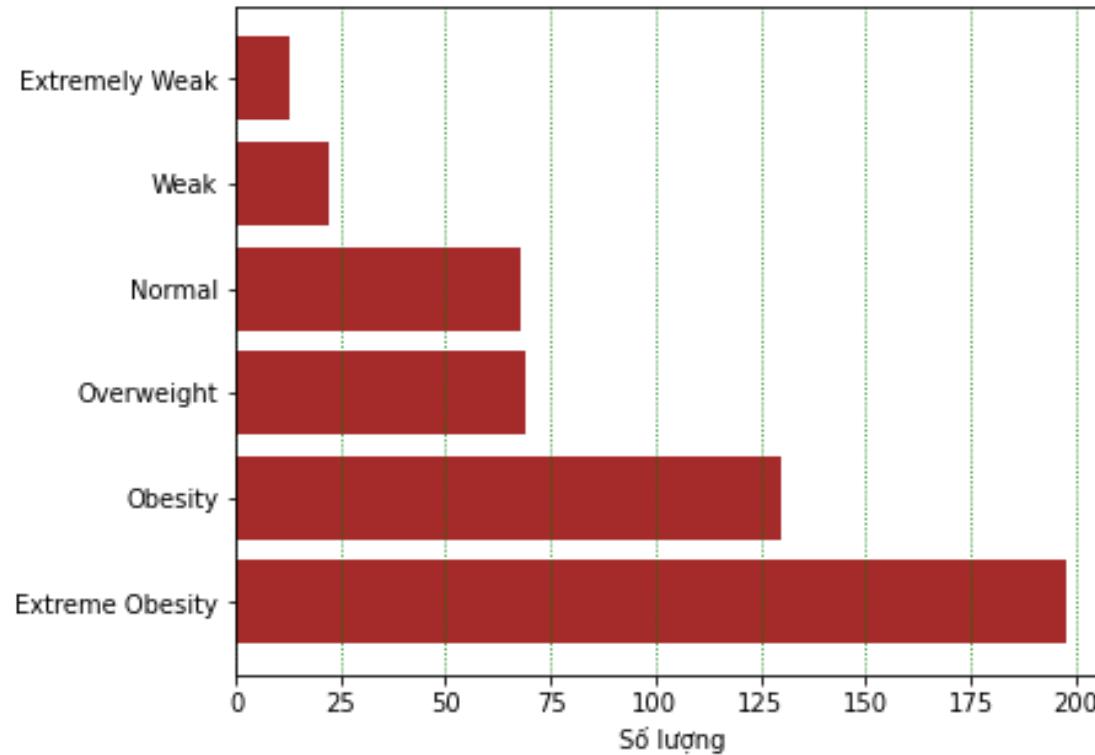


Thực hành

Yêu cầu 2:

Thực hiện thống kê số lượng theo chỉ số cơ thể (index) và trực quan hóa kết quả.

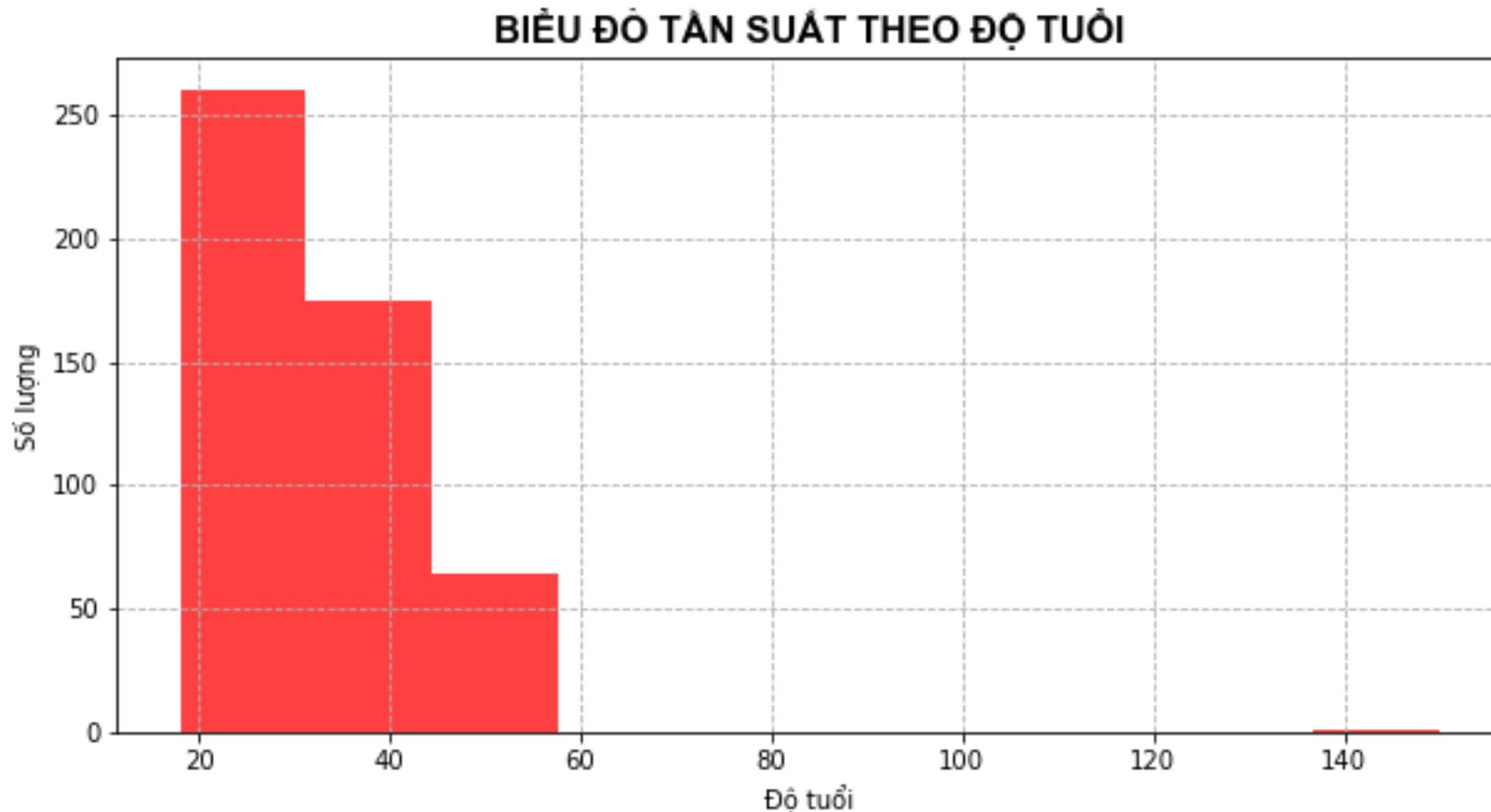
THỐNG KÊ DỮ LIỆU THEO TRẠNG THÁI CƠ THỂ TRONG TẬP DỮ LIỆU



Thực hành

Yêu cầu 3:

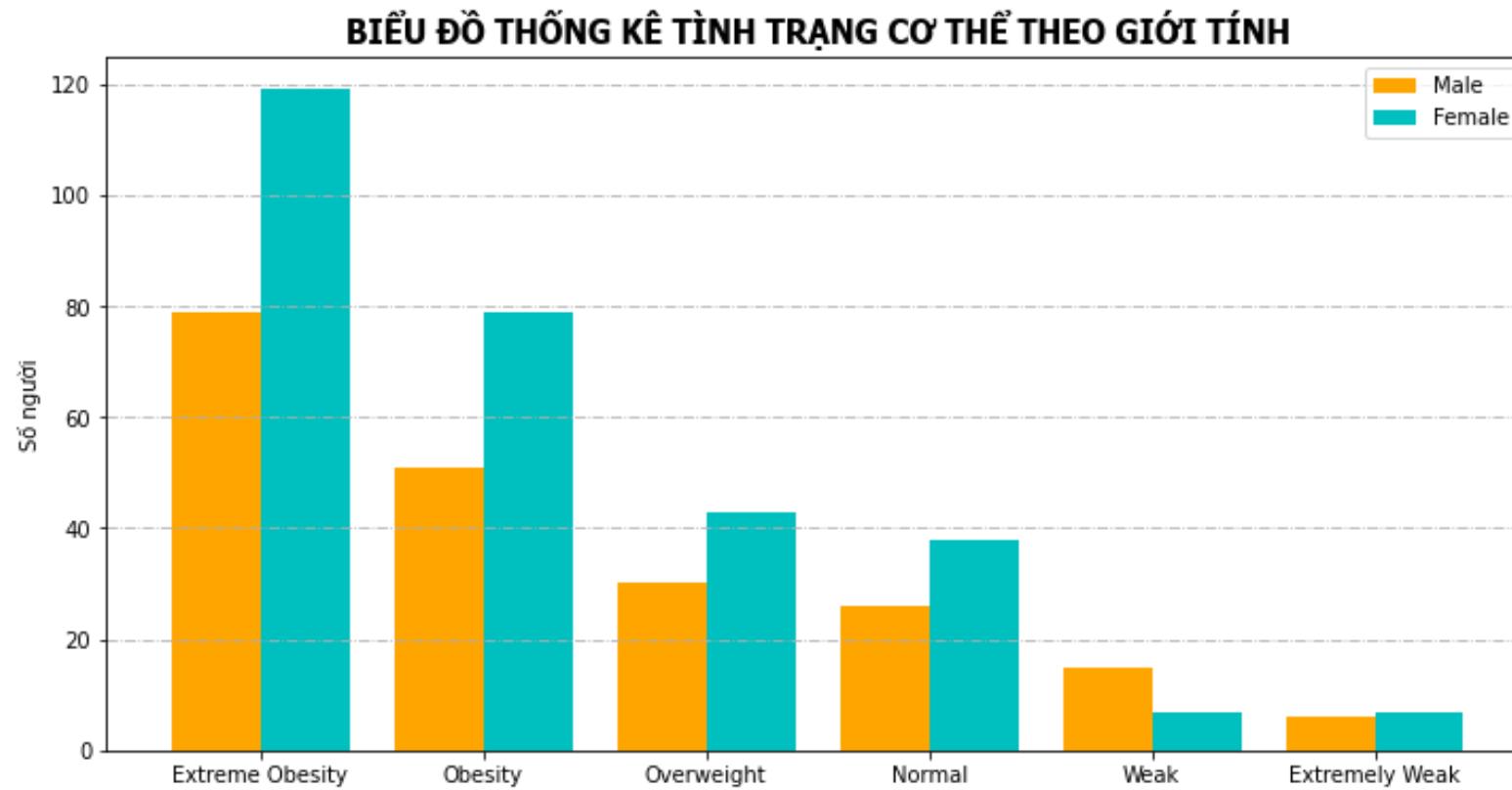
Trực quan hóa phân bố độ tuổi trong tập dữ liệu → cho nhận xét?



Thực hành

Yêu cầu 4:

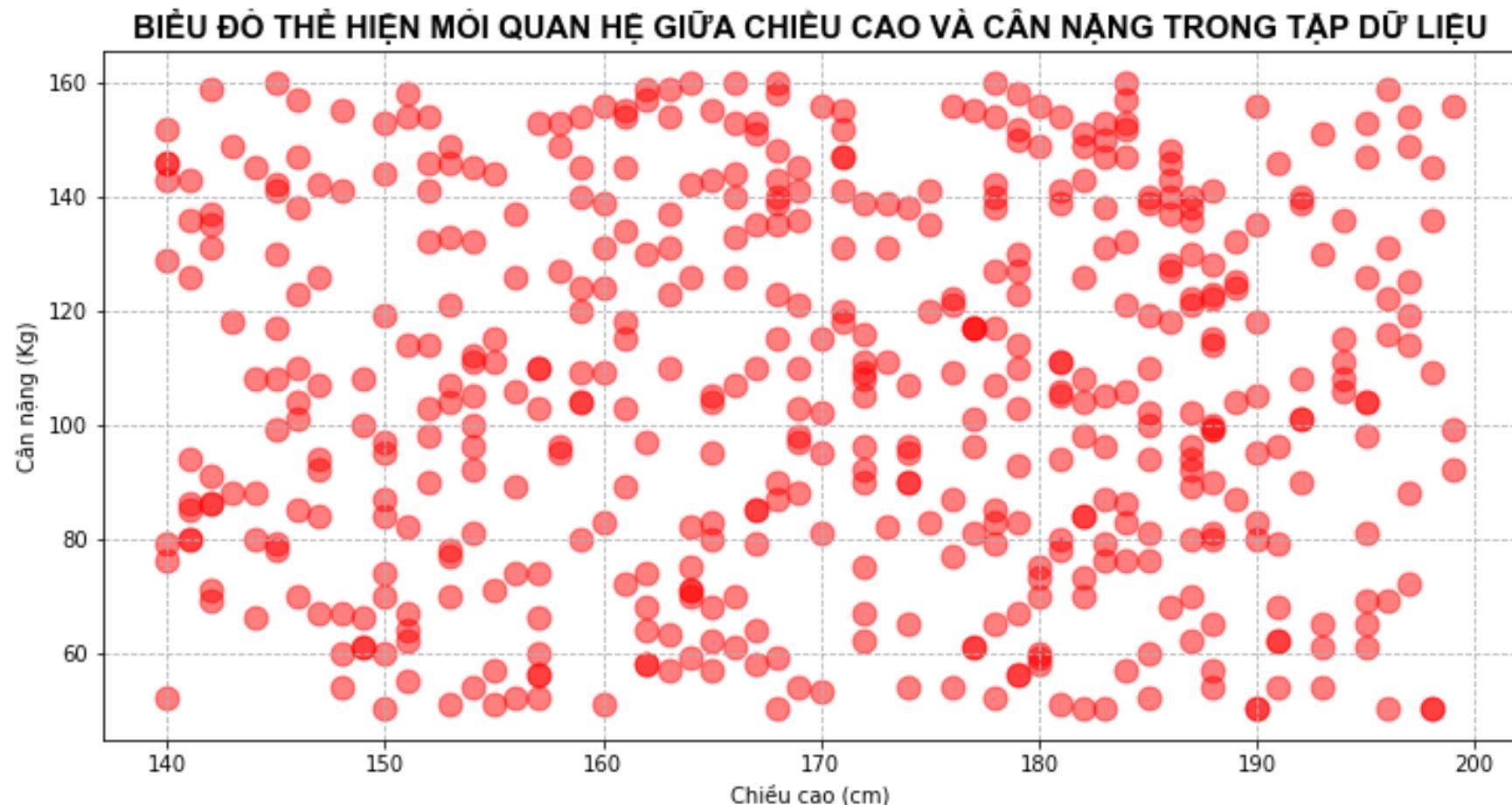
Thống kê chỉ số cơ thể theo từng giới tính và trực quan hóa → cho nhận xét?



Thực hành

Yêu cầu 5:

Vẽ biểu đồ thể hiện mối tương quan giữa Height và Weight của 500 người → cho nhận xét?





Q & A
Thank you!