Trực quan hóa dữ liệu với Matplotlib (2)

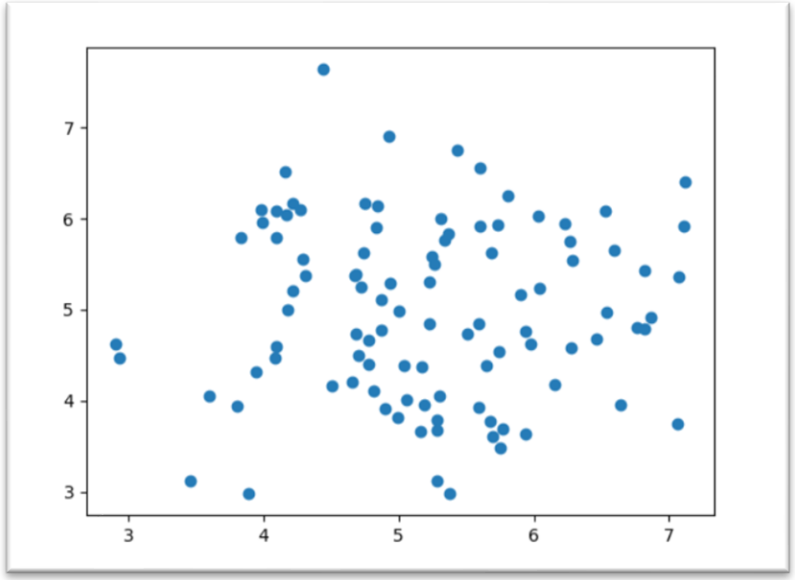
Giảng viên: **TS. Nguyễn Văn Quyết**

Nội dung

- Vẽ biểu đồ Scatter
- Vẽ biểu đồ Pie Chart
- Vẽ các Subplots
- Tùy biến Ticks, Labels, Legends
- Tùy biến Text và Annotation
- Trực quan hóa dữ liệu với Seaborn

5. Vẽ biểu đồ Scatter

- Biểu đồ Scatter (biểu đồ phân tán) là loại biểu đồ được dựng bởi các tọa độ tương quan giữa hai biến.
- Hai bộ dữ liệu được vẽ trên đồ thị
  - Trục tung Y: biến được dự đoán (còn gọi là biến phụ thuộc)
  - Trục hoành X: biến dùng để đưa ra dự đoán (còn gọi là biến độc lập



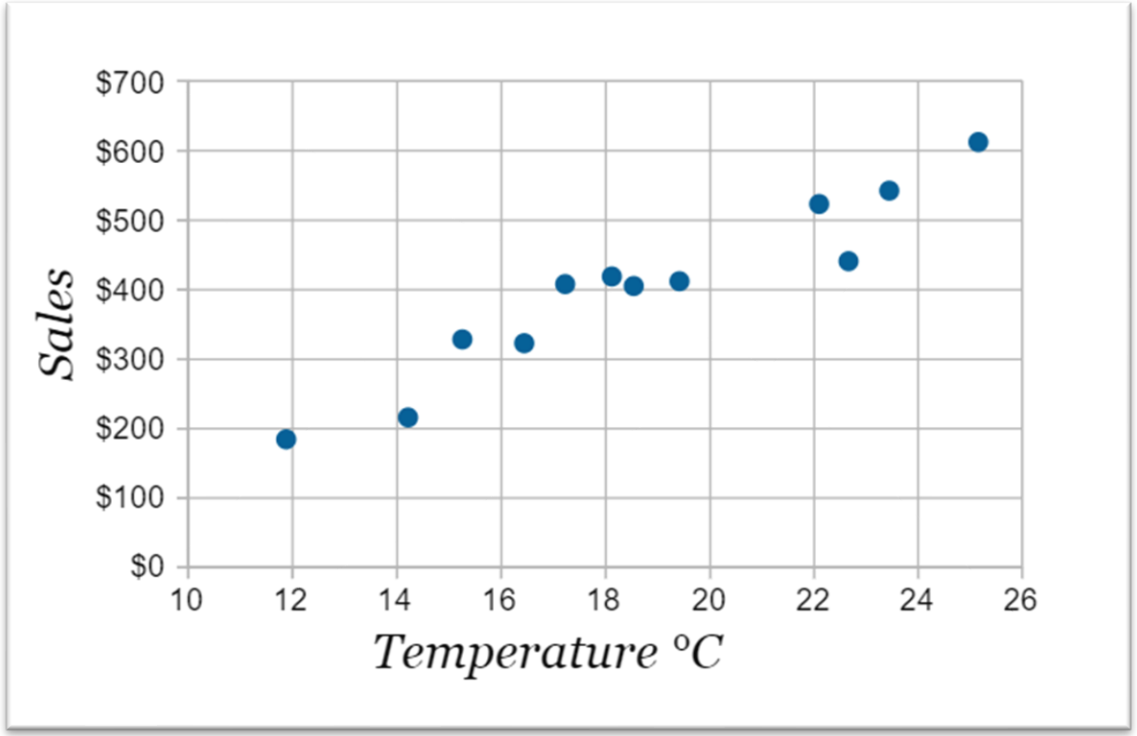
Khi nào sử dụng biểu đồ phân tán?

- Sử dụng biểu đồ phân tán khi bạn có cặp 2 dữ liệu (biến) và bạn muốn xác định xem hai biến có liên quan đến nhau hay không. Có liên quan thì liên quan nhiều hay ít thế nào.
  - Ví dụ: Một cửa hàng kem theo dõi doanh số bán hàng với nhiệt độ giữa trưa trong ngày và muốn tìm ra mối tương quan giữa chúng.

Biểu diễn dữ liệu bằng biểu đồ phân tán

Ice Cream Sales vs Temperature	
Temperature °C	**Ice Cream Sales**
14.2°	$215
16.4°	$325
11.9°	$185
15.2°	$332
18.5°	$406
22.1°	$522
19.4°	$412
25.1°	$614
23.4°	$544
18.1°	$421
22.6°	$445
17.2°	$408

- Biểu đồ Scatter tương ứng là:



Vẽ biểu đồ scatter

- Hàm **scatter** nhận vào các tham số chính sau:
  - x, y: lần lượt là các mảng dãy tọa độ tương ứng x, y của các điểm dữ liệu. x và y nếu là mảng phải có số lượng phần tử bằng nhau.
  - s: kích thước của các điểm đánh dấu.
  - c: màu của các điểm đánh dấu trên biểu đồ scatter.
  - marker: kiểu đánh dấu, quyết định hình dạng của các điểm đánh dấu. Mặc định *'o'*

Bài tập

- Bài toán 1: Sử dụng biểu đồ Scatter để biểu diễn sự tương quan (phụ thuộc) của doanh thu bán kem trong ngày với nhiệt độ buổi trưa trong ngày đó theo bảng sau:

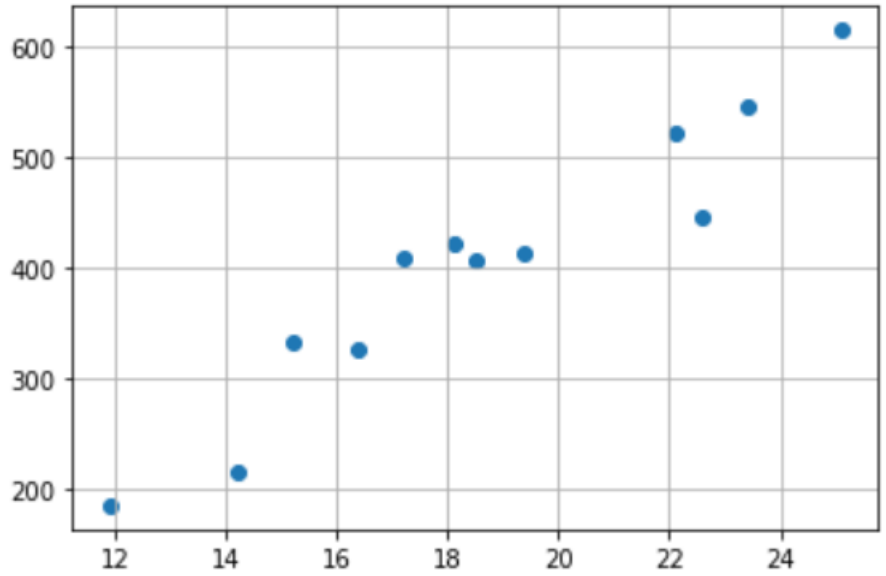
Ice Cream Sales vs Temperature	
Temperature °C	**Ice Cream Sales**
14.2°	$215
16.4°	$325
11.9°	$185
15.2°	$332
18.5°	$406
22.1°	$522
19.4°	$412
25.1°	$614
23.4°	$544
18.1°	$421
22.6°	$445
17.2°	$408

Bài tập

Ta chạy chương trình như sau:

```
x=np.array([14.2,16.4,11.9,15.2,
18.5,22.1,19.4,25.1,23.4,18.1,22
.6,17.2])

y=np.array([215,325,185,332,406,
522,412,614,544,421,445,408])

plt.scatter(x,y)

plt.grid()
```

- Biểu đồ sau khi chạy có dạng như sau:



Bài tập

- Bài toán 2: Sử dụng dữ liệu hoa Iris để so sánh sự tương quan giữa độ dài cánh hoa và đài hoa. Tải dữ liệu tại đây.

```
df=pd.read_csv("Iris.csv")
df
```

[2] ✓ 0.3s

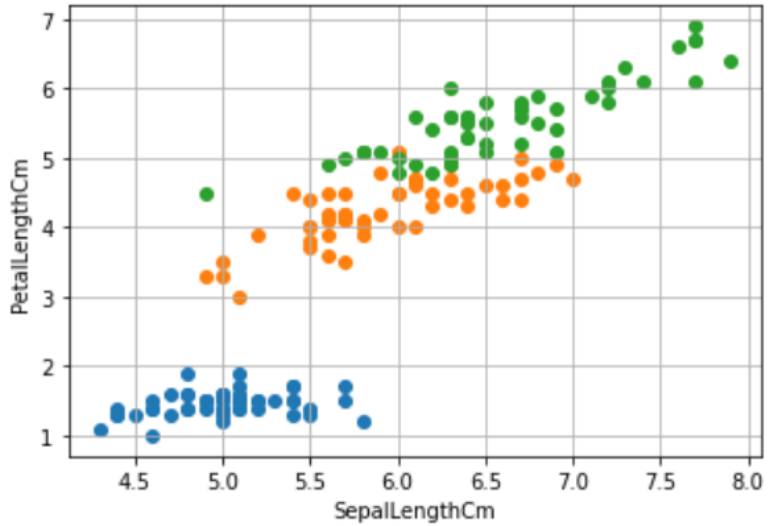
...

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...

Bài tập

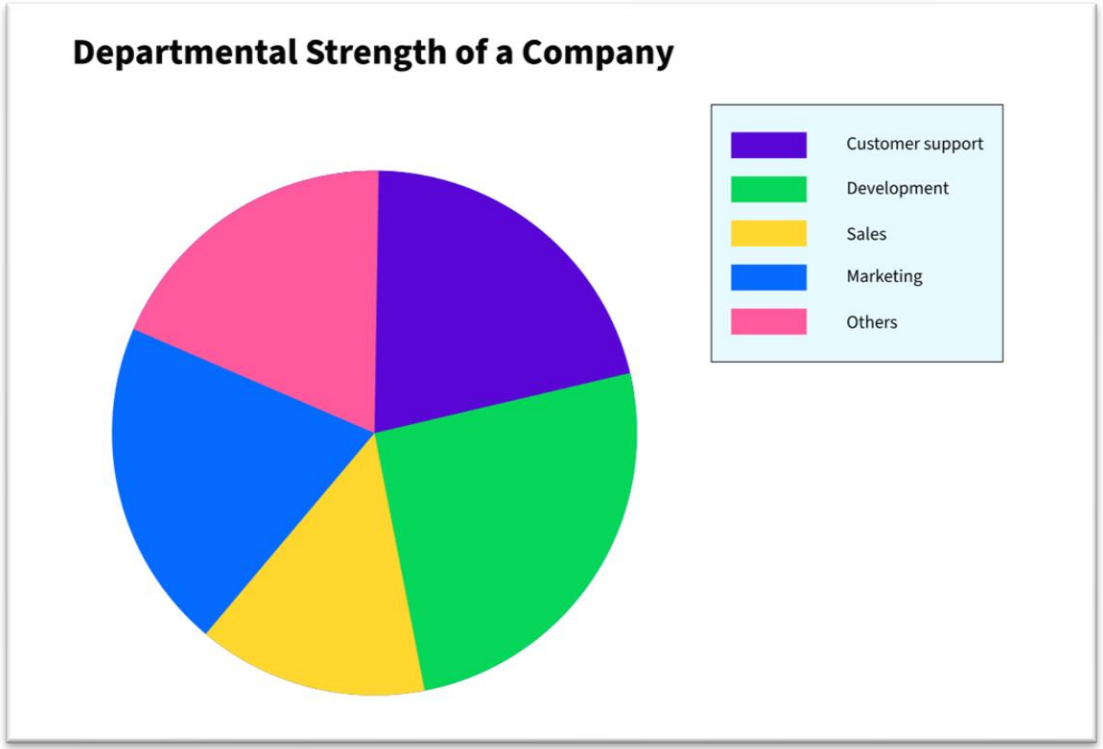
Ta thực hiện code như sau:

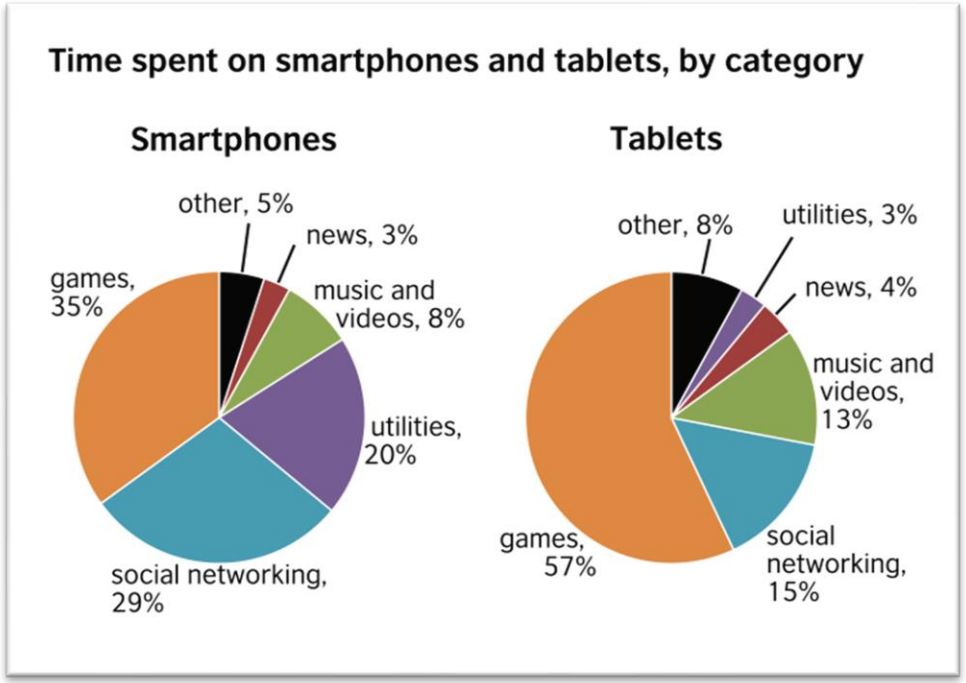
```
df=pd.read_csv("Iris.csv")
x1=df["SepalLengthCm"][df["Species"]=="Iris-setosa"]
y1=df["PetalLengthCm"][df["Species"]=="Iris-setosa"]
x2=df["SepalLengthCm"][df["Species"]=="Iris-versicolor"]
y2=df["PetalLengthCm"][df["Species"]=="Iris-versicolor"]
x3=df["SepalLengthCm"][df["Species"]=="Iris-virginica"]
y3=df["PetalLengthCm"][df["Species"]=="Iris-virginica"]
plt.scatter(x1,y1, c="blue")
plt.scatter(x2,y2, c="orange")
plt.scatter(x3,y3, c="green")
plt.xlabel("SepalLengthCm")
plt.ylabel("PetalLengthCm")
plt.grid()
```



6. Vẽ biểu đồ Pie

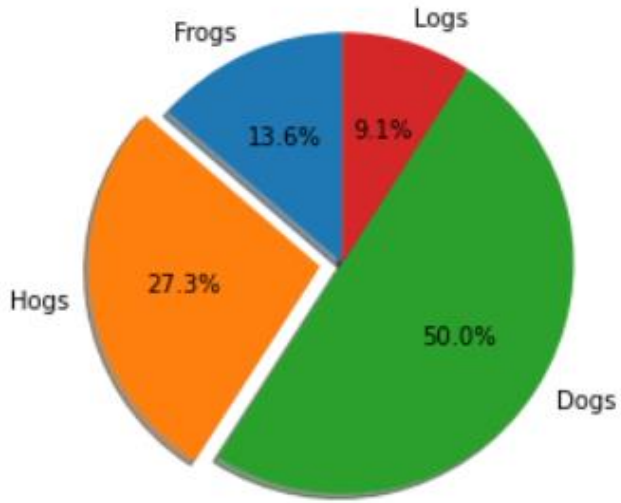
- Biểu đồ Pie là loại biểu đồ thống kê hình tròn, được chia ra để biểu diễn tỉ lệ phần trăm của các đối tượng được thống kê.





Hàm vẽ biểu đồ Pie

- Hàm pie nhận vào các tham số chính sau:
  - x: một dãy kích thước của các cung trong biểu đồ tròn.
  - explode: một dãy kích thước radius nhằm tạo khoảng cách giữa các cung với hình tròn.
  - labels: một dãy các xâu là các nhãn của mỗi cung.
  - colors: một dãy màu tùy chọn cho mỗi cung.
  - autopct: một xâu python định nghĩa format của các tỉ lệ trong các cung.



Bài tập

- Bài toán 1: Cho bảng thị phần của các công ty bán dẫn trên toàn thế giới. Hãy vẽ biểu đồ Pie biểu diễn tỉ lệ đó.

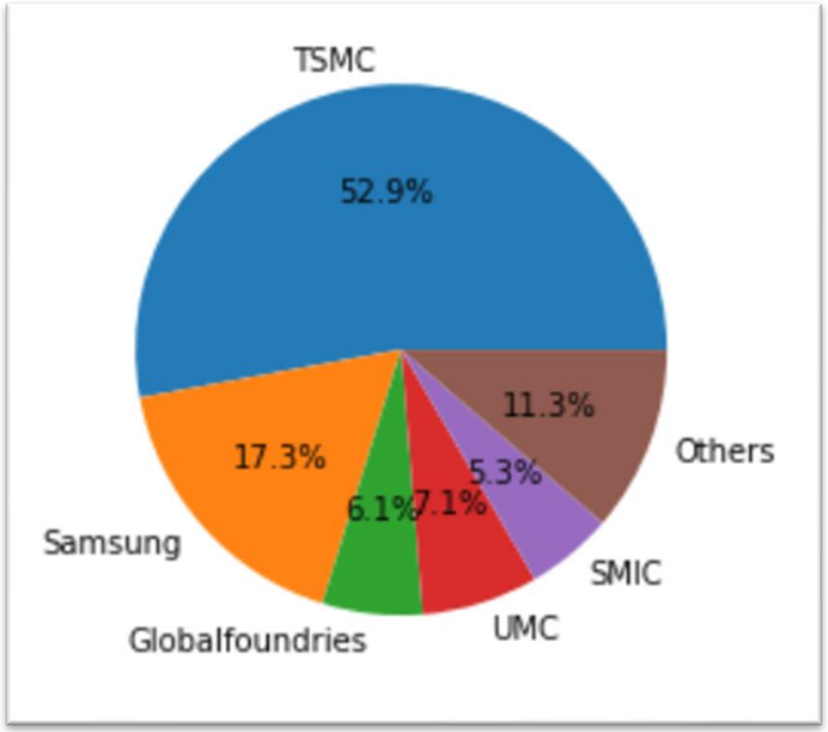
Công ty	Thị phần (%)
TSMC	52.9
Samsung	17.3
Globalfoundries	6.1
UMC	7.1
SMIC	5.3
Others	11.3

Bài tập

- Ta thực hiện code như sau:

```
x=np.array([52.9,17.3,6.1,7.1,5.3,11.3])
labels=["TSMC","Samsung","Globalfoundries","UMC
","SMIC","Others"]
plt.pie(x, labels=labels, autopct="%1.1f%%")
pass
```

- Ta có biểu đồ như sau:



Bài tập

- Ví dụ: Cho tệp dữ liệu bảng điểm: Vẽ biểu đồ pie biểu diễn tỉ lệ học sinh xuất sắc, giỏi, khá, trung bình, yếu. Tải tệp dữ liệu tại đây.

```
df=pd.read_csv("BangDiem.csv")
df
```

[29] ✓ 0.7s

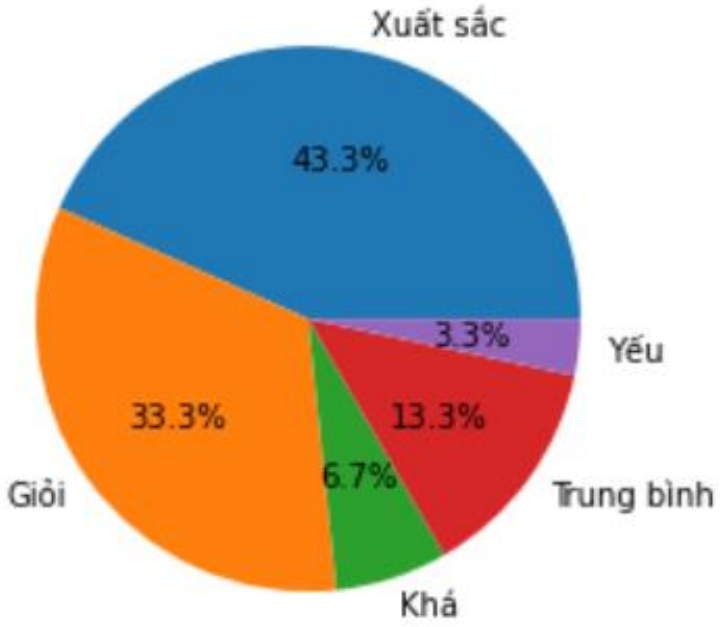
	STT	Mã SV	Họ và tên	Điểm
0	1	10117050	ĐỖ TUẤN ANH	9.0
1	2	10117056	BÙI ĐĂNG CƠ	9.0
2	3	10117252	NGUYỄN ĐÌNH DUY	8.0
3	4	10117029	NGUYỄN TIẾN ĐẠT	7.5
4	5	10117006	NGUYỄN HẢI ĐĂNG	8.5
5	6	10117009	NGUYỄN VĂN ĐỨC	9.0
6	7	10117005	NGUYỄN ĐỨC HIẾU	8.0

Bài tập

- Ta thực hiện như sau:

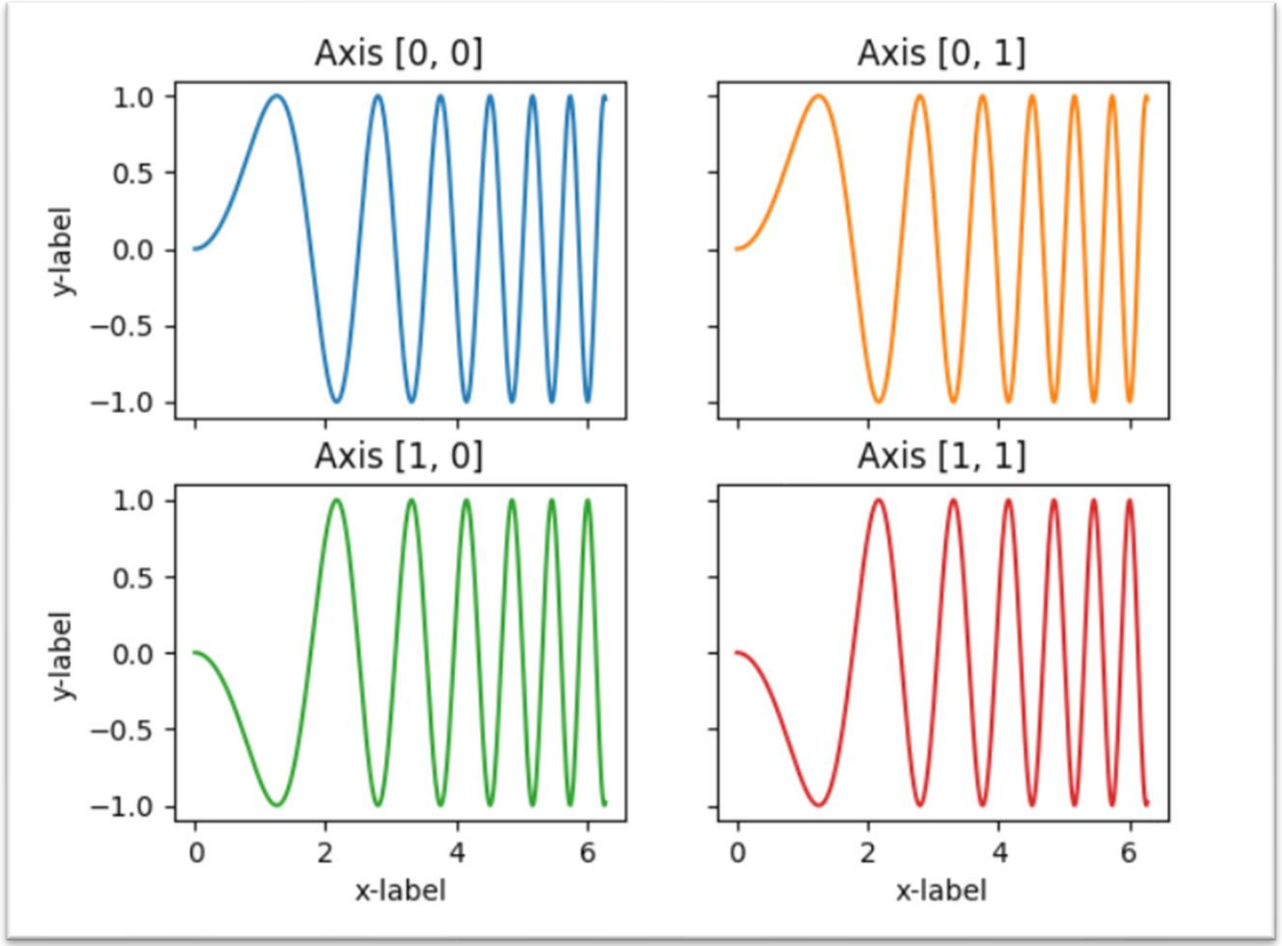
```
df=pd.read_csv("BangDiem.csv")
xuatsac=len(df["Điểm"][df["Điểm"]>=9.0])
gioi=len(df["Điểm"][(df["Điểm"]<9.0) &
(df["Điểm"]>=8.0)])
kha=len(df["Điểm"][(df["Điểm"]<8.0) &
(df["Điểm"]>=7.0)])
trungbinh=len(df["Điểm"][(df["Điểm"]<7.0)
& (df["Điểm"]>=6.0)])
yeu=len(df["Điểm"][df["Điểm"]<6.0])
quantities=[xuatsac,gioi,kha,trungbinh,ye
u]
labels=["Xuất sắc","Giỏi","Khá","Trung
bình","Yếu"]
plt.pie(quantities, labels=labels,
autopct="%1.1f%%")
pass
```

- Ta thu được hình vẽ như sau:



Khái niệm Subplot

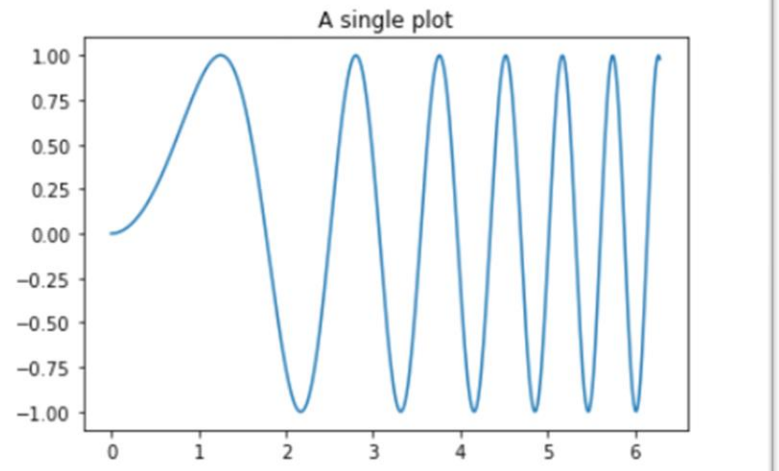
- Subplot là tập hợp các axis trong một axes.



Hàm tạo subplot

- Hàm pyplot.subplots tạo ra một figure và một grid các subplot cung cấp khả năng điều khiển các subplot một cách dễ dàng.
- Hàm subplots() nếu không truyền vào bất kì tham số nào sẽ trả về một figure và một Axes đơn. Đây là cách đơn giản nhất và được khuyến nghị khi tạo Figure và Axes.

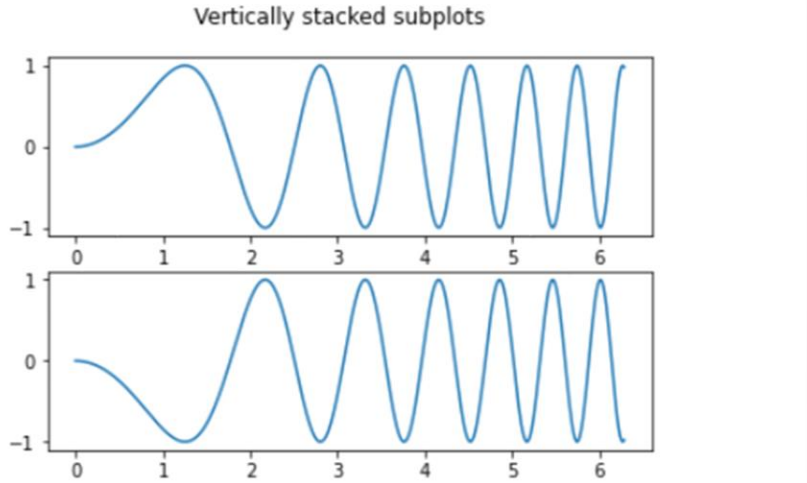
```
In [12]: fig, ax=plt.subplots()
         x = np.linspace(0, 2 * np.pi, 400)
         y = np.sin(x ** 2)
         ax.plot(x,y)
         ax.set_title('A single plot')
         pass
```



Chồng các subplots theo một chiều

- 2 tham số tự chọn đầu tiên của hàm pyplot.subplots định nghĩa số hàng và số cột của subplot grid.
- Khi chỉ truyền vào 1 tham số trong 2 tham số trên, tham số **ax** khi được trả về sẽ tạo ra 1 vector numpy một chiều chứa các subplot.

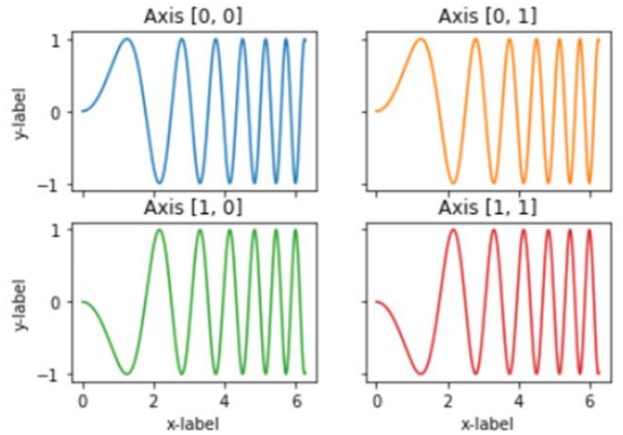
```
In [13]: fig, axs = plt.subplots(2)
         fig.suptitle('Vertically stacked subplots')
         axs[0].plot(x, y)
         axs[1].plot(x, -y)
         pass
```



Chồng các subplots theo 2 chiều

- Khi chồng các subplots theo 2 hướng khác nhau, **ax** được trả về là một vector numpy 2 chiều.

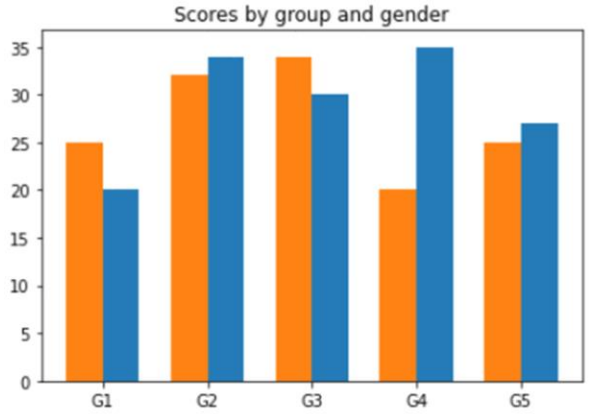
```
In [14]: fig, axs = plt.subplots(2, 2)
         axs[0, 0].plot(x, y)
         axs[0, 0].set_title('Axis [0, 0]')
         axs[0, 1].plot(x, y, 'tab:orange')
         axs[0, 1].set_title('Axis [0, 1]')
         axs[1, 0].plot(x, -y, 'tab:green')
         axs[1, 0].set_title('Axis [1, 0]')
         axs[1, 1].plot(x, -y, 'tab:red')
         axs[1, 1].set_title('Axis [1, 1]')

         for ax in axs.flat:
             ax.set(xlabel='x-label', ylabel='y-label')

         # Hide x labels and tick labels for top plots and y ticks for right plots.
         for ax in axs.flat:
             ax.label_outer()
```



7. Tùy biến Ticks, Labels, Legends

- Ticks là các giá trị được dùng để biểu diễn các điểm trên các trục tọa độ, các giá trị đó có thể là số hoặc là xâu.
  - Trong ví dụ về biểu đồ bar, ta đã tùy biến xticks bằng các giá trị mong muốn.

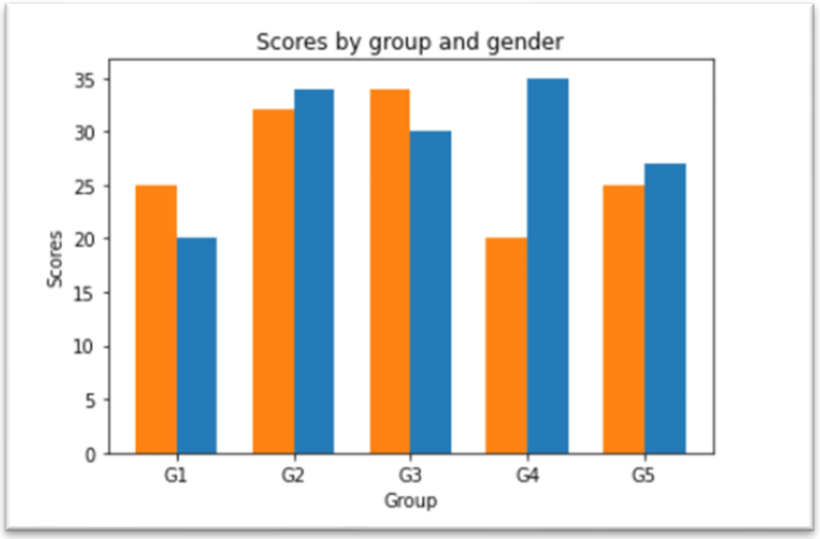
```
In [25]: labels=['G1','G2','G3','G4','G5']
         men=[20,34,30,35,27]
         women=[25,32,34,20,25]
         width=0.35 #width of a bar
         X=np.array([1,2,3,4,5])
         plt.bar(X+width/2,men,width,label="Men")
         plt.bar(X-width/2,women,width, label="Women")
         plt.title("Scores by group and gender")
         plt.xticks(X,labels)
         pass
```



Labels

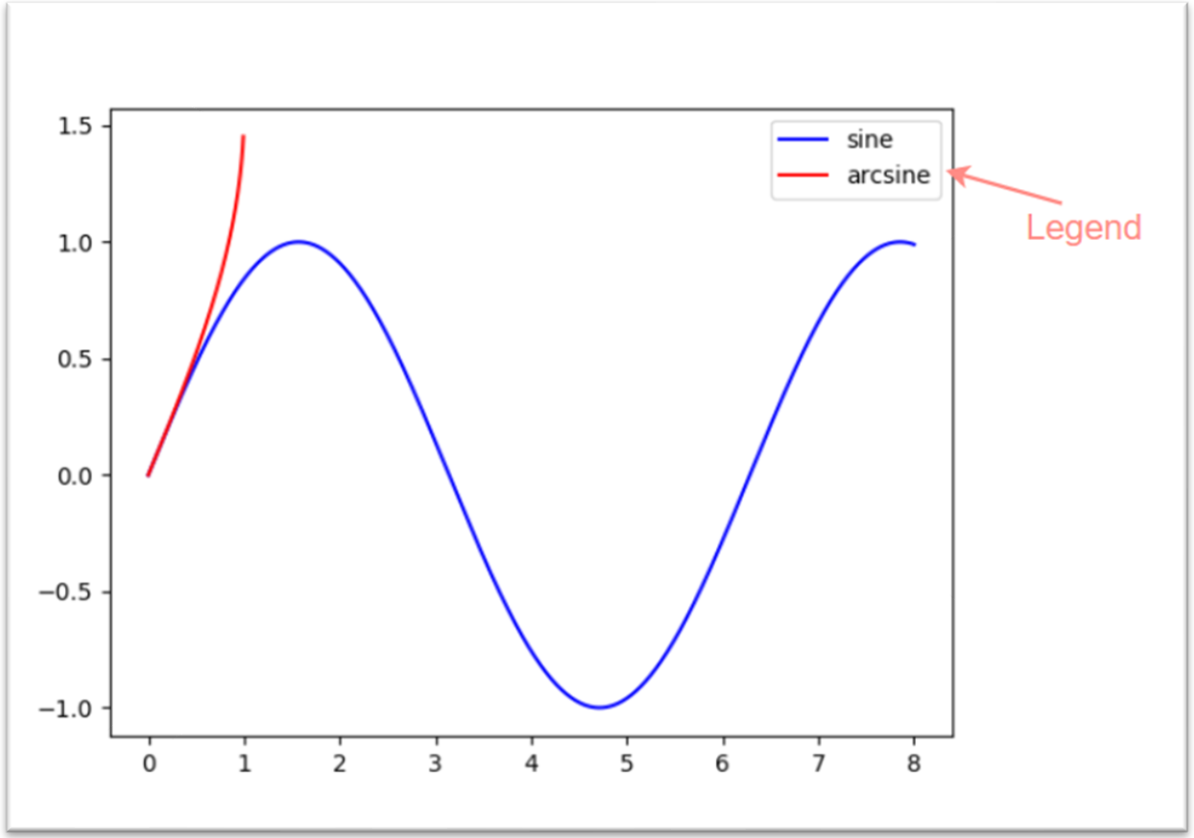
- Labels là các nhãn nằm trên các trục tọa độ, labels được dùng để giải thích ý nghĩa các trục.
  - Ví dụ: Tại ví dụ về biểu đồ bar, ta có thể thêm các label vào trục x và trục y tương ứng để giải thích.

```
plt.xlabel("Group")
plt.ylabel("Scores")
```



Legends

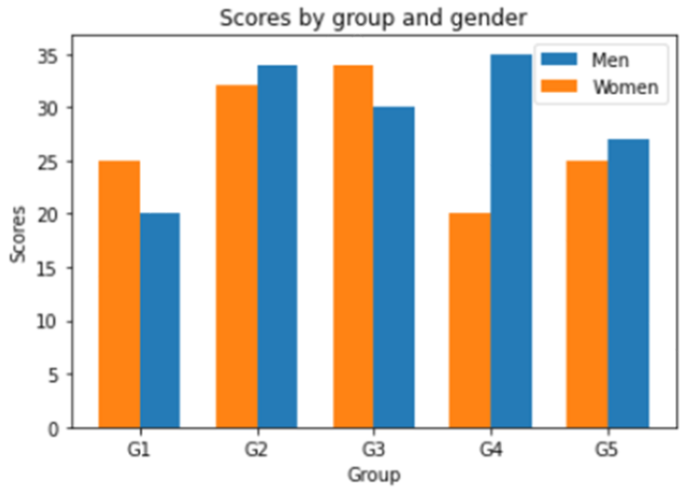
- Legends là bảng các chú thích các kí hiệu trong biểu đồ.



Ví dụ: Thêm legends vào biểu đồ bar

- Tại ví dụ về biểu đồ bar, ta sẽ thêm bảng chú thích vào trong biểu đồ.

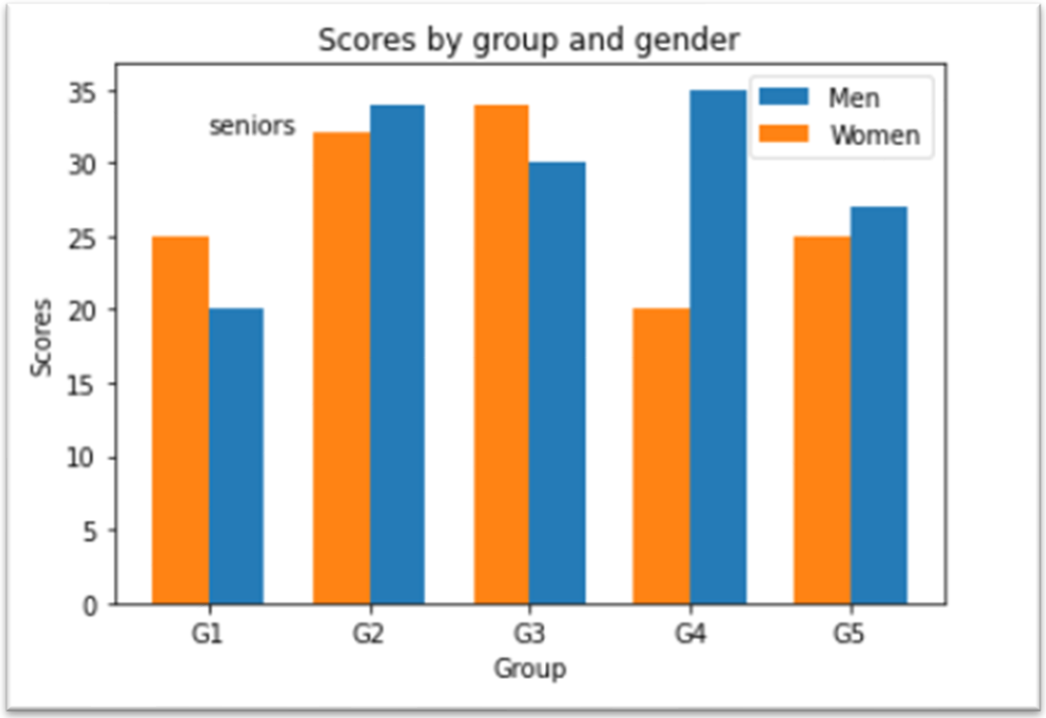
```
In [19]: labels=['G1','G2','G3','G4','G5']
men=[20,34,30,35,27]
women=[25,32,34,20,25]
width=0.35 #width of a bar
X=np.array([1,2,3,4,5])
plt.bar(X+width/2,men,width,label="Men")
plt.bar(X-width/2,women,width, label="Women")
plt.title("Scores by group and gender")
plt.xticks(X,labels)
plt.xlabel("Group")
plt.ylabel("Scores")
plt.legend()
pass
```



8. Text và Annotations

- Ta có thể thêm text vào tại một vị trí biểu đồ bằng hàm text nhằm mục đích chú thích cho biểu đồ.

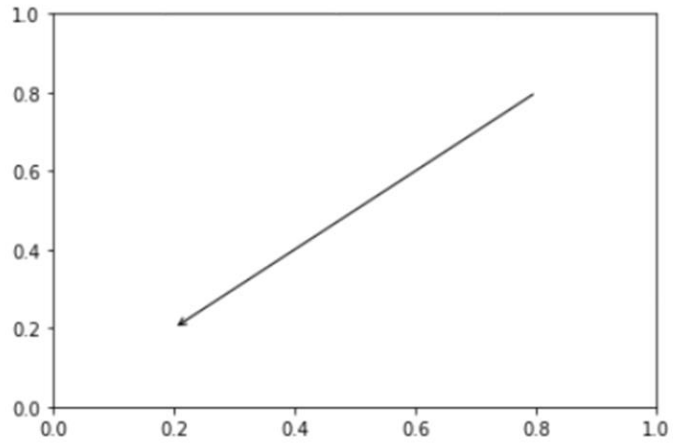
```
plt.legend()
plt.text(x=1,y=32,s="seniors")
pass
```

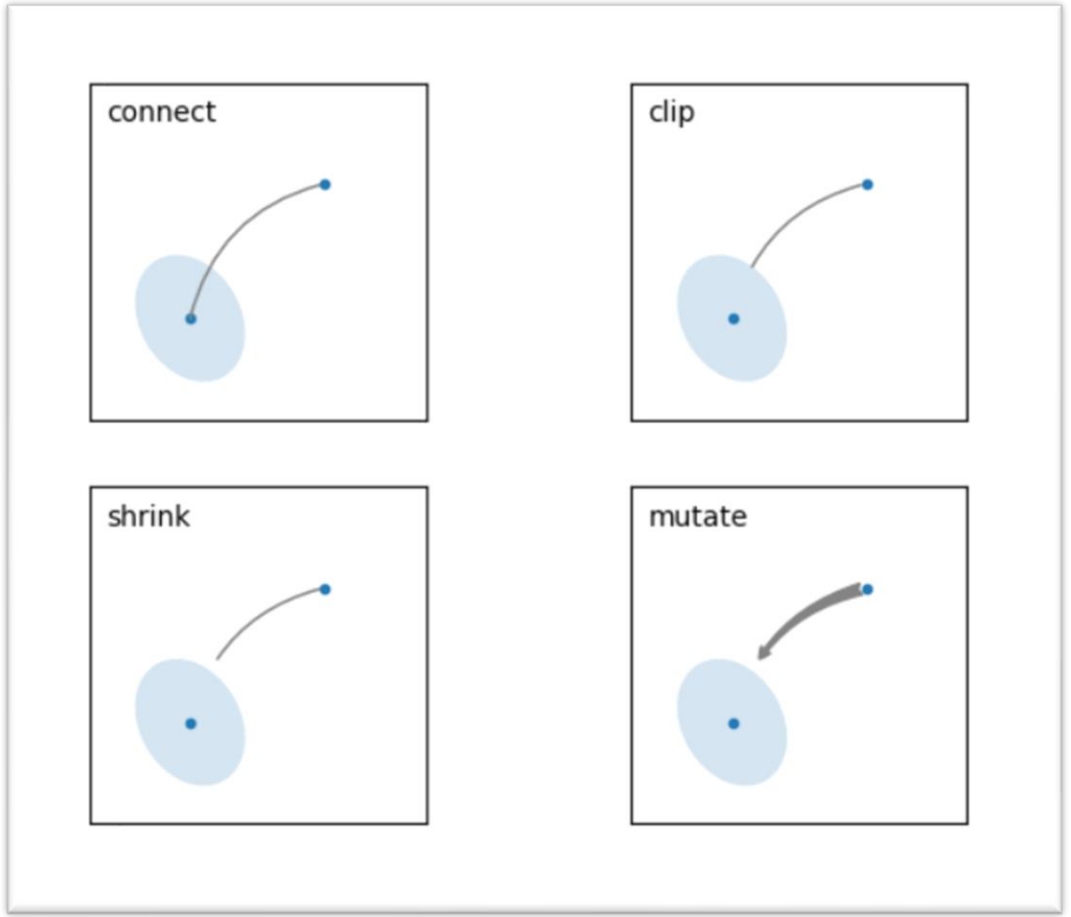


Annotate

- Hàm annnote giúp cho việc chú tích trở nên dễ dàng hơn.

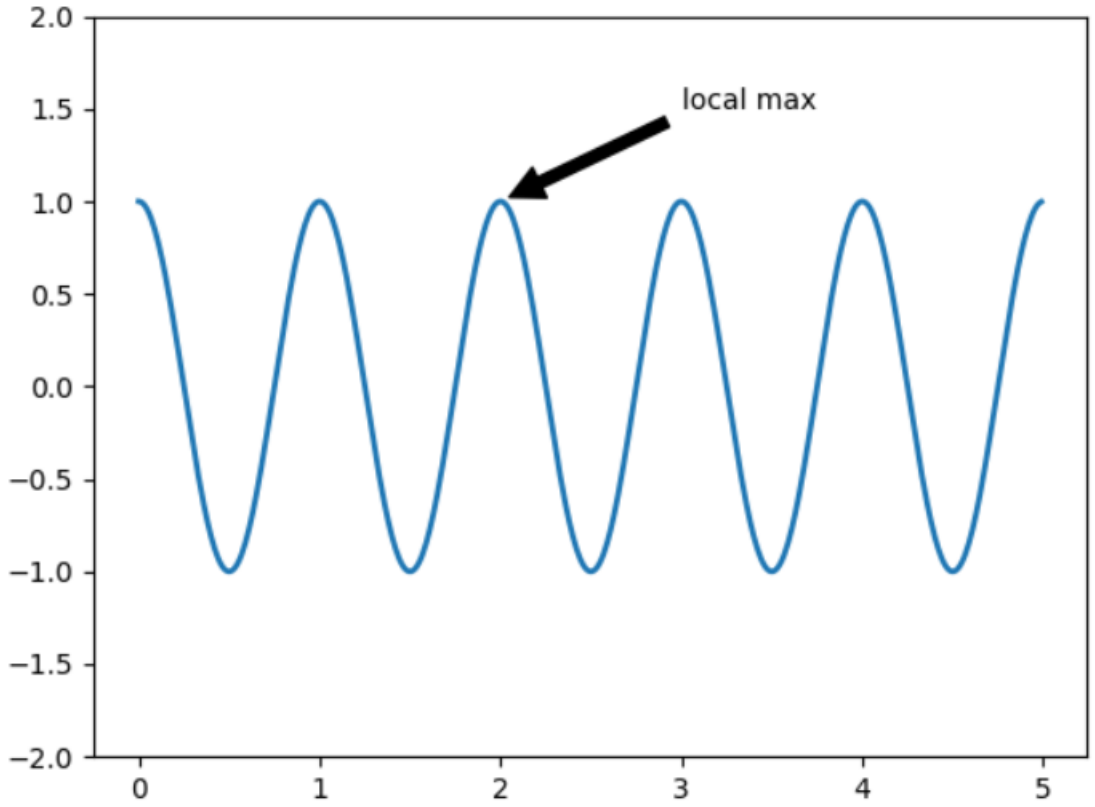
```
In [23]: fig, ax=plt.subplots()
         ax.annotate("",
                     xy=(0.2, 0.2), xycoords='data',
                     xytext=(0.8, 0.8), textcoords='data',
                     arrowprops=dict(arrowstyle="->",
                                     connectionstyle="arc3"),
                     )
         pass
```





Bài tập

- Ví dụ: Vẽ biểu đồ y=cos(x) và chú thích chỉ ra điểm cực đại địa phương và cực tiểu địa phương.

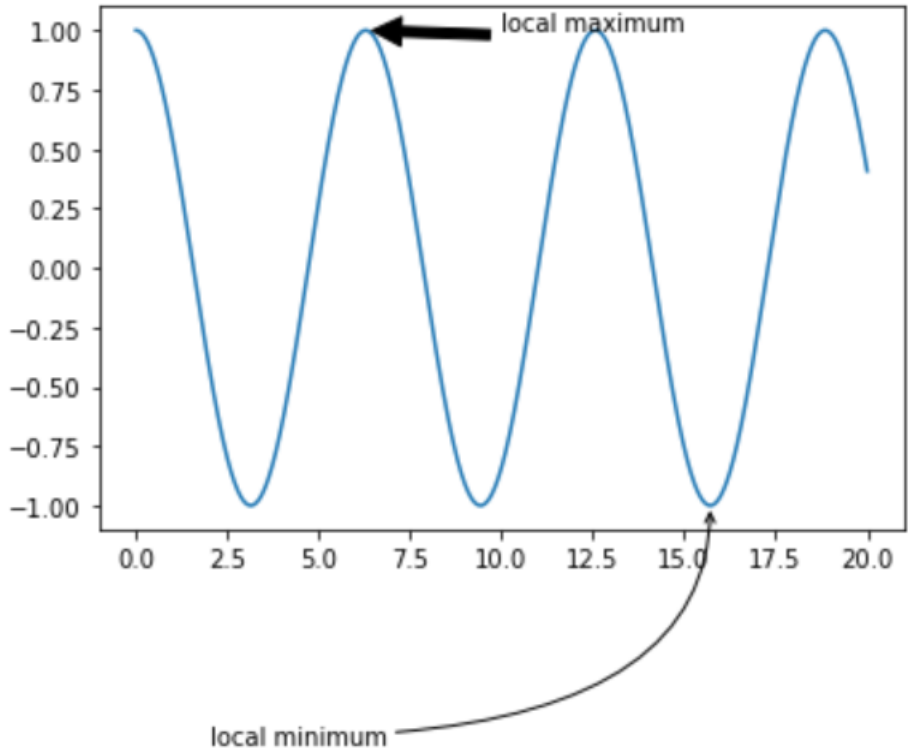


Bài tập

- Ta thực hiện như sau:

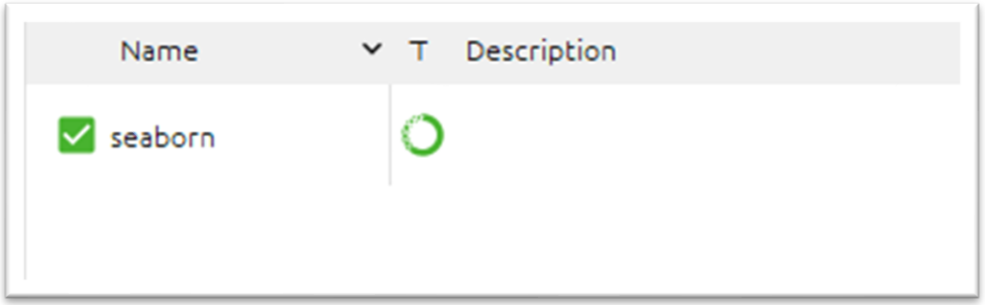
```
fig, ax = plt.subplots()
x = np.linspace(0, 20, 1000)
ax.plot(x, np.cos(x))
ax.annotate('local maximum', xy=(6.28, 1),
xytext=(10, 1),
arrowprops=dict(facecolor='black', shrink=0.05))
ax.annotate('local minimum', xy=(5 * np.pi, -1),
xytext=(2, -2),
arrowprops=dict(arrowstyle="->",
connectionstyle="angle3,angleA=0,angleB=-90"));
```

- Ta thu được đồ thị như sau:



9. Trực quan hóa dữ liệu với Seaborn

- Seaborn là thư viện được xây dựng trên matplotlib và hoạt động tốt với pandas.
- Ta cần cài đặt package seaborn để có thể sử dụng seaborn trên Anaconda.



Tại sao cần đến Seaborn?

- Matplotlib dù được coi là hữu dụng trong việc trực quan hóa dữ liệu nhưng vẫn tồn tại những điểm yếu khó có thể khắc phục của nó.
  - Matplotlib được ra đời trước pandas cả thập kỷ, vì vậy rõ ràng matplotlib không được thiết kế để làm việc với pandas.
  - Do được ra đời sớm, đồ họa của matplotlib chưa được đẹp mắt.
  - Việc tùy chỉnh đồ thị trên matplotlib còn chưa được thuận tiện.
- Đó là những lý do cho sự ra đời của seaborn, một công cụ có được sinh ra để giải quyết tất cả những vấn đề đó.

Cài đặt và khai báo seaborn

- Để sử dụng seaborn trước tiên ta phải cài đặt seaborn trên Anaconda.
  - conda install seaborn
- Cũng giống như numpy, pandas hay matplotlib, ta cũng có quy tắc khai báo seaborn như sau:
  - `import seaborn as sb`

Bài tập

- Bài toán 1: Vẽ biểu đồ bar bằng seaborn có các giá trị như sau. Cho nhận xét về màu sắc của Seaborn so với Matplotlib.

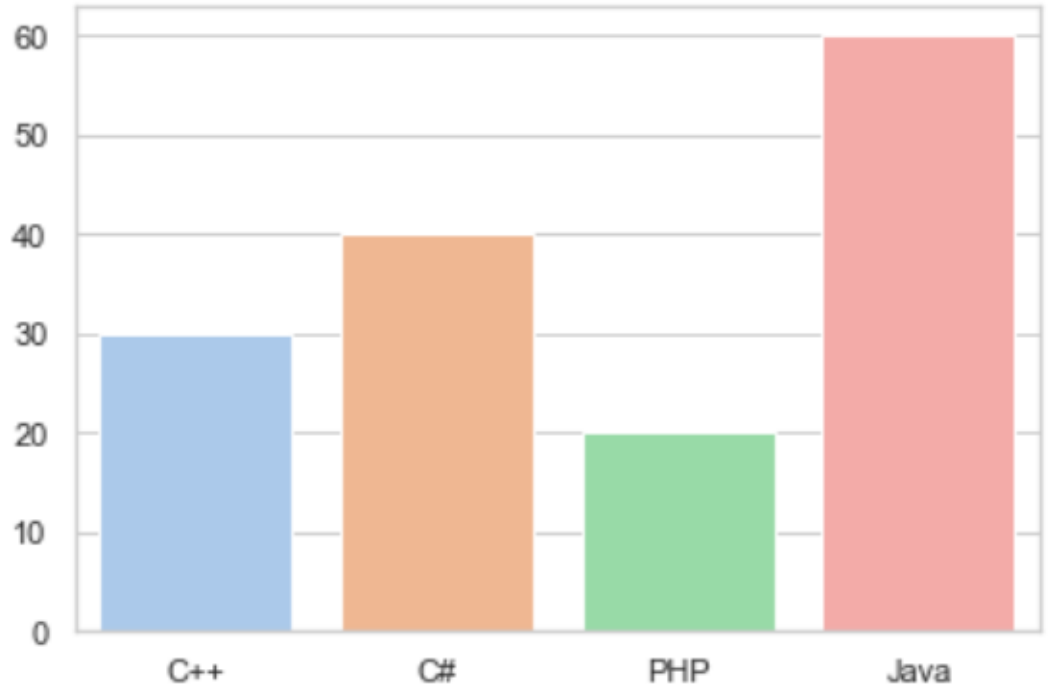
Language	Percent
C++	30
C#	40
PHP	20
Java	60

Bài tập

- Ta thực hiện như sau:

```
percent=[30,40,20,60]
labels=["C++","C#","PHP","Java"]
sb.set_theme(style="whitegrid",
palette="pastel")
sb.barplot(x=labels,y=percent)
pass
```

- Ta nhận được đồ thị như sau:



Tương tác tốt với Pandas

- Seaborn được thiết kế để tương tác với pandas. Các hàm của seaborn có thể nhận trực tiếp pandas là dữ liệu.
- Các hàm vẽ hình của Seaborn đều nhận 3 biến sau:
  - data: nhận vào một DataFrame là dữ liệu nguồn để vẽ đồ thị
  - x: vector hoặc key trong data biểu diễn tọa độ của trục hoành.
  - y: vector hoặc key trong data biểu diễn tọa độ của trục tung.

Bài tập

- Bài toán 2: Vẫn với dữ liệu của file Iris.csv, hãy vẽ biểu đồ Scatter biểu diễn sự tương quan giữa độ dài cánh hoa và đài hoa bằng Seaborn.

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa

Bài tập

- Ta thực hiện code như sau:

```
df=pd.read_csv("Iris.csv")
sb.scatterplot(data=df,
x="SepalLengthCm",
y="PetalLengthCm",
hue="Species")
pass
```

- Ta thu được đồ thị: