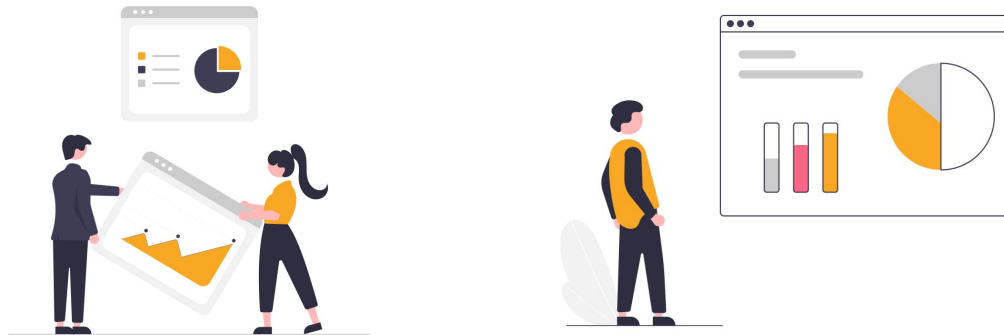




matplotlib

Python & Data Visualization

Data Visualization คืออะไร



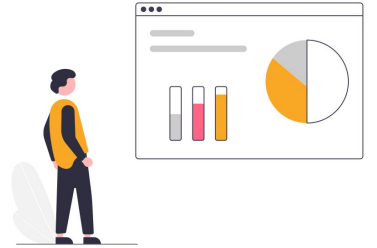
การนำข้อมูลมาแสดงผลในรูปแบบภาพกราฟิก กราฟ แผนภูมิ หรืออื่นๆ
เพื่อใช้อธิบายความสัมพันธ์ของข้อมูลได้ง่ายมากยิ่งขึ้นโดยข้อมูลที่สนใจมัก
อยู่ในรูปแบบอักขระ ข้อความ ตัวเลขหรือรูปแบบตาราง ซึ่งจำนวนข้อมูลมี
อย่างมากมายมหาศาล

Data Visualization คืออะไร

การใช้ Data Visualization จะช่วยให้การวิเคราะห์ข้อมูลทำได้ง่ายมากยิ่งขึ้นและเห็นภาพรวมของข้อมูลทั้งหมดดังคำกล่าวที่ว่า

“ ภาพหนึ่งภาพแทนคำพูดนับพันคำ ”

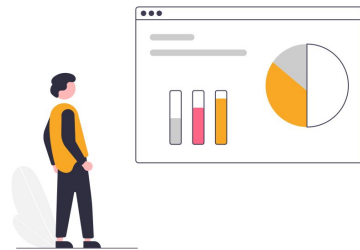
(A picture is worth a thousand words)



พื้นฐานที่ต้องเรียนมาก่อน

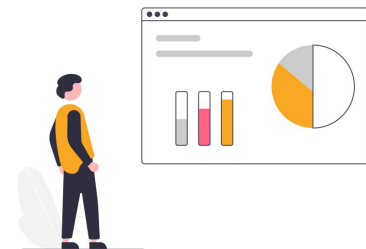
เข้าเรียนที่ช่อง KongRuksiam Official ได้เลย

- Python เบื้องต้น
- Numpy
- Pandas



ประเภทของการพล็อตกราฟ

- Line Plot / Line Graph
- Bar Graph
- Histogram
- Scatter Plot
- Pie Chart



คำสั่งสร้างกราฟ

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

```
#ข้อมูลแกน x , แกน y โดยต้องเป็นข้อมูลแบบชุด เช่น List , Numpy  
Array , DataFrame และ DataFrame
```

```
x = [1,2,3,4]
```

```
y = [1,2.5,3.5,3]
```

```
plt.bar(x,y) #กราฟแท่ง
```

```
plt.show() #แสดงกราฟ
```



กราฟเส้น

```
product1=[1.2,2.8,1.3,4.3,3.5];//ยอดขายสินค้า 1
```

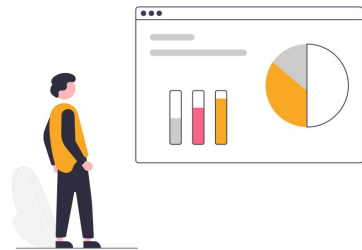
```
product2=[2,1.4,3.3,3.1,1.2]; // ยอดขายสินค้า 2
```

```
month=[1,2,3,4,5]; //เดือนที่
```

```
plt.plot(month,product1);
```

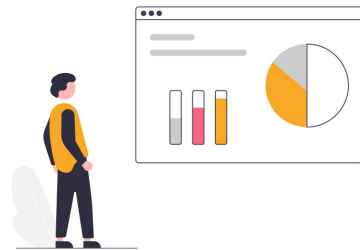
```
plt.plot(month,product2);
```

```
plt.show();
```



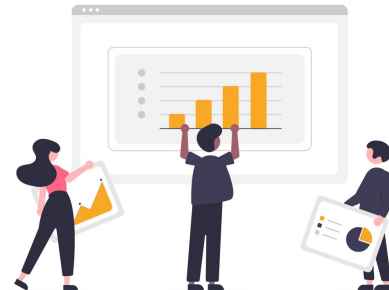
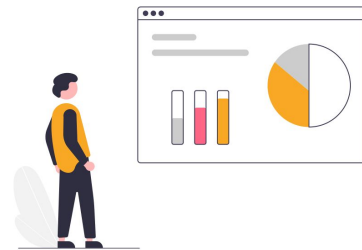
XLabel และ YLabel

```
import matplotlib.pyplot as plt  
plt.plot([1,2,3,4],[5,6,7,8])  
plt.xlabel("X Label")  
plt.ylabel("Y Label")  
plt.show()
```



บันทึกกราฟเป็นไฟล์

```
import matplotlib.pyplot as plt  
plt.plot([1,2,3,4],[5,6,7,8])  
plt.xlabel("X Label")  
plt.ylabel("Y Label")  
plt.savefig("matplot.png")  
plt.savefig("matplot.png",transparent=True)  
plt.show()
```



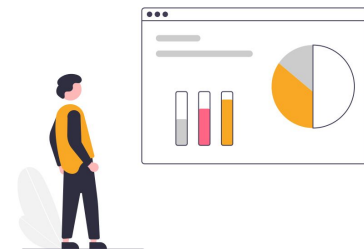
การกำหนดสี (Color)

Color Support

'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow

Color Support

'k'	black
'w'	white



กำหนดจุด Marker

Marker Support

‘.’	point
‘o’	cycle
‘x’	x marker
‘+’	plus
‘*’	star
‘s’	square

Marker Support

‘ ’	vline
‘_’	hline
‘v’	down
‘^’	up
‘>’	right
‘<’	left



รูปแบบเส้น

Line Support

‘_’

solid

‘_ _’

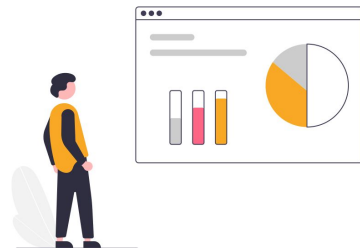
dashed

‘_.’

dash-dot

‘.’

dotted

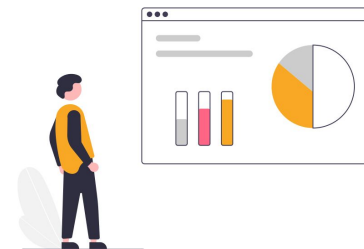


กราฟเส้น (Format String)

- [marker][line][color]
- [color][marker][line]

การใช้งาน

```
plt.plot([10,20,30],[1,2,3,4],'r+-')
```



กำหนดคุณสมบัติของ Label

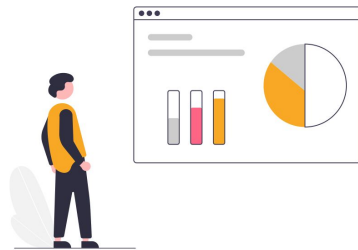
//คุณสมบัติผ่าน dict

```
data={"size":20,"color":"black"}
```

```
plt.xlabel("ข้อความ",data)
```

//คุณสมบัติผ่าน properties

```
plt.xlabel("ข้อความ",size=20,color="black",  
backgroundcolor="red")
```



Title (กำหนดหัวข้อ)

```
plt.xlabel("x label")
```

```
plt.ylabel("y label")
```

```
plt.title("สรุปผลข้อมูล")
```

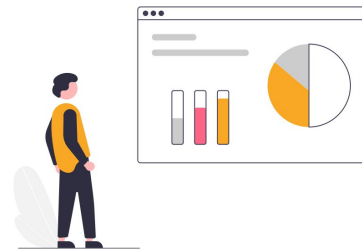
```
plt.title("สรุปผลข้อมูล",{"size":20,"color":"blue"})//dict
```

```
plt.title("สรุปผลข้อมูล",{"loc":"left","color":"blue"})//dict
```

```
plt.title("สรุปผลข้อมูล",loc="center")//location (default)
```

```
plt.title("สรุปผลข้อมูล",loc="left")
```

```
plt.title("สรุปผลข้อมูล",loc="right")
```



Legend Function

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

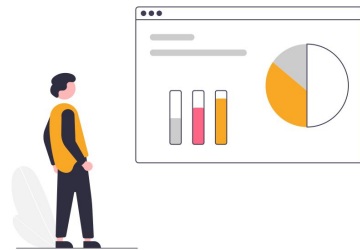
```
import numpy as np
```

```
t=np.array([1,2,3,4])
```

```
plt.plot(t**2,t,color="red")
```

```
plt.plot(t**3,t,color="green")
```

```
plt.legend(["Product1","Product2"])
```



กำหนดตำแหน่ง Legend

Format String Code

'best' (default)	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6

Format String Code

'center right'	7
'lower center'	8
'upper center'	9
'center'	10



กำหนดขนาดข้อความใน Legend

```
plt.legend(fontsize="xx-small")
```

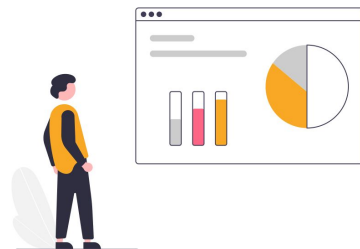
```
plt.legend(fontsize=20)
```

```
plt.legend(fontsize=40.5)
```

ค่าคงที่ **fontsize**

xx-small, x-small, small,

medium, large, x-large, xx-large



Face & Edge Color Legend

facecolor สีของแผ่น legend

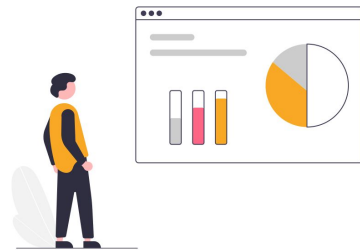
```
plt.legend(facecolor="red")
```

```
plt.legend(facecolor="yellow")
```

edgcolor สีของเส้น legend

```
plt.legend(edgecolor="red")
```

```
plt.legend(facecolor="yellow" , edgecolor="red")
```



Title Legend

title เป็นการกำหนดข้อความมีค่าเริ่มต้น คือ None

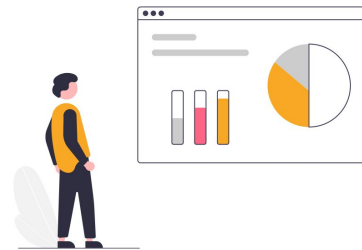
```
plt.legend(title="ข้อความ")
```

```
plt.legend(title=None)
```

```
plt.legend(title="ข้อความ",title_fontsize="small")
```

```
plt.legend(borderpad=3) // ขนาด default = 0.4
```

```
plt.legend(labelspacing=3) // ระยะ default = 0.5
```



Text Function

การวาดข้อความแบบระบุพิกัด

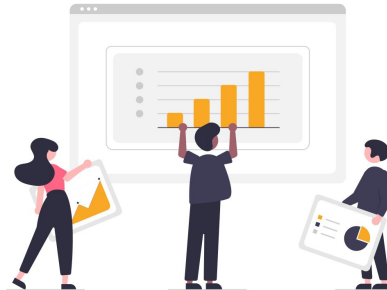
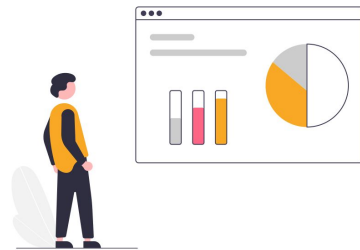
```
plt.text(x,y,str,args***)
```

ตัวอย่าง

```
plt.plot([10,20,30,40],marker="o")
```

```
plt.text(0,10,"Notes")
```

```
plt.text(0,10,"Notes",size=10,color="green")
```



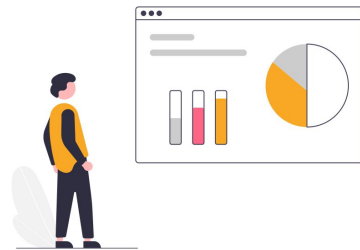
กราฟวงกลม

สำหรับแสดงข้อมูลที่มีสัดส่วนรวมกันได้ 100%
หรือเน้นส่วนที่สนใจเป็นพิเศษ

```
data=[15,30,45,60,40];
```

```
lang=['PHP','Flutter','Nodejs','C#','Java'];
```

```
plt.pie(data,labels=lang)
```



กราฟวงกลม - Color

```
data=[15,30,45,60,40];
```

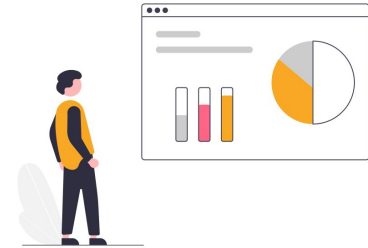
```
lang=['PHP','Flutter','Nodejs','C#','Java'];
```

```
colors=['green','orange','yellow','red','cyan'];
```

```
plt.pie(data,labels=lang,colors=colors
```

```
autopct='%1f%%'
```

```
);
```



กราฟวงกลม - Explode

```
colors=['green','orange','yellow','red','cyan'];
```

```
expl=[0,0,0,0.1];//ยิ่งค่าเยอะยิ่งห่างเยอะ
```

```
plt.pie(data,labels=lang,colors=colors
```

```
autopct='%1f%%',explode=explode,
```

```
shadow=True,startangle=90
```

```
);
```



กราฟการกระจาย (Scatter)

สำหรับการกระจายของข้อมูลหรือแสดงความสัมพันธ์ระหว่าง 2 ค่า (x,y)

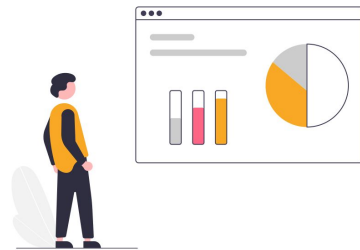
```
x=[1,2,3,4,5];
```

```
y=[1,2,5.6,4.5,7];
```

```
sizes=[10,20,12,15,10];
```

```
colors=["green","pink","red","blue","green"];
```

```
plt.scatter(x,y,s=sizes,c=colors);
```



ฮิสโตแกรม Histogram

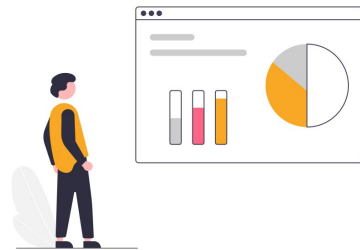
สำหรับแสดงความถี่ของข้อมูล เช่น แสดงความถี่ของ
อายุลูกค้าที่ใช้บริการระบบงานของเรา
สำหรับข้อมูลไม่ต่อเนื่อง

```
age =
```

```
[18,17,20,18,20,15,19,20,17,16,15,14,16,17,20]
```

```
plt.hist(age);
```

```
plt.show()
```



Histogram

สำหรับข้อมูลต่อเนื่อง

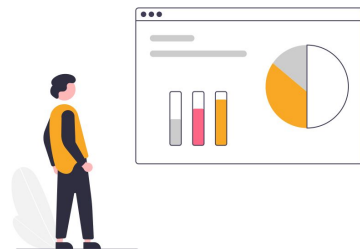
สุ่มข้อมูล 5000 ชุดมีค่าเฉลี่ย 10 และส่วนเบี่ยงเบน 2.5

```
data=np.random.normal(10,2.5,5000);
```

```
plt.hist(data);//ไม่กำหนดช่วงข้อมูล (default=10)
```

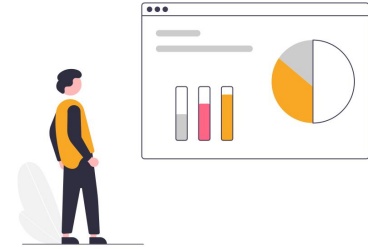
```
plt.hist(data,bins=30);//กำหนดช่วงเป็น 30 กลุ่ม
```

```
plt.show()
```



Bar Chart (กราฟแท่ง)

```
course=["Python","PHP","Java"];  
score=[80,50,99];  
c=["red","green","yellow"]; // สีของแท่ง  
plt.bar(course,score,colors=c);  
plt.xlabel("Course");  
plt.ylabel("Score");  
plt.title("Score 2021");  
plt.show();
```



กำหนดคุณสมบัติ Bar Chart

ความกว้างของ Bar

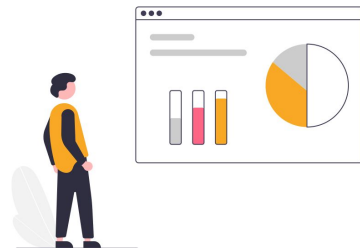
```
plt.bar(course,score,width=0.3); // default = 0.8
```

ความหนาของขอบ

```
plt.bar(course,score,linewidth=20);
```

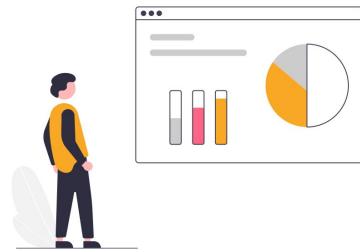
สีของขอบ

```
plt.bar(course,score,edgecolor="red");
```



Bar Chart (กราฟแท่งแนวนอน)

```
course=["Python","PHP","Java"];  
score=[80,50,99];  
c=["red","green","yellow"]; // สีของแท่ง  
plt.barh(course,score,colors=c);  
plt.xlabel("คอร์สเรียน");  
plt.ylabel("คะแนนสอบ");  
plt.title("ผลสอบประจำปี 2021");  
plt.show();
```



Stackbar

```
course=["Python","PHP","Java"];
```

```
boys=[30,40,20];
```

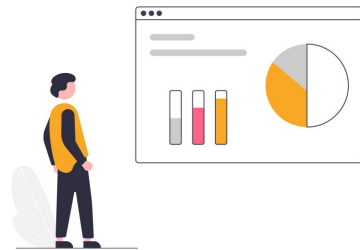
```
girls=[40,50,90];
```

```
plt.barh(course,boys,label="Boys");
```

```
plt.barh(course,girls,bottoms=boys,label="Girls");
```

```
plt.title("Student");
```

```
plt.show();
```



Workshop - รายงานยอดขายร้านชาบู

ข้อมูลโต๊ะร้านชาบู (Excel)

size = จำนวนลูกค้า

gender = เพศ (ชาย,หญิง,อื่นๆ)

day = วัน

price = ราคา

total=ยอดรวมแต่ละโต๊ะ

