

# Python可视化

- 1. Matplotlib
- 2. Seaborn
- 3. Plotly
- 4. Networkx
- 5. Pandas
- 6. Ggplot
- 7. Bokeh

1

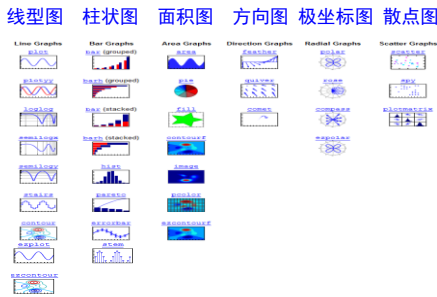
# Python可视化

- 二维曲线的绘制
- 特殊图形绘制
- 三维图形绘制
- 句柄图形

2

## 二维曲线的绘制

二维曲线功能很强大，主要包括



3

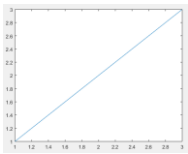
## 基本绘图命令

用plot(x) 绘制x向量曲线

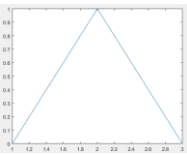
若x是长度为n的数值向量，则坐标系的**纵坐标为向量x**，横坐标为MATLAB系统根据x向量的元素序号自动生成的从1开始的向量。

plot(x) 命令用于在坐标系中顺序地**用直线段连接各点**，生成1条折线，当向量的元素充分多时，可以得到1条光滑的曲线。

plot([1 2 3])



plot([0 1 0])



4

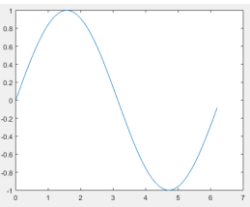
## 基本绘图命令

用plot(x, y) 命令绘制向量x和y的曲线

若参数x和y都是长度为n的向量，则x、y的长度必须相等，用plot(x, y) 命令绘制纵坐标为向量y、横坐标为向量x的曲线

```
x1 = 0 : 0.1 : 2*pi;  
y1 = sin(x1);  
plot(x1, y1);
```

把plot([0 1 0])用这个形式表达



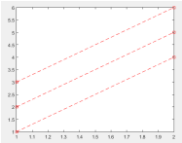
5

## 基本绘图命令

用plot(x) 命令绘制矩阵x的曲线

若x是1个mxn的矩阵，则plot(x) 命令为矩阵的**每一列画出1条线**，共n条曲线，各曲线自动地用不同颜色表示；每条线的横坐标为向量1:m，m是矩阵的行数，绘制方法与向量相同。

```
X1 = [1 2 3;4 5 6];  
plot(x1)
```



6

基本绘图命令

用plot(x, y) 命令绘制混合式曲线

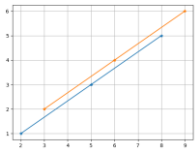
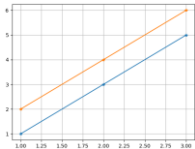
- If both x and y are 2D, they must have the same shape.
- If only one of them is 2D with shape (N, m) the other must have length N and will be used for every data set m.

7

基本绘图命令

```
x1 = [1 2 3]
y1 = [1 2
      3 4
      5 6]
plt.plot(x1,y1,'-k*')

x1 = [2 3
      5 6
      8 9]
y1 = [1 2
      3 4
      5 6]
plt.plot(x1,y1,'-k*')
```



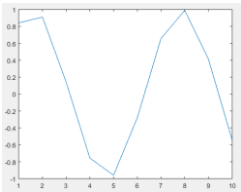
8

基本绘图命令

用plot() 命令绘制复向量曲线

当数据为复向量时，以实部作为横坐标，以虚部作为纵坐标。

```
a = np.arange(1,10)
b = np.sin(a)
z = a + 1j*b
plt.plot(z.real, z.imag)
```



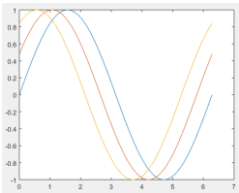
9

基本绘图命令

用plot(x1, y1, x2, y2, ...) 命令绘制多条曲线

plot命令还可以同时绘制多条曲线，用多个矩阵对为参数，MATLAB自动以不同的颜色绘制不同曲线。每一对矩阵(Xi, Yi) 均按照前面的方式解释，不同的矩阵对之间，其维数可以不同。

```
x = 0:pi/100:2*pi;
y1 = sin(x);
y2 = sin(x+5);
y3 = sin(x+1);
plot(x, y1, x, y2, x, y3);
```



10

绘制曲线的一般步骤

步 骤	内 容
1	曲线数据准备： 对于二维曲线，准备横坐标和纵坐标数据变量； 对于三维曲面，准备矩阵参变量和对应的函数值
2	指定图形窗口和子图位置： 默认时，打开Figure No. 1窗口或当前窗口、当前子图； 也可以打开指定的图形窗口和子图
3	设置曲线的绘制方式： 线型、色彩、数据点形
4	设置坐标轴： 坐标的范围、刻度和坐标分格
5	图形注释： 图名、坐标名、图例、文字说明
6	着色、明暗、灯光、材质处理（仅对三维图形使用）
7	视点、三度（横、纵、高）比（仅对三维图形使用）
8	图形的精细修饰（图形句柄操作）： 利用对象属性值进行设置； 利用图形窗工具条进行设置

11

多个图形绘制的方法

指定图形窗口：如果需要多个图形窗口同时打开时，可以使用figure语句。

```
H = figure(n) % 返回新图形窗口的句柄
```

同一窗口多个子图：如果需要在同一图形窗口中布置几幅独立的子图，则可以在plot命令前加上subplot命令以便将1个图形窗口划分为多个区域，每个区域1幅子图

```
H = subplot(m,n,k) % m x n幅子图中，按顺序第k幅成为当前图
% 返回坐标系的句柄。
```

```
subplot(2,3,1), subplot(2,3,2)
subplot(2,3,4), subplot(2,3,6)
```

12

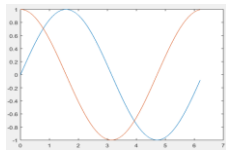
多个图形绘制的方法

同一窗口多次叠绘：在当前坐标系中绘图时，可以多次调用plot函数，最后才显示。

```
x1 = 0:0.1:2*pi;
plot(x1,sin(x1))

plot(x1,cos(x1))

show()
```



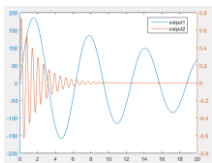
13

多个图形绘制的方法

双纵坐标图

twinx(x2,y2) % x1,y1 左y轴, x2,y2 右y轴

```
plt.plot(t, data1, 'r-')
ax2 = plt.gca().twinx()
ax2.plot(t, data2, 'g.-')
```



14

曲线的线形、颜色和数据点形

颜色		数据点间连线		数据点形	
类型	符号	类型	符号	类型	符号
黄色	y (Yellow)	实线 (默认)	—	实点标记	.
品红色 (紫色)	m (Magenta)	点线	:	圆圈标记	o
青色	c (Cyan)	点画线	-.	叉号形	x
红色	r (Red)	虚线	--	十字形	+
绿色	g (Green)			星号标记	*
蓝色	b (Blue)			方块标记	s
白色	w (White)			钻石形标记	d
黑色	k (Black)			向下的三角形标记	v
				向上的三角形标记	^
				向左的三角形标记	<
				向右的三角形标记	>
				五角星标记	p
				六连形标记	h

17

设置坐标轴和文字标注

分格线

grid(True) % 显示分格线  
grid(False) % 不显示分格线

文字标注

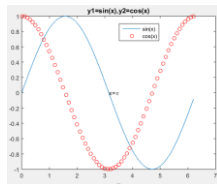
图形的文字标注是指在图形中添加标志性的注释，文字标注包括：

- 图名 (Title) : title( 'figure 1' )、
- 坐标轴名 (Label) : xlabel('X'), ylabel('Y')
- 文字注释 (Text) : text(x, y, string)
- 图例 (Legend) : legend([ 'line1' , 'line2'])

16

设置坐标轴和文字标注

```
plt.title('y1=sin(x),y2=cos(x)')
plt.xlabel('x')
plt.legend(['sin(x)', 'cos(x)'])
text(pi, sin(pi), 'x=\pi')
```



设置坐标轴和文字标注

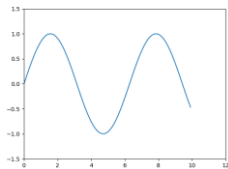
特殊符号：图形中的文字标志使用特殊字符，如希腊字母、数学符号，公式等

类别	命令	字符	命令	字符	命令	字符	命令	字符
希腊字母	\alpha	$\alpha$	\eta	$\eta$	\mu	$\mu$	\upsilon	$\upsilon$
	\beta	$\beta$	\theta	$\theta$	\xi	$\xi$	\Upsilon	$\Upsilon$
	\epsilon	$\epsilon$	\Theta	$\Theta$	\Xi	$\Xi$	\Phi	$\Phi$
	\gamma	$\gamma$	\iota	$\iota$	\pi	$\pi$	\chi	$\chi$
	\Gamma	$\Gamma$	\zeta	$\zeta$	\Pi	$\Pi$	\Psi	$\Psi$
	\delta	$\delta$	\kappa	$\kappa$	\rho	$\rho$	\Omega	$\Omega$
	\Delta	$\Delta$	\lambda	$\lambda$	\sigma	$\sigma$		
	\omega	$\omega$	\Lambda	$\Lambda$	\Sigma	$\Sigma$		
	\Omega	$\Omega$	\Sigma	$\Sigma$	\Xi	$\Xi$		
	\approx	$\approx$	\lesseqgtr	$\lesseqgtr$	\leq	$\leq$		
数学符号	\approx	$\approx$	\lesseqgtr	$\lesseqgtr$	\leq	$\leq$		
	\asymp	$\asymp$	\lesseqgtr	$\lesseqgtr$	\leq	$\leq$		
	\sim	$\sim$	\lesseqgtr	$\lesseqgtr$	\leq	$\leq$		
	\sim	$\sim$	\lesseqgtr	$\lesseqgtr$	\leq	$\leq$		
	\cup	$\cup$	\cap	$\cap$	\vee	$\vee$	\wedge	$\wedge$
	\cup	$\cup$	\cap	$\cap$	\vee	$\vee$	\wedge	$\wedge$
	\cup	$\cup$	\cap	$\cap$	\vee	$\vee$	\wedge	$\wedge$
	\cup	$\cup$	\cap	$\cap$	\vee	$\vee$	\wedge	$\wedge$
	\cup	$\cup$	\cap	$\cap$	\vee	$\vee$	\wedge	$\wedge$
	\cup	$\cup$	\cap	$\cap$	\vee	$\vee$	\wedge	$\wedge$
箭头	\uparrow	$\uparrow$	\downarrow	$\downarrow$	\rightarrow	$\rightarrow$	\leftarrow	$\leftarrow$
	\leftarrow	$\leftarrow$	\rightarrow	$\rightarrow$	\leftarrow	$\leftarrow$	\rightarrow	$\rightarrow$

# 设置坐标轴区间

坐标区间:

```
plt.xlim(0, 12)      or plt.gca().set_xlim([0, 12])
plt.ylim(0, 12)      or plt.gca().set_ylim([0, 12])
```



19

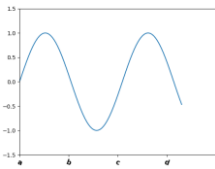
# 设置坐标轴和文字标注

坐标刻度: 默认的刻度是自动等距离分隔的, 但有些刻度需要特别标注出来, 因此需要使用坐标刻度专门标注。

- 通过设置xtick和ytick属性可以划分坐标刻度。
- 通过设置xticklabel和yticklabel属性可以标注将坐标刻度的标志

```
plt.xticks(np.arange(0,12,3), ['a','b','c','d'],
rotation=-20,
fontsize=12,
fontweight='bold')
```

```
ax.set_xticks()
ax.set_xticklabels()
```



20

# 图形对象的操作

对象句柄的获取

```
gcf      %获取当前图形窗口句柄
gca      %获取当前坐标轴句柄
```

```
h = plt.figure()
h.axes
h.get_axes()
h.gca()
h.clf()
h.add_axes()
h.add_subplot
```

21

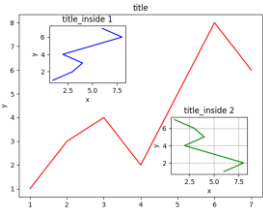
# 图形对象的操作

left, bottom, width, height = 0.2, 0.6, 0.25, 0.25

```
ax = fig.add_axes([left, bottom, width, height])
```

```
ax.plot(y, x, 'b')
```

```
ax.plot(y, x, 'b')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_title('title_inside 1')
```



22

# 图形对象的操作

查找对象。用命令 findobj可以快速查找所有对象, 以及获取指定属性值的对象句柄。

```
h=findobj      %返回根对象和所有子对象的句柄
h=findobj(h_obj) %返回指定对象的句柄
h=findobj('PropertyName', PropertyValue) %返回符合指定属性值的对象句柄
```

```
import matplotlib.text as text
for obj in fig.findobj(text.Text):
    if obj.get_text() == 'x':
        break
obj.set_color('red')
obj.set_fontsize(15)
```

23

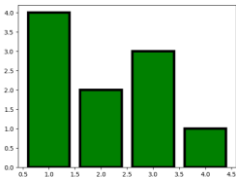
# 特殊图形绘制

条形图: 常用于对统计的数据进行作图, 特别适用于少量且离散的数据

```
bar(x, height, width=0.8, align='center')
```

```
plt.bar(x = [1, 2, 3, 4], height = [4, 2, 3, 1],
width=0.8, edgecolor='black',
linewidth=4, align='center', color='g')
```

```
tick_label=['a','b','c','d']
```



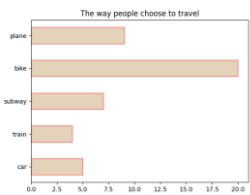
24

## 特殊图形绘制

**条形图：** 常用于对统计的数据进行作图，特别适用于少量且离散的数据

```
y=[1,2,3,4,5]
width=[5,4,7,20,9]
label=['car','train','subway','bike','plane']

plt.barh(y,width,facecolor='tan',height=
0.5,edgecolor='r',alpha=0.6,tick_label=l
abel)
```



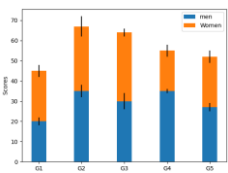
25

## 特殊图形绘制

**条形图：** 常用于对统计的数据进行作图，特别适用于少量且离散的数据

```
ax.bar(labels,men_means,width,yerr=men_std,label='men')

ax.bar(labels,women_means,width,
yerr=women_std,bottom=men_means,
label='Women')
```

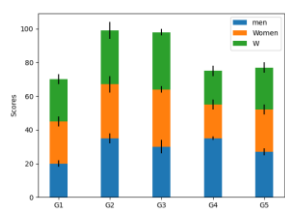


26

## 特殊图形绘制

**条形图：** 常用于对统计的数据进行作图，特别适用于少量且离散的数据

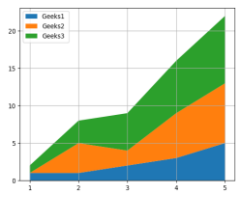
How to get this?



27

## 特殊图形绘制

```
x = [1, 2, 3, 4, 5]
y1 = [1, 1, 2, 3, 5]
y2 = [0, 4, 2, 6, 8]
y3 = [1, 3, 5, 7, 9]
labels = ["Geeks1 ", "Geeks2", "Geeks3"]
ax = plt.gca()
ax.stackplot(x, y1, y2, y3, labels=labels)
```

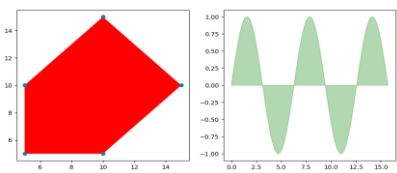


28

## 特殊图形绘制

```
a = [5, 10, 15, 10, 5]
b = [5, 5, 10, 15, 10]
plt.fill(a, b, 'r')
```

```
x=np.linspace(0,5*np.pi,1000)
y1=np.sin(x)
plt.fill(x,y1,color="g",alpha=0.3)
```



29

## 特殊图形绘制

matplotlib.pyplot.fill\_between(x, y1, y2=0, where=None)

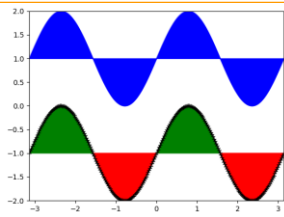
- Fill the area between two horizontal curves.
- The curves are defined by the points (x, y1) and (x, y2). This creates one or multiple polygons describing the filled area.
- You may exclude some horizontal sections from filling using where.

30

## 特殊图形绘制

```
X = np.linspace(-np.pi, np.pi, n, endpoint=True)
Y = np.sin(2 * X)
plt.fill_between(X, 1, Y + 1, color='blue')

plt.fill_between(X, -1, Y - 1, where=-1 > Y - 1, color='red')
plt.fill_between(X, -1, Y - 1, where=-1 < Y - 1, color='green')
```



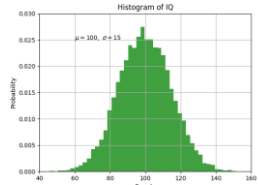
31

## 特殊图形绘制

直方图：统计数据的频率

```
np.random.seed(19680801)
mu, sigma = 100, 15
x = mu + sigma *
np.random.randn(10000)

plt.hist(x, 50, density=True, facecolor='g',
alpha=0.75)
```

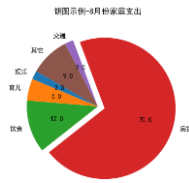


32

## 特殊图形绘制

饼形图：pie(x, explode=None, labels=None, colors=None)

```
labels = ['娱乐','育儿','饮食','房贷','交通','其它']
sizes = [2,5,12,70,2,9]
explode = (0,0,0,0,1,0,0)
plt.pie(sizes,explode=explode,
labels=labels,startangle=150)
```

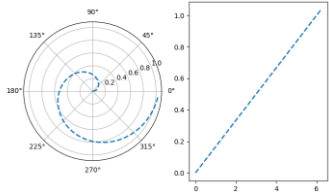


33

## 特殊图形绘制

极坐标图

```
theta = np.arange(0, 2*np.pi, 2*np.pi/N)
ax1 = plt.subplot(121, projection='polar')
ax1.plot(theta,theta/6,'--',lw=2)
ax2.plot(theta,theta/6,'--',lw=2)
```



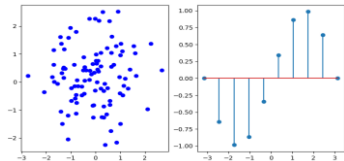
34

## 离散数据图

stem(x, y, 'filled') %画火柴杆图  
scatter(x, y) %画点图

```
x = np.random.randn(100)
y = np.random.randn(100)
plt.scatter(x, y, c='b')

x = np.linspace(-np.pi, np.pi, n)
y = np.sin(x)
plt.stem(x, y)
```



35

## 练习

- 1、绘制函数  $y = e^x$  在区间  $[-2, 2]$  上的图像。
- 2、在同一坐标系上绘制  $y = \cos 2x$  和  $y = \frac{x}{x+1}$  在区间  $[0, 2\pi]$  上的图像，并要求  $y = \cos 2x$  用红色\*号描绘，  $y = \frac{x}{x+1}$  用蓝色虚线描绘。
- 3、用subplot，画  $\sin(x)$ ,  $\cos(x)$ ,  $\sin(x)+\cos(x)$
- 4、有一组测量数据满足  $y = \exp(-at)$ ，t的变化范围为  $0 \sim 10$ ，用不同的线型和标记点画出  $a=0.1$ 、 $a=0.2$  和  $a=0.5$  三种情况下的曲线。

36

