Julia 使用 PyPlot库 画图

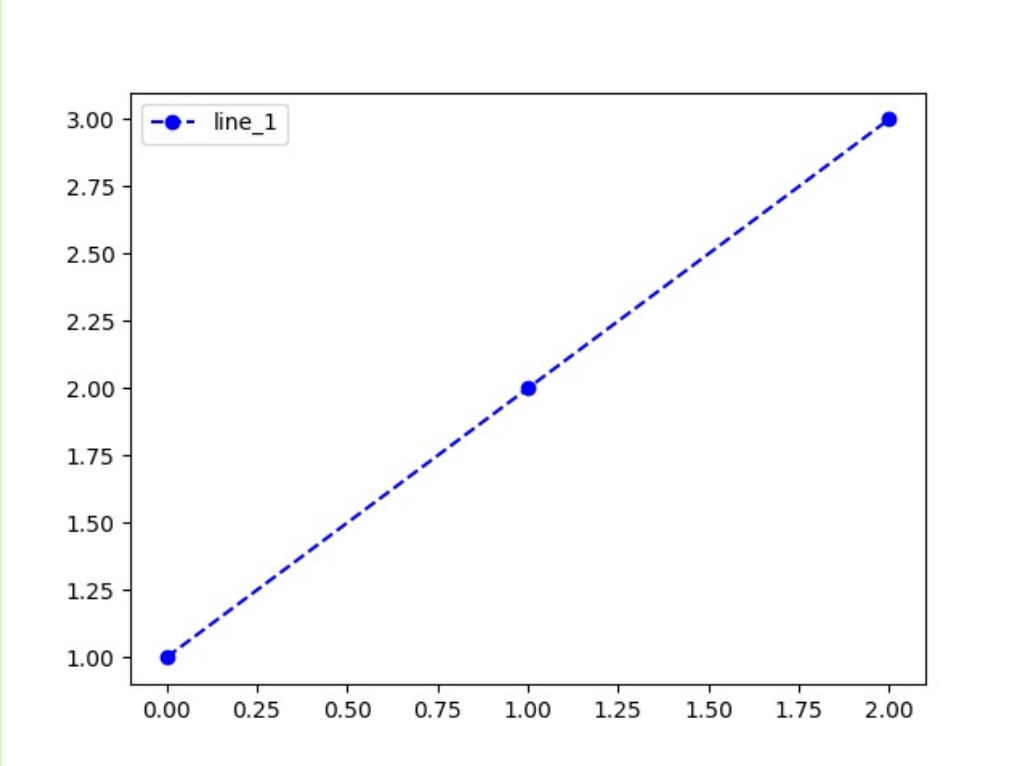
❖ 引入PyPlot

- using PyPlot //加载内容较多,耗时较长

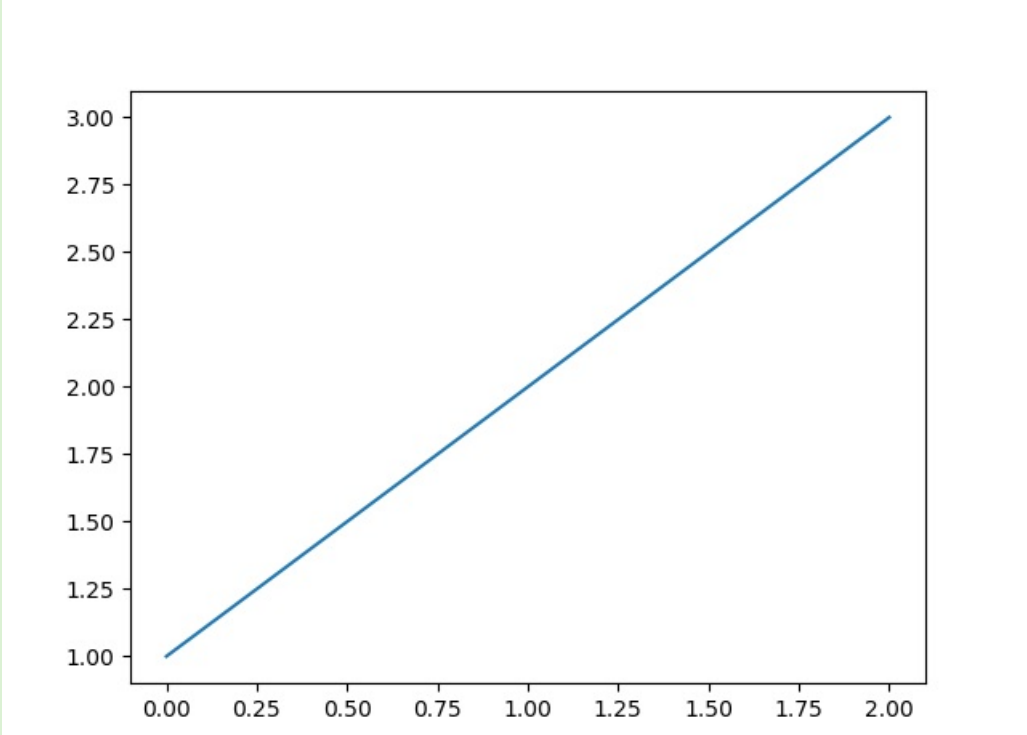
简单图 展示

❖ 点图,线图,虚线图

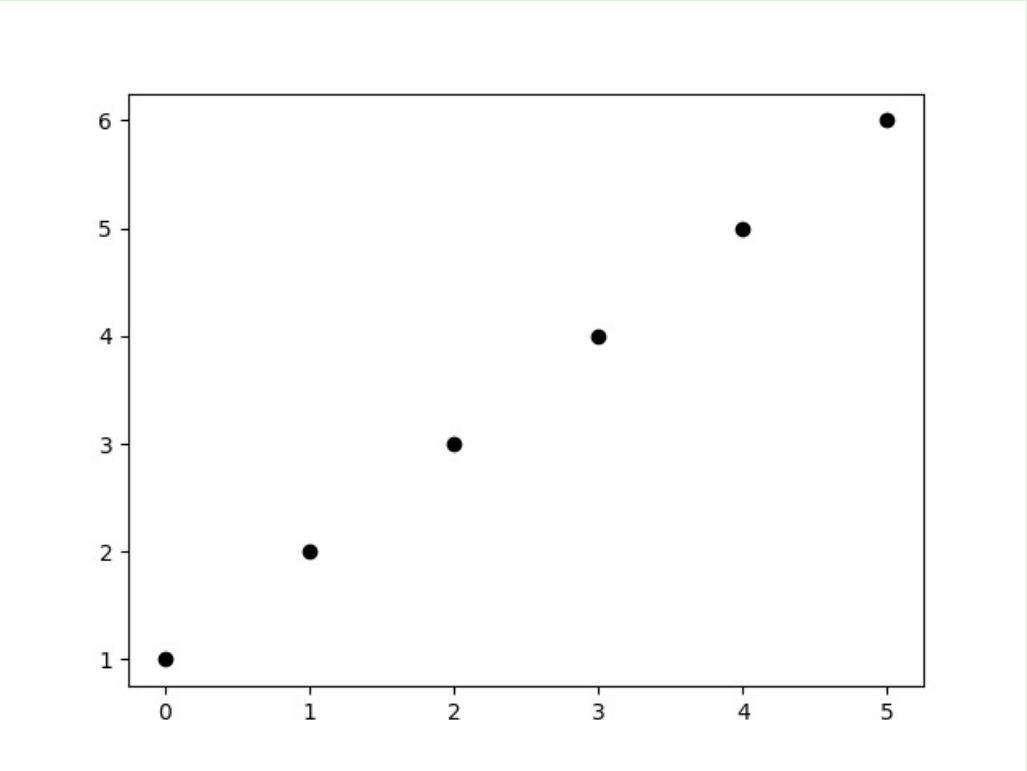
```
>>>plot([1,2,3],"bo--",label="line_1"); #蓝色圆点虚线bo--
>>>legend()  #显示lable
 show()  #非交互式环境,显示图片命令
```




```
>>> plot([1,2,3])  #默认是直线图:"b-"
```

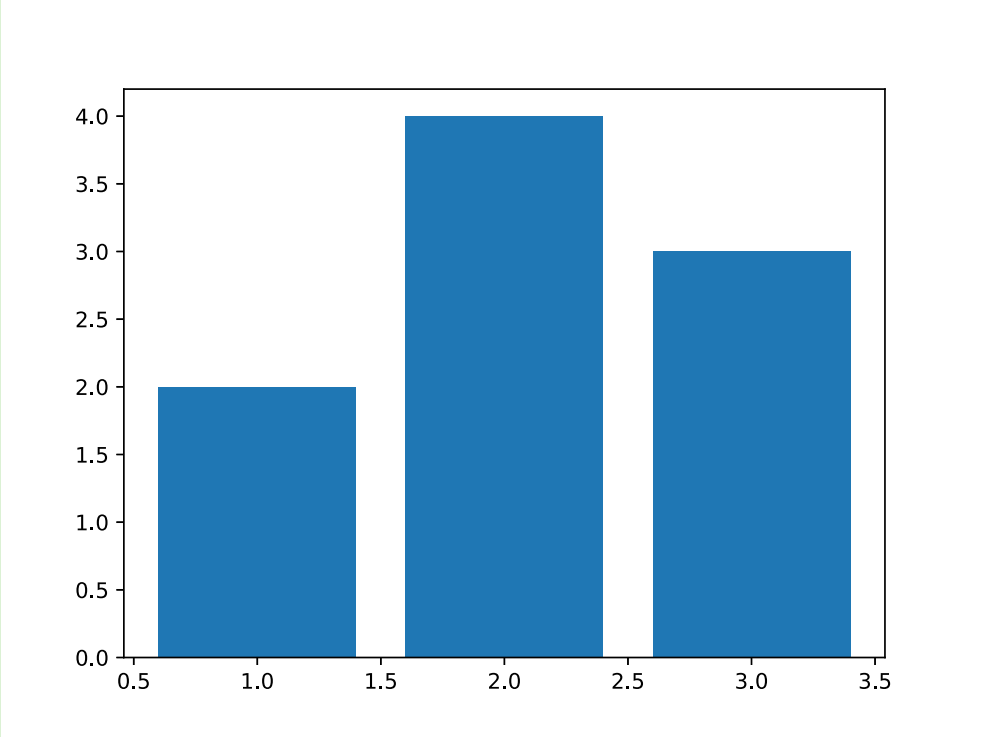


```
>>> plot([1,2,3,4,5,6],"ko")
```



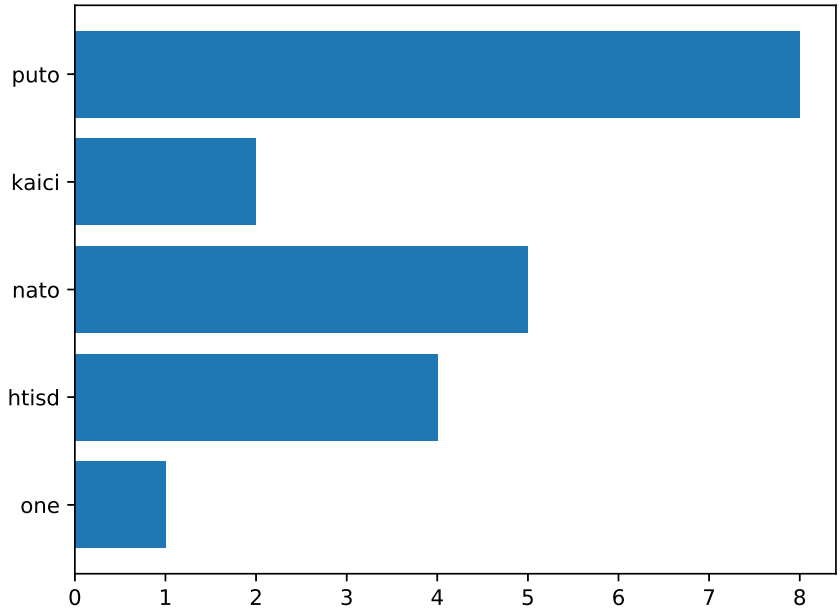
❖ 条形图

```
>>> bar([1,2,3],[2,4,3])
```



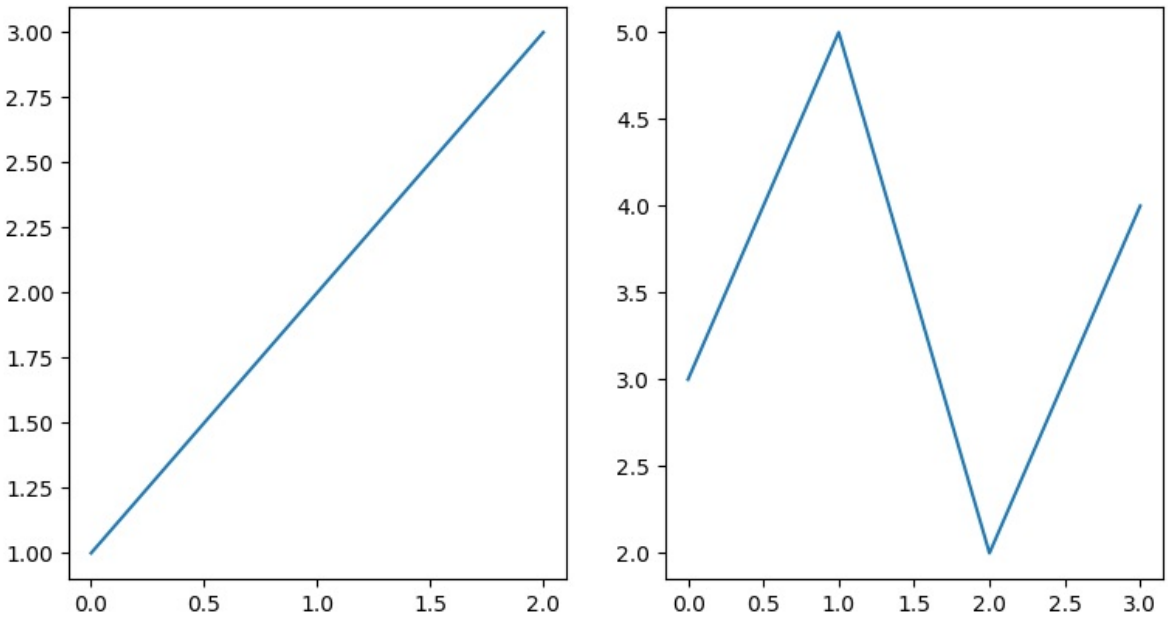
❖ 横向条形图

```
>>> barh(["one","htisd","nato","kaici","puto"],[1,4,5,2,8])
```



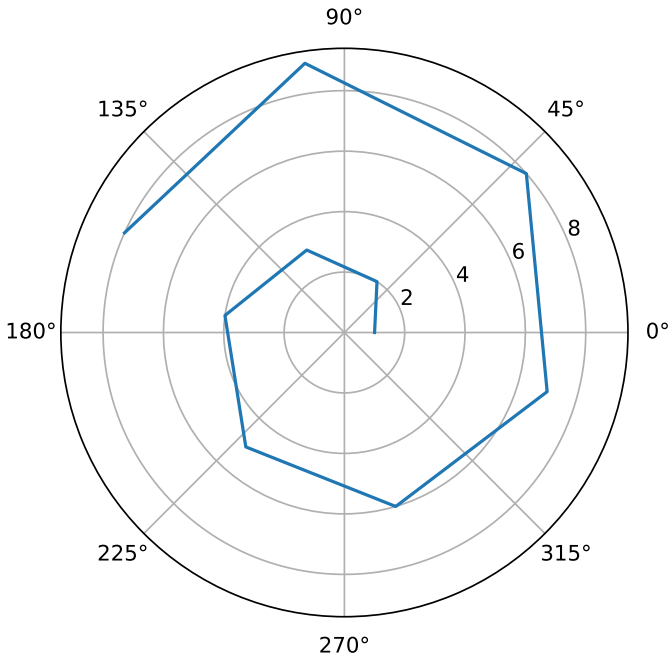
❖ 子图:一行两列

```
>>> subplot(121);plot([1,2,3]);subplot(122);plot([3,5,2,4])
```

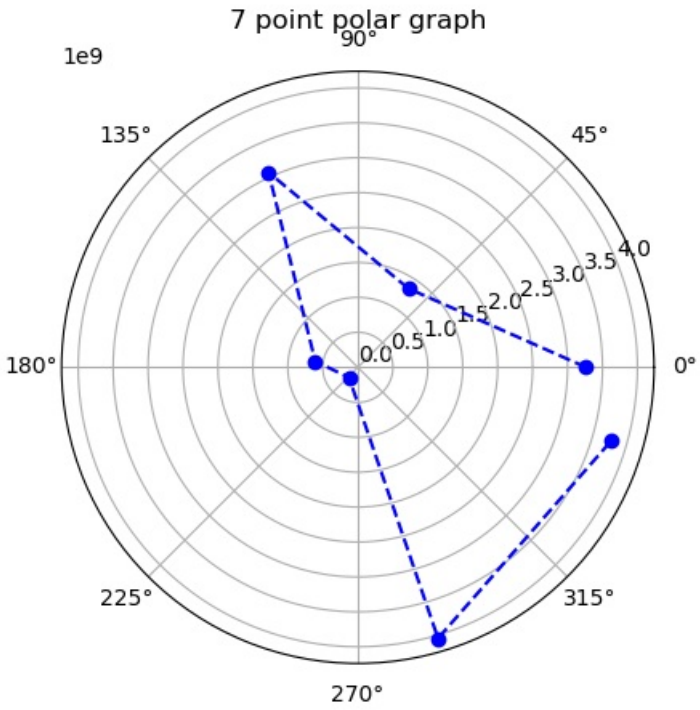


❖ 极坐标图

```
>>>polar([1,2,3,4,5,6,7,8,9,8])
```

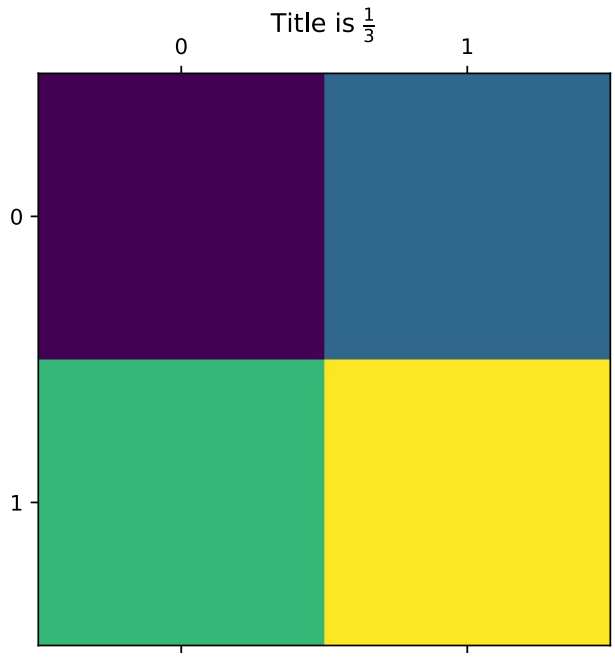



```
>>>polar(rand(UInt32,7),"bo--");title("7 point polar graph")
```



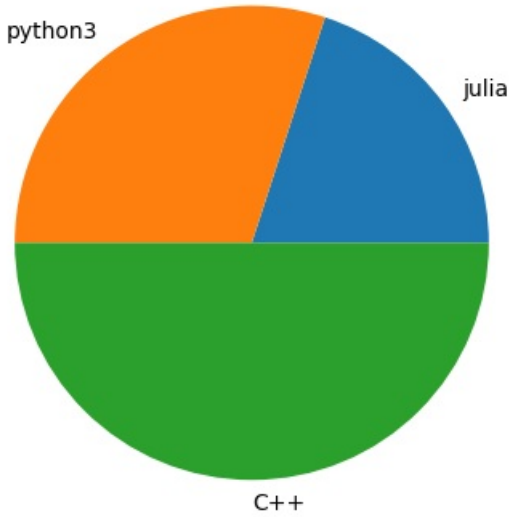
❖ 矩阵图

```
>>>  matshow([[1,2],[3,4]]);title(L"this is $\frac{1}{3}$")  #添加标题
```

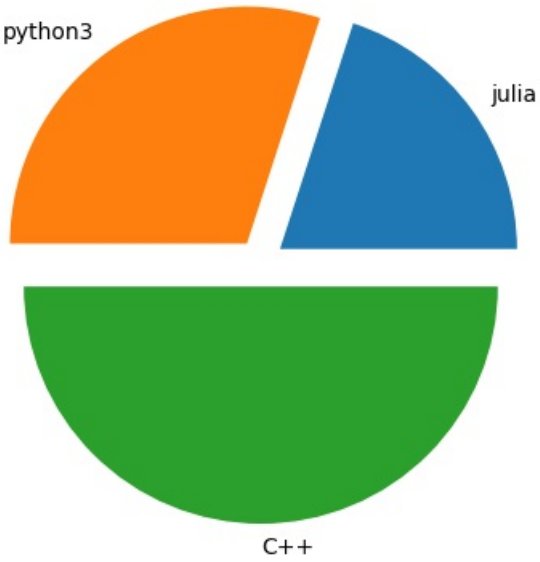


❖ 饼图

```
>>> pie([20,30,50],labels=["julia","python3","C++"])
```

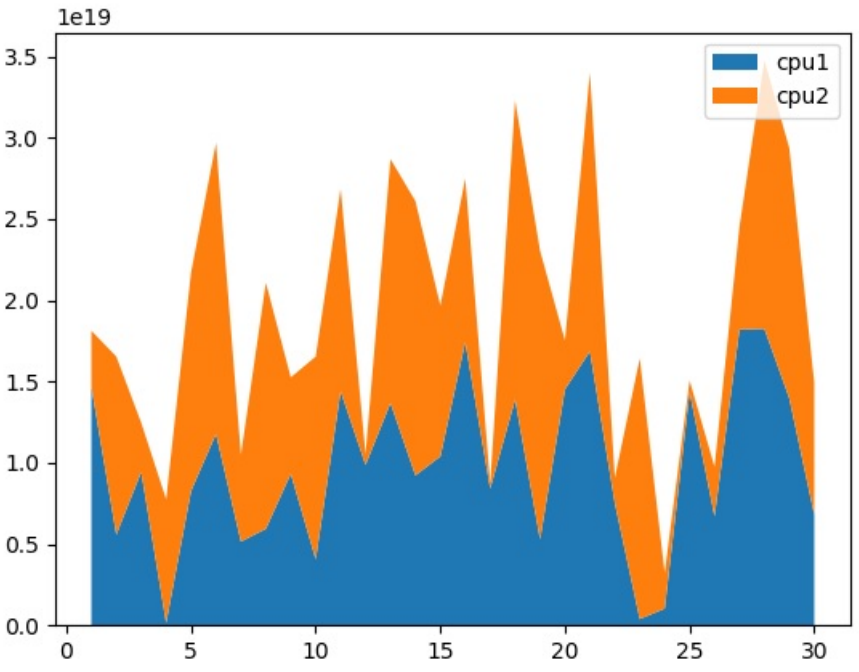



```
>>> pie([20,30,50],labels=["julia","python3","C++"],explode=[0.1,0.1,0.1])
```



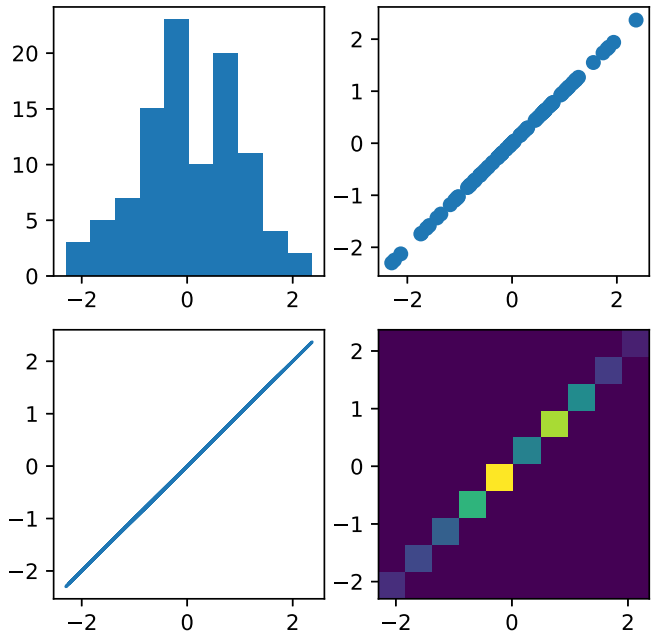
❖ stack_plot

```
>>>stackplot([i for i=1:30],rand(UInt,30),rand(UInt,30),labels=["cpu1","cpu2"]);legend()
```



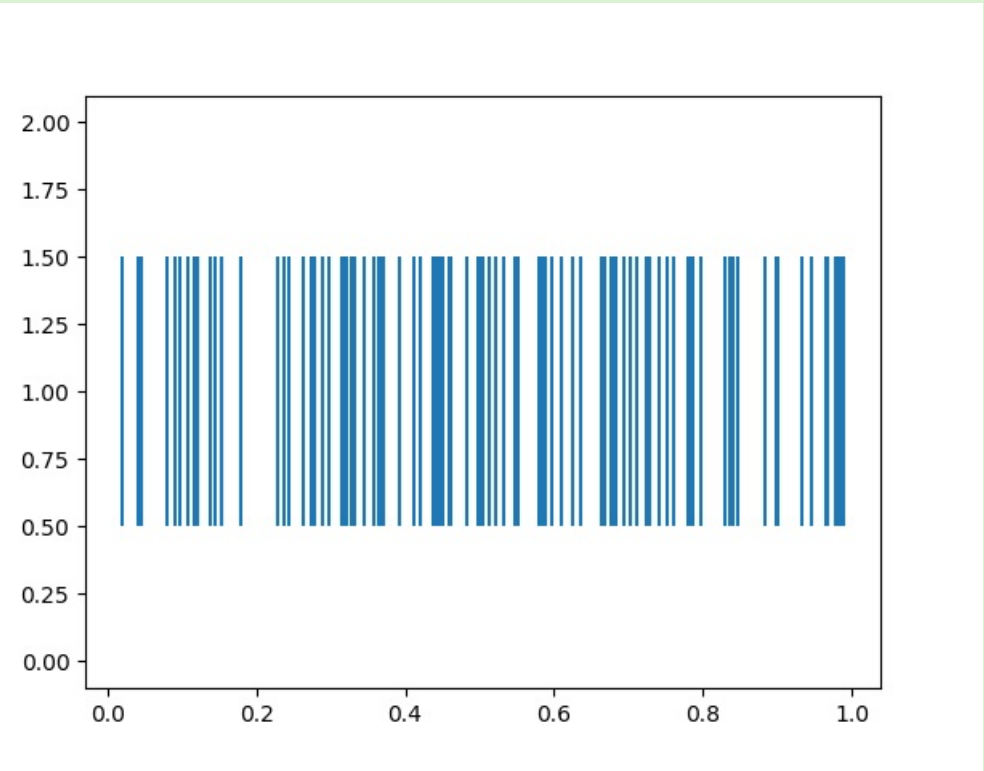
❖ 2x2 的子图

```
>>> data=randn(100)
>>>fig, axs = plt.subplots(2, 2, figsize=(5, 5));
axs[1,1].hist(data);
axs[1, 2].scatter(data, data);
axs[2, 1].plot(data, data);
axs[2,2].hist2d(data, data)
# suptitle("主标题")
```

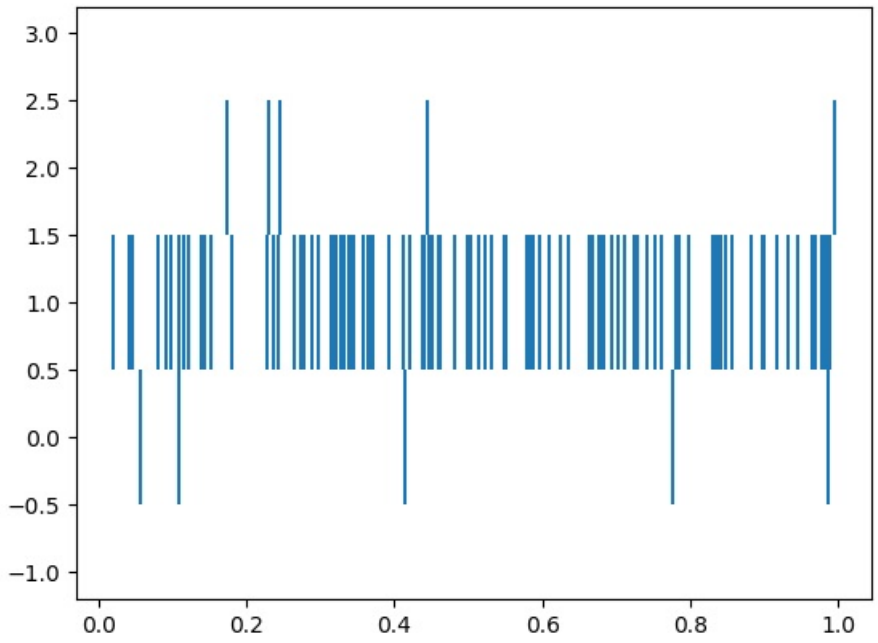


❖ 条形码???

```
>>> eventplot(rand(5,10))
```

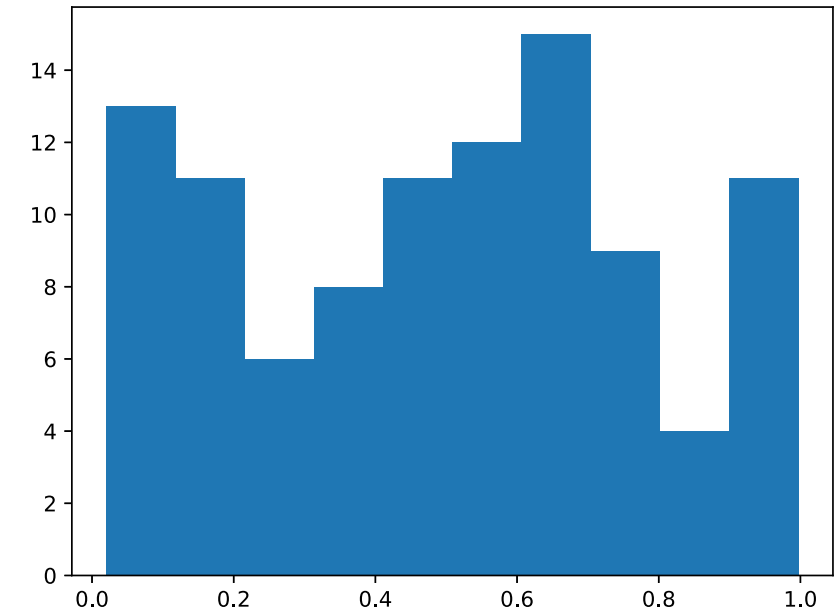



```
>>> eventplot(rand(3,5))
```



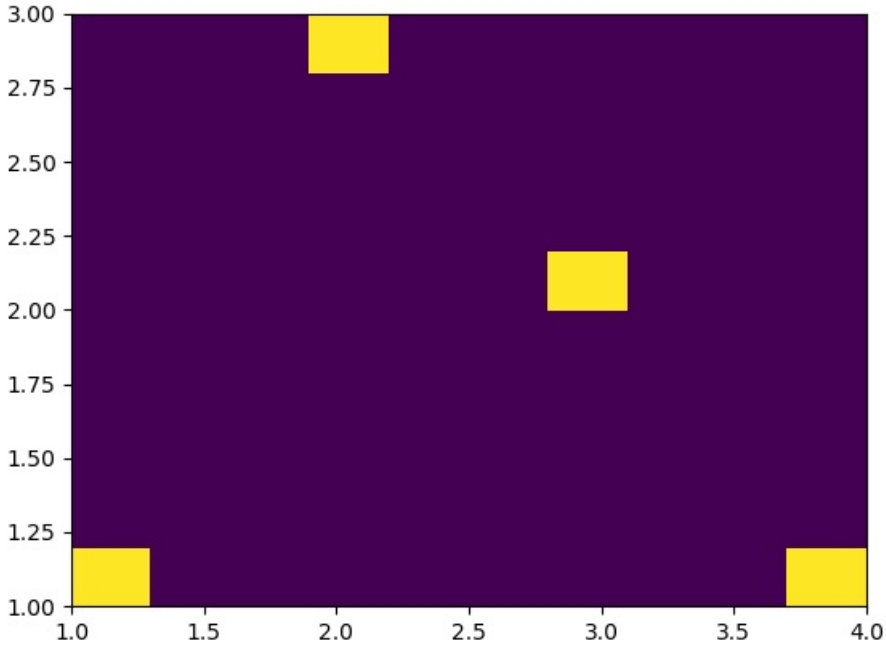
❖ 直方图

```
>>>hist(rand(100))
```



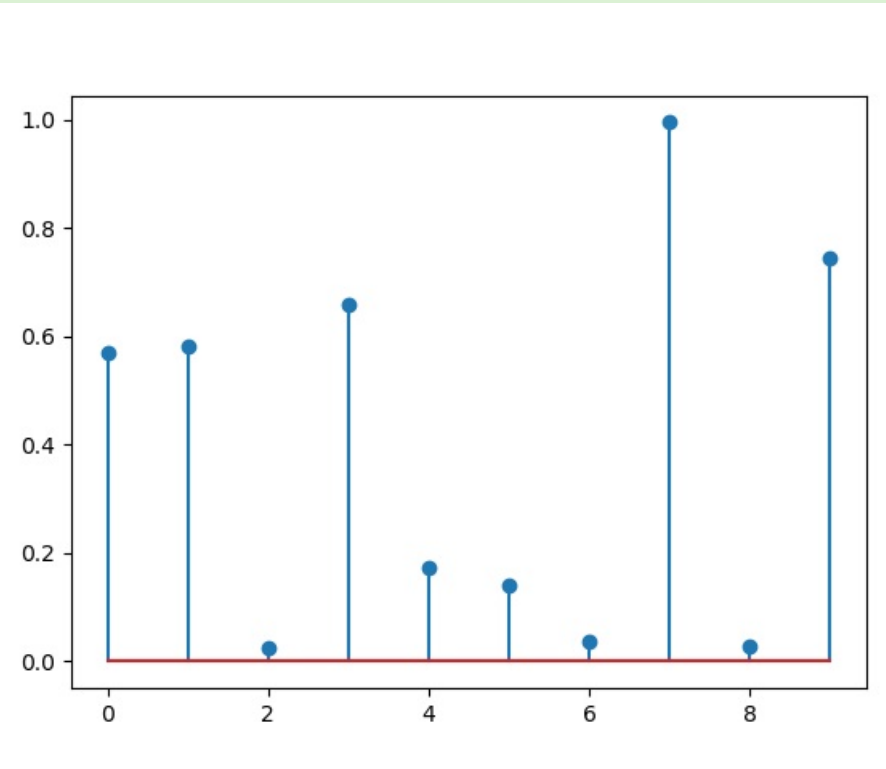
❖ 二维直方图

```
>>>hist2D([1,2,3,4],[1,3,2,1])
```

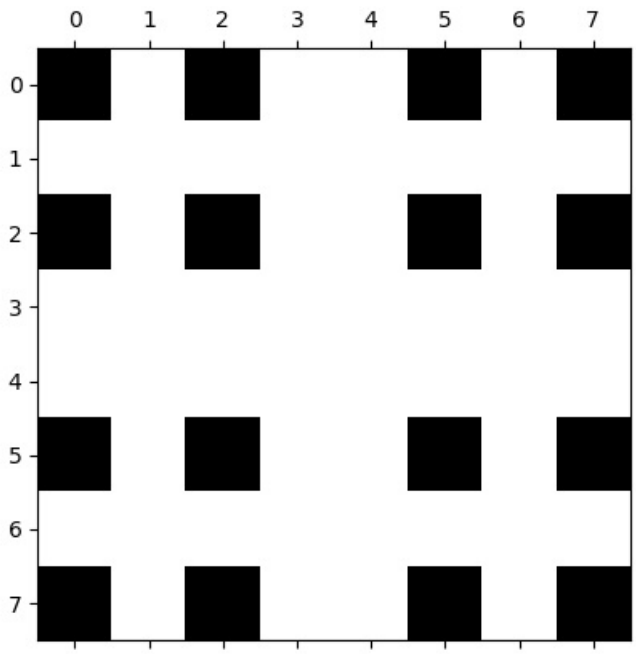


❖ stem 茎图

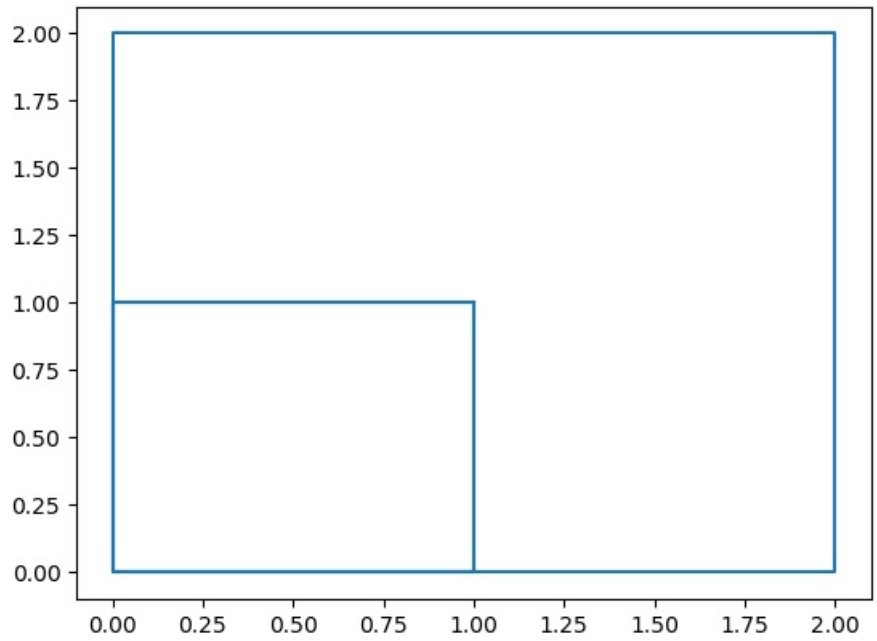
```
>>>stem(rand(10))
```



❖ spy() #展示稀疏模式

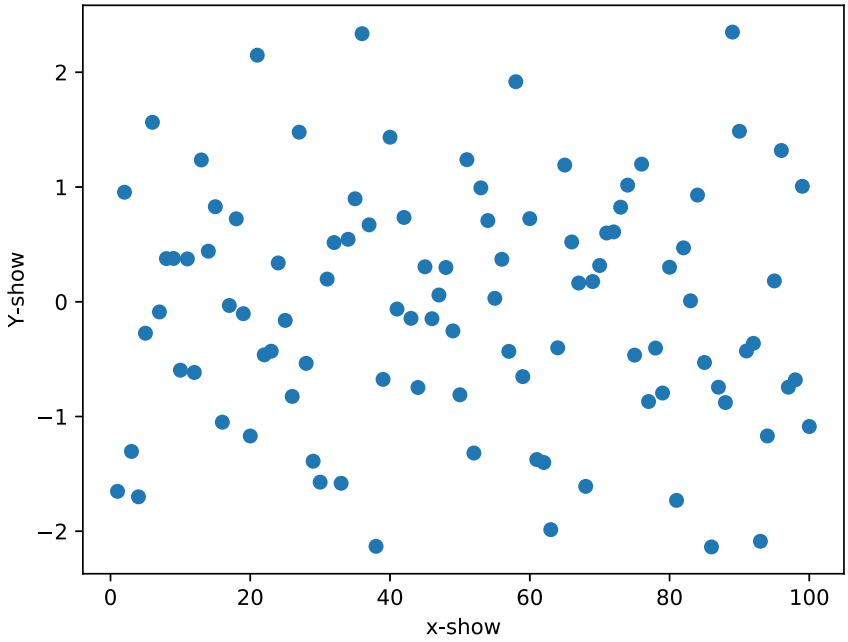


❖ step()

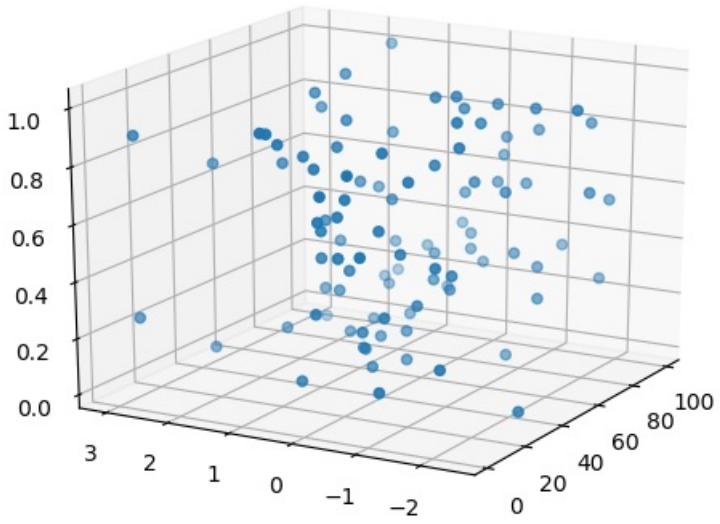


❖ 二维散点图和三围散点图

```
>>> scatter([i for i=1:100],randn(100));xlabel("x-show");ylabel("Y-show")
```




```
>>>scatter3D(rand(100),rand(100),rand(100))
```



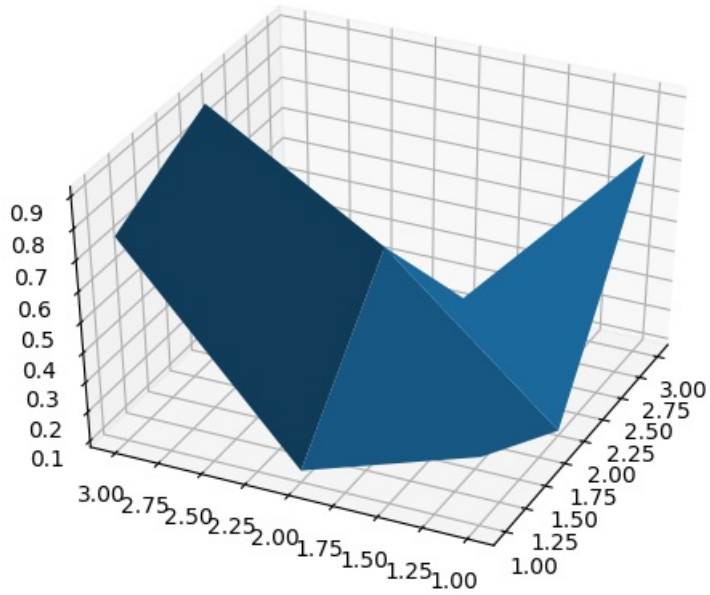
❖ 保存图片命令

- savefig("filename.svg")

3D图形

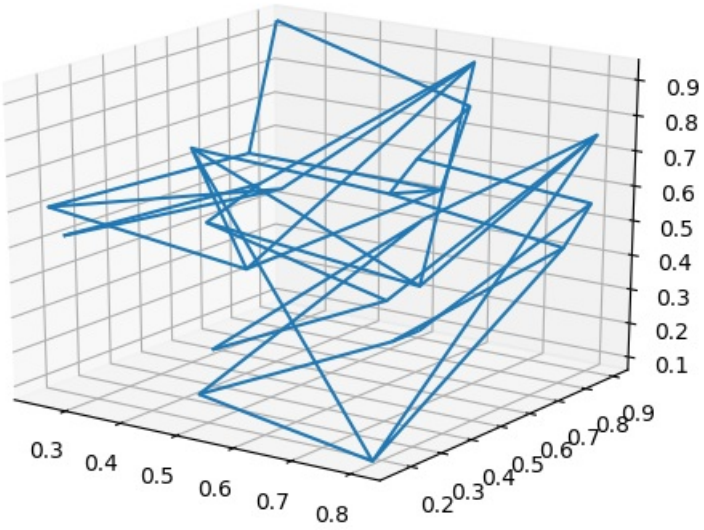
❖ 三维平面

```
>>>surf(rand(3,4))
```

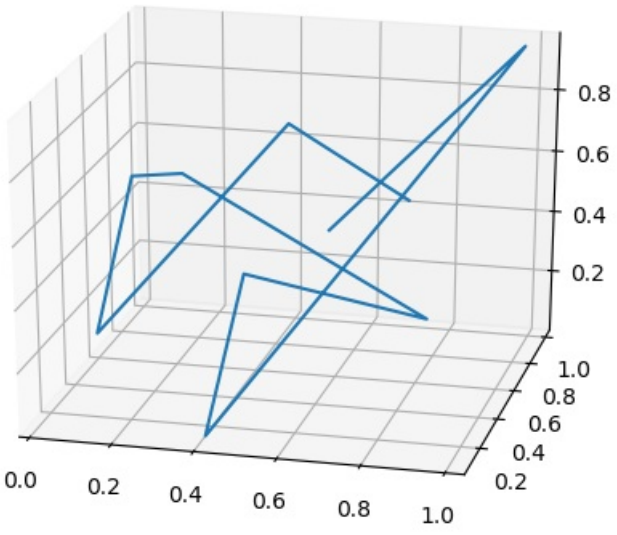


❖ 折线

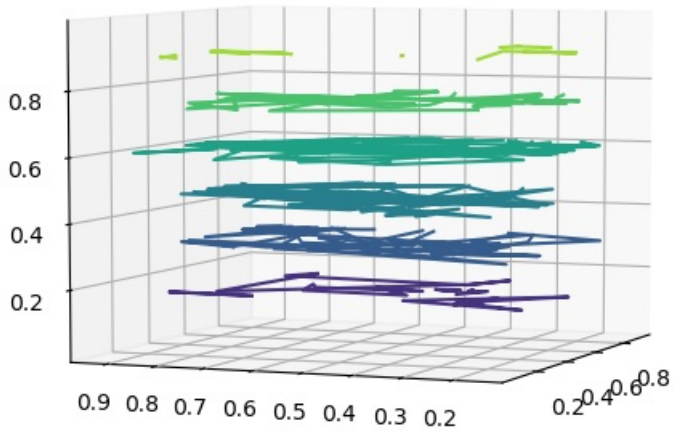
```
>>> mesh(rand(5),rand(5),rand(5,5))
```



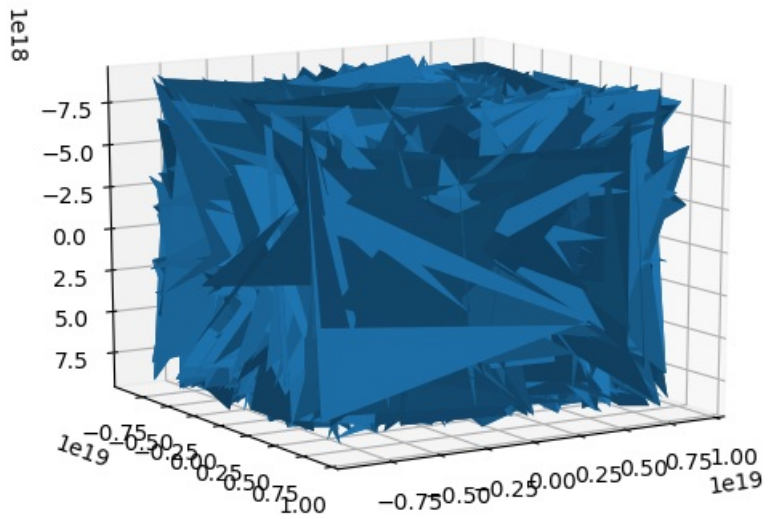

```
>>> plot3D(rand(10),rand(10),rand(10))
```



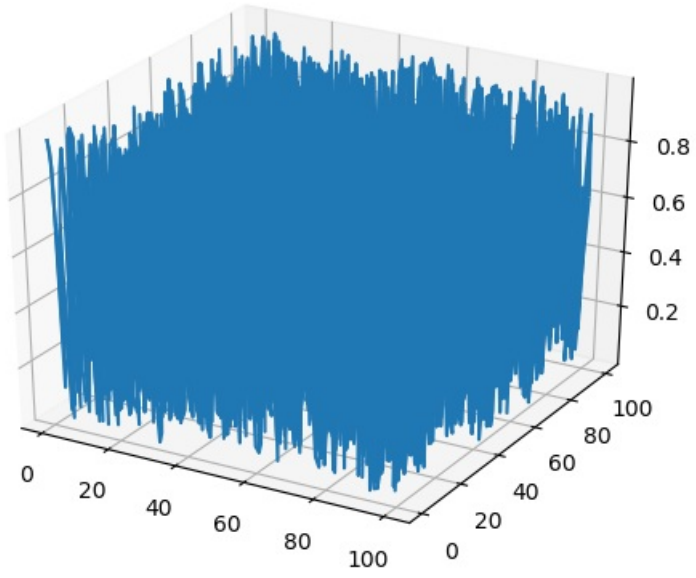

```
>>> contour3D(rand(10),rand(10),rand(10,10))
```




```
>>>asd=[i for i=1:100];plot_surface(asd,asd,asd*asd')
```




```
>>> plot_wireframe([i for i=1:100],[i for i=1:100],rand(100,100))
```



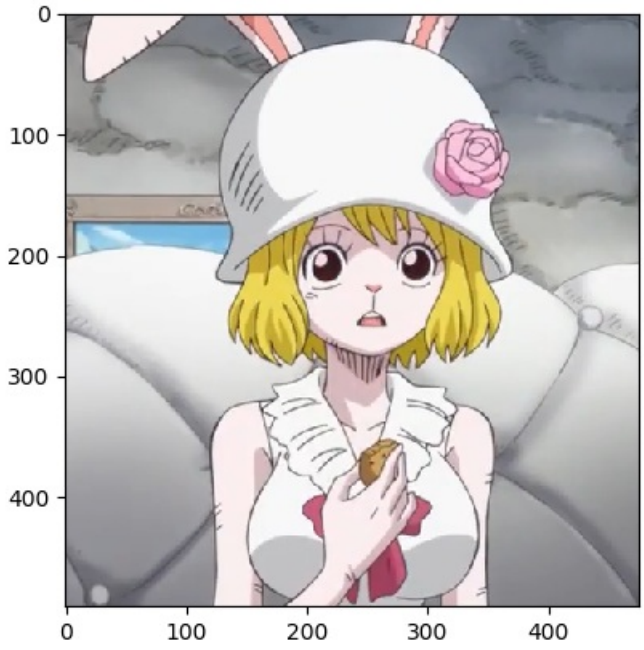
❖ 存疑?

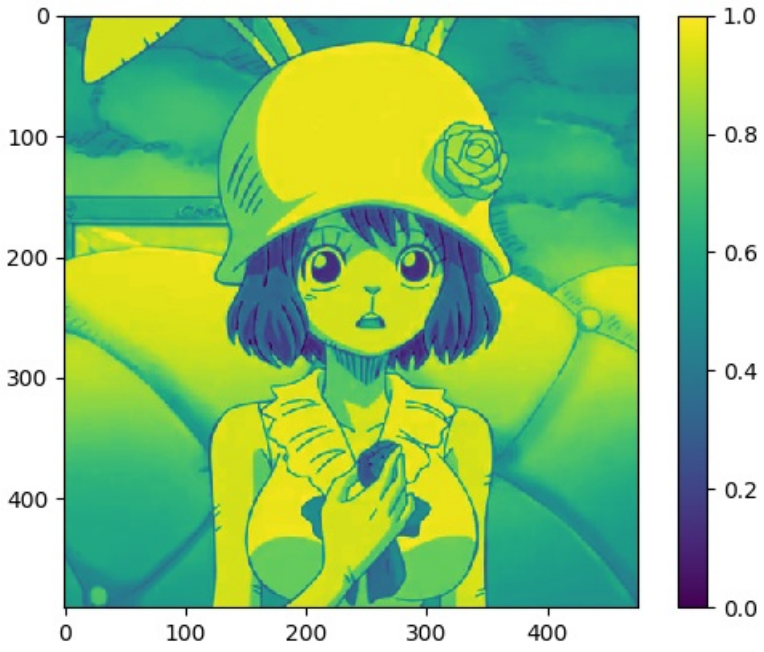
- plotfile() :有意思
- psd() ?
- streamplot() 空气流动图
- table() #图表
- triplot()

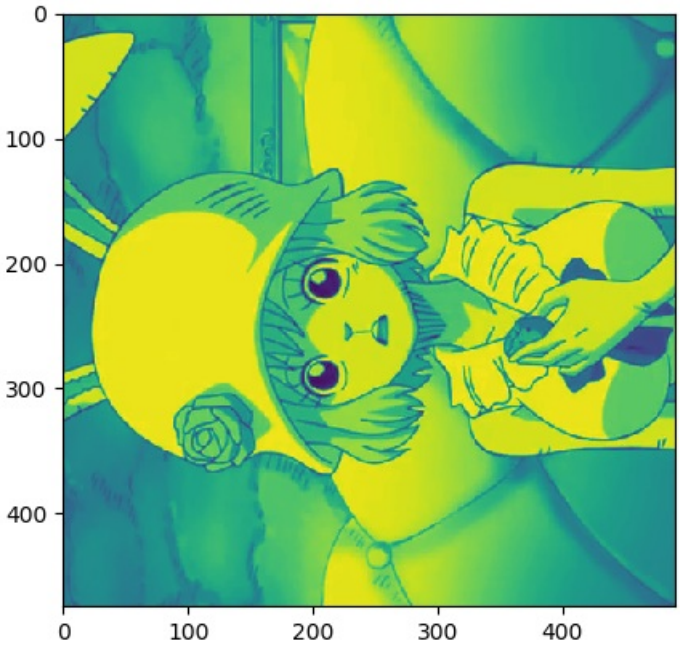
图片操作

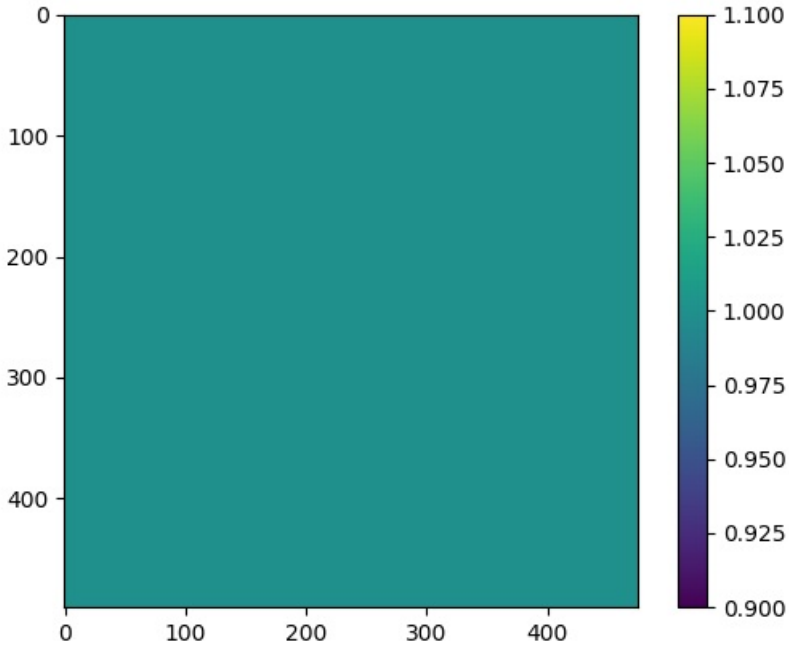
- asd=imread("filename")
- imshow(asd)
- imsave()

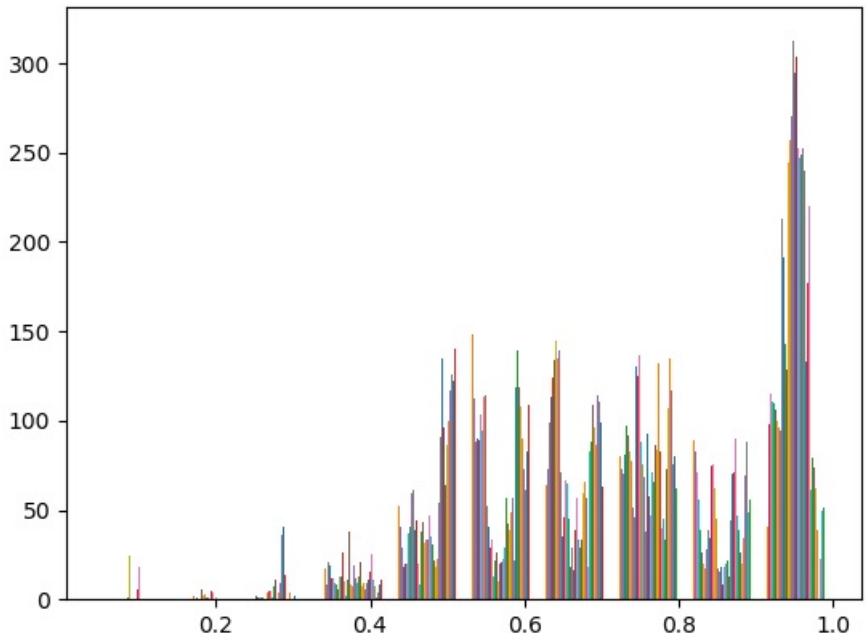
❖ 鼠标在图片上停留

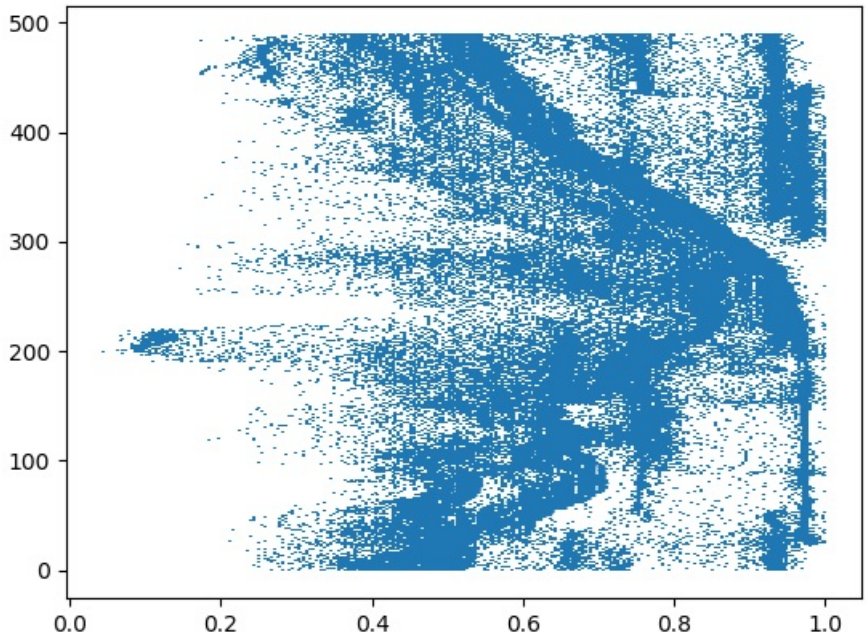

0
100
200
300
400
0
100
200
300
400


0
100
200
300
400
0
100
200
300
400
1.0
0.8
0.6
0.4
0.2
0.0


0
100
200
300
400
0
100
200
300
400


0
100
200
300
400
0
100
200
300
400
1.100
1.075
1.050
1.025
1.000
0.975
0.950
0.925
0.900





其他

- grid(true) 显示网格
- text() 添加文字
- close() 关闭显示的图像
- draw() 重新画
- 开关交互式环境: ion() /\ ioff()
- subplot_tool() #调节子图的位置和间距

使用对象(.)调用

- plt.plot([1, 2, 3, 4]);plt.axis([0, 6, 0, 20]);plt.show()
  - 横坐标0->6
  - 纵坐标0->20

- julia> t = [1,2,3,4,5,6,7,8,9];plt.plot(t, t, "r-", t, t.^2, "bs-", t, t.^3, "g^-")
  - 一个plot中画三条线

❖ 在图中放置文本

- text(5,125,L"this line = x^3")
  - 在x=5,y=125 处放置文本

❖ 改变风格:

```
import matplotlib.pyplot as plt
print(plt.style.available ) //查看可用的风格
style_one=['grayscale','fivethirtyeight','bmh','dark_background',
        'seaborn-whitegrid', 'Solarize_Light2','seaborn-notebook',
         'seaborn-paper', 'fast']
for iter in style_one :
    with plt.style.context(iter):
        plt.plot([1,2,3,4,5,6,7,8,5,4,3,9], 'k-o')
        plt.title(iter)
    plt.show()
```

❖ 箭头 解释

```
plt.annotate('local max', xy=(2, 1), xytext=(3, 1.5),
             arrowprops=dict(facecolor='black', shrink=0.05),
             )
```

例子

❖ draw() 刷新当前图片

```
asda=[]
for i=1:100
       push!(asda,i)
       polar(asda)
       draw()  # 要打开交互式环境 ion()
       sleep(1)
end
```