数据分析September28

numpy:数值计算

matplotlib：数据可视化

scipy：科学计算

pandas：时间日期序列

1 numpy的基本特点

1.矢量化运算：将包含多个数据的集合作为一个整体参与运算

2.称为几乎所有与数值分析，科学计算以及人工智能有关的功能模块的底层模块

3.绝大部分代码用标准C语言编写，更有一小部分代码直接使用汇编语言编写，性能卓越。

4.完全开源且免费

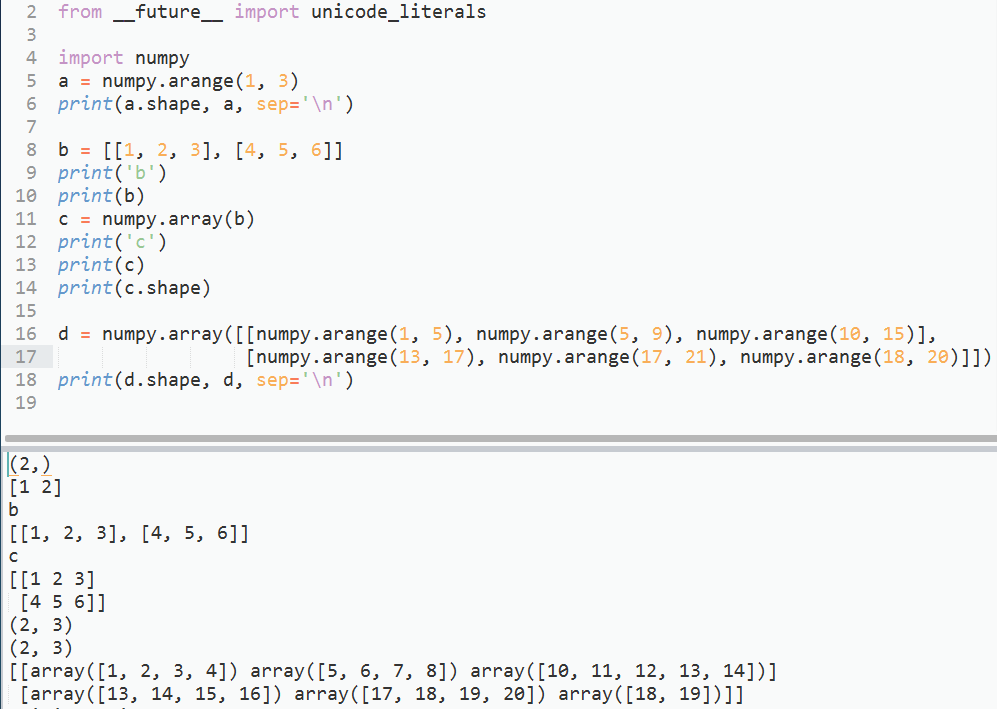
5.数据集合中的元素必须同质，牺牲灵活性换取高性能

2 数组（矢量化运算库的核心类型）

1.numpy中的数组实际上就是numpy.ndarray类类型的对象，用于表示任意维度的数据结构。其维度信息通过shape属性访问，ndarray是一个通用的同构数据多维容器，其中所有的元素必须是相同类型。该对象由两部分组成：

A.实际数据：数据元素本身

B.元数据：对实际数据类型，结构等信息的描述

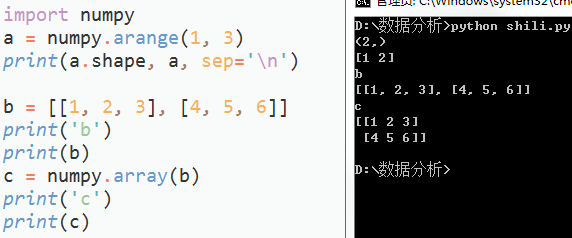


创建数组的两个常用函数：

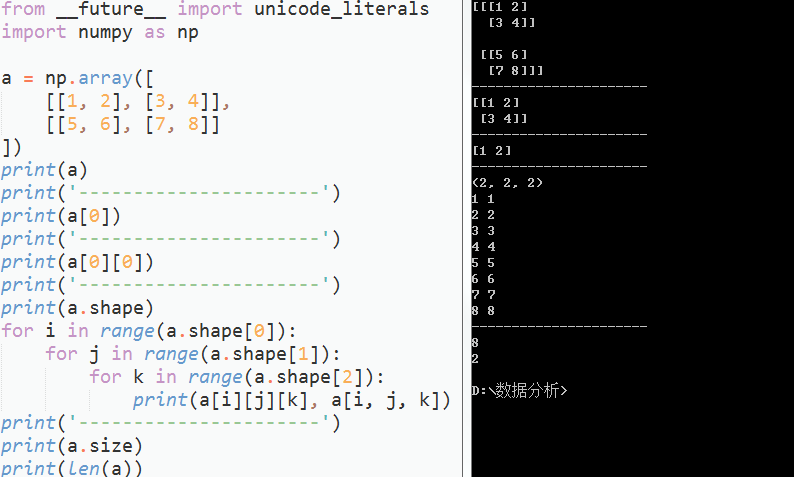
numpy.arange（起始值，终止值，步长）->获得数组对象缺省起始值取0，缺省步长为1

numpy.array（任何同质容器）->获取的数组对象

所创建数组的维度与参数容器的维度严格一致，每个数组都有一个shape属性，一个用于表示各维度大小的元组。



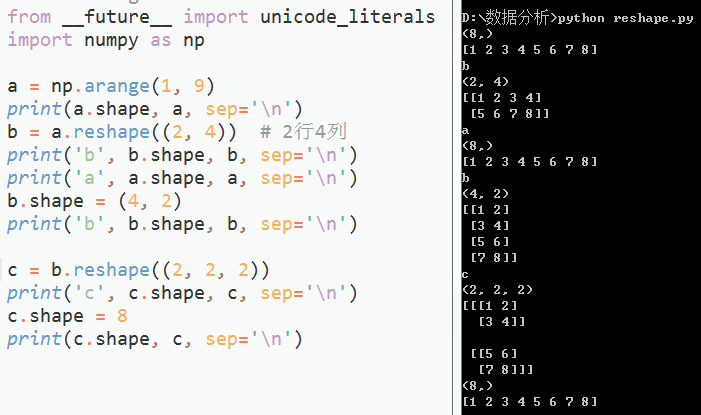
2.numpy中的数组可以使用基0的下标访问其元素。通过len()/numpy.ndarray.size获取元素个数们对于高纬数组，Python中的len()返回的是第一维数组。而size返回的是所有维数组的乘积



3.数组的维度可以在定义以后根据需要在合理的前提下作出调整：reshape（（新维度））/shape=新维度，

第一种方法并不会改变调用数组的维度，而是得到一个新的数组，

第二种方法直接修改了所操作的数组对象本身



4.ndarry对象的dtype属性反映了元素的数据类型，可以通过该属性读取或修改数组元素的数据类型。

数组对象.astype（新数据类型）->新数组对象

3 numpy的内置数据类型

1.布尔：bool8/bool\_

2.整型：int8/int16/int32/int64

intN：[-2^(N-1),2^(N-1)-1]

int8:[-128,127]

uint8/uint16/uint32/uint64 无符号整数

uintN：[0,2^N-1]

uint8：[0,255]

3.浮点型

float16/float32/float32/float64/float96/float128

特别：float96和float128在个别系统上才有

4.复数

complex64/complex128

5.字符串

str\_ 字符串就是一个unicode序列

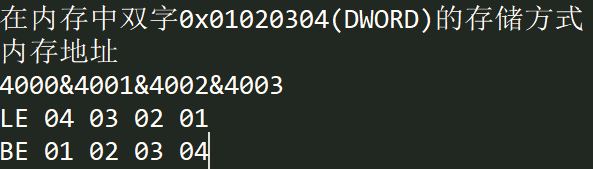
6.numpy的类型字符码

|  |  |
| --- | --- |
| ? | 布尔型 |
| i | 有符号整型 |
| u | 无符号整型 |
| f | 浮点型 |
| c | 复数 |
| u | 字符串 |
| M | 日期时间 |
| O | Python对象 |

7.numpy的类型字符串

<字节序><维度><类型字符码><字节数>

|  |  |
| --- | --- |
| > | 大端字节序 |
| < | 小端字节序 |
| = | 系统字节序 |



4 切片

数组[起始：终止：步长，起始：终止：步长，…] #在每个维度上使用切片

缺省起始：步长>0，首元素；步长<0，尾元素

缺省终止：步长>0，尾元素后；步长<0，首元素前

缺省步长：1

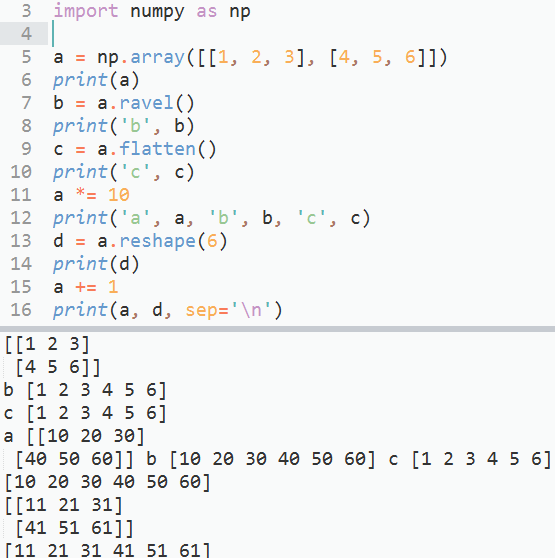
某几个维度全部使用

5 展平

任意维度->一维

1.a.ravel()返回a数组的一维视图（数据还是a数组的数据）

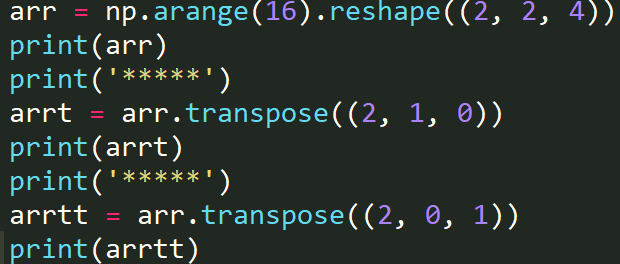
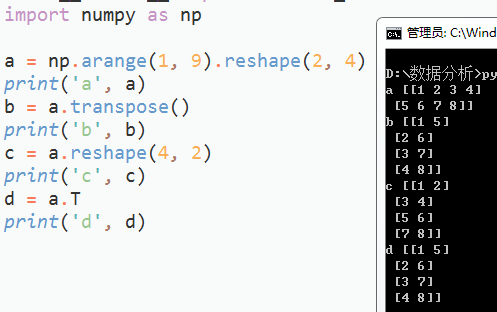
2.a.flatten()返回a数组的一维副本（flatten会请求分配内存来保存结果）



6 转置

|  |  |
| --- | --- |
| 数组.transpose() | 转置视图的方法 |
| 数组.T | 转置视图属性 |

至少是二维数组才支持转置



7 组合与拆分

|  |  |
| --- | --- |
| 垂直组合 | vstack((上，下)) |
| 垂直分割 | vsplit（数组，分个数）->子数组元组 |
| 水平组合 | hstack（（左，右）） |
| 深度组合 | dstack（（前，后）） |
| 深度分割 | dsplit（数组，分割数）->子数组元组 |
| 行组合 | row\_stack（（上，下）） |
| 列组合 | column\_stack（（左，右）） |

除了合并轴上的维度，其他维度必须相等

8 数组的属性

dtype 元素类型

shape 数组维度

T 转置视图

size 元素个数

ndim 数组维数

itemsize 每个元素字节数

nbytes 数组总字节数=size\*itemsize

real 实部数组

imag 虚部数组

flat 扁平迭代器

flat属性将返回一个numpy.flatiter对象

numpy.ndarry.tolist()->返回列表对象

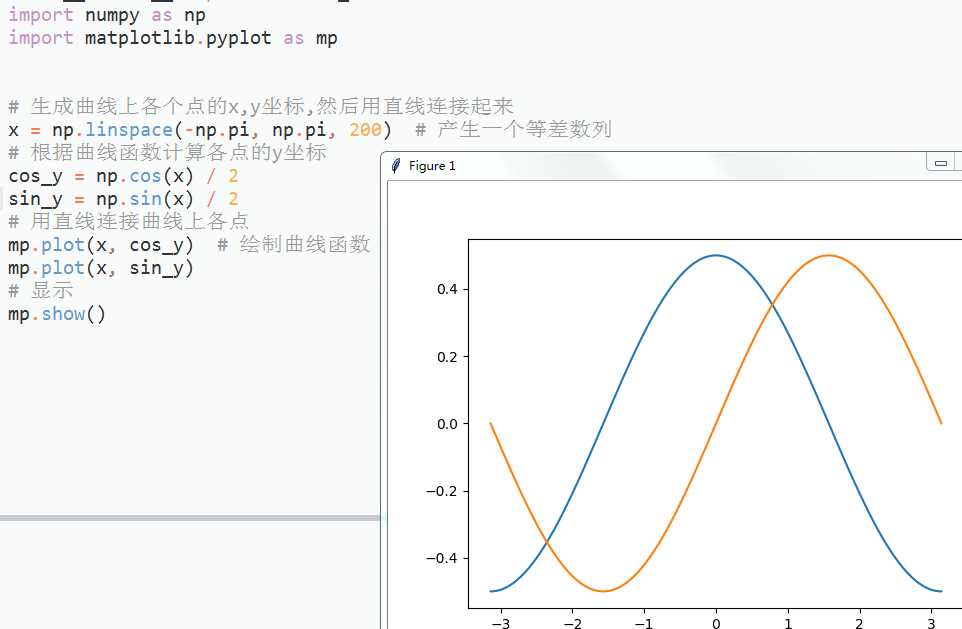
matplotlib September29

基于matplotlib的数据可视化，matplotlib.pyplot提供基于Python语言的绘图函数

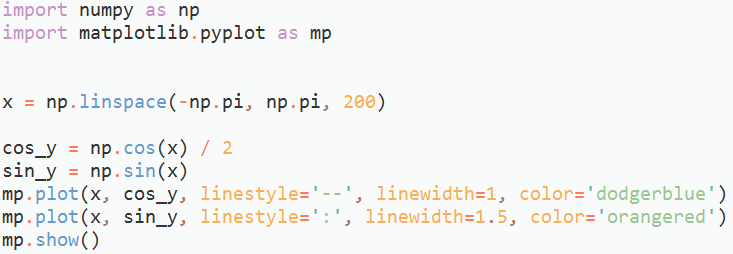
1 基本图形绘制

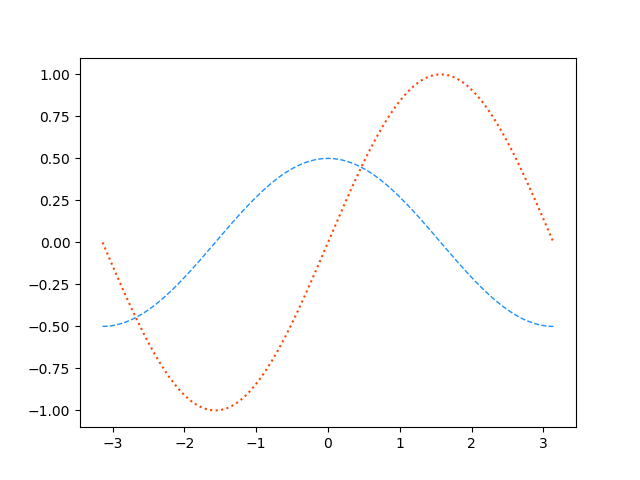
1.1缺省样式

mp.plot(x,y)



1.2设置线性、线宽和颜色

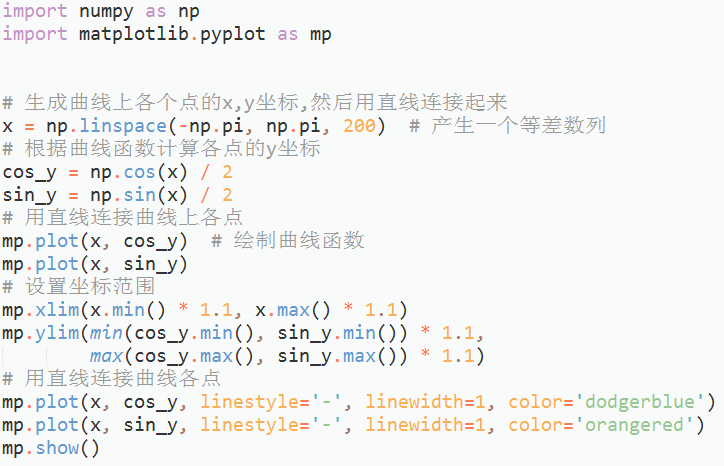




1.3设置坐标范围

设置水平坐标范围：mp.xlim（最小值，最大值）

设置垂直坐标范围：mp.ylim（最小值，最大值）



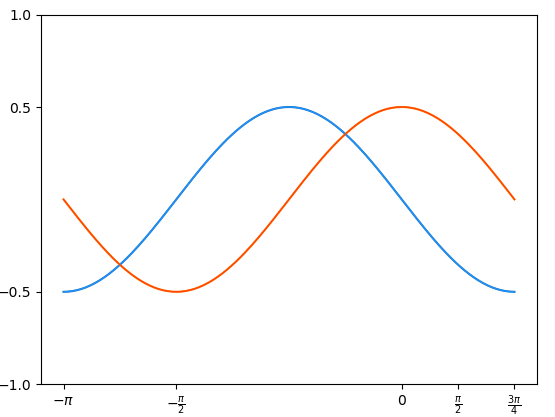
1.4设置坐标轴刻度标签

mp.xticks（刻度标签位置，刻度标签文本）

mp.yticks（刻度标签位置，刻度标签文本）

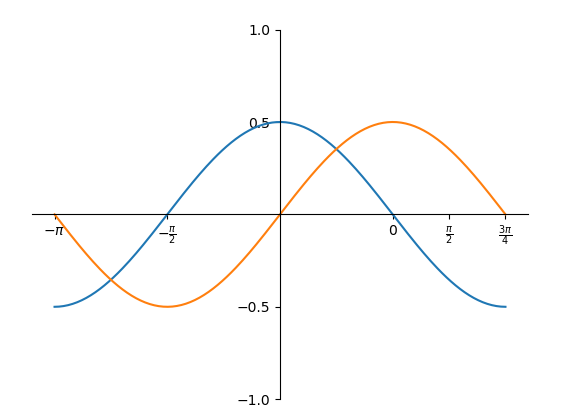
有几个文本位置就给几个文本

文本可以不给，默认值是刻度值



1.5将矩形坐标轴改为十字坐标轴



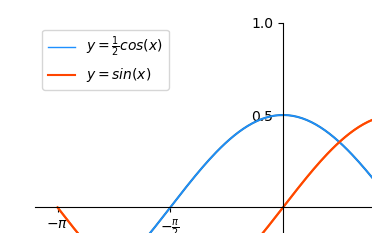


1.6图例显示

mp.plot（…,label=图例文本）

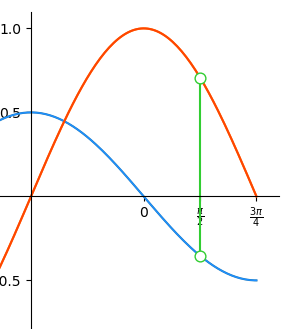
mp.legend(loc = 图例位置)





1.7添加特殊点

mp.scatter（点集水平坐标数组，点集垂直坐标数组，…）



1.8添加注释

mp.annotate(注释文本， #字符串

xy=被注释点的坐标， #加在哪个目标上

xycoords=被注释点的坐标属性， #一般用data,表示参考数据坐标系，而不是窗口坐标系

xytext=注释文本的坐标，

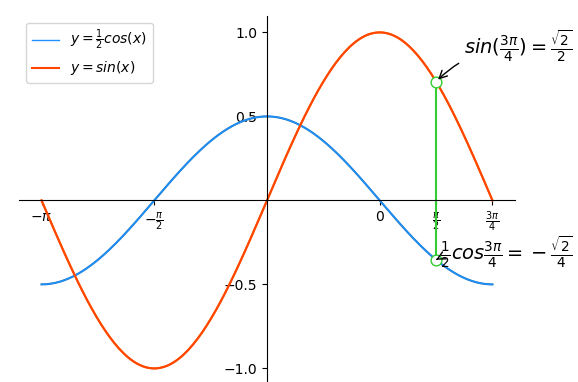
textcoords=注释文本坐标的属性，#绝对坐标或相对坐标

fontsize=字体大小，

arrowprops=dict(arrowstyle=箭头形状，connectionstyle=箭头连线的风格)

)





2 图形对象

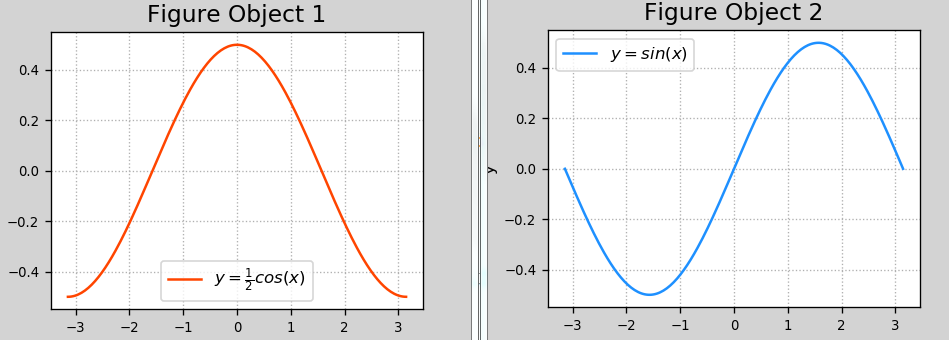
一个图形对象实际上就可以被看成窗口显示图形的窗口

函数：

mp.figure(图姓名，facecolor=背景色)

创建新的或获取已有的图形对象

图形名既可以作为显示在图形窗口标题栏中的文本，同时也是该对象的名称



3 子坐标图

mp.subplot(ABC)

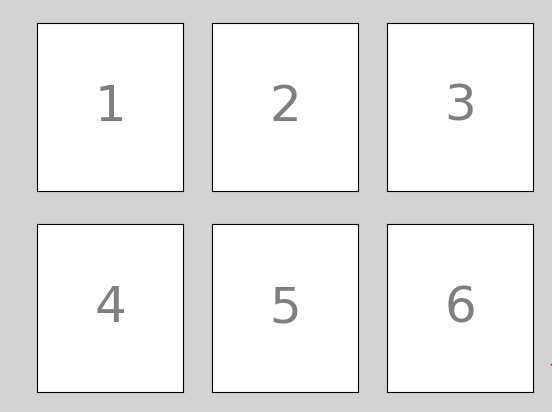
A：总行数

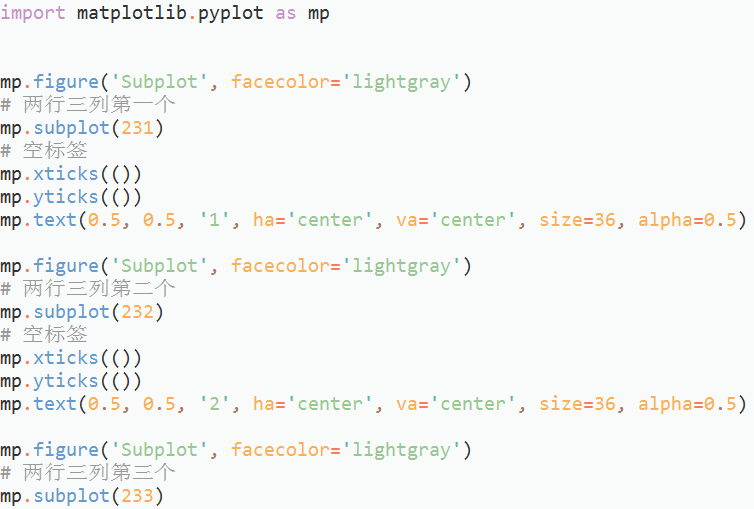
B：总列数

C：图形号（从1开始）

[231][232][233]

[234][235][236]





4 栅格布局

import matplotlib.gridspec as mg

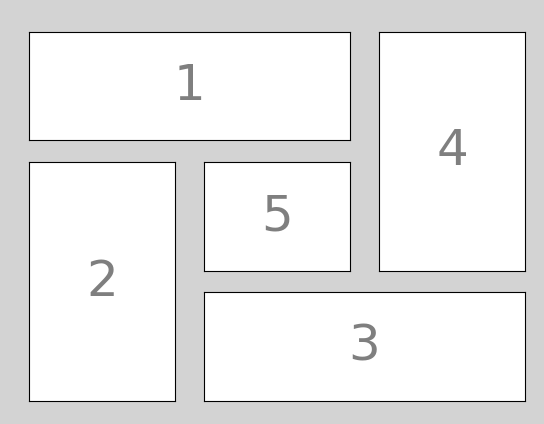
gs = mg.GridSpec(行数，列数)---》返回一个栅格布局对象

mp.subplot(gs[0,0])---》画在第0行，第0列

mp.subplot(gs[0,:2])---》第0行，第1列和第2列





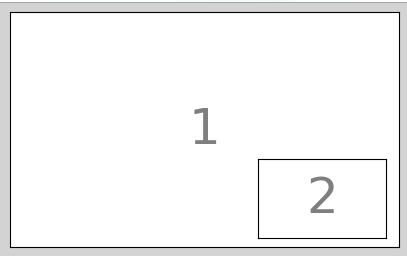


5 自由布局

mp.axes([左，底，宽，高])

#4个参数是相对于水平和垂直坐标的比例值，到左边框，到底边框的距离比，图形的宽度和高度





6 刻度定位器

ax=mp.gca()

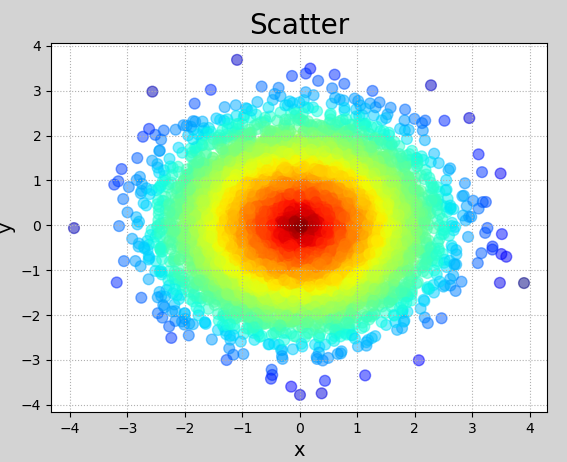
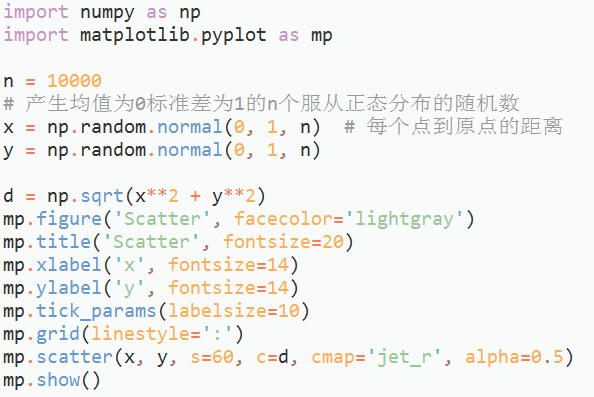
ax.xaxis…

ax.yaxis…

|  |  |
| --- | --- |
| set\_major\_locator() | 设置主刻度定位器 |
| set\_minor\_locarot() | 设置次刻度定位器 |
| NullLocator() | 空，不做刻度标记 |
| MaxNLocator() | 指定最多刻度数 |
| FixedLocator() | 由参数指定刻度 |
| AutoLocator() | 默认的，自动选择合理的刻度 |
| IndexLocator() | 根据偏移和增量定位刻度 |
| LogLocator() | 根据指定的底数和指数定位刻度 |

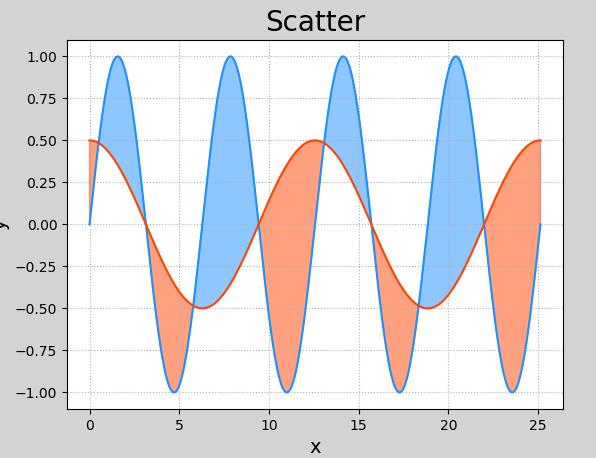
7 散点图

mp.scatter()



8 区域填充

mp.fill\_between（填充区域水平坐标，下限垂直坐标，上限垂直坐标，where=条件表达式，color=颜色，alpha=透明度）



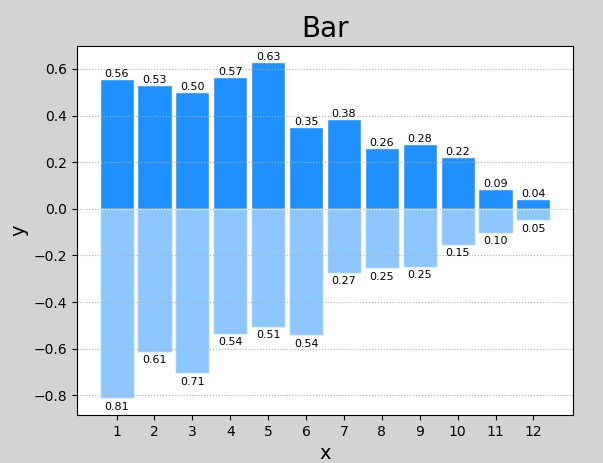


9 柱状图

mp.bar(水平坐标数组，高度坐标数组，宽度比例，ec=勾边色，fc=填充色，label=图例标签)

注意：

高度可以为负、宽度比例以柱子间不留空白



matplotlib September30

1 热图

1.1缺省样式

mp.imshow(z,cmap=颜色映射，origin=垂直轴向)

2 饼图

mp.pie（值数组，间隙数组，标签数组，颜色数组，标注格式，shadow=是否绘制阴影，startangle=起始角度）

每个扇形的值，扇形间的间隙，每个扇形的标签，扇形颜色，标注的字体格式，阴影，其实扇形偏离的角度

3 三维曲面图

ax=mp.gca(projection=’3d’)

#获取一个可以画3D效果的轴对象（坐标图对象）

ax.plot\_surface(x,y,z,rstride=行步距，cstride=列步距，cmap=颜色距)

4 三维线框图

ax=mp.gca(projection=’3d’)

ax.plot\_wireframe(x,y,z,rstride=行步距，cstride=列步距，color=线框颜色)

5 动画

FuncAnimation(图形对象，绘制函数，interval=时间间隔(ms))

def FuncAnimation():

…

for data in