Python与机器学习

——Python库的使用

华算科技 黄老师

2022年1月18日

目录

1. Python库介绍

2. NumPy库

3. pandas库

4. 案例：谱数据平滑

5. matplotlib库与数据可视化

目录

1. Python库介绍

2. NumPy库

3. pandas库

4. 案例：谱数据平滑

5. matplotlib库与数据可视化

库

库是完成一定功能的代码的集合

scikit learn

NumPy

pandas

tkinter

SciPy

matplotlib
Version 3.4.3

Keras

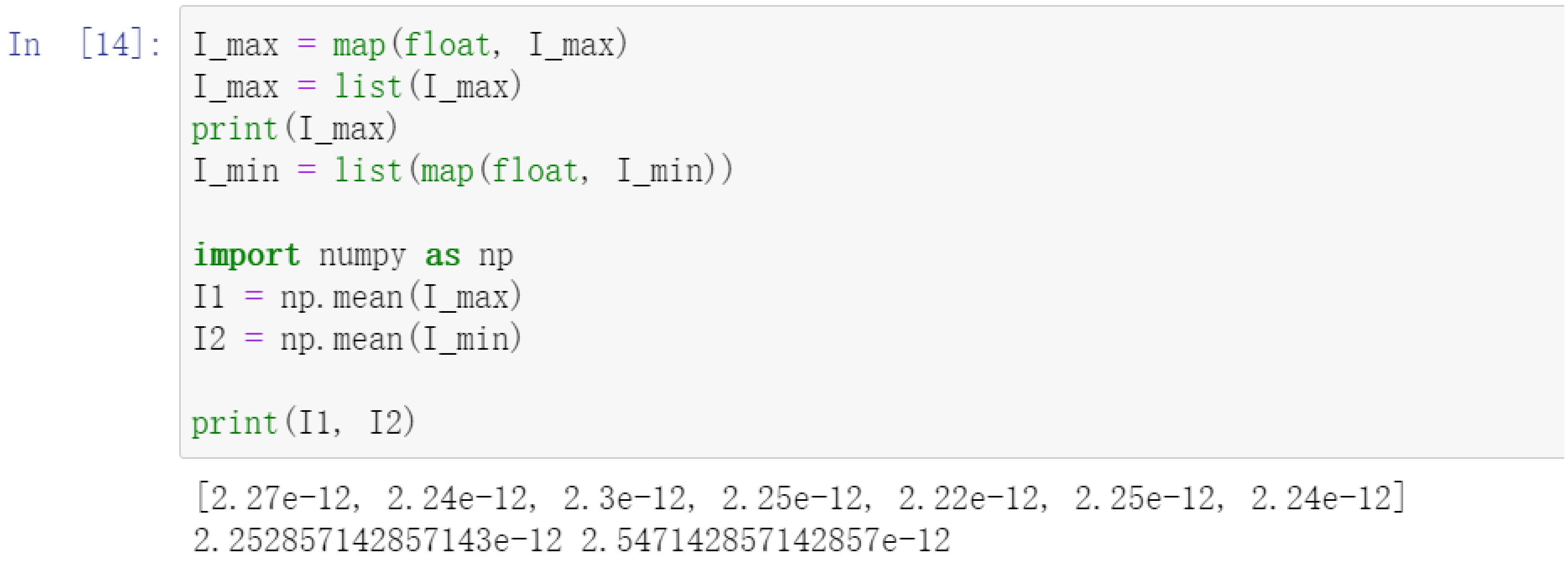
pymatgen

matminer

库的优势

提高开发效率

例如：前面的求list平均

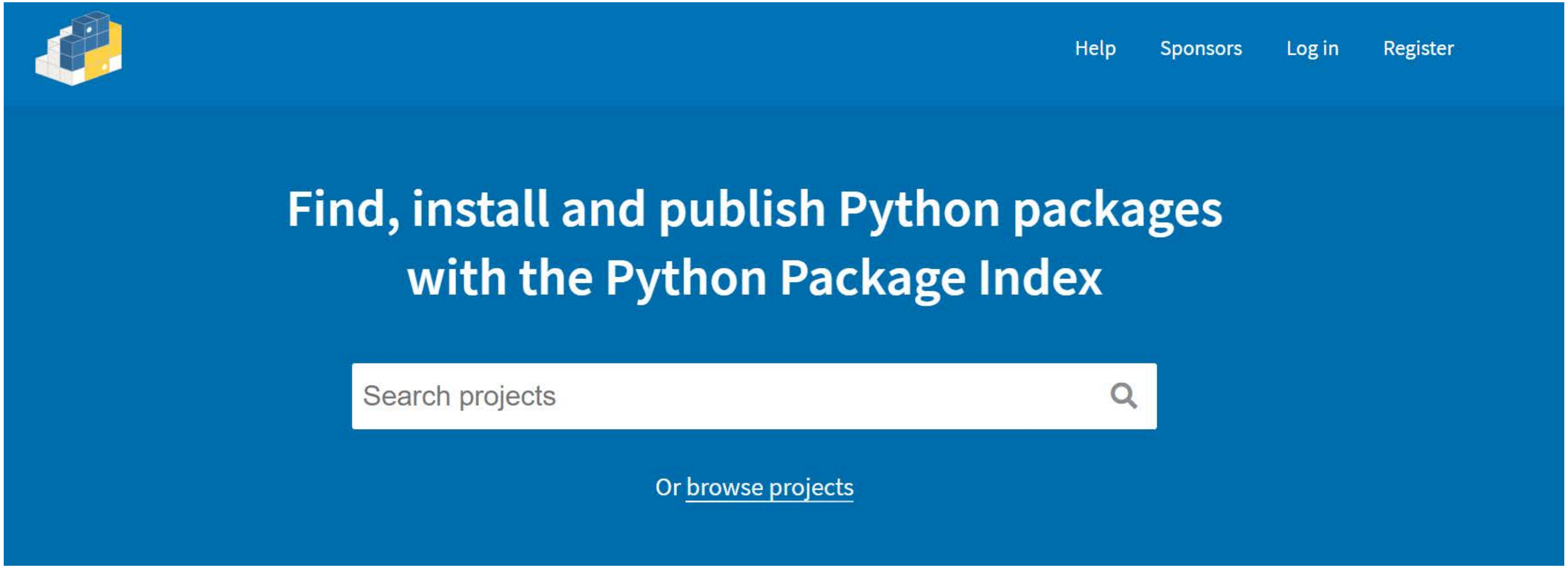


```
In [14]: I_max = map(float, I_max)
         I_max = list(I_max)
         print(I_max)
         I_min = list(map(float, I_min))

         import numpy as np
         I1 = np.mean(I_max)
         I2 = np.mean(I_min)

         print(I1, I2)

         [2.27e-12, 2.24e-12, 2.3e-12, 2.25e-12, 2.22e-12, 2.25e-12, 2.24e-12]
         2.252857142857143e-12 2.547142857142857e-12
```

库的安装

pip install xxx命令

Python库官方索引网站：PyPI(Python Package Index)   https://pypi.org



Find, install and publish Python packages with the Python Package Index

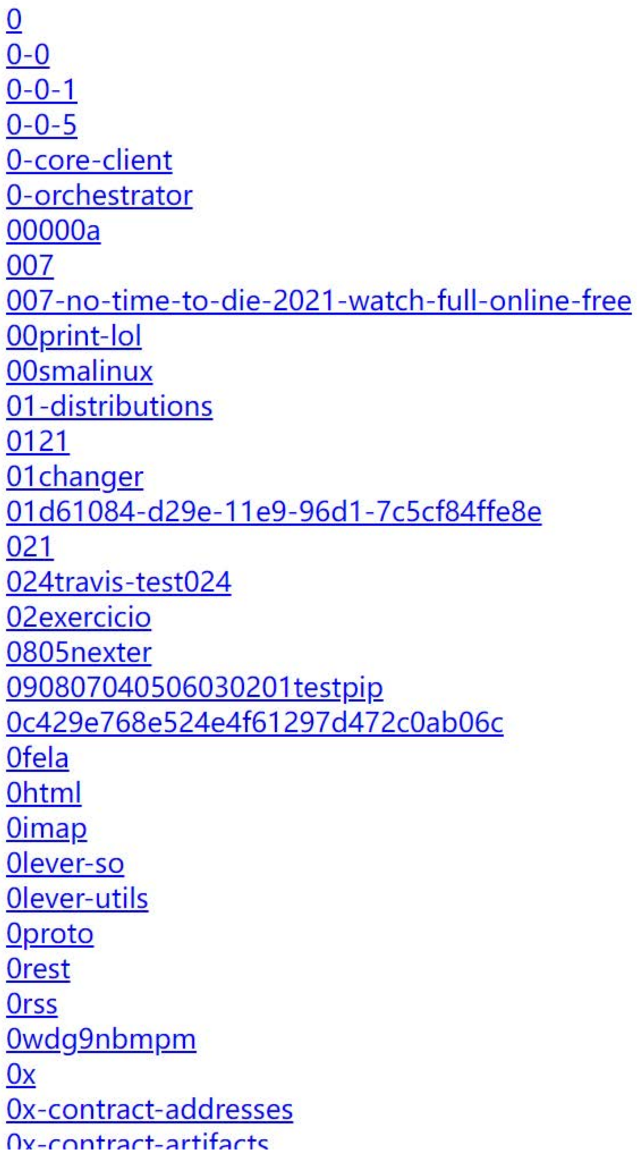
镜像源

清华：https://pypi.tuna.tsinghua.edu.cn/simple

阿里云：http://mirrors.aliyun.com/pypi/simple

中国科技大学 https://pypi.mirrors.ustc.edu.cn/simple

豆瓣：http://pypi.douban.com/simple



0
0-0
0-0-1
0-0-5
0-core-client
0-orchestrator
00000a
007
007-no-time-to-die-2021-watch-full-online-free
00print-lol
00smalinux
01-distributions
0121
01changer
01d61084-d29e-11e9-96d1-7c5cf84ffe8e
021
024travis-test024
02exercicio
0805nexter
090807040506030201testpip
0c429e768e524e4f61297d472c0ab06c
0fela
0html
0imap
0lever-so
0lever-utils
0proto
0rest
0rss
0wdg9nbmpm
0x
0x-contract-addresses
0x-contract-artifacts

库的导入

1. import 语句

使用方法：import A　　(导入A模块，例如导入numpy模块，import numpy)

可添加别名，例如 import numpy as np，程序中则可使用np代表numpy

2. from ... import ... 语句

使用方法：from A import a1　(在内存中创建并加载A模块中a1工具的副本，例如导入numpy模块中的zeros函数，from numpy import zeros)

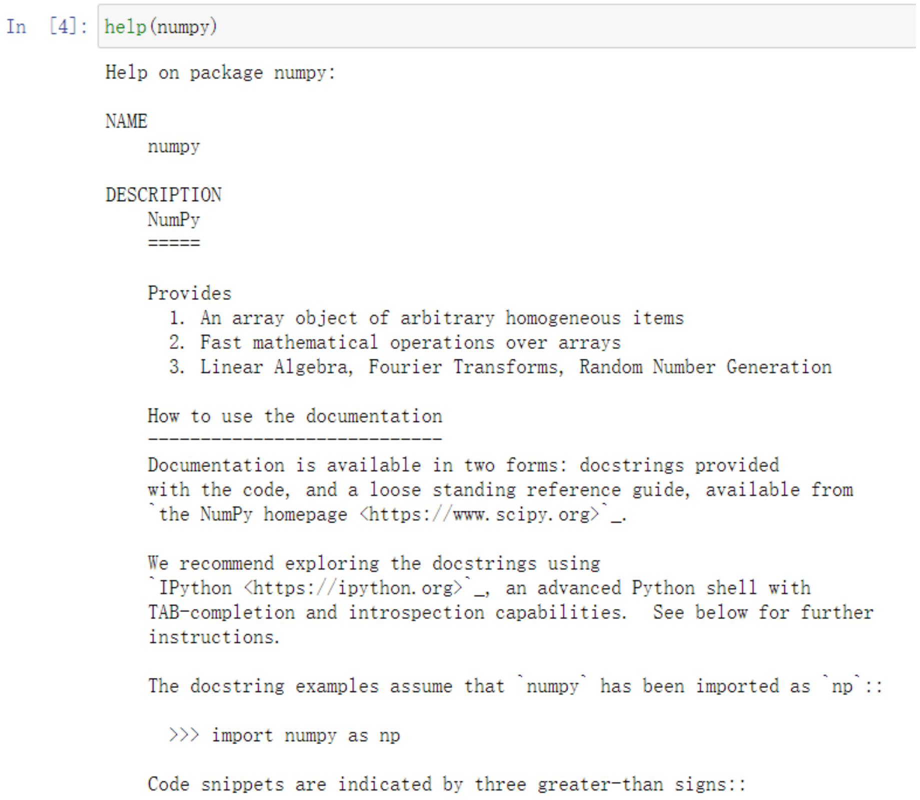
与import A.a1的区别，前者可直接调用，后者只能使用全名

查看帮助文件

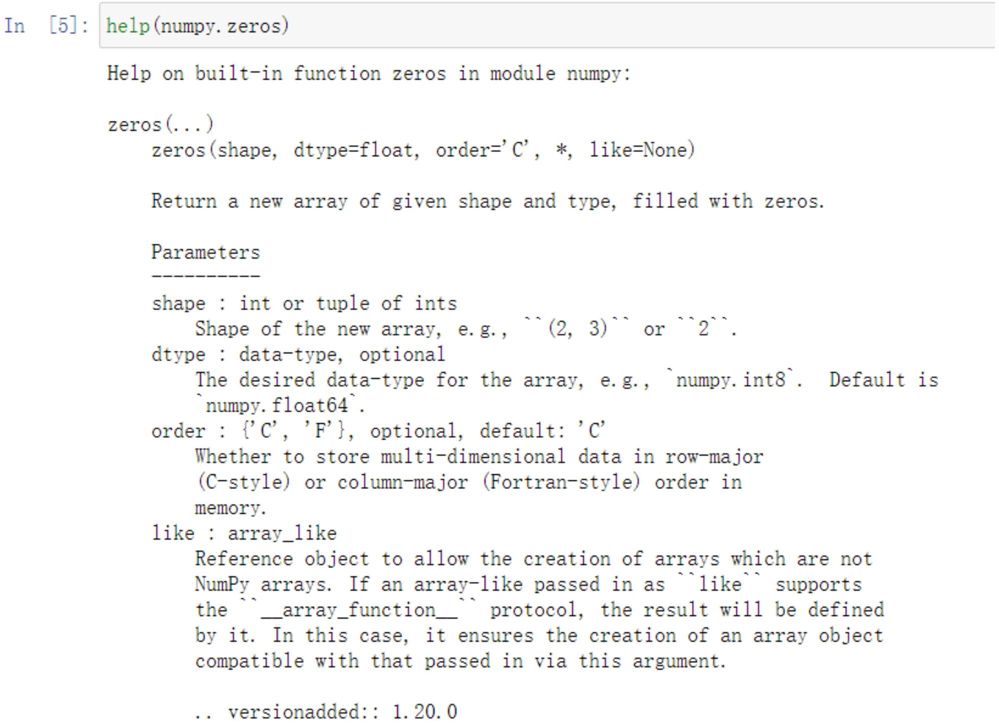
help 语句

使用方法：help(A)　（查看A模块下的所有函数，help(numpy)）

help(A.a1)　（查看A模块中的a函数，help(numpy.zero)）



```
In  [4]: help(numpy)

Help on package numpy:

NAME
    numpy

DESCRIPTION
    NumPy
    =====

    Provides
      1. An array object of arbitrary homogeneous items
      2. Fast mathematical operations over arrays
      3. Linear Algebra, Fourier Transforms, Random Number Generation

    How to use the documentation
    ----------------------------
    Documentation is available in two forms: docstrings provided
    with the code, and a loose standing reference guide, available from
    `the NumPy homepage <https://www.scipy.org>`_.

    We recommend exploring the docstrings using
    `IPython <https://ipython.org>`_, an advanced Python shell with
    TAB-completion and introspection capabilities.  See below for further
    instructions.

    The docstring examples assume that `numpy` has been imported as `np`::

      >>> import numpy as np

    Code snippets are indicated by three greater-than signs::
```



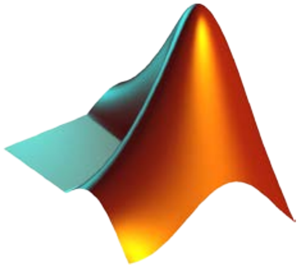
```
In  [5]: help(numpy.zeros)

Help on built-in function zeros in module numpy:

zeros(...)
    zeros(shape, dtype=float, order='C', *, like=None)

    Return a new array of given shape and type, filled with zeros.

    Parameters
    ----------
    shape : int or tuple of ints
        Shape of the new array, e.g., ``(2, 3)`` or ``2``.
    dtype : data-type, optional
        The desired data-type for the array, e.g., `numpy.int8`.  Default is
        `numpy.float64`.
    order : {'C', 'F'}, optional, default: 'C'
        Whether to store multi-dimensional data in row-major
        (C-style) or column-major (Fortran-style) order in
        memory.
    like : array_like
        Reference object to allow the creation of arrays which are not
        NumPy arrays. If an array-like passed in as ``like`` supports
        the ``__array_function__`` protocol, the result will be defined
        by it. In this case, it ensures the creation of an array object
        compatible with that passed in via this argument.

        .. versionadded:: 1.20.0
```

科学计算库

NumPy，SciPy，matplotlib的组合，广泛用于替代MATLAB

最优化、线性代数等科学计算

可视化，图像输出

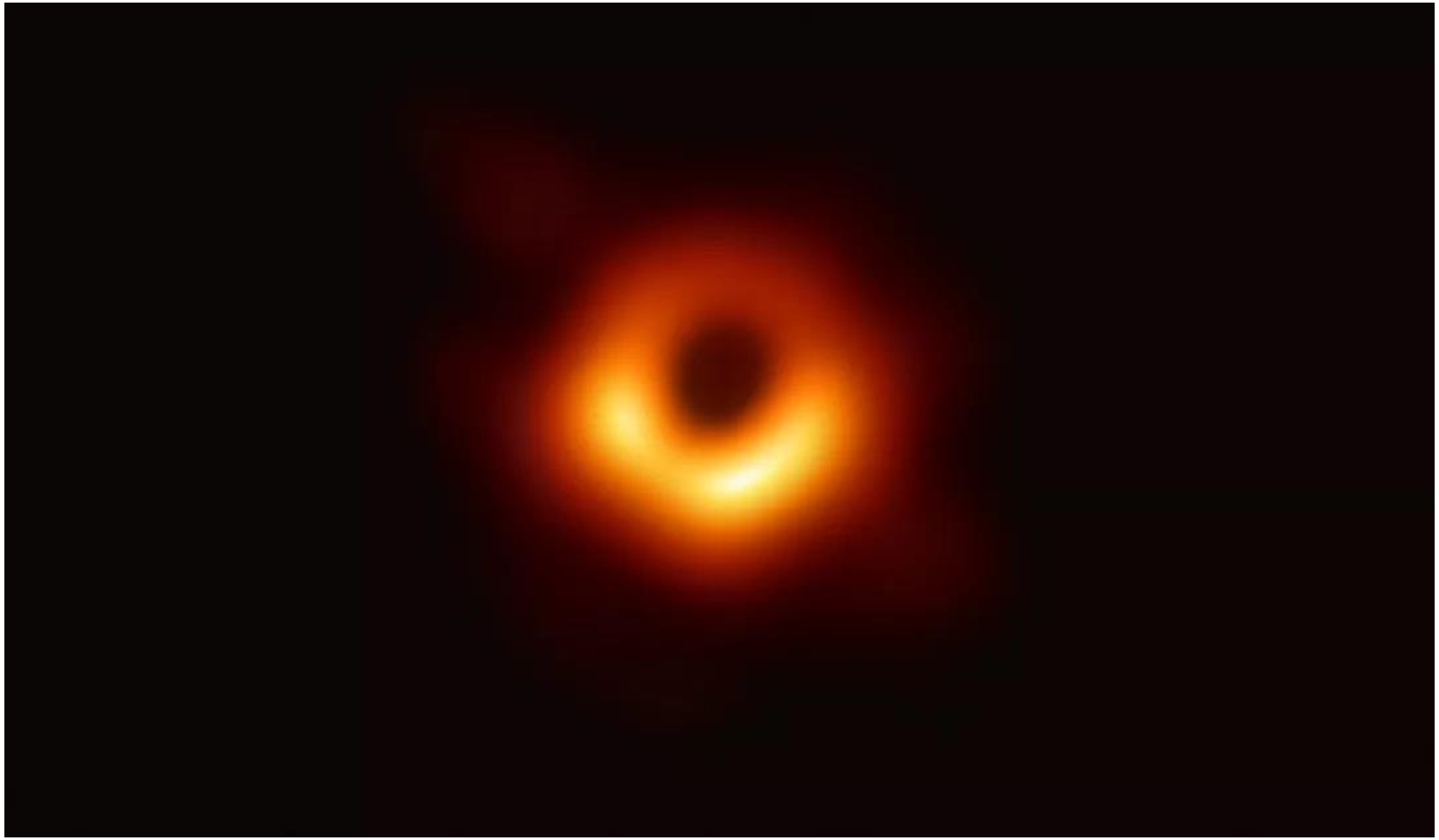


MATLAB

黑洞照片

2019年，人类首次拍到了黑洞的照片

https://github.com/achael/eht-imaging



目录

1. Python库介绍

2. NumPy库

3. pandas库

4. 案例：谱数据平滑

5. matplotlib库与数据可视化

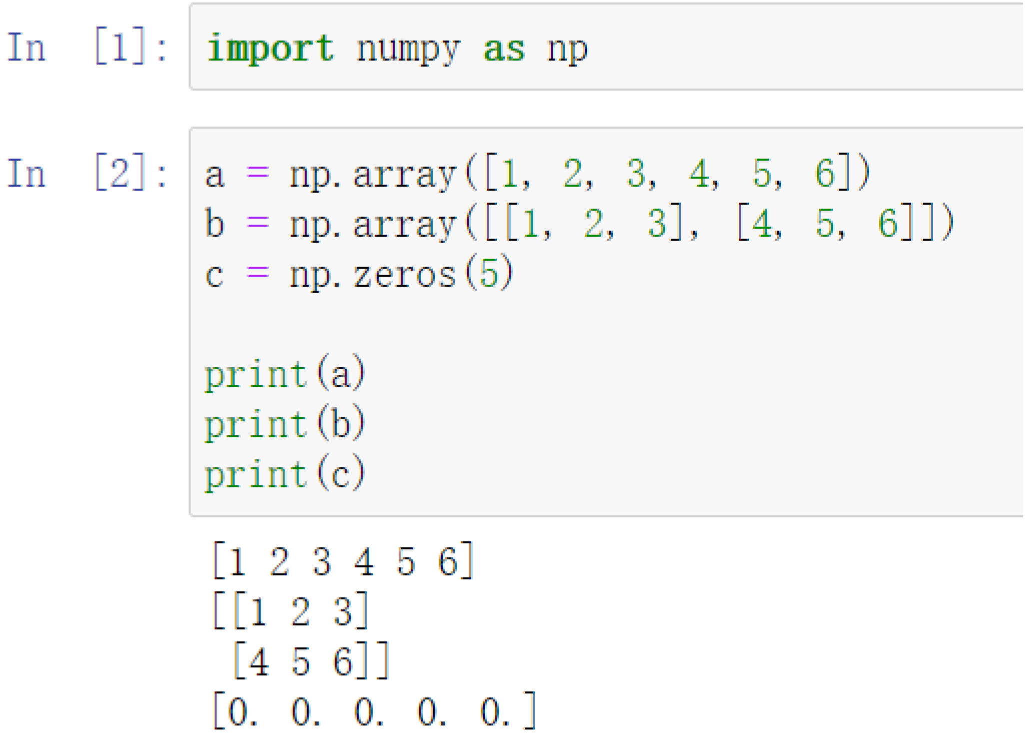
NumPy库的使用

NumPy中封装有array（数组），其结构与Python中自带的list相似，但在科学计算中更为方便与强大。

list = [3.3, 2.2, 2.5, 3.1, 3.0, 'eV']

a = np.array([3.3, 2.2, 2.5, 3.1, 3.0])

array中存放的数据类型相同



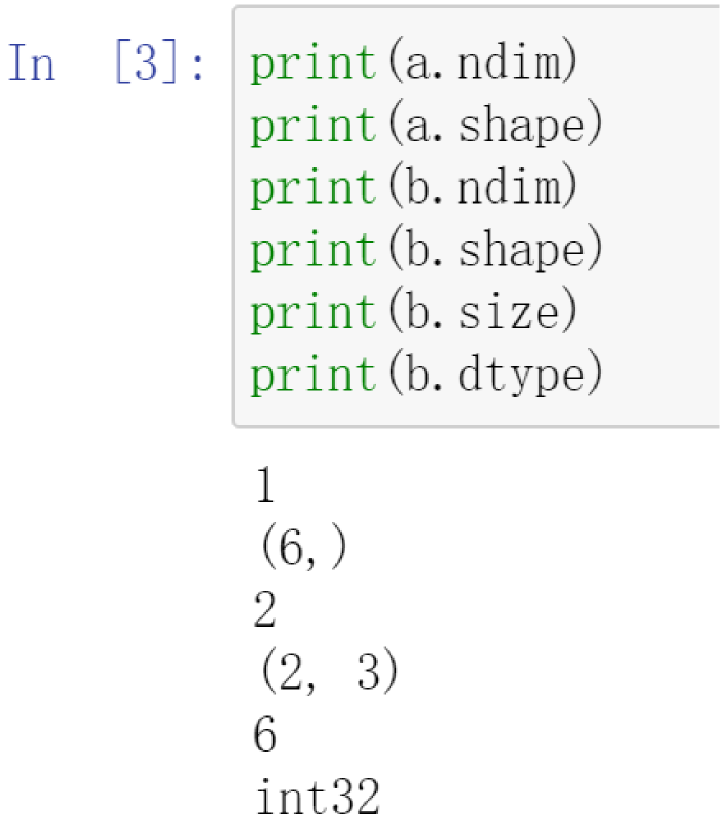
```
In [1]: import numpy as np
```

```
In [2]: a = np.array([1, 2, 3, 4, 5, 6])
        b = np.array([[1, 2, 3], [4, 5, 6]])
        c = np.zeros(5)

        print(a)
        print(b)
        print(c)
```

```
[1 2 3 4 5 6]
[[1 2 3]
 [4 5 6]]
[0. 0. 0. 0. 0.]
```

NumPy库的使用

数组属性



```
In  [3]: print(a.ndim)
         print(a.shape)
         print(b.ndim)
         print(b.shape)
         print(b.size)
         print(b.dtype)

1
(6,)
2
(2, 3)
6
int32
```

ndim：数组维度

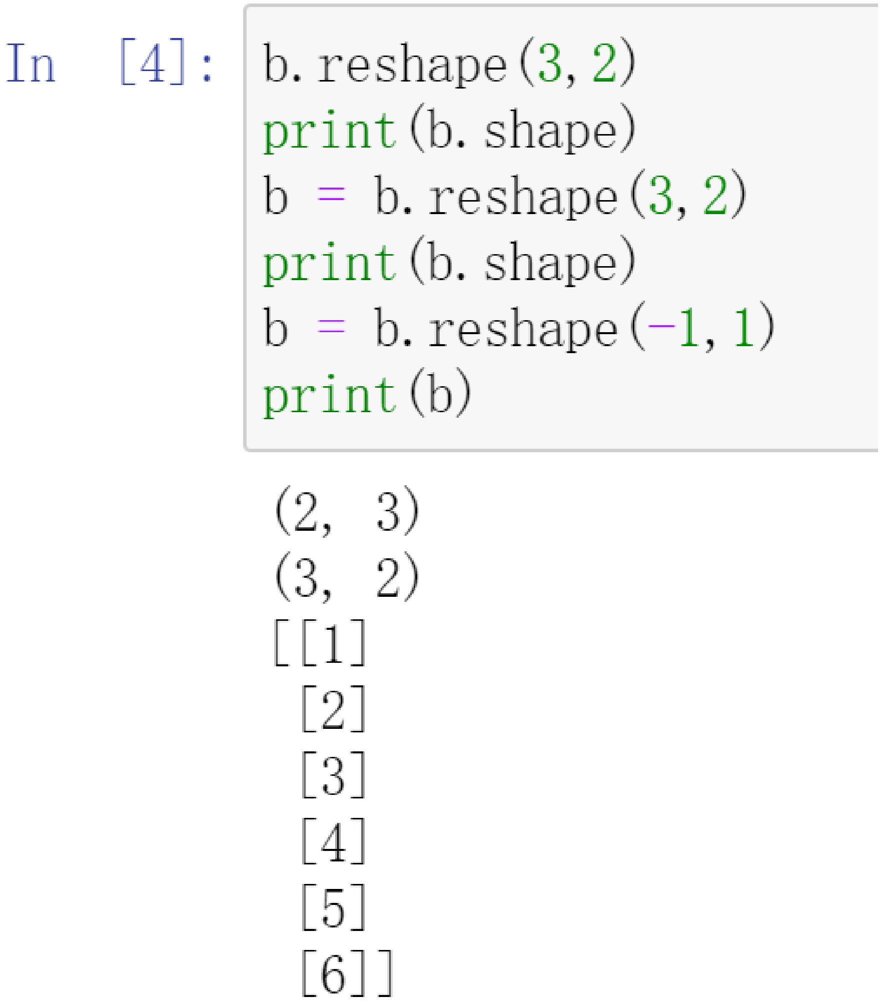
shape：n行m列

size：总元素个数

dtype：数据类型

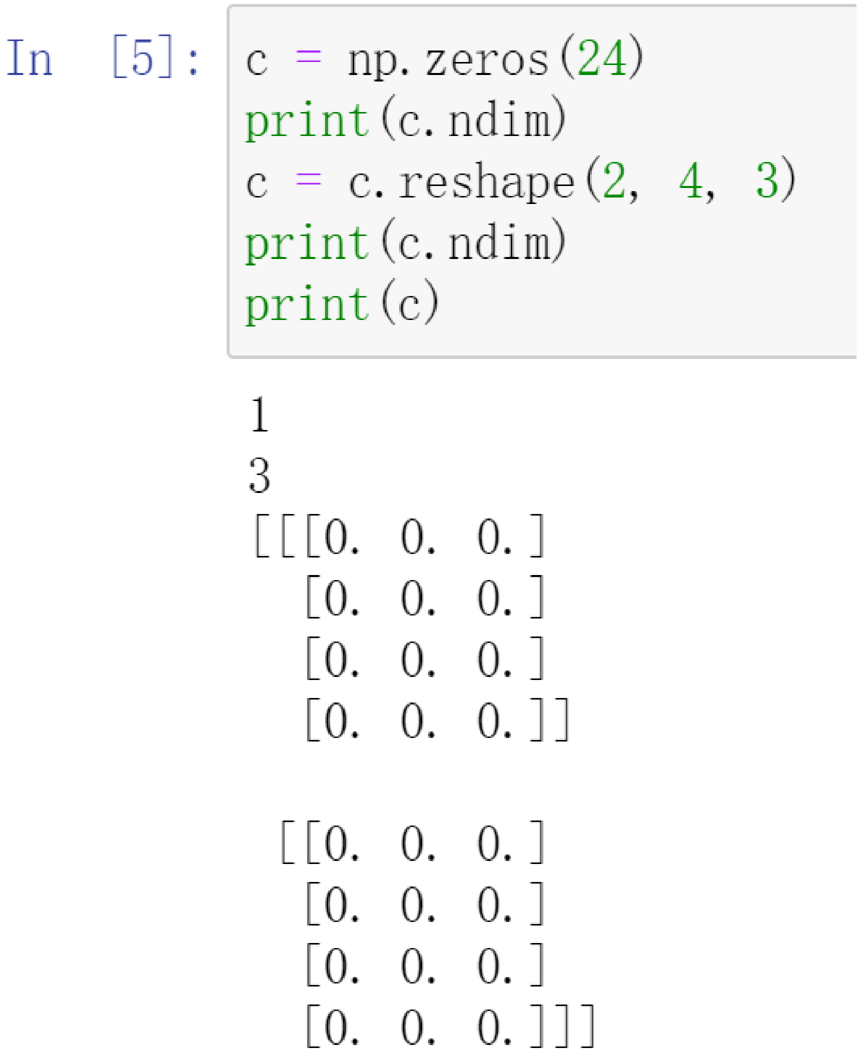
NumPy库的使用

维度与形状调整



```
In [4]: b.reshape(3,2)
        print(b.shape)
        b = b.reshape(3,2)
        print(b.shape)
        b = b.reshape(-1,1)
        print(b)
```

```
(2, 3)
(3, 2)
[[1]
 [2]
 [3]
 [4]
 [5]
 [6]]
```



```
In [5]: c = np.zeros(24)
        print(c.ndim)
        c = c.reshape(2, 4, 3)
        print(c.ndim)
        print(c)
```

```
1
3
[[[0. 0. 0.]
  [0. 0. 0.]
  [0. 0. 0.]
  [0. 0. 0.]]

 [[0. 0. 0.]
  [0. 0. 0.]
  [0. 0. 0.]
  [0. 0. 0.]]]
```

目录

1. Python库介绍

2. NumPy库

3. pandas库

4. 案例：谱数据平滑

5. matplotlib库与数据可视化

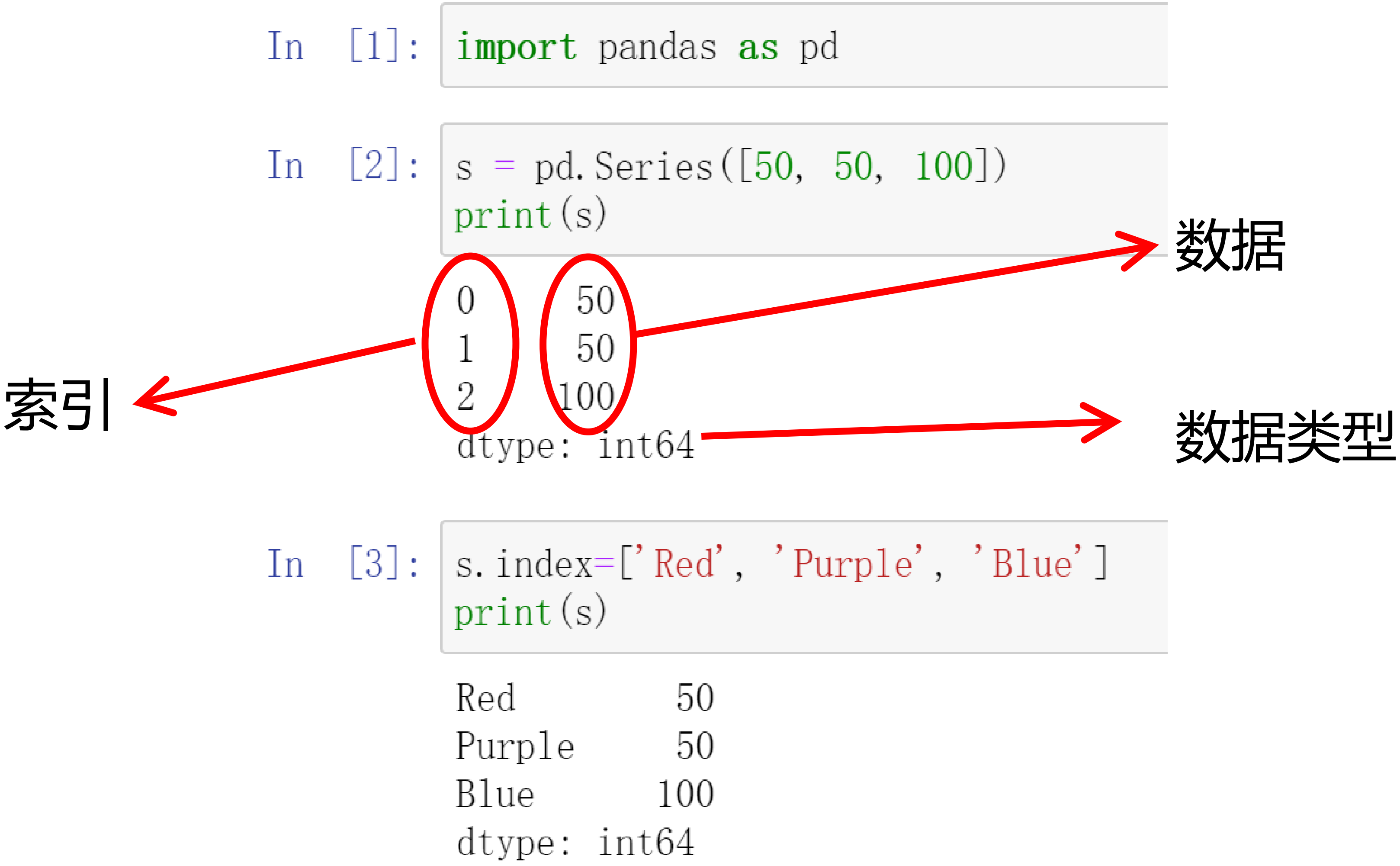
pandas数据结构

Series

DataFrame

Series

Series:　　由一组数据以及一组与之相关的标签构成



```
In  [1]: import pandas as pd
```

```
In  [2]: s = pd.Series([50, 50, 100])
         print(s)
```

```
0     50
1     50
2    100
dtype: int64
```

索引

数据

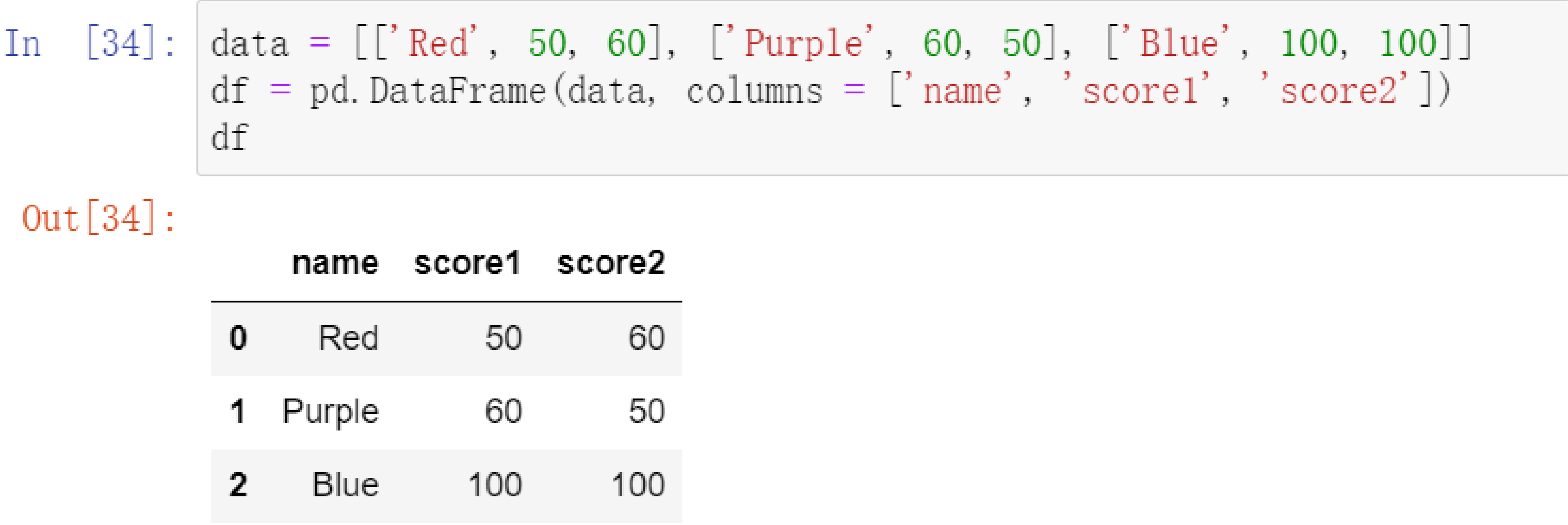
数据类型

```
In  [3]: s.index=['Red', 'Purple', 'Blue']
         print(s)
```

```
Red        50
Purple     50
Blue      100
dtype: int64
```

DataFrame

DataFrame:　　由若干个共用一个索引的Series组成



In [34]:

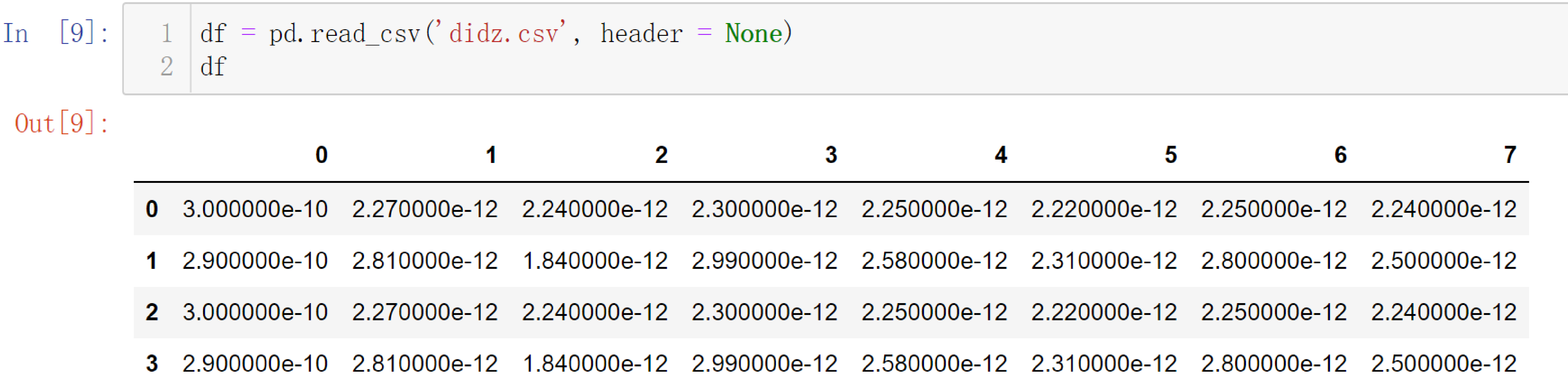
```python
data = [['Red', 50, 60], ['Purple', 60, 50], ['Blue', 100, 100]]
df = pd.DataFrame(data, columns = ['name', 'score1', 'score2'])
df
```

Out[34]:

	name	score1	score2
0	Red	50	60
1	Purple	60	50
2	Blue	100	100

pandas读取csv

read_csv()函数　　把csv文件读入DataFrame中



```
In [9]:  df = pd.read_csv('didz.csv', header = None)
         df
```

Out[9]:

	0	1	2	3	4	5	6	7
0	3.000000e-10	2.270000e-12	2.240000e-12	2.300000e-12	2.250000e-12	2.220000e-12	2.250000e-12	2.240000e-12
1	2.900000e-10	2.810000e-12	1.840000e-12	2.990000e-12	2.580000e-12	2.310000e-12	2.800000e-12	2.500000e-12
2	3.000000e-10	2.270000e-12	2.240000e-12	2.300000e-12	2.250000e-12	2.220000e-12	2.250000e-12	2.240000e-12
3	2.900000e-10	2.810000e-12	1.840000e-12	2.990000e-12	2.580000e-12	2.310000e-12	2.800000e-12	2.500000e-12

目录

1. Python库介绍

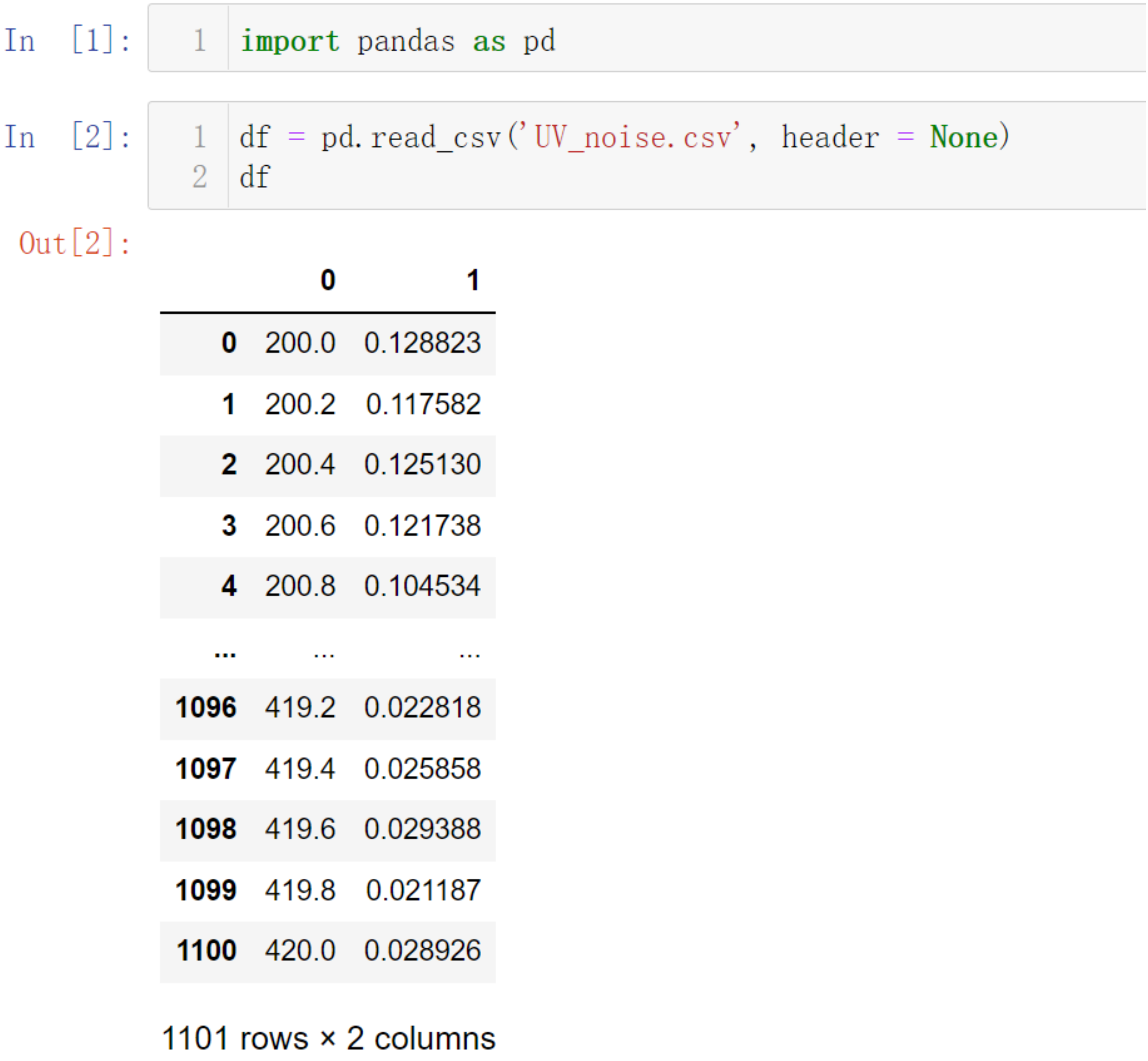
2. NumPy库

3. pandas库

4. 案例：谱数据平滑

5. matplotlib库与数据可视化

实操：谱图平滑

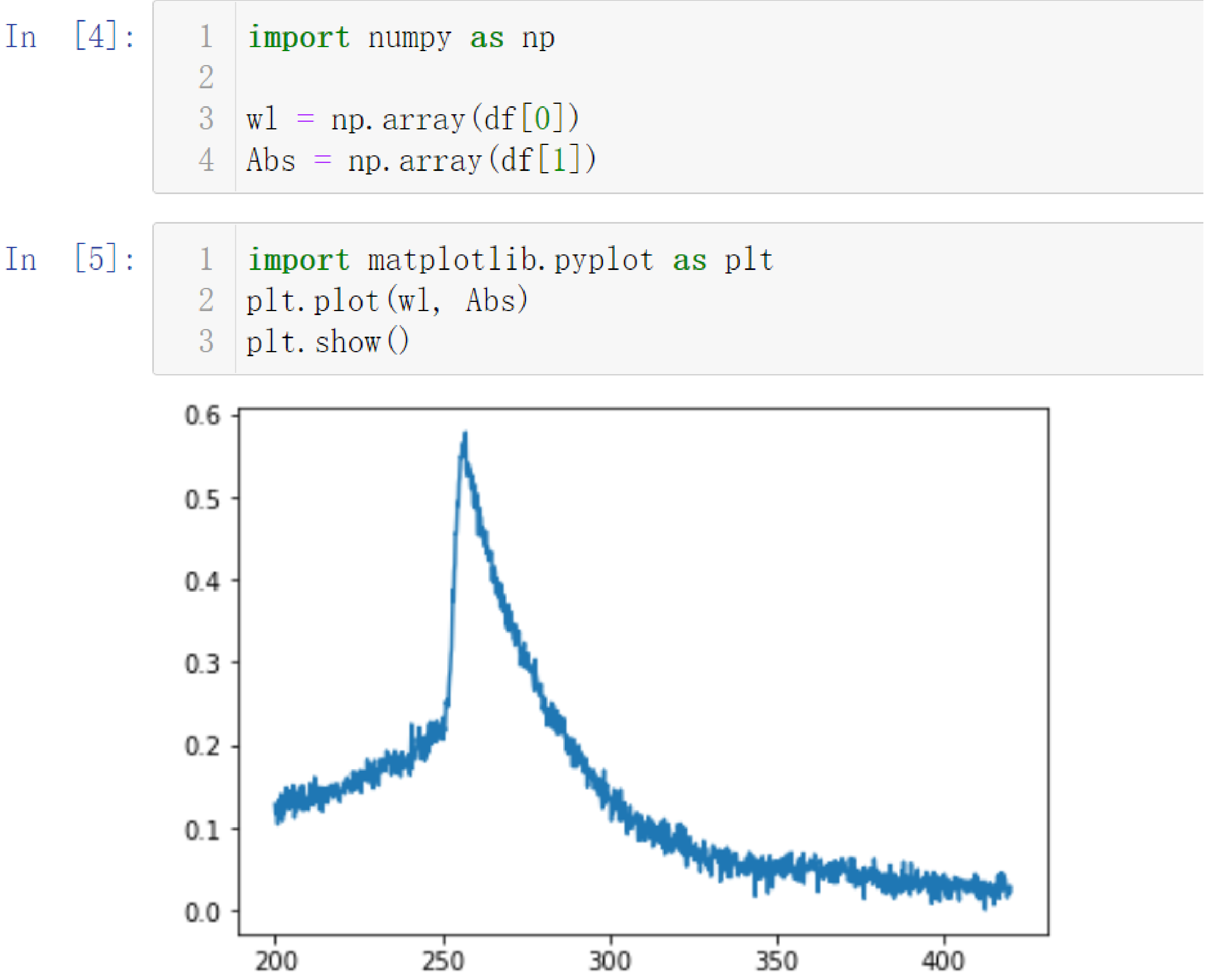


```
In  [1]:  import pandas as pd
```

```
In  [2]:  df = pd.read_csv('UV_noise.csv', header = None)
          df
```

```
Out[2]:
            0          1
   0    200.0   0.128823
   1    200.2   0.117582
   2    200.4   0.125130
   3    200.6   0.121738
   4    200.8   0.104534
 ...      ...        ...
1096    419.2   0.022818
1097    419.4   0.025858
1098    419.6   0.029388
1099    419.8   0.021187
1100    420.0   0.028926

1101 rows × 2 columns
```

实操：谱图平滑



In [4]:

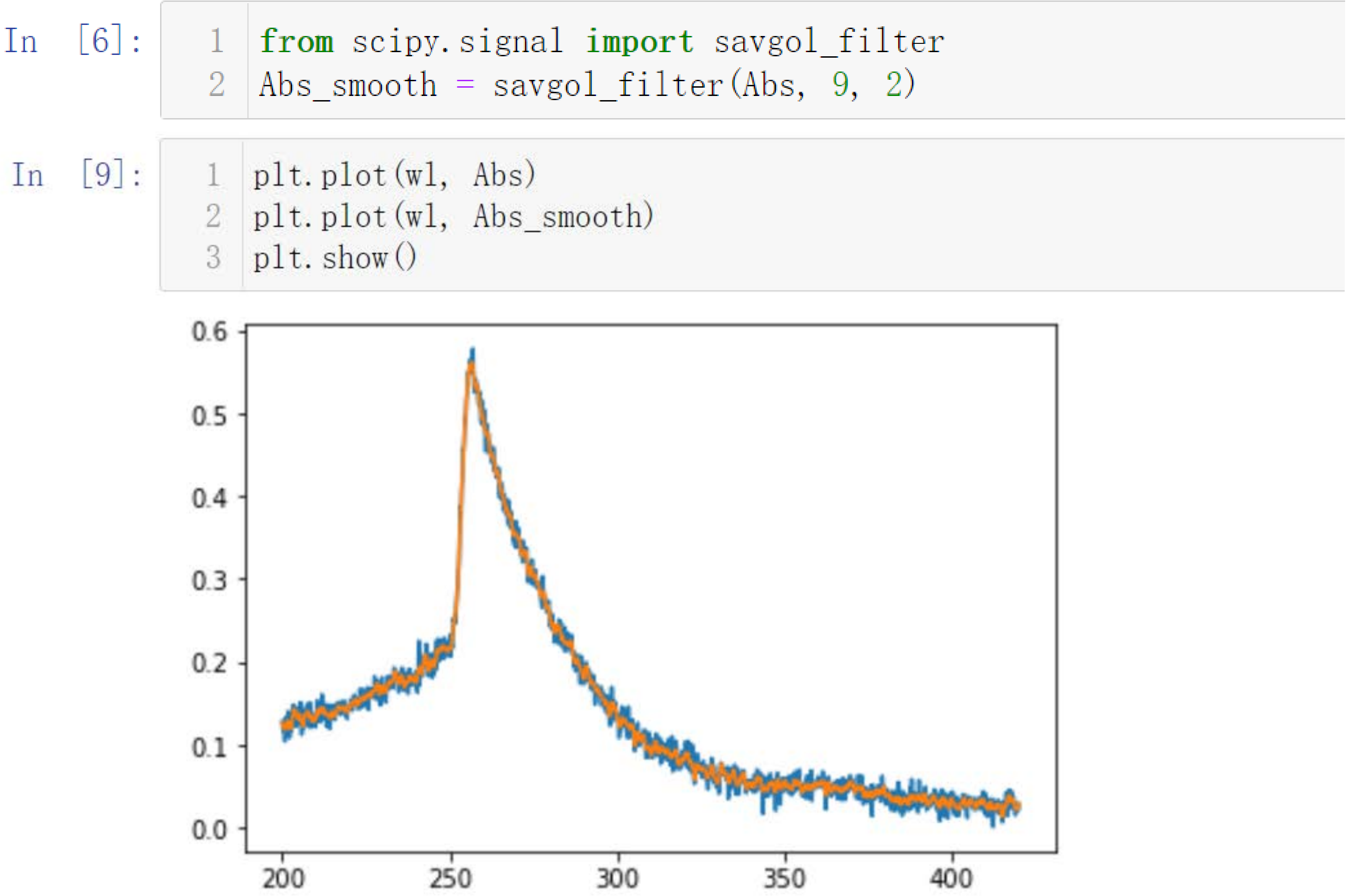
```python
import numpy as np

wl = np.array(df[0])
Abs = np.array(df[1])
```

In [5]:

```python
import matplotlib.pyplot as plt
plt.plot(wl, Abs)
plt.show()
```

实操：谱图平滑

Savitzky-Golay平滑法



In [6]:

```
from scipy.signal import savgol_filter
Abs_smooth = savgol_filter(Abs, 9, 2)
```

In [9]:

```
plt.plot(wl, Abs)
plt.plot(wl, Abs_smooth)
plt.show()
```

目录

1. Python库介绍

2. NumPy库

3. pandas库

4. 案例：谱数据平滑

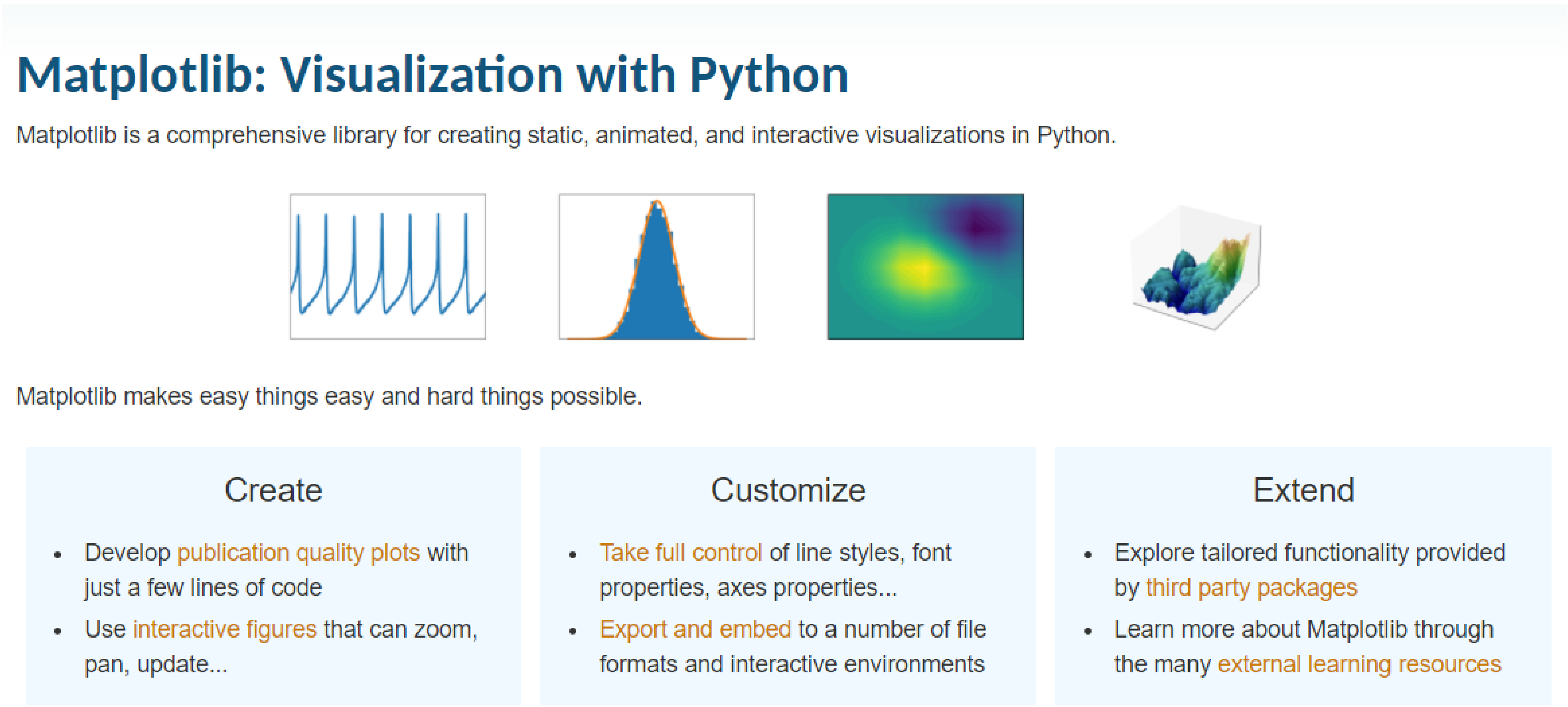
5. matplotlib库与数据可视化

可视化

matplotlib
Version 3.4.3

matplotlib库

https://matplotlib.org



Matplotlib: Visualization with Python

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Matplotlib makes easy things easy and hard things possible.

Create

- Develop publication quality plots with just a few lines of code
- Use interactive figures that can zoom, pan, update...

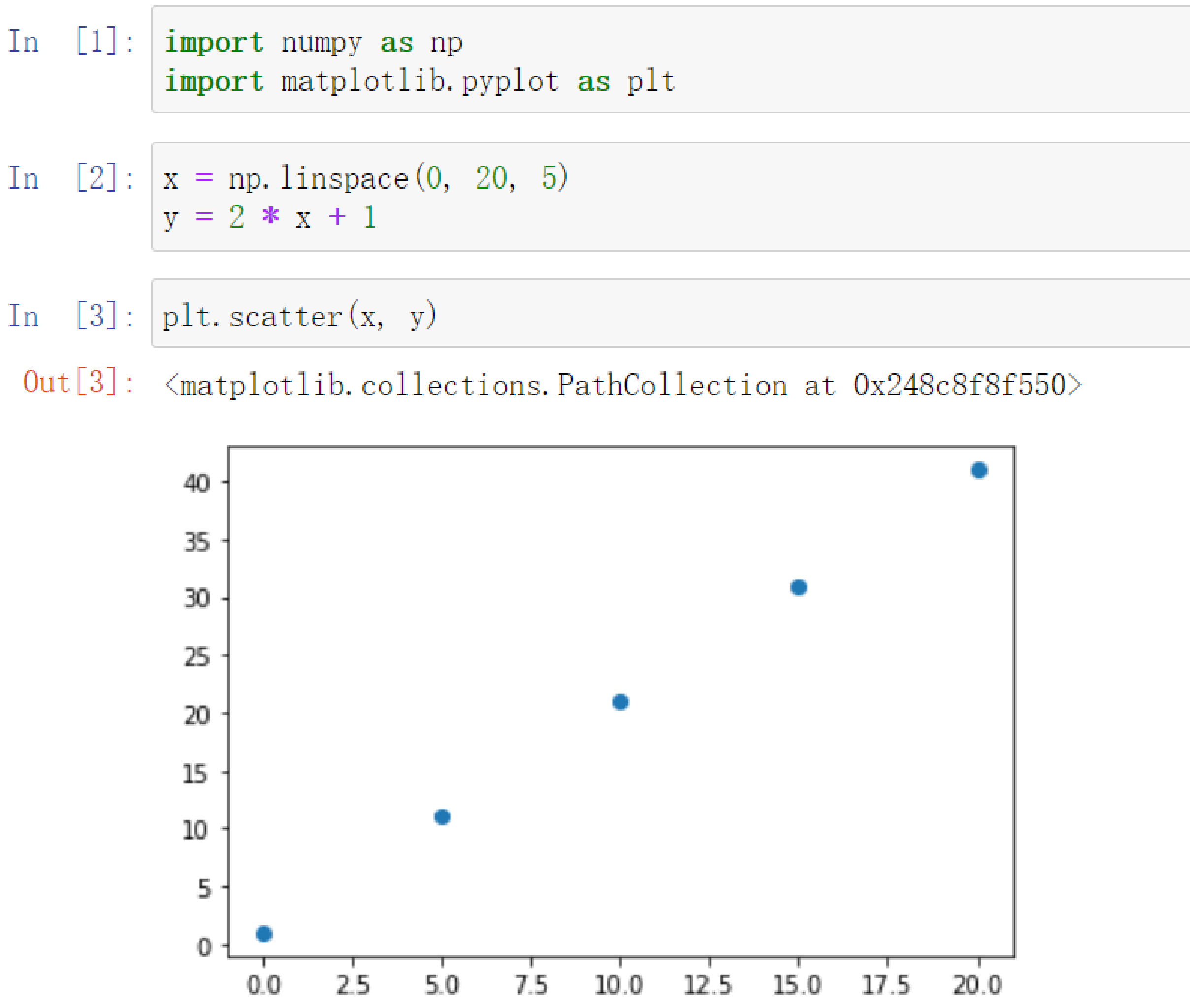
Customize

- Take full control of line styles, font properties, axes properties...
- Export and embed to a number of file formats and interactive environments

Extend

- Explore tailored functionality provided by third party packages
- Learn more about Matplotlib through the many external learning resources

实操：散点图的绘制



```
In  [1]: import numpy as np
         import matplotlib.pyplot as plt

In  [2]: x = np.linspace(0, 20, 5)
         y = 2 * x + 1

In  [3]: plt.scatter(x, y)

Out[3]: <matplotlib.collections.PathCollection at 0x248c8f8f550>
```

linspace()
在指定的间隔内返回均匀间隔的数字

x = [0, 5, 10, 15, 20]

y = [1, 11, 21, 31, 41]

图像点调整

调整点的大小

s=200

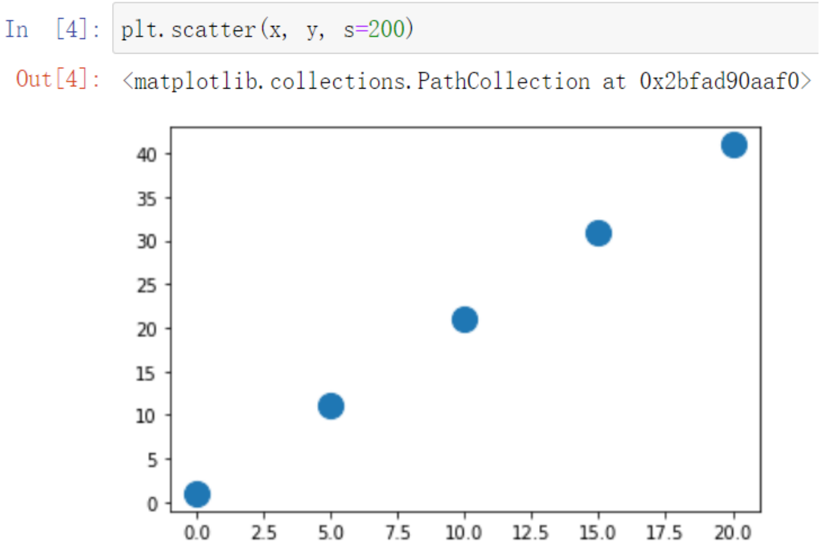
调整点的形状

marker=','

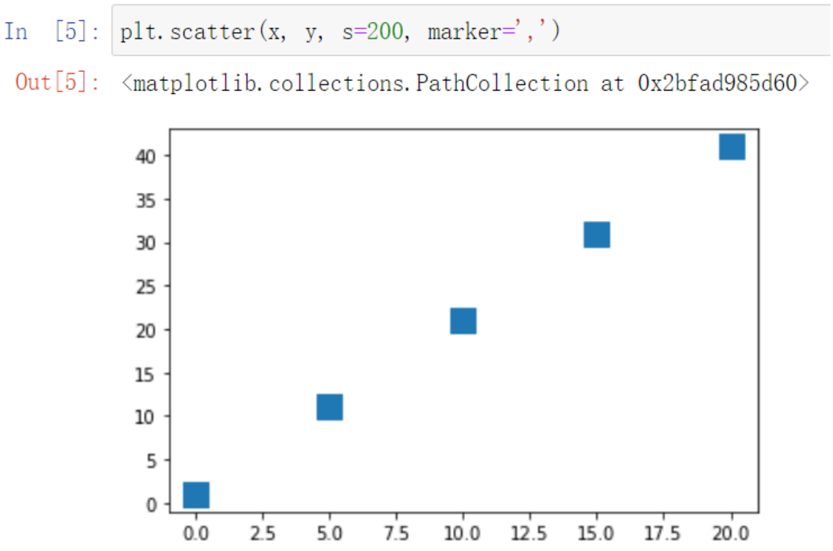
marker='v'

调整点的颜色

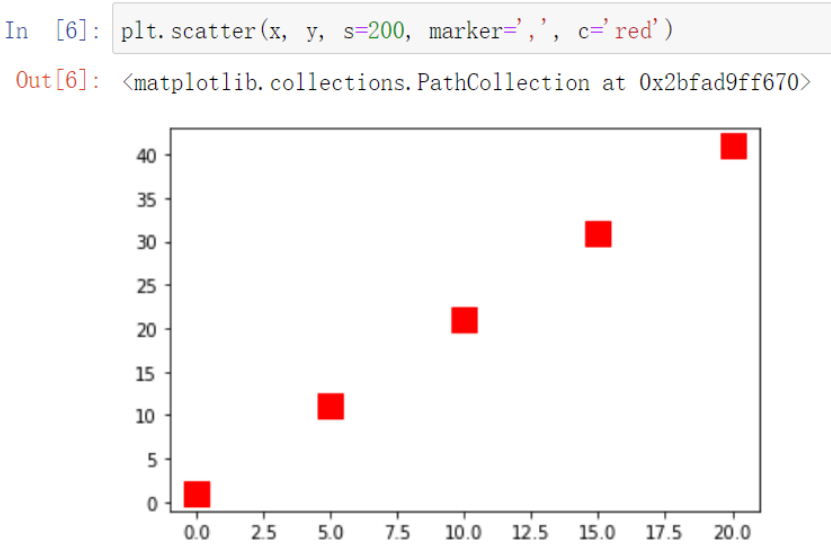
c='red'



```
In  [4]: plt.scatter(x, y, s=200)

Out[4]: <matplotlib.collections.PathCollection at 0x2bfad90aaf0>
```



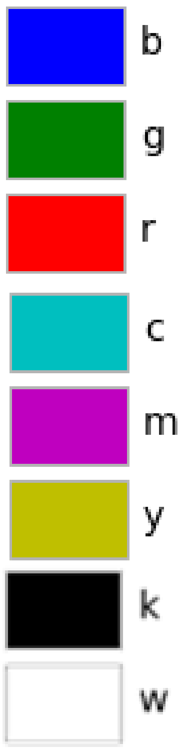
```
In  [5]: plt.scatter(x, y, s=200, marker=',')

Out[5]: <matplotlib.collections.PathCollection at 0x2bfad985d60>
```



```
In  [6]: plt.scatter(x, y, s=200, marker=',', c='red')

Out[6]: <matplotlib.collections.PathCollection at 0x2bfad9ff670>
```

常用颜色

color = 或c =	颜色	样例
None	靛青色	●
‘b’或’blue’	蓝色	●
‘g’或’green’	绿色	●
‘r’或’red’	红色	●
‘c’	青色(cyan)	●
‘m’	品红色(magenta)	●
‘y’	黄色(yellow)	●
‘k’或’black’	黑色	●
‘w’或’white’	白色	



坐标调整

调整坐标轴大小

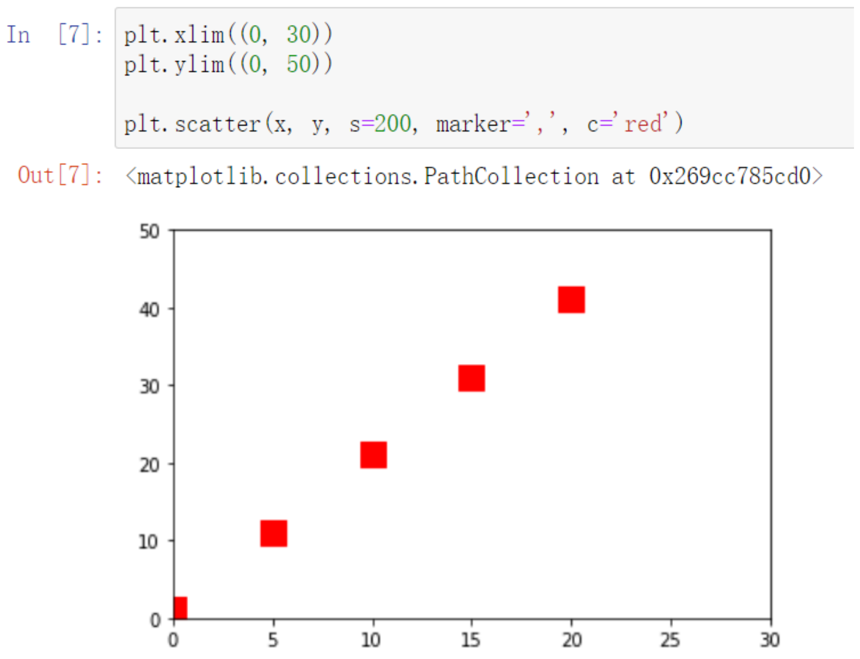
plt.xlim((0, 30))

调整坐标轴刻度

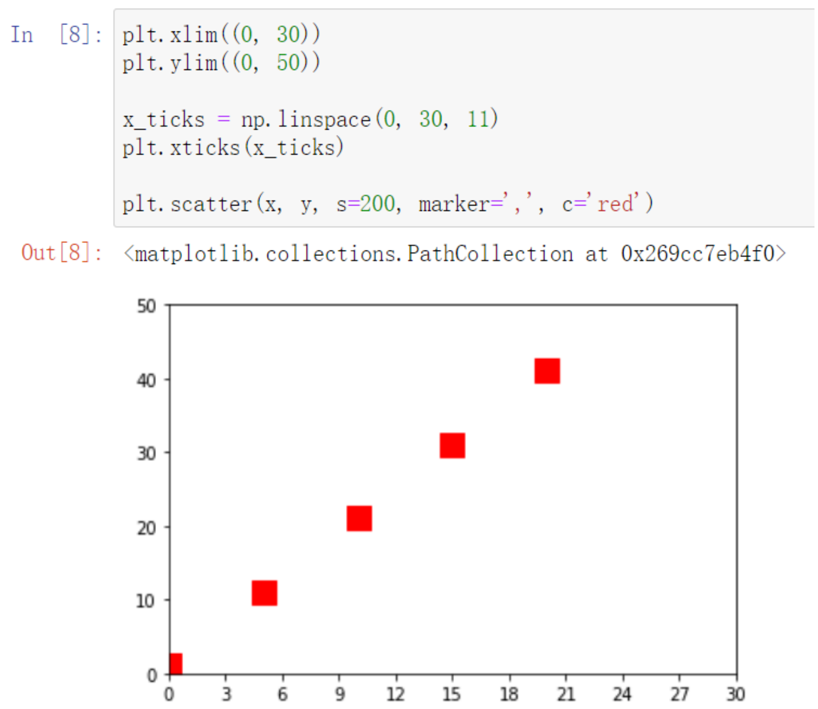
plt.xticks()

调整坐标轴标签

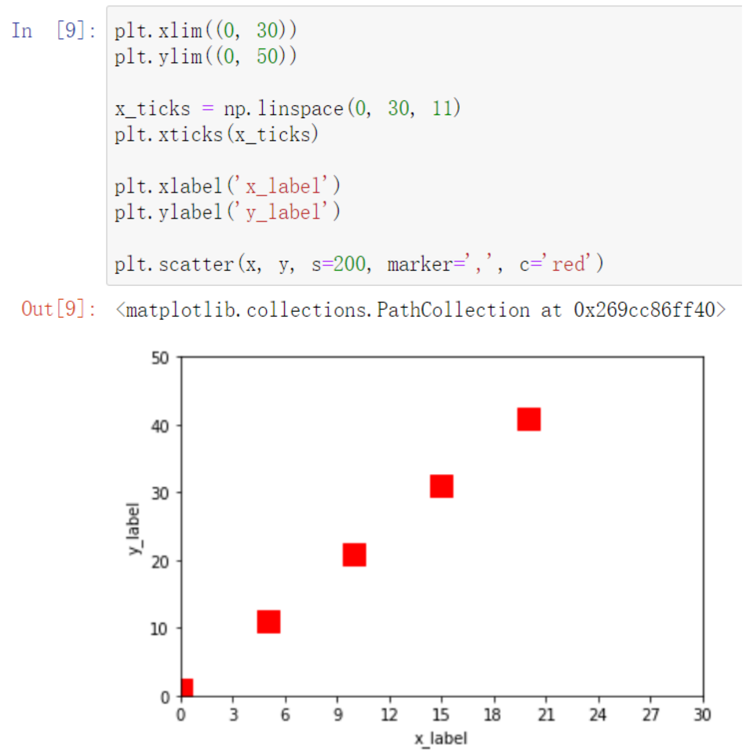
plt.xlabel()



```
In [7]: plt.xlim((0, 30))
        plt.ylim((0, 50))

        plt.scatter(x, y, s=200, marker=',', c='red')

Out[7]: <matplotlib.collections.PathCollection at 0x269cc785cd0>
```



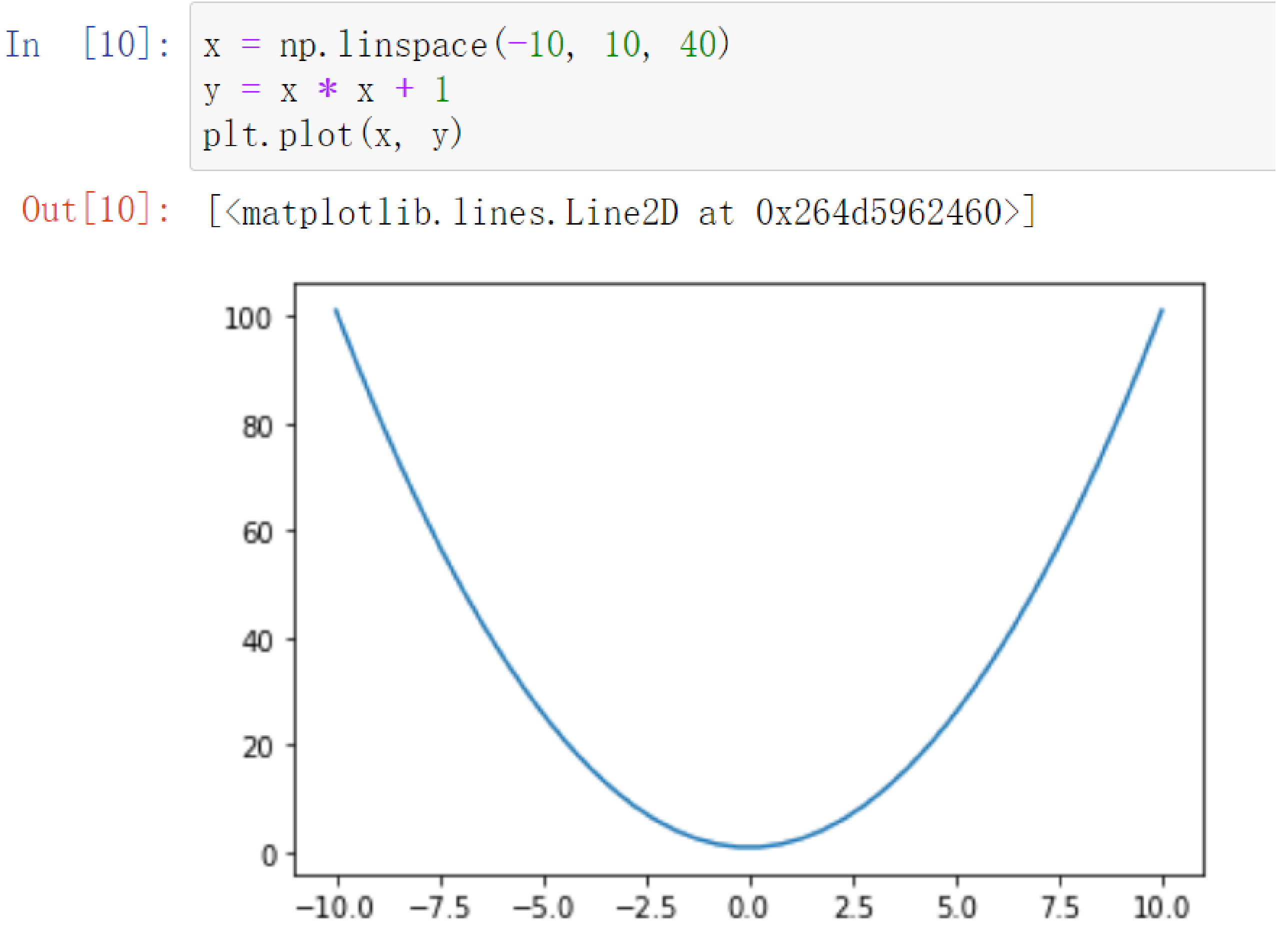
```
In [8]: plt.xlim((0, 30))
        plt.ylim((0, 50))

        x_ticks = np.linspace(0, 30, 11)
        plt.xticks(x_ticks)

        plt.scatter(x, y, s=200, marker=',', c='red')

Out[8]: <matplotlib.collections.PathCollection at 0x269cc7eb4f0>
```



```
In [9]: plt.xlim((0, 30))
        plt.ylim((0, 50))

        x_ticks = np.linspace(0, 30, 11)
        plt.xticks(x_ticks)

        plt.xlabel('x_label')
        plt.ylabel('y_label')

        plt.scatter(x, y, s=200, marker=',', c='red')

Out[9]: <matplotlib.collections.PathCollection at 0x269cc86ff40>
```

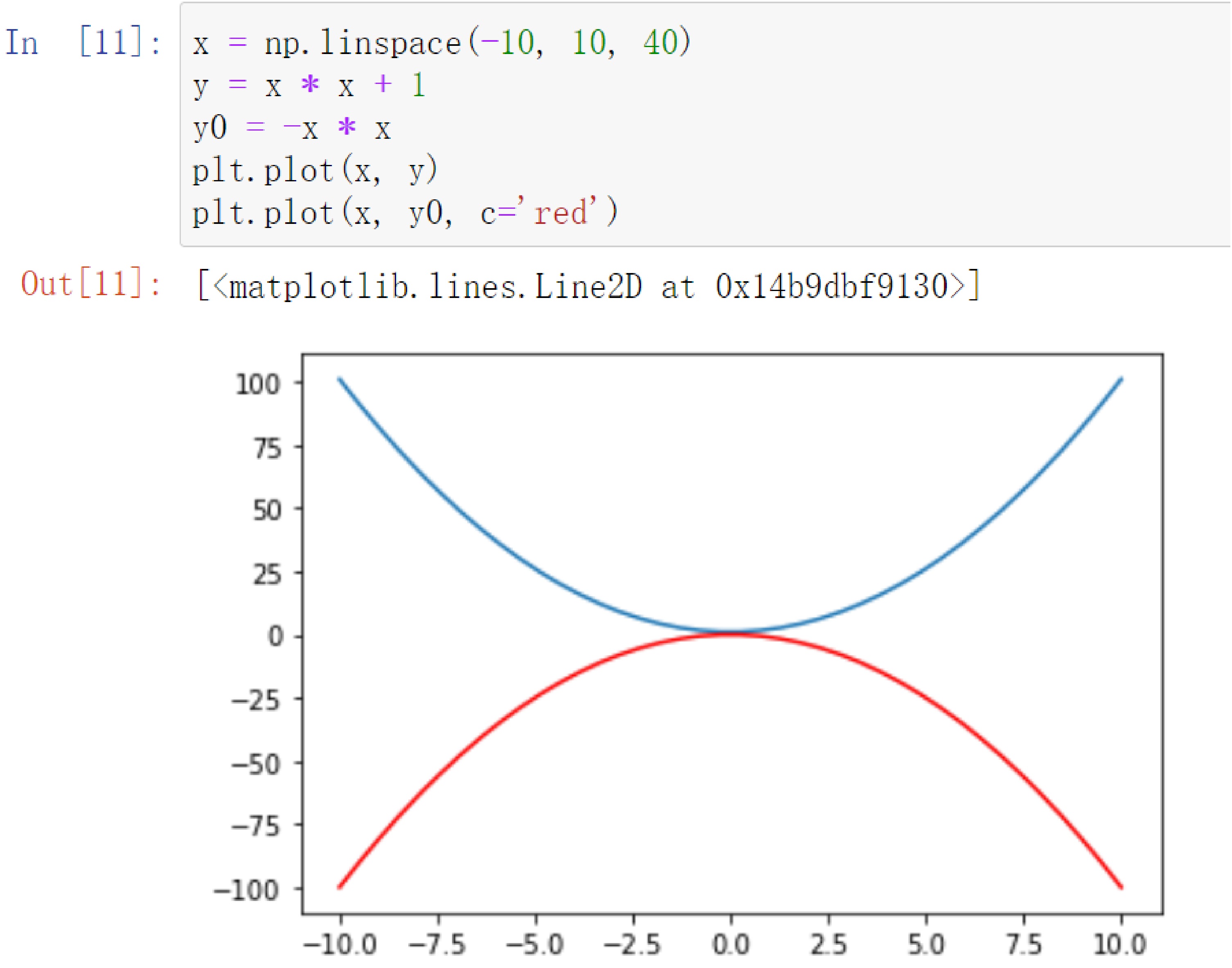
实操：简单函数绘图

y = x^2 + 1



```
In  [10]: x = np.linspace(-10, 10, 40)
          y = x * x + 1
          plt.plot(x, y)

Out[10]: [<matplotlib.lines.Line2D at 0x264d5962460>]
```

同一个图绘制多个函数



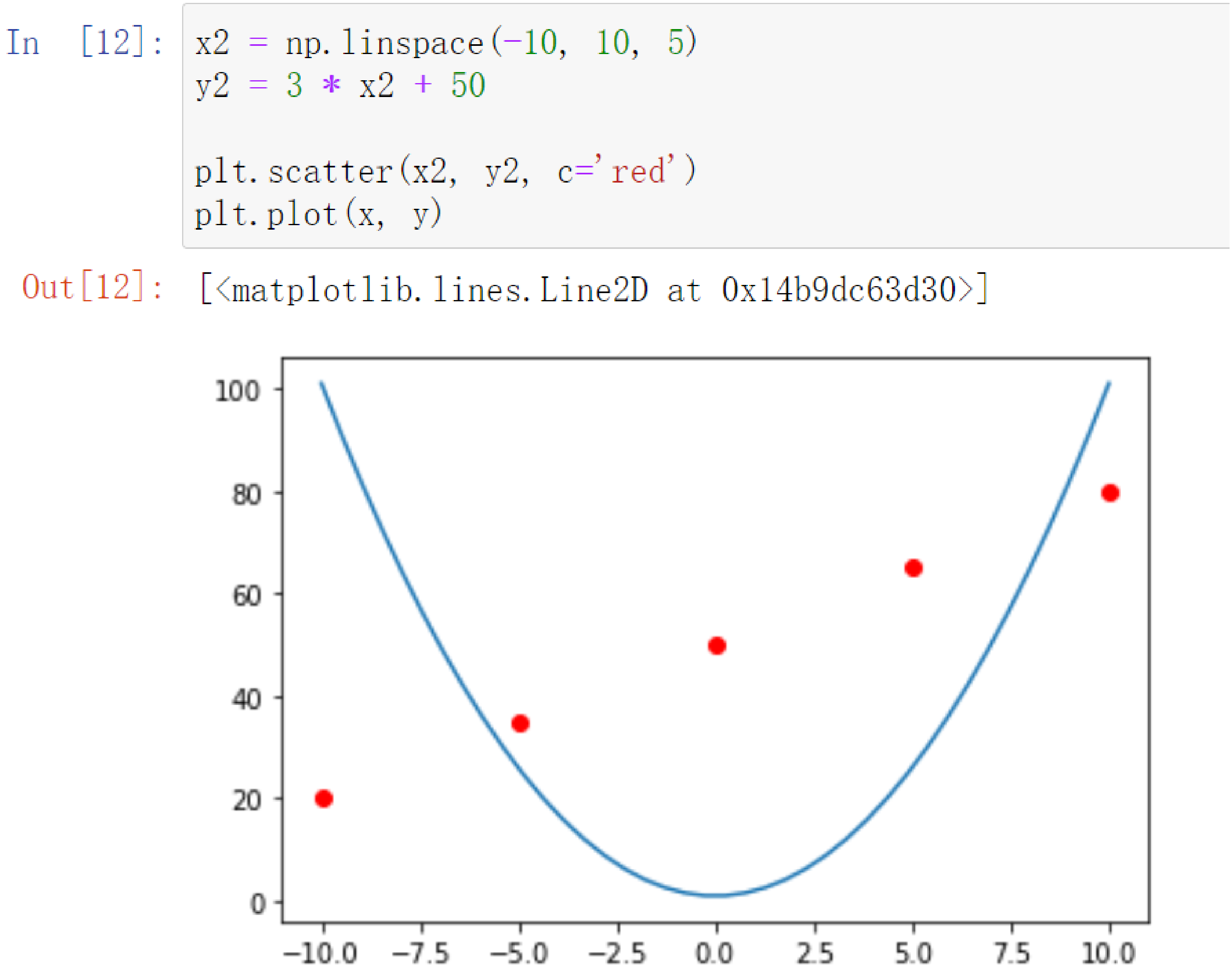
In [11]:

```
x = np.linspace(-10, 10, 40)
y = x * x + 1
y0 = -x * x
plt.plot(x, y)
plt.plot(x, y0, c='red')
```

Out[11]:

```
[<matplotlib.lines.Line2D at 0x14b9dbf9130>]
```

同时绘制散点与线



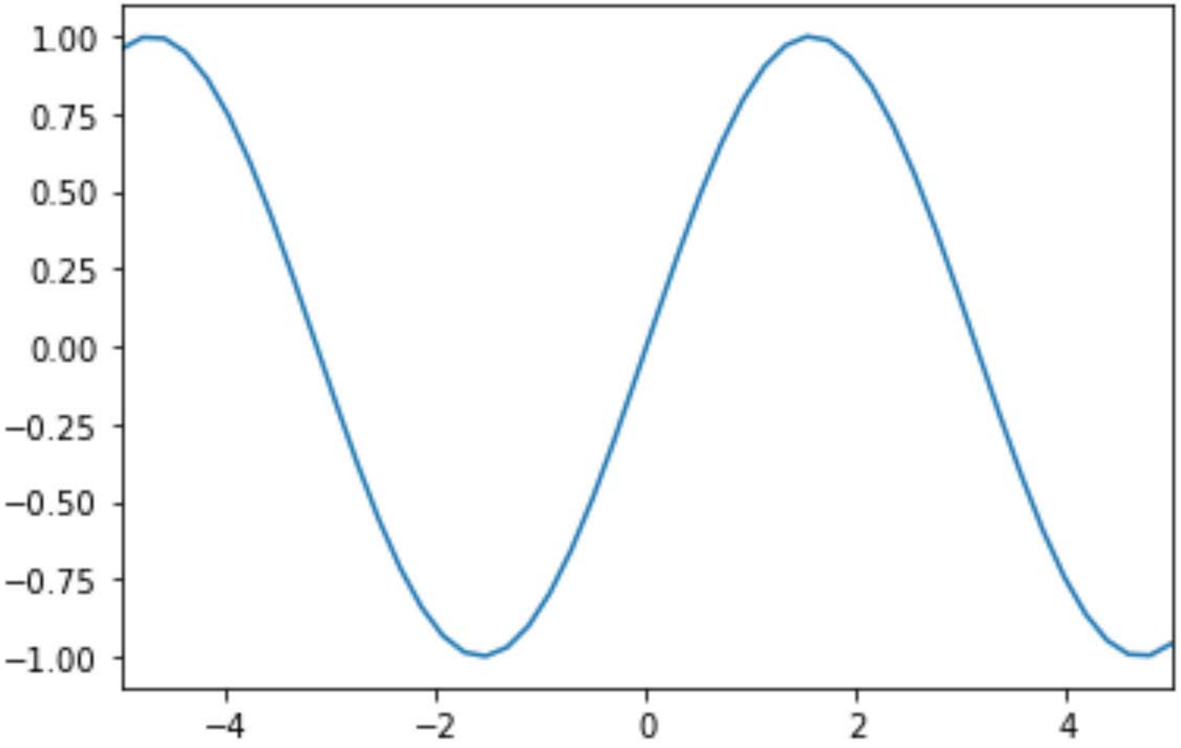
```
In  [12]: x2 = np.linspace(-10, 10, 5)
          y2 = 3 * x2 + 50

          plt.scatter(x2, y2, c='red')
          plt.plot(x, y)
```

```
Out[12]: [<matplotlib.lines.Line2D at 0x14b9dc63d30>]
```

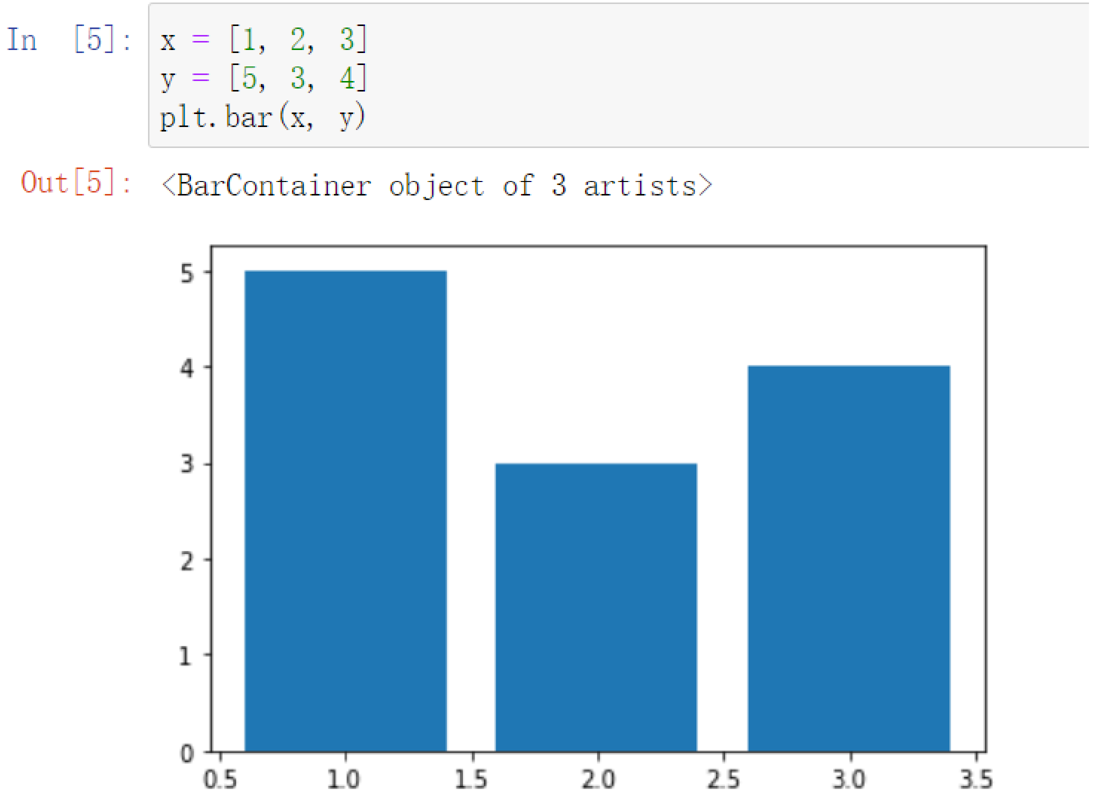
练习：绘制正弦函数图像

绘制x在(-5,5)范围内正弦函数 y = sin(x) 的图像

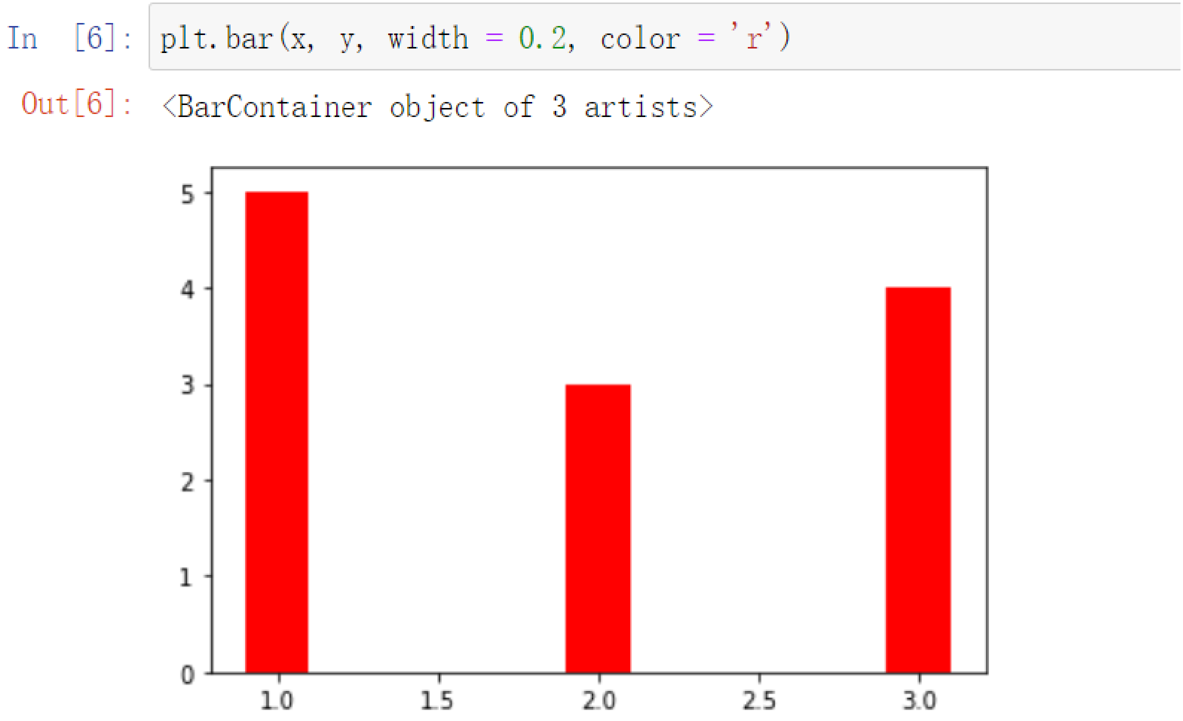
提示：sin函数调用方法为np.sin()



柱状图

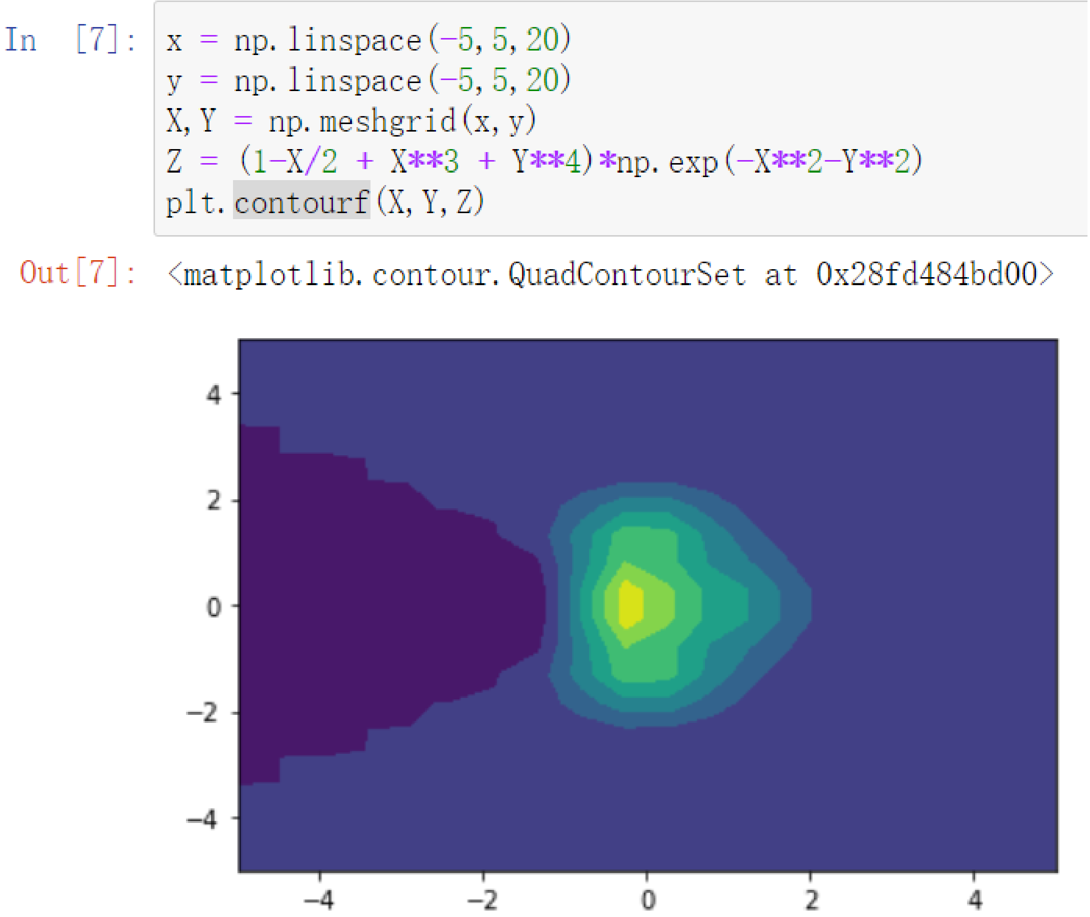


```
In  [5]: x = [1, 2, 3]
         y = [5, 3, 4]
         plt.bar(x, y)

Out[5]: <BarContainer object of 3 artists>
```

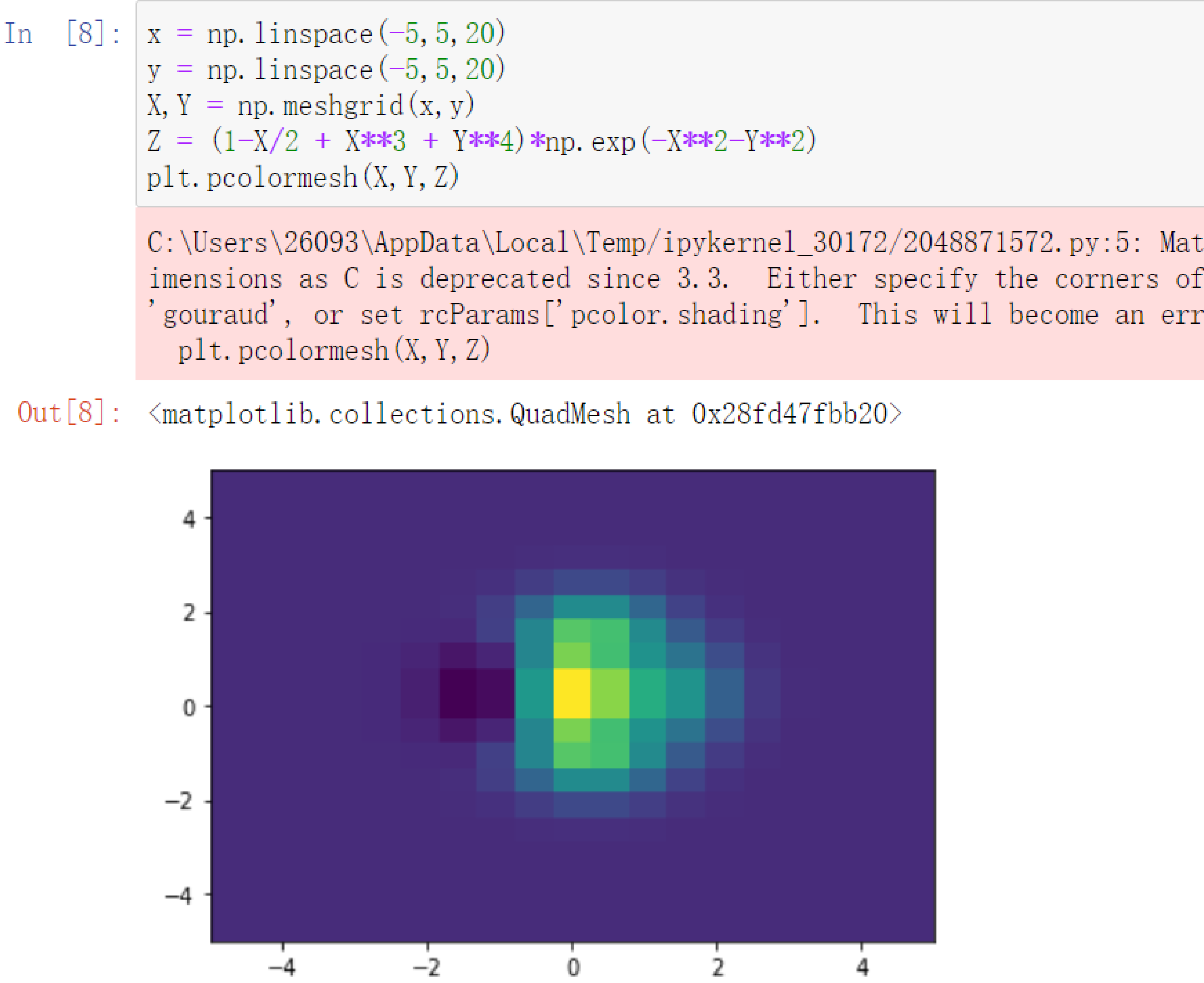


```
In  [6]: plt.bar(x, y, width = 0.2, color = 'r')

Out[6]: <BarContainer object of 3 artists>
```

阅读材料：二维图



```
In  [7]: x = np.linspace(-5,5,20)
         y = np.linspace(-5,5,20)
         X,Y = np.meshgrid(x,y)
         Z = (1-X/2 + X**3 + Y**4)*np.exp(-X**2-Y**2)
         plt.contourf(X,Y,Z)

Out[7]: <matplotlib.contour.QuadContourSet at 0x28fd484bd00>
```



```
In  [8]: x = np.linspace(-5,5,20)
         y = np.linspace(-5,5,20)
         X,Y = np.meshgrid(x,y)
         Z = (1-X/2 + X**3 + Y**4)*np.exp(-X**2-Y**2)
         plt.pcolormesh(X,Y,Z)

C:\Users\26093\AppData\Local\Temp/ipykernel_30172/2048871572.py:5: Mat
imensions as C is deprecated since 3.3.  Either specify the corners of
'gouraud', or set rcParams['pcolor.shading'].  This will become an err
  plt.pcolormesh(X,Y,Z)

Out[8]: <matplotlib.collections.QuadMesh at 0x28fd47fbb20>
```