数据计算（2）

Pandas主要数据结构

- Series类型
- DataFrame类型

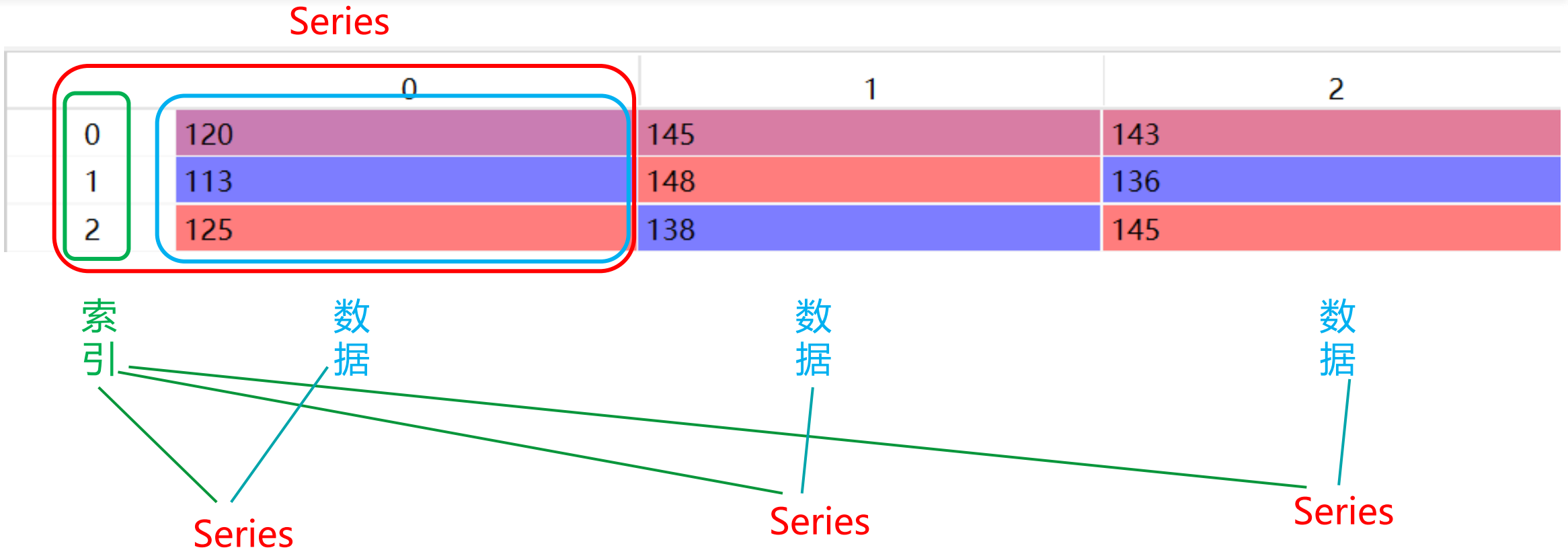
DataFrame类型：带有标签的二维异构表格

由二维数据及与之相关的行列索引（标签）组成

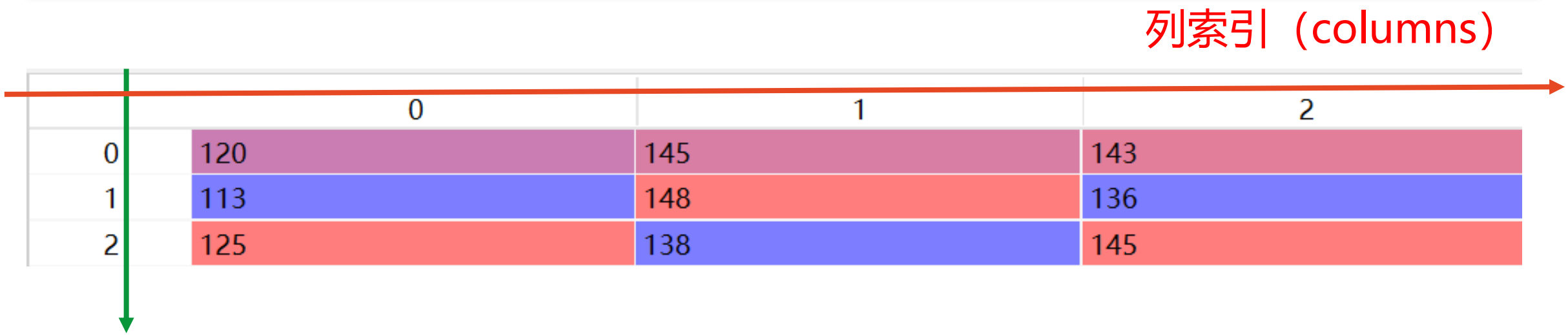
df = {DataFrame: (3, 3)} 0 1 2 [0 120 145 143] [1 113 148 136] [2 125 138 145]

	0	1	2
0	120	145	143
1	113	148	136
2	125	138	145

DataFrame类型

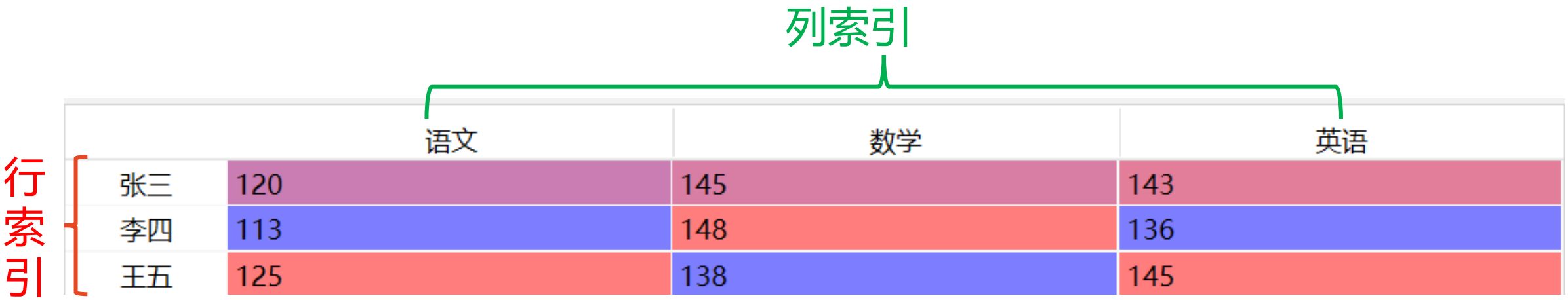


DataFrame类型




```
>>> df.index = ['张三','李四','王五']
>>> df.columns = ['语文','数学','英语']
```

DataFrame类型




```
>>> print(df)
      语文    数学    英语
张三   120   145   143
李四   113   148   136
王五   125   138   145
```

```
>>> df.index = ['张三','李四','王五']
>>> df.columns = ['语文','数学','英语']
```

DataFrame对象：创建

df=pandas.DataFrame(data=None, index=None, columns=None,dtype=None, copy=False)

- data：数据，支持列表、字典、numpy数组、series对象
- index: 行标签（索引）
- columns：列标签（索引）
- dtype：每一列的数据类型
- copy：是否创建输入数据的副本

DataFrame对象创建：由二维数组创建

```
>>> data = [[120,145,143],[113,148,136],[125,138,145]]
>>> df = pd.DataFrame(data)
```

隐式索引

```
>>> print(df)
     0    1    2
0  120  145  143
1  113  148  136
2  125  138  145
```

DataFrame对象创建：由字典创建

```
>>> dict1={'语文':[120,113,125],'数学':[145,148,138],'英语':[143,136,145]}
>>> df1 = pd.DataFrame(dict1)
```

```
>>> pd.set_option('display.unicode.east_asian_width',True)
```

```
>>> print(df1)
    语文   数学   英语
0  120  145  143
1  113  148  136
2  125  138  145
```

```
>>> print(df1)
    语文   数学   英语
0  120  145  143
1  113  148  136
2  125  138  145
```

DataFrame对象创建：由字典创建

```
>>> dict1={'语文':[120,113,125],'数学':[145,148,138],'英语':[143,136,145]}
>>> index2 = ['张三','李四','王五']
>>> columns2 = ['语文','数学','英语','班级']
>>> df2 = pd.DataFrame(dict1,index=index2,columns=columns2)
>>> df2['班级']='高三五班'
>>> print(df2)
```

	语文	数学	英语	班级
张三	120	145	143	NaN
李四	113	148	136	NaN
王五	125	138	145	NaN

```
>>> print(df2)
```

	语文	数学	英语	班级
张三	120	145	143	高三五班
李四	113	148	136	高三五班
王五	125	138	145	高三五班

DataFrame对象创建：由字典创建

```
>>> print(df1)
      语文   数学   英语     班级
张三   120   145   143   高三五班
李四   113   148   136   高三五班
王五   125   138   145   高三五班
```

```
>>> print(df1.dtypes)
语文      int64
数学      int64
英语      int64
班级     object
dtype: object
```

通用类型

DataFrame对象的属性和方法

pandas.DataFrame — pandas 1.3.3 documentation (pydata.org)

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html

DataFrame对象的常见运算和操作

- Numpy二维数组运算及操作
- Python字典操作
- 特有操作

DataFrame对象查看

```
>>> df2.describe()
              语文         数学         英语
count    3.000000    3.000000    3.000000
mean   119.333333  143.666667  141.333333
std      6.027714    5.131601    4.725816
min    113.000000  138.000000  136.000000
25%    116.500000  141.500000  139.500000
50%    120.000000  145.000000  143.000000
75%    122.500000  146.500000  144.000000
max    125.000000  148.000000  145.000000
```

```
>>> df2.values
array([[120, 145, 143, '高三五班'],
       [113, 148, 136, '高三五班'],
       [125, 138, 145, '高三五班']], dtype=object)
>>> df2.index
Index(['张三', '李四', '王五'], dtype='object')
>>> df2.columns
Index(['语文', '数学', '英语', '班级'], dtype='object')
```

DataFrame对象的索引结构

		0	1	2	3	4	5
		A	B	C	D	E	F
0	A						
1	B						
2	C						
3	D						
4	E						
5	F						

DataFrame数据抽取：抽取一行数据

```
>>> print(df1)
      语文   数学   英语     班级
张三   120   145   143   高三五班
李四   113   148   136   高三五班
王五   125   138   145   高三五班
```

```
>>> print(df1.loc['张三'])
语文        120
数学        145
英语        143
班级      高三五班
Name: 张三, dtype: object
```

```
>>> print(df1.iloc[1])
语文        113
数学        148
英语        136
班级      高三五班
Name: 李四, dtype: object
```

DataFrame数据抽取：抽取多行数据

```
>>> print(df1)
      语文   数学   英语     班级
张三   120   145   143  高三五班
李四   113   148   136  高三五班
王五   125   138   145  高三五班
```

```
>>> print(df1.loc[['张三','王五']])
      语文   数学   英语     班级
张三   120   145   143  高三五班
王五   125   138   145  高三五班
```

```
>>> print(df1.iloc[[0,2]])
      语文   数学   英语     班级
张三   120   145   143  高三五班
王五   125   138   145  高三五班
```

DataFrame数据抽取：抽取连续多行数据

```
>>> print(df1.loc['张三':'王五'])
      语文   数学   英语      班级
张三   120   145   143   高三五班
李四   113   148   136   高三五班
王五   125   138   145   高三五班
>>> print(df1.loc[:'李四':])
      语文   数学   英语      班级
张三   120   145   143   高三五班
李四   113   148   136   高三五班
```

```
>>> print(df1.iloc[0:2])
      语文   数学   英语      班级
张三   120   145   143   高三五班
李四   113   148   136   高三五班
>>> print(df1.iloc[1::])
      语文   数学   英语      班级
李四   113   148   136   高三五班
王五   125   138   145   高三五班
```

DataFrame数据抽取：抽取指定列数据

```
>>> print(df1)
      语文   数学   英语     班级
张三   120   145   143  高三五班
李四   113   148   136  高三五班
王五   125   138   145  高三五班
```

```
>>> print(df1[['语文','数学']])
      语文   数学
张三   120   145
李四   113   148
王五   125   138
```

DataFrame数据抽取：抽取指定列数据

```
>>> print(df1.loc[:,['语文','英语']])
      语文   英语
张三    120    143
李四    113    136
王五    125    145
```

```
>>> print(df1.iloc[:,[0,2]])
      语文   英语
张三    120    143
李四    113    136
王五    125    145
```

```
>>> print(df1.loc[:,'数学':])
      数学   英语      班级
张三    145    143   高三五班
李四    148    136   高三五班
王五    138    145   高三五班
```

```
>>> print(df1.iloc[:,:2])
      语文   数学
张三    120    145
李四    113    148
王五    125    138
```

DataFrame数据抽取：抽取指定行列数据

```
>>> print(df1.loc['李四','英语'])
136

>>> print(df1.iloc[1,2])
136

>>> print(df1.loc[['李四'],['英语']])
      英语
李四   136

>>> print(df1.iloc[[1],[2]])
      英语
李四   136
```

```
>>> print(df1.loc['李四':,['数学']])
      数学
李四   148
王五   138

>>> print(df1.iloc[1:,[2]])
      英语
李四   136
王五   145

>>> print(df1.iloc[:,2])
张三    143
李四    136
王五    145
Name: 英语, dtype: int64
```

DataFrame数据抽取：按指定条件抽取

```
>>> print(df1)
      语文   数学   英语       班级
张三   120   145   143   高三五班
李四   113   148   136   高三五班
王五   125   138   145   高三五班
```

```
>>> print(df1.loc[(df1['语文']>115) & (df1['数学']>140)])
      语文   数学   英语       班级
张三   120   145   143   高三五班
```

DataFrame索引设置：修改行列索引

```
DataFrame.reindex(labels=None, index=None, columns=None, axis=None, method=None,
copy=True, level=None, fill_value=nan, limit=None, tolerance=None)
```

DataFrame索引设置：修改行列索引

```
pd.set_option('display.unicode.east_asian_width', True)
data = [[110,105,99],[105,88,115],[109,120,130]]
index=['mr001','mr003','mr005']
columns = ['语文','数学','英语']
df = pd.DataFrame(data=data, index=index,columns=columns)
print(df)
```

```
      语文  数学  英语
mr001   110   105    99
mr003   105    88   115
mr005   109   120   130
```

DataFrame索引设置：修改行索引

```
>>> print(df.reindex(['mr001','mr002','mr003','mr004','mr005']))
        语文    数学    英语
mr001  110.0  105.0   99.0
mr002    NaN    NaN    NaN
mr003  105.0   88.0  115.0
mr004    NaN    NaN    NaN
mr005  109.0  120.0  130.0
```

DataFrame索引设置：修改列索引

```
>>> print(df.reindex(columns=['语文','物理','数学','英语']))
        语文   物理   数学   英语
mr001    110    NaN    105     99
mr003    105    NaN     88    115
mr005    109    NaN    120    130
```

DataFrame索引设置：修改行列索引

```
>>> print(df.reindex(index=['mr001','mr002','mr003','mr004','mr005'],columns=['语文','物理','数学','英语']))
        语文   物理    数学    英语
mr001  110.0   NaN  105.0   99.0
mr002    NaN   NaN    NaN    NaN
mr003  105.0   NaN   88.0  115.0
mr004    NaN   NaN    NaN    NaN
mr005  109.0   NaN  120.0  130.0
```

DataFrame索引设置：设置某列为索引

```
DataFrame.set_index(keys, drop=True, append=False, inplace=False, verify_integrity=False)
```

- keys：设置为索引的列
- drop: 是否从DataFrame中删除已设置为索引的列
- append：是否将索引列增加到DataFrame中
- inplace：是否替代原DataFrame
- verify_integrity：检查新索引是否存在重复项

DataFrame索引设置：设置某列为索引

```
df = pd.DataFrame({'month': [1, 4, 7, 10],
                   'year': [2012, 2014, 2013, 2014],
                   'sale': [55, 40, 84, 31]})
```

```
>>> df
   month  year  sale
0      1  2012    55
1      4  2014    40
2      7  2013    84
3     10  2014    31
```

```
>>> df.set_index('month')
       year  sale
month
1      2012    55
4      2014    40
7      2013    84
10     2014    31
```

```
>>> df.set_index(['year', 'month'])
           sale
year month
2012 1       55
2014 4       40
2013 7       84
2014 10      31
```

DataFrame数据修改：按列增加数据

```
pd.set_option('display.unicode.east_asian_width', True)
data = [[110,105,99],[105,88,115],[109,120,130],[112,115,140]]
name = ['张三','李四','王五','赵六']
columns = ['语文','数学','英语']
df = pd.DataFrame(data=data, index=name, columns=columns)
```

```
>>> df
     语文  数学  英语
张三   110   105    99
李四   105    88   115
王五   109   120   130
赵六   112   115   140
```

```
>>> df.loc[:,'物理'] = [88,79,60,50]
>>> df
     语文  数学  英语  物理
张三   110   105    99    88
李四   105    88   115    79
王五   109   120   130    60
赵六   112   115   140    50
```

```
>>> df.loc[:,'物理'] = 55
>>> df
     语文  数学  英语  物理
张三   110   105    99    55
李四   105    88   115    55
王五   109   120   130    55
赵六   112   115   140    55
```

DataFrame数据修改：按列增加数据

```
>>> df
      语文   数学   英语
张三   110   105    99
李四   105    88   115
王五   109   120   130
赵六   112   115   140
```

```
wl =[88,79,60,50]
df.insert(1,'物理',wl)
```

```
>>> df
      语文   物理   数学   英语
张三   110    88   105    99
李四   105    79    88   115
王五   109    60   120   130
赵六   112    50   115   140
```

DataFrame数据修改：按行增加数据

```
>>> df
      语文   数学   英语
张三   110   105    99
李四   105    88   115
王五   109   120   130
赵六   112   115   140
```

```
>>> df.loc['钱多多'] = [100,120,99]
>>> df
      语文   数学   英语
张三   110   105    99
李四   105    88   115
王五   109   120   130
赵六   112   115   140
钱多多  100   120    99
```

DataFrame数据修改：修改行数据

```
>>> df
      语文   数学   英语
张三   110   105    99
李四   105    88   115
王五   109   120   130
赵六   112   115   140
```

```
>>> df.loc['李四']=[120,115,109]
>>> df
      语文   数学   英语
张三   110   105    99
李四   120   115   109
王五   109   120   130
赵六   112   115   140
```

```
>>> df.loc['张三']=df.loc['张三']+10
>>> df
      语文   数学   英语
张三   120   115   109
李四   120   115   109
王五   109   120   130
赵六   112   115   140
```

DataFrame数据修改：修改列数据

```
>>> df
      语文   数学   英语
张三   110   105    99
李四   105    88   115
王五   109   120   130
赵六   112   115   140
```

```
>>> df.loc[:,'语文']=[115,108,112,118]
>>> df
      语文   数学   英语
张三   115   105    99
李四   108    88   115
王五   112   120   130
赵六   118   115   140
```

```
>>> df.loc[:,'数学']=df.loc[:,'数学']*1.1
>>> df
      语文    数学   英语
张三   115  115.5    99
李四   108   96.8   115
王五   112  132.0   130
赵六   118  126.5   140
```

DataFrame数据修改：删除数据

```
DataFrame.drop(labels=None, axis=0, index=None, columns=None, level=None, inplace=False,
errors='raise')                                                                    [source]
```

- labels：要删除数据的行标签或列标签
- axis: 执行删除的轴
- index：要删除数据的行标签
- columns：要删除数据的列标签
- level：对存在多级索引的数据，指明按哪级索引进行删除
- inplace：是否替换原DataFrame
- error：错误处理

DataFrame数据修改：删除列数据

```
>>> df
      语文   数学   英语
张三   110   105    99
李四   105    88   115
王五   109   120   130
赵六   112   115   140
```

```
>>> df.drop(labels='数学',axis=1)
>>> df.drop(columns='数学')
>>> df.drop(['数学'],axis=1)
```

```
      语文   英语
张三   110    99
李四   105   115
王五   109   130
赵六   112   140
```

DataFrame数据修改：删除行数据

```
>>> df
      语文   数学   英语
张三    110    105     99
李四    105     88    115
王五    109    120    130
赵六    112    115    140
```

```
>>> df.drop(['王五'])
>>> df.drop(index='王五')
>>> df.drop(labels='王五',axis=0)
```

```
      语文   数学   英语
张三    110    105     99
李四    105     88    115
赵六    112    115    140
```

DataFrame数据修改：按条件删除行

```
>>> df
      语文   数学   英语
张三   110   105    99
李四   105    88   115
王五   109   120   130
赵六   112   115   140
```

```
>>> df.drop(index=df[df['数学'].isin([88,120])].index)
      语文   数学   英语
张三   110   105    99
赵六   112   115   140
```

```
>>> df.drop(index=df[df['英语']<120].index,inplace=True)
>>> df
      语文   数学   英语
王五   109   120   130
赵六   112   115   140
```

DataFrame对象排序：按索引排序

DataFrame.sort_index(axis=0, ascending=True, inplace=False)

```
>>> df2
     name  quantity  price  sold
3     pen       200    9.9    50
1   ruler       400    3.1    30
2  rubber       800    2.3    78
```

```
>>> df2.sort_index()
     name  quantity  price  sold
1   ruler       400    3.1    30
2  rubber       800    2.3    78
3     pen       200    9.9    50
```

```
>>> df2.sort_index(axis=1)
     name  price  quantity  sold
3     pen    9.9       200    50
1   ruler    3.1       400    30
2  rubber    2.3       800    78
```

DataFrame对象排序：按值排序

DataFrame.sort_values(by, axis=0, ascending=True, inplace=False)

```
>>> print(df2)
      语文   数学   英语      班级
张三   120   145   143  高三五班
李四   113   148   136  高三五班
王五   125   138   145  高三五班

>>> df2.sort_values(by='语文')
      语文   数学   英语      班级
李四   113   148   136  高三五班
张三   120   145   143  高三五班
王五   125   138   145  高三五班
```

```
>>> df2.sort_values(by=['英语','数学'],ascending=False)
      语文   数学   英语      班级
王五   125   138   145  高三五班
张三   120   145   143  高三五班
李四   113   148   136  高三五班
>>> df2.sort_values(by=['数学','英语'],ascending=False)
      语文   数学   英语      班级
李四   113   148   136  高三五班
张三   120   145   143  高三五班
王五   125   138   145  高三五班
```

DataFrame对象排序：按值排序

DataFrame.sort_values(by, axis=0, ascending=True, inplace=False)

```
>>> df2
     name  quantity  price  sold
3     pen     500.0    9.9  50.0
1   ruler     300.0    3.1  30.0
2  rubber     500.0    2.3  78.0
```

```
>>> df2.sort_values(by=['quantity','sold'],ascending=[True,False])
     name  quantity  price  sold
1   ruler     300.0    3.1  30.0
2  rubber     500.0    2.3  78.0
3     pen     500.0    9.9  50.0
```

DataFrame对象排序：按值排序

DataFrame.sort_values(by, axis=0, ascending=True, inplace=False)

```
>>> df
   b  a  c
2  1  4  1
0  2  3  3
1  3  2  8
3  2  1  2
```

```
>>> df.sort_values(by=[3,0],axis=1,ascending=[True,False])
   a  c  b
2  4  1  1
0  3  3  2
1  2  8  3
3  1  2  2
```

DataFrame对象的算术运算

```
>>> df3 = pd.DataFrame(np.arange(15).reshape(3,5))
>>> df4 = pd.DataFrame(np.arange(16).reshape(4,4))
>>> print(df3)
    0   1   2   3   4
0   0   1   2   3   4
1   5   6   7   8   9
2  10  11  12  13  14
>>> print(df4)
    0   1   2   3
0   0   1   2   3
1   4   5   6   7
2   8   9  10  11
3  12  13  14  15
```

```
>>> print(df3+df4)
      0     1     2     3   4
0   0.0   2.0   4.0   6.0 NaN
1   9.0  11.0  13.0  15.0 NaN
2  18.0  20.0  22.0  24.0 NaN
3   NaN   NaN   NaN   NaN NaN
```

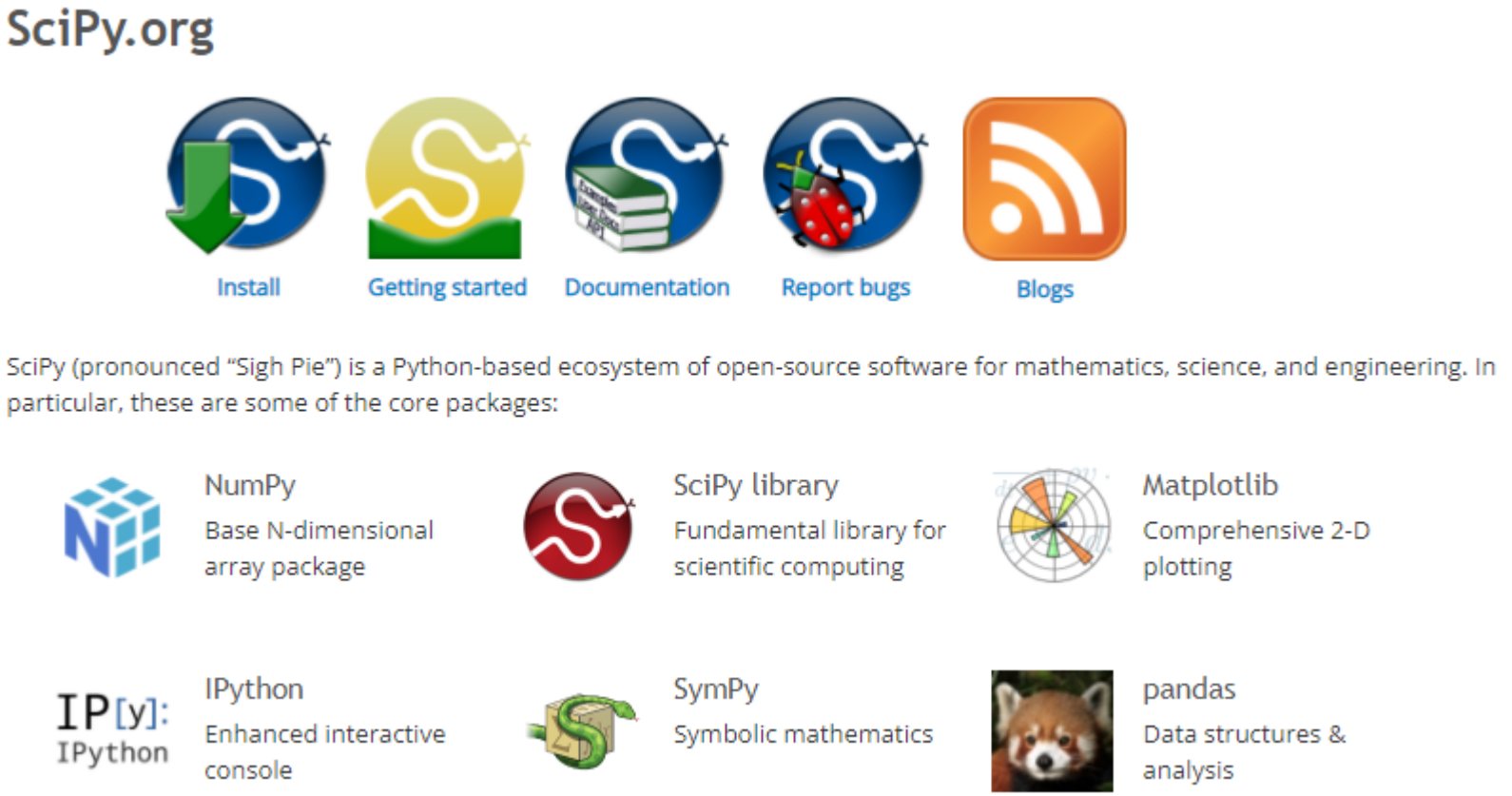
```
>>> print(df3.add(df4,fill_value=100))
       0      1      2      3      4
0    0.0    2.0    4.0    6.0  104.0
1    9.0   11.0   13.0   15.0  109.0
2   18.0   20.0   22.0   24.0  114.0
3  112.0  113.0  114.0  115.0    NaN
```

内容

- NumPy
- Pandas
- Scipy

SciPy

- SciPy 是一个开源的 Python 算法库和数学工具包
- Scipy 基于 Numpy 的科学计算库，用于数学、科学、工程学等领域，很多有一些高阶抽象和物理模型需要使用 Scipy
- SciPy 包含的模块有最优化、线性代数、积分、插值、特殊函数、快速傅里叶变换、信号处理和图像处理、常微分方程求解和其他科学与工程中常用的计算



https://www.scipy.org/

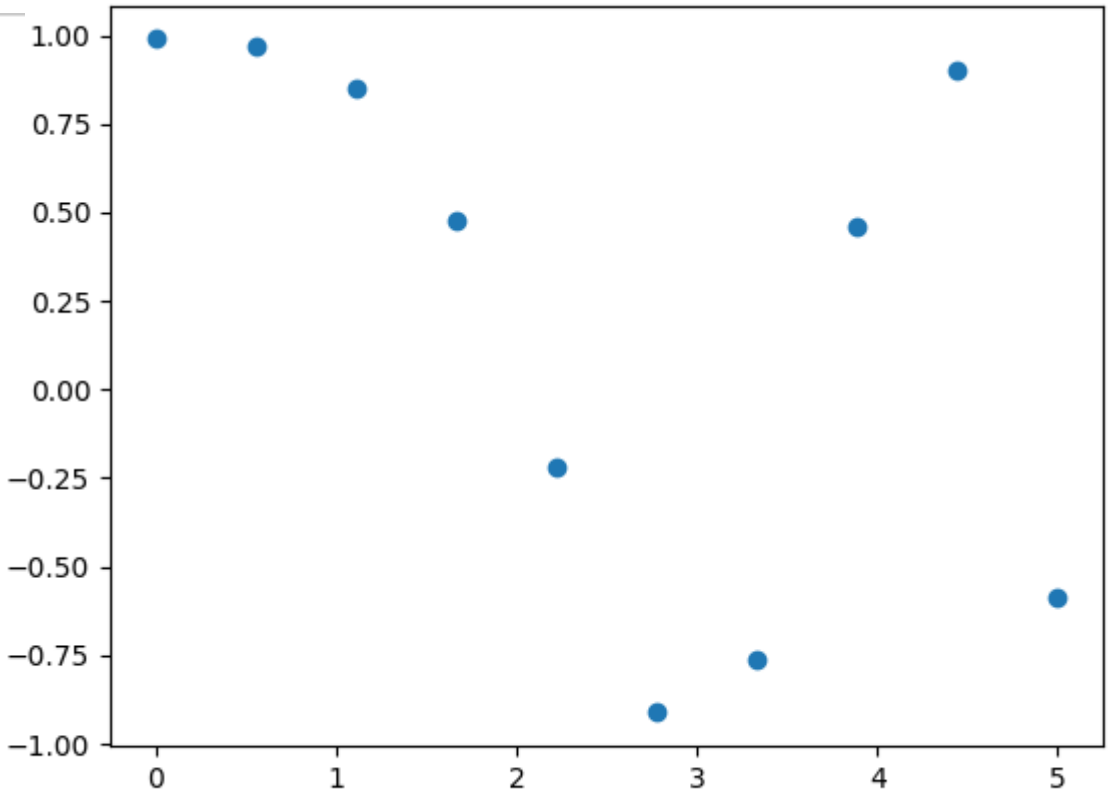
SciPy常用模块

模块	说明
cluster	聚类
constants	物理和数学常数
fft	快速傅里叶变换
integrate	积分和常微分方程求解器
interpolate	插值
io	数据输入和输出
linalg	线性代数例程
misc	图像处理

模块	说明
ndimage	n维图像包
odr	正交距离回归
optimize	优化算法
signal	信号处理
sparse	稀疏矩阵
spatial	空间数据结构和算法
special	特殊函数
stats	统计函数

SciPy：插值（interpolate）

```
import numpy as np
from scipy import interpolate
import matplotlib.pyplot as plt
x = np.linspace(0, 5, 10)    #在0到5之间生成10个数据
y = np.sin(x**2/3+8)     #y是x的某种三角函数
plt.plot(x, y,'o')  #使用matplotlib库生成图形 plt.show()
plt.show()
```

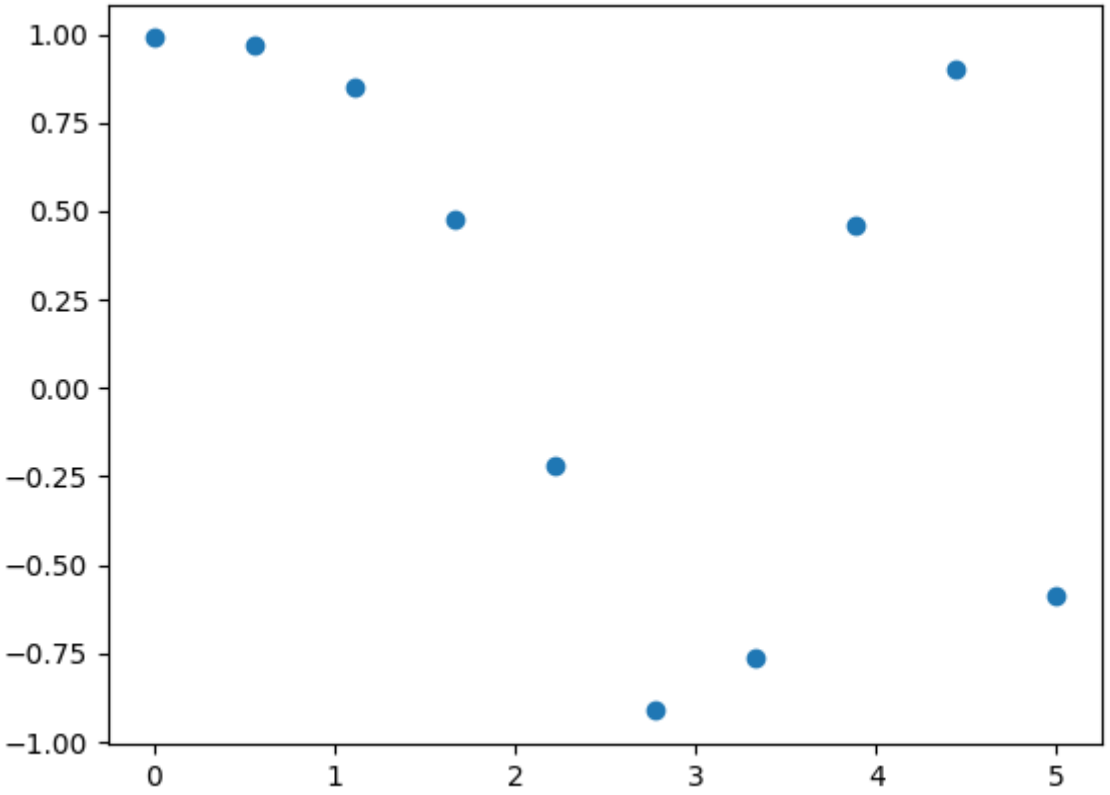


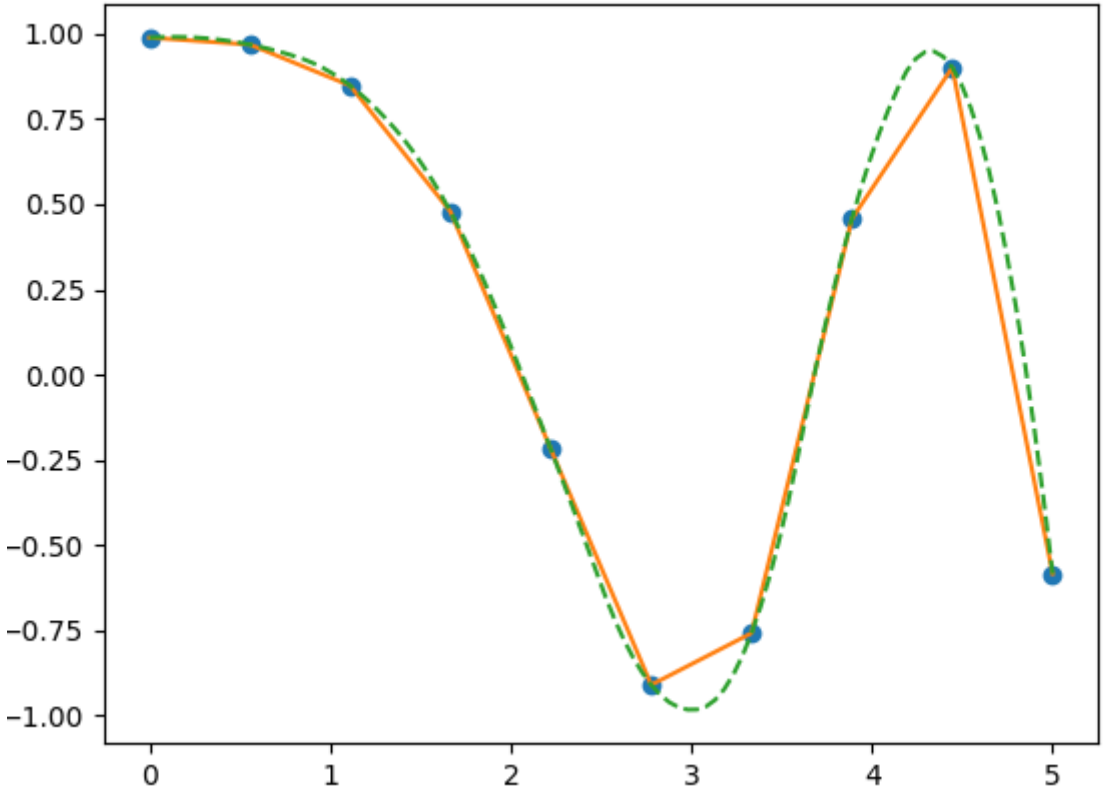
SciPy：插值（interpolate）

scipy.interpolate中的interp1d类，是一种创建基于固定数据点的函数的便捷方法，可以使用插值方法在给定数据定义区域内的任意位置评估该函数

```
f1 = interpolate.interp1d(x, y,kind = 'linear') #线性插值
f2 = interpolate.interp1d(x, y, kind = 'quadratic') #2阶样条插值
xnew = np.linspace(0, 5,100)
plt.plot(x, y, 'o', xnew, f1(xnew), '-', xnew, f2(xnew), '--')
plt.show()
```

SciPy：插值（interpolate）





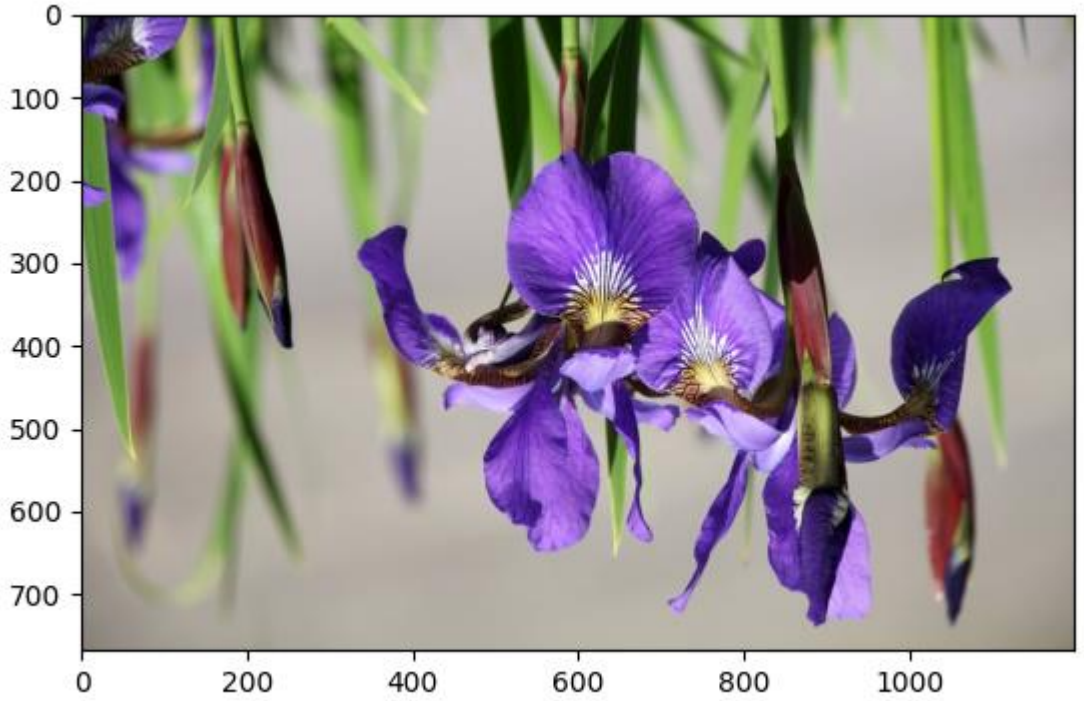
SciPy：图像处理（ndimage）

```
from matplotlib import pyplot as plt
from scipy import ndimage
from imageio import imread

iris_path = r"C:\Users\wangjing\Pictures\iri
iris = imread(iris_path)
plt.imshow(iris)
plt.show()
```

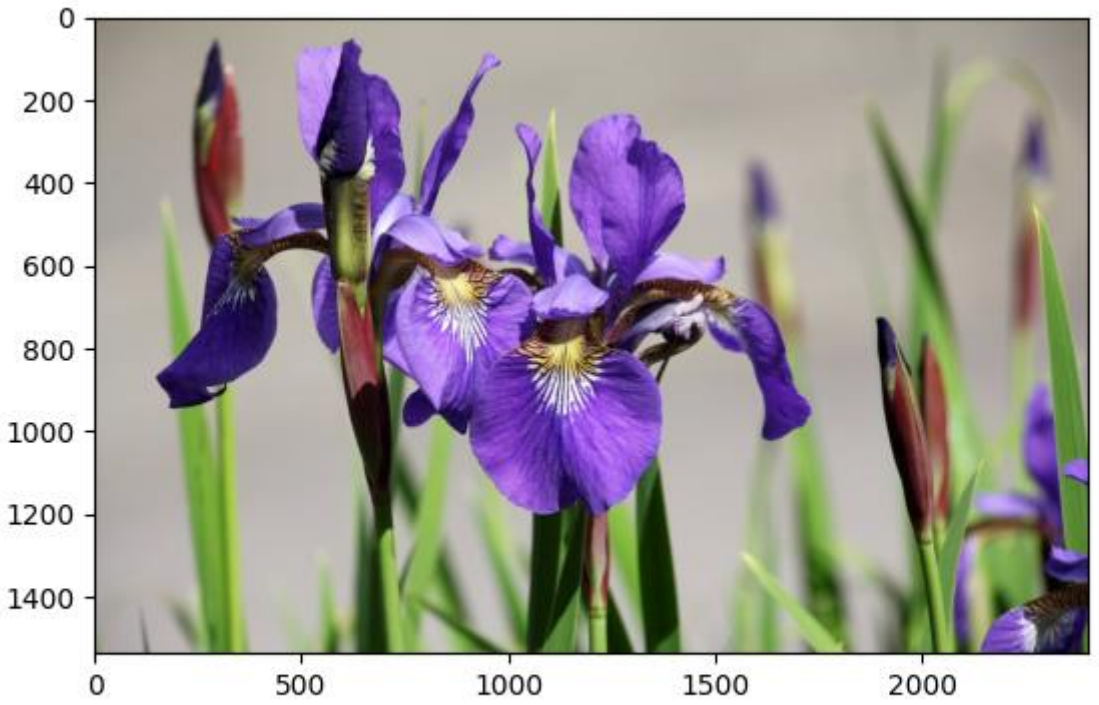
SciPy：图像处理（ndimage）


```
iris2 = ndimage.rotate(iris,angle = 180)
plt.imshow(iris2)
```



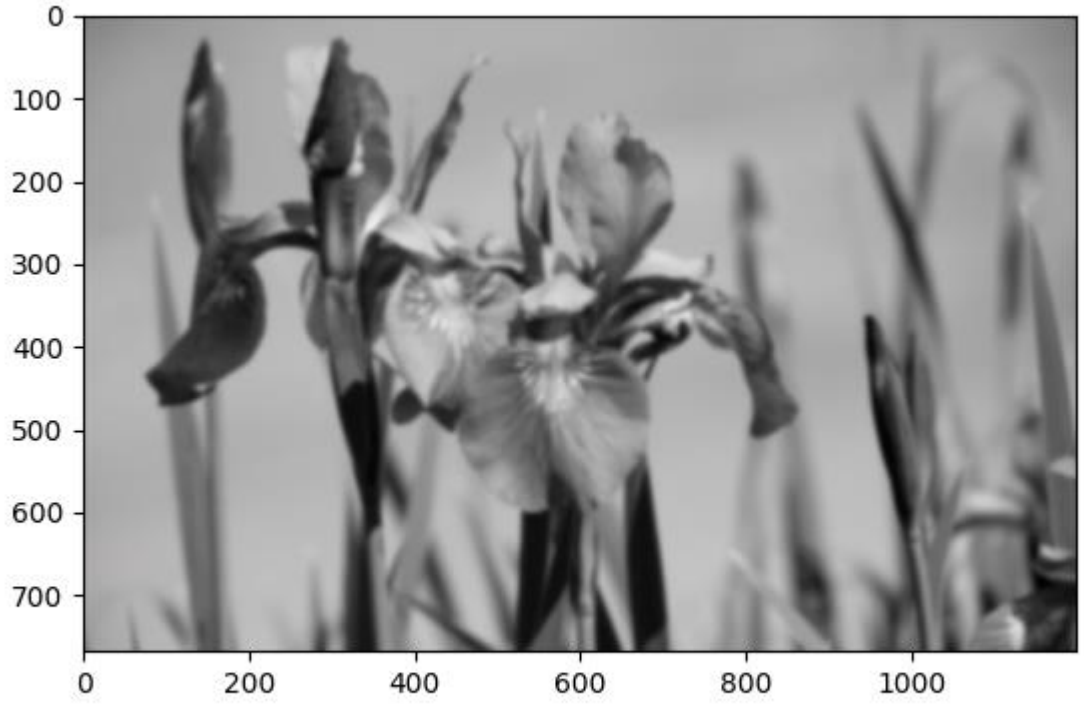
SciPy：图像处理（ndimage）


```
iris3 = ndimage.zoom(iris, zoom =[2,2,1])
plt.imshow(iris3)
plt.show()
```



SciPy：图像处理（ndimage）


```
iris4 = ndimage.gaussian_filter(iris, sigma = 3)
plt.imshow(iris4)
plt.show()
```



数据计算：总结

- Numpy：Python数据分析和机器学习的基础库，核心为数组运算
- Numpy：数组的概念、创建、运算、索引切片、增删改、重塑、统计分析
- Pandas：Python核心数据分析支持库
- Pandas主要数据结构：Series和DataFrame
- Pandas基本数据处理：创建、数据抽取、索引设置、数据增删改、排序