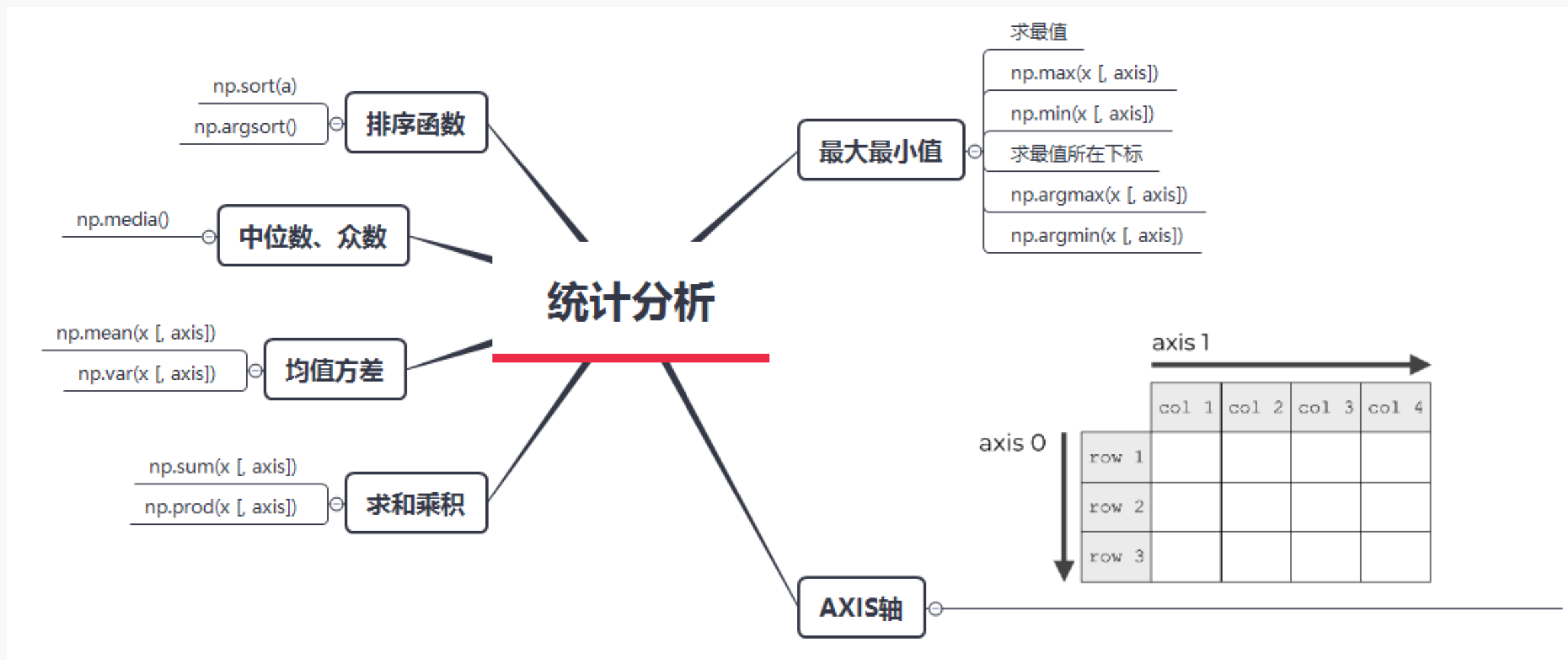


Python科学计算库Numpy之

06-统计函数





1、np.max()

作用：找出最大值

```
1 >>> import numpy as np
2 >>> x = np.array([[ -1, 3, 0], [ 5, 2, 7]])
3 >>> x
4 array([[ -1,  3,  0],
5         [ 5,  2,  7]])
6
7 >>> np.max(x)
8 7
```

2、np.min()

作用：找出最小值

```
1 >>> np.min(x)
2 -1
```

3、np.sum()

作用：计算所有元素之和

```
1 >>> np.sum(x)
2 16
```

4、np.prod()

作用：计算数组中所有元素之积

```
1 >>> np.prod(x)
2 0
```

5、np.std()

作用：计算元素的标准差

```
1 >>> np.std(x)
2 2.748737083745107
```

6、np.var()

作用：计算元素的方差

```
1 >>> np.var(x)
2 7.555555555555556
```

```
array([[ -1,  3,  0],
       [ 5,  2,  7]])
```

7、np.mean()

作用：计算元素的平均值

```
1 >>> np.mean(x)
2 2.6666666666666665
```

8、np.median()

作用：计算元素的中位数

```
1 >>> np.median(x)
2 2.5
```

最大值方法与函数

```
data = np.array([1,2,3])
```

data

1
2
3

data

1
2
3

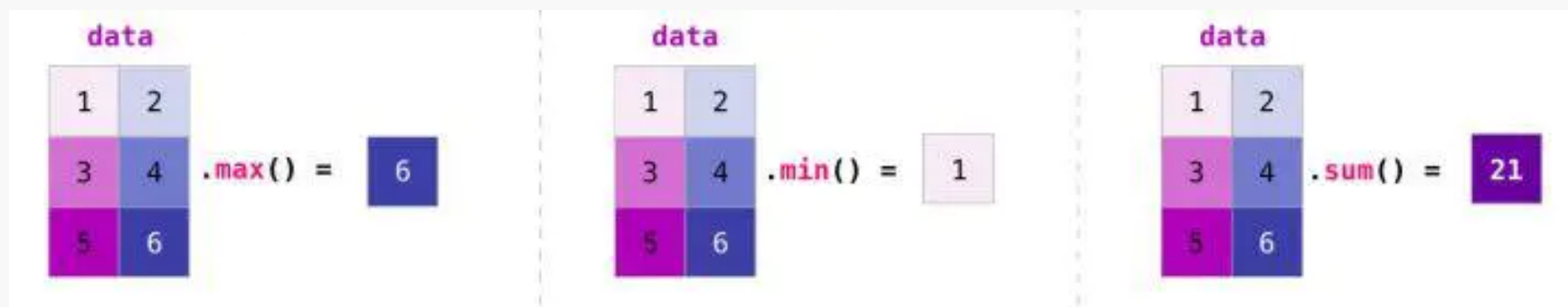
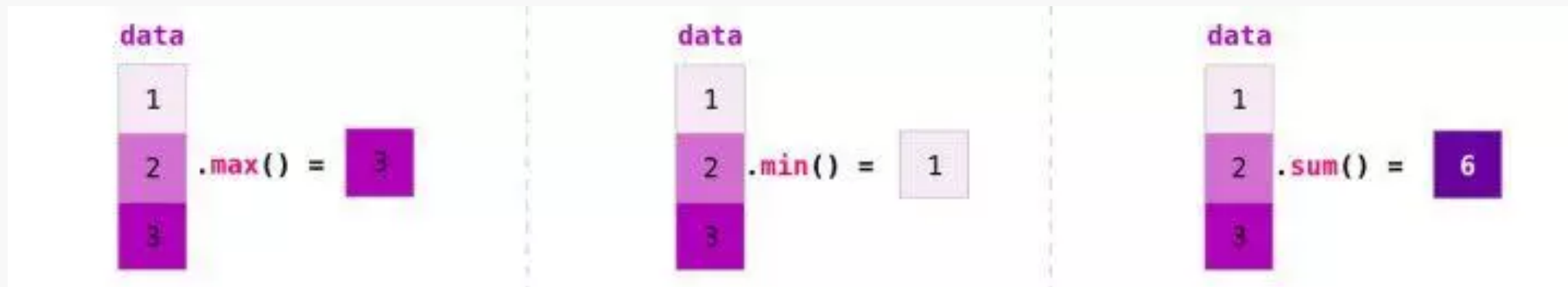
.max() = 3

data

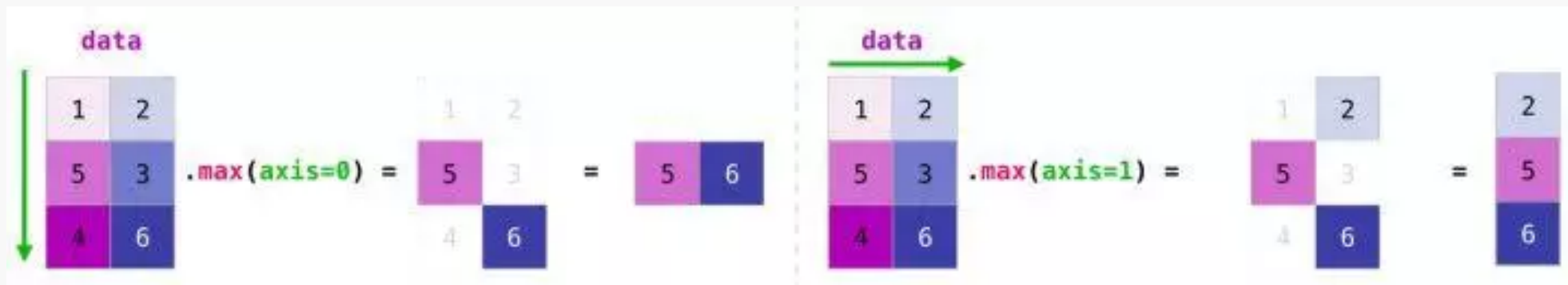
1
2
3

np.max()

最小、最大、求和



AXIS轴



axis 0 是沿行 (row) 向下的轴

axis 0



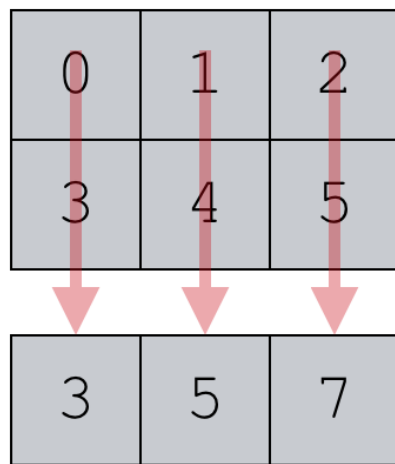
	col 1	col 2	col 3	col 4
row 1				
row 2				
row 3				

axis 1

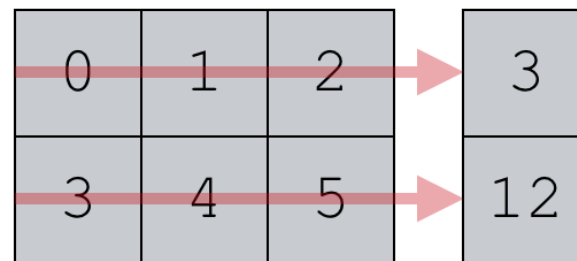


axis 1 是沿列 (columns) 横穿的轴

WHEN WE SET `axis = 0, np.sum()` COLLAPSES THE ROWS AND CALCULATES THE SUM

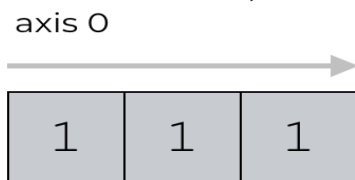


WHEN WE SET `axis = 1, np.sum()` COLLAPSES THE COLUMNS AND CALCULATES THE SUM



In a 1-d array, there is only one axis

一维NumPy数组只有一个轴（即axis=0）



均值、中位数、众数

平均数：

- (1)需要全组所有数据来计算
各数据的总和除以数据的个数；
- (2)易受数据中极端数值的影响。

$$\bar{x} = \frac{1}{n} (x_1 + x_2 + \cdots + x_n)$$

中位数：

- (1)把数据按顺序排列,取中间的位置；
- (2)不易受数据中极端数值的影响。

原始数据：10 20 60 80 70

重新排列：10 20 60 70 80

众数：

- (1)通过计数得到；
- (2)不易受数据中极端数值的影响
- (3)诸如“最佳”、“最受欢迎”、“最满意”等

原始数据：10 20 60 80 70 50

重新排列：10 20 50 60 70 80

均值与方差

甲、乙两名射击运动员在某场测试中各射击20次，测试成绩如下表，则选派哪位选手参加比赛？

甲的成绩				
环数	7	8	9	10
频数	4	6	6	4

乙的成绩		
环数	7	8
频数	6	4

分别计算甲乙两个人的方差可得，甲的方差小于乙的方差，结合方差的意义（反映数据波动的剧烈程度），可得甲最稳定。

甲的平均数 = $(7 \times 4 + 8 \times 6 + 9 \times 6 + 10 \times 4) \div 20 = 8.5$

乙的平均数 = $(7 \times 6 + 8 \times 4 + 9 \times 4 + 10 \times 6) \div 20 = 8.5$

$$S_{\text{甲}}^2 = [4 \times (7-8.5)^2 + 6 \times (8-8.5)^2 + 6 \times (9-8.5)^2 + 4 \times (10-8.5)^2] \div 20 = 1.05$$

$$S_{\text{乙}}^2 = [4 \times (8-8.5)^2 + 6 \times (7-8.5)^2 + 6 \times (10-8.5)^2 + 4 \times (9-8.5)^2] \div 20 = 1.45$$

$$\therefore S_{\text{甲}}^2 < S_{\text{乙}}^2$$

样本平均数公式：

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

样本方差公式：

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}$$

计算均方误差

均方误差MSE公式，它是监督机器学习、模型处理、回归问题的核心公式：

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

在NumPy中实现这一点是轻而易举：

```
error = (1/n) * np.sum(np.square(predictions - labels))
```

计算均方误差

```
error = (1/3) * np.sum(np.square(
    predictions - labels
))
```

predictions	labels
1	1
1	2
1	3

predictions和labels向量都包含三个值。这意味着n的值为3。在我们执行减法后，我们最终得到如下值：

```
error = (1/3) * np.sum(np.square(
    [0, -1, -2]
))
```

然后我们可以对矢量中的值进行平方：

```
error = (1/3) * np.sum(
    [0, 1, 4]
)
```

然后求和，结果就是均方误差：

```
error = (1/3) * 5
```

谢谢!

