**Numpy的数组各行，各列的求和，平均值，最大值，最小值，最大最小值差，标准差，方差等的计算**

*🔗 原文链接：* [*https://blog.csdn.net/qq\_18351157/a...*](https://blog.csdn.net/qq_18351157/article/details/103890205)

*⏰ 剪存时间：2022-09-27 12:16:57 (UTC+8)*

*✂️ 本文档由* [*飞书剪存*](https://www.feishu.cn/hc/zh-CN/articles/606278856233?from=in_ccm_clip_doc) *一键生成*

[**Numpy**](https://so.csdn.net/so/search?q=Numpy&spm=1001.2101.3001.7020) **的数组各行，各列的求和，平均值，最大值，最小值，最大最小值差，标准差，方差等的计算**

函数numnumpy.sum()可以算出ndarray [数组](https://so.csdn.net/so/search?q=%E6%95%B0%E7%BB%84&spm=1001.2101.3001.7020) 中所有元素的和，函数numpy.mean()可以算出ndarray数组中所有元素的平均值。  
默认的情况下是算出数组中所有元素的和与平均值，但是也可以使用参数 [axis](https://so.csdn.net/so/search?q=axis&spm=1001.2101.3001.7020) ，对行或列进行计算。

在此，对一下的内容进行说明。

* numpy.sum() 求和
* numpy.mean() 平均值
* numpy.min() 最小值/numpy.max() 最大值
* numpy.ptp() 最大值与最小值的差（最大值-最小值）
* numpy.std() 标准差/numpy.var() 方差
* 多维数组的参数axis

首先，准备一个3x4的数组。

|  |
| --- |
| Python import numpy as np  a = np.arange(12).reshape(3, 4) print(a.shape) print(a) # (3, 4) # [[ 0 1 2 3] # [ 4 5 6 7] # [ 8 9 10 11]] |

**numpy.sum() 求和**

把刚刚生成的数组a放入函数np.sum()中，返回得到数组中所有元素的和。

|  |
| --- |
| Python print(np.sum(a)) # 66 |

参数axis＝0时，返回数组各列的和，参数axis＝1时，返回数组各行的和。

|  |
| --- |
| Python print(np.sum(a, axis=0)) print(np.sum(a, axis=1)) # [12 15 18 21] # [ 6 22 38] |

其实不使用numpy的函数也是可以的，因为ndarndarray数组中也有相同的求和方法sum()。并且也可以通过参数axis指定行或列。

|  |
| --- |
| Python print(a.sum()) # 66  print(a.sum(axis=0)) print(a.sum(axis=1)) # [12 15 18 21] # [ 6 22 38] |

**numpy.mean() 平均值**

numpy.mean()的使用方法与numpy.sum()相同，也可以通过参数axis指定行或列。

|  |
| --- |
| Python print(np.mean(a)) # 5.5  print(np.mean(a, axis=0)) print(np.mean(a, axis=1)) # [ 4. 5. 6. 7.] # [ 1.5 5.5 9.5] |

和sum()一样，ndarray数组中也有相同的求平均值的方法mean()。并且也可以通过参数axis指定行或列。

|  |
| --- |
| Python print(a.mean()) # 5.5  print(a.mean(axis=0)) print(a.mean(axis=1)) # [ 4. 5. 6. 7.] # [ 1.5 5.5 9.5] |

**numpy.min() 最小值/numpy.max() 最大值**

在分别使用numpy.min()和numpy.max()求数组中元素的最小值与最大值的时候，也可以通过参数axis指定行或列。并且，为了使用方便，还可以直接使用函数numpy.amin()和numpy.amax()进行计算，所得到的结果是一样的。

|  |
| --- |
| Python print(np.min(a)) print(np.min(a, axis=0)) print (np.amin(a,0)) # 0 # [0 1 2 3] # [0 1 2 3]  print(a.max()) print(a.max(axis=1)) print(np.amax(a,1)) # 11 # [ 3 7 11] # [ 3 7 11] |

**numpy.ptp() 最大值与最小值的差（最大值-最小值）**

函数numpy.ptp()可以返回得到数组中最大值与最小值之间的差（最大值-最小值），也可以通过参数axis指定行或列。

|  |
| --- |
| Python print(np.ptp(a)) #11  print(np.ptp(a, axis=1)) #[3 3 3]  print(np.ptp(a, axis=0)) #[8 8 8 8] |

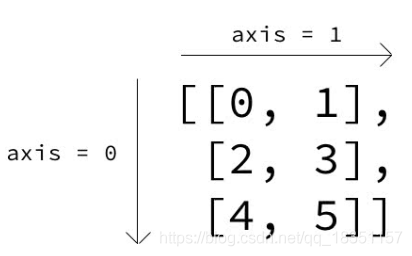
**numpy.std() 标准差/numpy.var() 方差**

求标准差和方差的函数分别为numpy.std()和numpy.var()。使用方法和之前相同，也可以通过参数axis指定行或列。

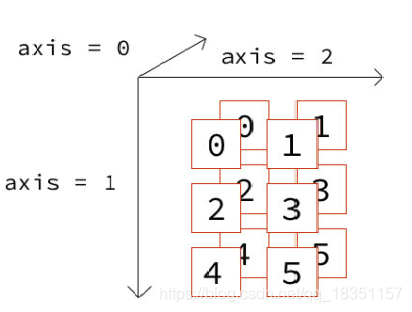
|  |
| --- |
| Python print(np.std(a)) #3.452052529534663  print(np.var(a)) #11.916666666666666 |

**多维数组的参数axis**

2维数组时，参数axis＝0或1时，分别指定行或列进行计算。下面对多维数组参数axis的使用方法进行简单的说明介绍。



2维数组时的参数axis。



3维数组时的参数axis。

首先，准备一个2x3x4的数组b。

|  |
| --- |
| Python b = np.arange(24).reshape(2, 3, 4) print(b.shape) print(b) # (2, 3, 4) # [[[ 0 1 2 3] # [ 4 5 6 7] # [ 8 9 10 11]] # # [[12 13 14 15] # [16 17 18 19] # [20 21 22 23]]] |

当axis＝0时，结果维3x4的数组。

|  |
| --- |
| Python print(b.sum(axis=0)) # [[12 14 16 18] # [20 22 24 26] # [28 30 32 34]] |

当axis＝1时，结果维2x4的数组。

|  |
| --- |
| Python print(b.sum(axis=1)) # [[12 15 18 21] # [48 51 54 57]] |

当axis＝2时，结果维2x3的数组。

|  |
| --- |
| Python print(b.sum(axis=2)) # [[ 6 22 38] # [54 70 86]] |

asix还可以进行双数值的指定。结果如下。

|  |
| --- |
| Python print(b.sum(axis=(0, 1))) # [60 66 72 78]  print(b.sum(axis=(0, 2))) # [ 60 92 124]  print(b.sum(axis=(1, 2))) # [ 66 210] |