



链滴

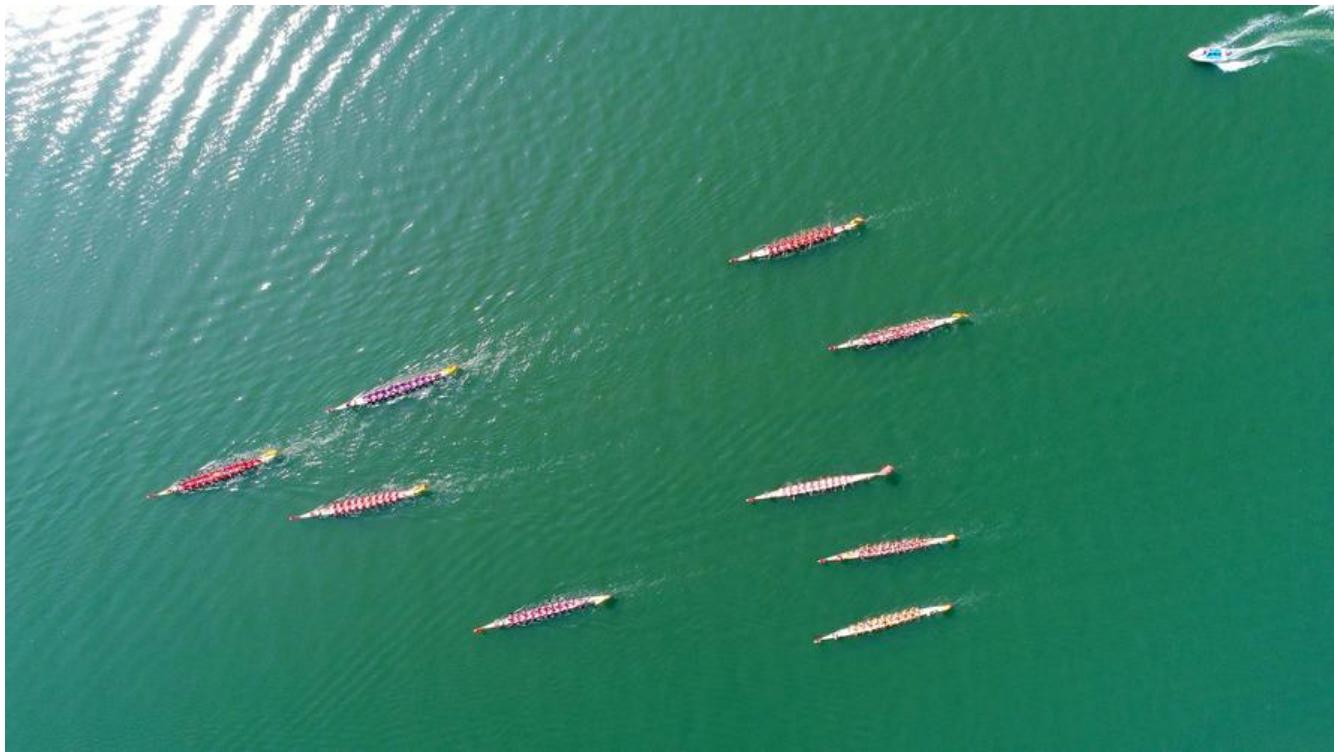
Python 数据科学 (1)——NumPy(3.Arrays 逻辑篇)

作者: [hsxian](#)

原文链接: <https://ld246.com/article/1556006965729>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



比较和排序

我们知道，在算法中，排序总是离不开比较的，那么在 numpy 中是否也是如此呢？让我们来看看。

比较、掩码和布尔逻辑

比较

```
In [1]: import numpy as np  
In [2]: a = np.array([3,4,5])
```

```
In [3]: a < 4  
Out[3]: array([ True, False, False], dtype=bool)
```

```
In [4]: (2 * a) == (a ** 2)  
Out[4]: array([False, False, False], dtype=bool)
```

numpy 中比较函数和对应运算符等价

运算符 应函数	对应函数	运算符
<code>==</code> <code>qual</code>	<code>np.equal</code>	<code>!=</code> <code>np.not_</code>
<code><</code> <code>ua</code>	<code>np.less</code>	<code><=</code> <code>np.less_e</code>
<code>></code> <code>ter_e</code>	<code>np.greater</code>	<code>>=</code> <code>np.gre</code>

使用比较运算符，可以实现一些过滤操作：

```
In [5]: a = np.array([-5,-4,-3, 0,3,4,5])
```

```
In [6]: np.count_nonzero(a < 0) # 小于0的元素个数  
Out[6]: 3
```

```
In [8]: np.sum(a < 0)# 小于0的元素个数  
Out[8]: 3
```

```
In [9]: np.any(a > 0)# 是否存在大于0的元素  
Out[9]: True
```

布尔逻辑

同样的， numpy 矩阵支持布尔逻辑运算

```
In [17]: np.sum((a > -1) & (a < 1))# 位于-1到1之间的元素个数  
Out[17]: 1
```

布尔矩阵作为掩码

类似于过滤，我们可以使用布尔矩阵作为矩阵索引，获取相应的数据

```
In [18]: a = np.array([[5,0,3,3],[7,9,3,5], [2,4,7,6]])
```

```
In [19]: a[a<5]  
Out[19]: array([0, 3, 3, 3, 2, 4])
```

排序

numpy 中的排序

快速排序： sort 和 argsort

```
In [20]: x = np.array([2, 1, 4, 3, 5])  
In [21]: x.sort()  
In [22]: x  
Out[22]: array([1, 2, 3, 4, 5])#可以看出， 快速排序是原址排序
```

```
In [23]: x = np.array([2, 1, 4, 3, 5])  
In [24]: x.argsort()  
In [26]: i = x.argsort()#只返回索引  
In [27]: i  
Out[27]: array([1, 0, 3, 2, 4])
```

```
In [28]: x[i]#利用排序后的index， 我们可获得排序结果  
Out[28]: array([1, 2, 3, 4, 5])
```

部分排序

寻找 k 个最小值

```
In [34]: x = np.array([7, 2, 3, 1, 6, 5, 4])
```

```
...: ...: np.partition(x, 3)
```

```
Out[34]: array([2, 1, 3, 4, 6, 5, 7])
```

```
In [35]: rand = np.random.RandomState(42)
```

```
...: X = rand.randint(0, 10, (4, 6))
```

```
In [36]: np.partition(X, 2, axis=1)
```

```
Out[36]:
```

```
array([[3, 4, 6, 7, 6, 9],  
       [2, 3, 4, 7, 6, 7],  
       [1, 2, 4, 5, 7, 7],  
       [0, 1, 4, 5, 9, 5]])
```