add pandas
由 LiuZhangshu 提交于 17 分钟前

8285c391

numpy.md 5.5 KB

numpy

1.创建array,并定义数据类型

```
## 创建array,并定义数据类型(int,float)
import numpy as np
a = np.array([[2,2,3,4],
              [4,6,8,9]],dtype=np.int)
print(a)
print(a.dtype)
## 定义全0矩阵
a = np.zeros((3,5))
print(a)
## 定义全1
a = np.ones((3,5))
print(a,"\n")

##定义空
b = np.empty((3,5),dtype=float)
print("null is \n ",b)

##定义序列arange(start,stop,step)
a = np.arange(10,20,3)
print(a)
## 修改形状 array.reshape
a = np.arange(12).reshape((3,4))
print(a)
## 分段（start, end,split num）
print(np.linspace(1,10,5))
```

```
[[2 2 3 4]
 [4 6 8 9]]
int32
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
[[1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]]

null is
  [[1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]]
[10 13 16 19]
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
[ 1.    3.25  5.5   7.75 10.  ]
```

2.基础运算

```
##基础运算
import numpy as np
a = np.array([10,20,30,40])
b = np.arange(4)
## 减法
c = a - b
print(a)
print(b)
print(c)
## 指数**
print(b**3)
## 三角函数
print( 10*np.sin(a))

## <,>,==
print(b<3)
## 矩阵运算
a = np.array([[1,1],
             [0,1]])
b = np.arange(4).reshape(2,2)
##元素逐个相乘
c = a*b
## 矩阵的乘法
c_dot = np.dot(a,b)
c_dot2= a.dot(b)
print(c,"\n")
print(c_dot)
```

```
[10 20 30 40]
[0 1 2 3]
[10 19 28 37]
[ 0  1  8 27]
[-5.44021111  9.12945251 -9.88031624  7.4511316 ]
[ True  True  True False]
[[0 1]
 [0 3]]

[[2 4]
 [2 3]]
```

```
## 随机生成(0,1)参数为形状
a = np.random.random((2,4))
print(a)
## axis=1行 axis=0列
print(np.sum(a,axis=0))
print(np.min(a))
print(np.max(a))
```

```
[[0.0063358  0.07472742 0.91447883 0.55993582]
 [0.31040072 0.61862646 0.32844487 0.88354428]]
[0.31673652 0.69335388 1.2429237  1.4434801 ]
0.006335799942833953
0.9144788310361289
```

3.基本属性查看

```
##基本属性查看
import numpy as np
##数组转换为矩阵
A = np.arange(2,14).reshape((3,4))
print('matrix is\n',A)
print('\nnumber of dim:',A.ndim)
print('\nshape:',A.shape)
print('\nsize:',A.size)
print('\nmin index: ',np.argmin(A))
print('\nmax index: ',np.argmax(A))
print('\nmean:' ,np.mean(A,axis=0))
print('\nmean:' ,np.average(A))
print('\nmedian:',np.median(A,axis=0))
print('\n累加:',np.cumsum(A))
print('\n累差: ',np.diff(A))
print('\nsort', np.sort(A))
print('转置',np.transpose(A))
print('转置',A.T)
print('截取, 小于5=5大于9=9',np.clip(A,5,9))
```

```
matrix is
 [[ 2  3  4  5]
 [ 6  7  8  9]
 [10 11 12 13]]

number of dim: 2

shape: (3, 4)

size: 12

min index:  0

max index:  11

mean: [6. 7. 8. 9.]

mean: 7.5

median: [6. 7. 8. 9.]

累加: [ 2  5  9 14 20 27 35 44 54 65 77 90]

累差:  [[1 1 1]
 [1 1 1]
 [1 1 1]]

sort [[ 2  3  4  5]
 [ 6  7  8  9]
 [10 11 12 13]]
转置 [[ 2  6 10]
 [ 3  7 11]
 [ 4  8 12]
 [ 5  9 13]]
转置 [[ 2  6 10]
 [ 3  7 11]
 [ 4  8 12]
 [ 5  9 13]]
截取，小于5=5大于9=9 [[5 5 5 5]
 [6 7 8 9]
 [9 9 9 9]]
```

4.numpy的索引

```
## numpy的索引
import numpy as np

A = np.arange(3,15).reshape((3,4))
print('A is',A)
print('A[2] is',A[2])
print('A[2][1] is ',A[2][1])
print('第一列的所有数',A[:,1])
print('第二行的所有数',A[2,:])
print('第一行从1到2',A[1,1:3])
#循环打印
i=0
for row in A :
    i=i+1
    print('第',i,'行',row)
i=0

for column in A.T :
    i = i +1
    print('第',i,'列',column)
#迭代元素
#转换矩阵为数组
print (A.flatten())
## flat迭代器
for item in A.flat :
    print(item)
```

```
A is [[ 3  4  5  6]
 [ 7  8  9 10]
 [11 12 13 14]]
A[2] is [11 12 13 14]
```

```
A[2][1] is  12
第一列的所有数 [ 4  8 12]
第二行的所有数 [11 12 13 14]
第一行从1到2 [8 9]
第 1 行 [3 4 5 6]
第 2 行 [ 7  8  9 10]
第 3 行 [11 12 13 14]
第 1 列 [ 3  7 11]
第 2 列 [ 4  8 12]
第 3 列 [ 5  9 13]
第 4 列 [ 6 10 14]
[ 3  4  5  6  7  8  9 10 11 12 13 14]
3
4
5
6
7
8
9
10
11
12
13
14
```

5.array的合并、分割

```
## array的合并
A = np.array([1,1,1])
B = np.array([2,2,2])
##上下
C = np.vstack((A,B))
print(C)
##左右
D=np.hstack((A,B))
print(D)
##多个合并
C = np.concatenate((A,B,B,A),axis=0)
print (C)
```

```
[[1 1 1]
 [2 2 2]]
[1 1 1 2 2 2]
[1 1 1 2 2 2 2 2 2 1 1 1]
```

```
## array 分割
A = np.arange(12).reshape((3,4))
print(A)
#分割
print(np.split(A,2,axis=1))
#不等分割
print(np.array_split(A,3,axis=1))
#纵向分割
print(np.hsplit(A,2))
#横向分割
print(np.vsplit(A,3))
```

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
[array([[0, 1],
       [4, 5],
       [8, 9]]), array([[ 2,  3],
       [ 6,  7],
       [10, 11]])]
[array([[0, 1],
       [4, 5],
       [8, 9]]), array([[ 2],
       [ 6],
       [10]]), array([[ 3],
       [ 7],
       [11]])]
[array([[0, 1],
       [4, 5],
       [8, 9]]), array([[ 2,  3],
       [ 6,  7],
       [10, 11]])]
[array([[0, 1, 2, 3]]), array([[4, 5, 6, 7]]), array([[ 8,  9, 10, 11]])]
```

6.copy deepcopy

```
import numpy as np
a = np.array([1,2,3,4,5,6]).reshape((2,3))
b = a ##point same address
b = a.copy() ##只关联值
```