

## Python-numpy常用函数

### import numpy as np

#### 1. multiply

```
[in] x1 = [1,2,3]
      x2 = [4,5,6]
      np.multiply(x1,x2)
```

```
[out] [4,10,18]
```

multiply函数得到的结构是对应位置上面元素的乘积

#### 2. std标准差, var方差

```
[in] b = [1,3,5,6]
      np.var(b)
      np.std(b)
      l1 = [[1,2,3,4,5,6],[3,4,5,6,7,8]]
      np.var(l1[0])
      np.var(l1,0) # 第二个参数为0,表示按列求方差
      np.var(l1,1) # . . . . .,表示按行求方差
```

```
[out] 3.6875
      1.920286436967152
      2.9166666666666665
      [1. 1. 1. 1. 1. 1. ]
      [2.916666667 2.916666667]
```

#### 3. mean

```
[in] b = [1,3,5,6]
      l1 = [[1,2,3,4,5,6],[3,4,5,6,7,8]]
      np.mean(b)
      np.mean(l1) #求全部元素均值
      np.mean(l1,0) #按列求均值
      np.mean(l1,1) #按行求均值
```

```
[out] 3.75
      4.5
      [2. 3. 4. 5. 6. 7.]
      [3.5 3.5]
```

mean函数得向量的均值

#### 4. sum

```
[in] x = [[0,1,2],[2,1,0]]
      b = [1,3,5,6]
      np.sum(b)
      np.sum(x)

[out] 15
      6
```

sum求向量的和。也可以求矩阵所以元素的和

## 5. cov()

```
[in] b = [1,3,5,6]
      print cov(b)

[out] array(4.91666667)
```

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$$

$$S = \frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{X})^2$$

$$C = \frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y})$$

cov求的是样本协方差

- 公式一 样本均值
- 公式二 样本方差
- 公式三 样本协方差

```
[in] x = [[0,1,2],[2,1,0]]
      print cov(x)

[out] array([[ 1., -1.],
             [-1.,  1.]])
```

cov的参数是矩阵，输出结果也是矩阵！输出的矩阵为协方差矩阵！

## 6. corrcoef 该函数得到相关系数矩阵

```
[in] vc = [1,2,39,0,8]
      vb = [1,2,38,0,8]
      np.corrcoef(vc,vb)

[out] [[1.          0.99998623]
       [0.99998623  1.          ]]
```

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X - \mu_X)(Y - \mu_Y))}{\sigma_X \sigma_Y},$$

其中， $E$ 是数学期望， $\text{cov}$ 表示协方差， $\sigma_X$ 和 $\sigma_Y$ 是标准差。

因为 $\mu_X = E(X)$ ， $\sigma_X^2 = E(X^2) - E^2(X)$ ，同样地，对于 $Y$ ，可以写成

$$\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)} \sqrt{E(Y^2) - E^2(Y)}}。$$

相关系数公式：

## 7. vdot向量的点积

```
# vdot 返回两向量的点积
[in] l1 = [1,2,3]
      l2 = [4,5,6]
      np.vdot(l1,l2)

[out] 32
```

对应位置相乘求和

## 8. mat

```
[in] l1 = [1,2,3]
      l2 = [4,5,6]
      l1 = [l1,l2]
      np.vdot(l1,l2)
      np.mat(l1)

[out] 32
      [[1 2 3]
       [4 5 6]]
```

mat函数把列表转换成矩阵形式。

## 9. shape

```
#矩阵有一个shape属性，是一个(行,列)形式的元组
[in] a = np.array([[1,2,3],[4,5,6]])
      a.shape

[out] (2,3)
```

## 10.ones

```
#返回按要求的矩阵
[in] ones = np.ones((2,1))
      ones

[out] array([[1.],
             [1.]])
```

ones返回指定行列数的全意矩阵

## 11. range

```
[in] for i in range(3):
      print i
      test = [1,2,3,4]
      test[:]
      test[2:3]

      for i in range(2,5):
          print i

[out] 0
      1
      2
      [1,2,3,4]
      [3]
      2
      3
      4
```

range用于循环中,参数为一个整数的话,可循环遍历小于该参数的值。两个参数,则循环遍历两个整数之间的值。

test[:]则表示获取test列表中的所有元素 test[2:3] 则表示获取从第2个位置到第3个位置间的元素

## 12.strptime

```
import time
from datetime import datetime,date

dd = datetime.strptime('2019-11-08T10:53:49.875z', "%Y-%m-%dT%H:%M:%S.%fZ")
time.mktime(dd.timetuple())

[out]1573181629.0
```

## 13. tuple和数组转换成字符串

```
[in] tuple = (1,2,3)
      tuple[len(tuple)-1]
      tuple[-1]
      9.99.__repr__()
      str(9.99)

[out] 3
      3
      '9.99'
      '9.99'
```

tuple是一个元组，访问元素的时候，可以通过[index]这种方式访问。  
访问最后一个元素的时候，可以通过[-1]访问。  
另外，数字转换成字符串有两种方式：

```
__repr__()
str()
```

## 14. transpose和.T

```
[in] aa = [[1],[2],[3]]
      aa = np.mat(aa)#将列表转换成矩阵，并存放在aa中
      aa.transpose()#将矩阵进行转置
      aa.T#将矩阵进行转置

[out] matrix([[1],
              [2],
              [3]])
      matrix([[1, 2, 3]])
      matrix([[1, 2, 3]])
```

## 15. zeros() ones()

```
[in] np.zeros((2,1))
      np.ones((2,3))

[out] [[ 0.]
       [ 0.]]
       [[ 1.  1.  1.]
       [ 1.  1.  1.]]
```

zeros返回指定行列全零矩阵  
ones返回指定行列的全一矩阵

## 16. 列表 数组 linspace

*#列表和数组的区别*

*#列表: [1,2,3,4]*

*#数组: [1 2 3 4]*

数组中间元素没有分隔符,列表逗号分隔

```
[in] np.linspace(0,3,6)
array([0. , 0.6, 1.2, 1.8, 2.4, 3. ])
```

## 17. argsort排序索引

```
[in] ary=np.array(np.zeros(4))
    ary[0]=0.1
    ary[1]= 0.6
    ary[2]= 0.5
    ary[3]= 0.7
    #有-号, 降序排列
    #无-号, 升序排列
    sortindex = np.argsort(-ary)
    for id in sortindex:
        print '索引: ',id
    for i in ary:
        print i
```

```
[out] 索引: 3
    索引: 1
    索引: 2
    索引: 0
    0.1
    0.6
    0.5
    0.7
```