

- `A=np.array(data)` 把数据变成矩阵，如果要用浮点型，data整型数据要加.0
- `A[::-1]` 将A逆序
- `A[[a,b],[c,d]]` 取矩阵A的第a行b列和c行d列
`A[a:b,c:d]` 取矩阵A的a到(b-1)行，c到(d-1)列
`A[[a,b]][:[c,d]]` 分两步 取矩阵A的第a, b行和c, d列
(可以用 `range(start,end+1,step)`) 整数
`np.arange(start,end+step,step)` 浮点型
- `np.transpose(数组)` .T也行 转置
`np.transpose([a])` !!! 一维矩阵转置
- `np.linalg.inv(A)` 计算A的逆矩阵
`np.linalg.pinv(A)` 如果矩阵A不是正方形则用这种方法，svd算法，（自己按网上公式写有时候不对，也不知道为啥，所以还是用库函数吧）
- 矩阵乘法 `dot(a,b)` @也行
点乘直接 * 就行
- `np.hstack((a,b))` 把a和b进行矩阵横着拼接 双括号
`np.vstack((a,b))` 是纵着拼接
- `A[:,a]` 取A的a列
`A[:,a]` 取A的a列变成行(前边写的转置)
- 矩阵.shape没有括号 输出行数列数 一维的行向量只输出 列数 长度用 `len(矩阵)`
`A.reshape(a,b)` 变形别的维度
- `A.max(axis=0)` 返回每列最大值 (返回结果为数组)
`A.argmax(axis=0)` 返回每列最大值的索引 1为行求均值
`A.mean(axis=1))` #行
`A.mean(axis=0))` #列
- `np.zeros(5)` 生成一行五列矩阵
`np.zeros((3,4))` 生成四行四列矩阵 (双括号) (一律双括号就完事了)
`np.random.random((1,3))` 生成一行三列随机矩阵

- `A.reshape(a,-1)` 转化成a行 (横着顺次数) (变换完还是二维矩阵, 不是数组)
`A.reshape(-1,a)` 转换成a列 (横着顺次数) (要变数组需要加个[0,:])
`np.reshape(A,(a,b))` 把A变成a行b列
- `np.ravel(A)` 把A拉成一维向量
`np.concatenate((A,B))` 把两个向量放在一起 (双括号)
- `A = np.insert(B,0,np.ones(3),axis=1)` 第一个0代表第几行/第几列 然后是size axis=1代表列 (不写直接转成一维)
- `np.linspace(a,b,c)` 创建等差数列, 从a到b一共c个点
- `y = np.delete(x, i, axis = 0)` 删除矩阵x第i行并赋值给y
- `np.clip(arr,a,b)` 将数组arr限制在[a,b]范围内, 如果有数小于a就等于a, 如果有数大于b就等于b
- `np.where(条件)` 返回满足“条件”的索引 (返回一个数组) 自己看吧, n维的返回n个
`np.where(条件, A, B)` 满足条件执行A, 不满足执行B

```

1  # 例如通过where()函数将a数组中负值设为0, 正值不变
2
3  >>> a = np.array([[1,2,3],[4,5,-1],[-2,1,-3]])
4  >>> a
5  array([[ 1,  2,  3],
6         [ 4,  5, -1],
7         [-2,  1, -3]])
8  >>> b = np.where(a>0,a,0)
9  >>> b
10 array([[1, 2, 3],
11         [4, 5, 0],
12         [0, 1, 0]])

```

- `np.sort(A)` 给矩阵A排序 (从小到大)
`np.argsort(A)` 返回矩阵A的排序索引 (从小到大)
- `data = np.fft.fft(filtedData)/len(filtedData)` 后边除的东西是归一化, 快速傅里叶变换
- `np.savetxt(path, A, fmt='%.2f', delimiter=',')` 将矩阵A写入path的文件中, 浮点型小数点后两位, 分隔符为','
`np.loadtxt(path)`, 从文件读取

- `A = np.unique(data)` 删除data数组的重复元素

`A = np.unique(data, axis=0)` 删除重复列

`A = np.unique(data, axis=1)` 删除重复行

```
1 data = np.array([[1,8,3,3,4],
2                 [1,8,9,9,4],
3                 [1,8,3,3,4]])
4 # 删除整个数组的重复元素
5 uniques = np.unique(data)
6 print( uniques)
7 array([1, 3, 4, 8, 9])
8 # 删除重复行
9 uniques = np.unique(data, axis=0)
10 print( uniques)
11 array([[1,8,3,3,4],
12        [1,8,9,9,4]])
13 # 删除重复列
14 uniques = np.unique(data, axis=1)
```

- `np.isnan(number)` 判定数值是不是Nan（无法表示的数）比如除以0

- `np.around(A,decimals=2)` 保留数组A小数点后2位, `decimals=` 可不写