In [229]:
```
#pip install numpy
```

In [230]:
```
import numpy as np
```

In [231]:
```
arr1 = np.arange(1000)
```

In [232]:
```
%time arr2 = arr1 * 2
```

```
Wall time: 999 µs
```

In [233]:
```
lst1 = list(range(1000))
```

In [234]:
```
%time lst2 = [i * 2 for i in lst1]
```

```
Wall time: 0 ns
```

In [235]:
```
x = np.array([[1, 2] , [3, 4]])
```

In [236]:
```
x
```

Out[236]:
```
array([[1, 2],
       [3, 4]])
```

In [237]:
```
x.ndim
```

Out[237]:
```
2
```

In [238]:
```
x.shape
```

Out[238]:
```
(2, 2)
```

In [239]:
```
x.dtype
```

Out[239]:
```
dtype('int32')
```

In [240]:
```
x * 3
```

Out[240]:
```
array([[ 3,  6],
       [ 9, 12]])
```

In [241]:
```
x + x
```

Out[241]:
```
array([[2, 4],
       [6, 8]])
```

In [242]:
```
x * x
```

Out[242]:
```
array([[ 1,  4],
       [ 9, 16]])
```

In [243]:
```
1 / x
```

Out[243]:
```
array([[1.        , 0.5       ],
       [0.33333333, 0.25      ]])
```

In [244]:
```
x ** 5
```

Out[244]:
```
array([[   1,   32],
       [ 243, 1024]], dtype=int32)
```

In [245]:
```
y = np.array([[-2, 6],[8, 3]])
```

In [246]:
```
y
```

Out[246]:
```
array([[-2,  6],
       [ 8,  3]])
```

In [247]:
```
x
```

Out[247]:
```
array([[1, 2],
       [3, 4]])
```

In [248]:
```
x < y
```

Out[248]:
```
array([[False,  True],
       [ True, False]])
```

In [249]:
```
lst = [2, 4.5 , 6]
type(lst)
```

Out[249]:
```
list
```

In [250]:
```
arr = np.array(lst)
arr
```

Out[250]:
```
array([2. , 4.5, 6. ])
```

In [251]:
```
arr.dtype
```

Out[251]:
```
dtype('float64')
```

In [252]:
```
lst2 = [[1, 2, 3] , [4, 5, 6]]
arr2 = np.array(lst2)
arr2
```

Out[252]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [253]:
```
arr2.shape
```

Out[253]:
```
(2, 3)
```

In [254]:
```
arr2.ndim
```

Out[254]:
```
2
```

In [255]:
```
arr  = np.zeros(3)
arr
```

Out[255]:
```
array([0., 0., 0.])
```

In [256]:
```
arr = np.full(3, 2)
arr
```

Out[256]:
```
array([2, 2, 2])
```

In [257]:
```
x = np.full((3, 2), 4)
x
```

Out[257]:
```
array([[4, 4],
       [4, 4],
       [4, 4]])
```

In [258]:
```
a = np.identity(3)
a
```

Out[258]:
```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

In [259]:
```
lst = [1, 2]
arr = np.array(lst, dtype=np.int32)
arr.dtype
```

Out[259]:
```
dtype('int32')
```

In [260]:
```
arr2 = np.array(lst, dtype=np.float64)
arr2.dtype
```

Out[260]:
```
dtype('float64')
```

In [261]:
```
arr2
```

Out[261]:
```
array([1., 2.])
```

In [262]:
```
x = np.array(lst)
x.dtype
```

Out[262]:
```
dtype('int32')
```

In [263]:
```
y = x.astype(np.float64)
y.dtype
```

Out[263]:
```
dtype('float64')
```

In [264]:
```
lst2 = [1.6, 3.2, 0.3]
a = np.array(lst2)
a
```

Out[264]:
```
array([1.6, 3.2, 0.3])
```

In [265]:
```
a.astype(np.int32)
```

Out[265]:
```
array([1, 3, 0])
```

indexing and slicing

In [266]:
```
arr = np.arange(8)
arr
```

Out[266]:
```
array([0, 1, 2, 3, 4, 5, 6, 7])
```

In [267]:
```
arr[3]
```

Out[267]:
```
3
```

In [268]:
```
arr[2:5]
```

Out[268]:
```
array([2, 3, 4])
```

In [269]:
```
arr[2:5] = 13
arr
```

Out[269]:
```
array([ 0,  1, 13, 13, 13,  5,  6,  7])
```

In [270]:
```
x = arr[2:6]
x
```

Out[270]:
```
array([13, 13, 13,  5])
```

In [271]:
```
x[1] = 17
x
```

Out[271]:
```
array([13, 17, 13,  5])
```

In [272]:
```
arr
```

Out[272]:
```
array([ 0,  1, 13, 17, 13,  5,  6,  7])
```

In [273]:
```
x[:] = 64
x
```

Out[273]:
```
array([64, 64, 64, 64])
```

In [274]:
```
arr
```

Out[274]:
```
array([ 0,  1, 64, 64, 64, 64,  6,  7])
```

In [275]:
```
arr = np.arange(8)
arr
```

Out[275]:
```
array([0, 1, 2, 3, 4, 5, 6, 7])
```

In [276]:
```
arr[2:6].copy()
```

Out[276]:
```
array([2, 3, 4, 5])
```

In [277]:
```
a = np.array([[1, 2, 3] , [4, 5, 6], [7, 8, 9]])
a
```

Out[277]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [278]:
```
a[2]
```

Out[278]:
```
array([7, 8, 9])
```

In [279]:
```
a[0]
```

Out[279]:
```
array([1, 2, 3])
```

In [280]:
```
a[0][2]
```

Out[280]: 3

In [281]:
```
a[2][1]
```

Out[281]: 8

In [282]:
```
a[2][2]
```

Out[282]: 9

In [283]:
```
a[2, 2]
```

Out[283]: 9

In [284]:
```
a.ndim
```

Out[284]: 2

arr3d

In [285]:
```
a = np.array([[[1, 2, 3] , [4, 5, 6]],[[7, 8, 9],[10, 11, 12]]])
a
```

Out[285]:
```
array([[[ 1,  2,  3],
        [ 4,  5,  6]],

       [[ 7,  8,  9],
        [10, 11, 12]]])
```

In [286]:
```
a.ndim
```

Out[286]: 3

In [287]:
```
a.shape
```

Out[287]: (2, 2, 3)

In [288]:
```
a[0]
```

Out[288]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [289]:
```
a[1]
```

Out[289]:
```
array([[ 7,  8,  9],
       [10, 11, 12]])
```

In [290]:
```
a[0][0]
```

Out[290]:
```
array([1, 2, 3])
```

In [291]:
```
a[0][1]
```

Out[291]:
```
array([4, 5, 6])
```

In [292]:
```
a[1][0]
```

Out[292]:
```
array([7, 8, 9])
```

In [293]:
```
a[1][1]
```

Out[293]:
```
array([10, 11, 12])
```

In [294]:
```
a[1][1][1]
```

Out[294]:
```
11
```

In [295]:
```
a[1][1][0]
```

Out[295]:
```
10
```

In [296]:
```
a[0]
```

Out[296]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [297]:
```
y = a[0].copy()
y
```

Out[297]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [298]:
```
a[0] = 88
a
```

Out[298]:
```
array([[[88, 88, 88],
        [88, 88, 88]],

       [[ 7,  8,  9],
        [10, 11, 12]]])
```

In [299]:
```
y
```

Out[299]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [300]:
```
a[0] = y
a
```

Out[300]:
```
array([[[ 1,  2,  3],
        [ 4,  5,  6]],

       [[ 7,  8,  9],
        [10, 11, 12]]])
```

arr2d

In [301]:
```
x = np.array([[1, 2, 3],[4, 5, 6],[7, 8, 9]])
x
```

Out[301]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [302]:
```
x.shape
```

Out[302]:
```
(3, 3)
```

In [303]:
```
x.ndim
```

Out[303]:
```
2
```

In [304]:
```
x[:2]
```

Out[304]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [305]:
```
x[:1]
```

Out[305]:
```
array([[1, 2, 3]])
```

In [306]:
```
x[:]
```

Out[306]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [307]:
```
x[:2, 1:]
```

Out[307]:
```
array([[2, 3],
       [5, 6]])
```

In [308]:
```
x[:2, 1:] = 0
x
```

Out[308]:
```
array([[1, 0, 0],
       [4, 0, 0],
       [7, 8, 9]])
```

In [309]:
```
x = np.array([[1, 2, 3],[4, 5, 6],[7, 8, 9]])
x
```

Out[309]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [310]:
```
x[:2, 0]
```

Out[310]:
```
array([1, 4])
```

In [311]:
```
x[:1, 1]
```

Out[311]:
```
array([2])
```

In [312]:
```
x[:, :1]
```

Out[312]:
```
array([[1],
       [4],
       [7]])
```

In [313]:
```
x[:, :2]
```

Out[313]:
```
array([[1, 2],
       [4, 5],
       [7, 8]])
```

Boolean indexing

In [314]:
```
n = np.array(['ali', 'sara', 'taha', 'ali'])
n
```

Out[314]:
```
array(['ali', 'sara', 'taha', 'ali'], dtype='<U4')
```

In [315]:
```
n == 'ali'
```

Out[315]:
```
array([ True, False, False,  True])
```

In [316]:
```
d = np.random.randn(4,3)
d
```

Out[316]:
```
array([[-1.94271337e-01,  1.44970384e+00,  4.18066230e-01],
       [ 2.30788009e-02,  3.73243267e-01,  6.28415961e-01],
       [ 3.60239728e-01, -9.28641521e-01,  2.77348124e+00],
       [-1.36042377e+00, -4.05192079e-04, -1.51775711e+00]])
```

In [317]:
```
d [n == 'ali']
```

Out[317]:
```
array([[-1.94271337e-01,  1.44970384e+00,  4.18066230e-01],
       [-1.36042377e+00, -4.05192079e-04, -1.51775711e+00]])
```

In [318]:
```
d[n=='ali' , 1:]
```

Out[318]:
```
array([[ 1.44970384e+00,  4.18066230e-01],
       [-4.05192079e-04, -1.51775711e+00]])
```

In [319]:
```
d
```

Out[319]:
```
array([[-1.94271337e-01,  1.44970384e+00,  4.18066230e-01],
       [ 2.30788009e-02,  3.73243267e-01,  6.28415961e-01],
       [ 3.60239728e-01, -9.28641521e-01,  2.77348124e+00],
       [-1.36042377e+00, -4.05192079e-04, -1.51775711e+00]])
```

In [320]:
```
d [~(n == 'ali')]
```

Out[320]:
```
array([[ 0.0230788 ,  0.37324327,  0.62841596],
       [ 0.36023973, -0.92864152,  2.77348124]])
```

In [321]:
```
c = n == 'ali'
d[~c]
```

Out[321]:
```
array([[ 0.0230788 ,  0.37324327,  0.62841596],
       [ 0.36023973, -0.92864152,  2.77348124]])
```

In [322]:
```
d
```

Out[322]:
```
array([[-1.94271337e-01,  1.44970384e+00,  4.18066230e-01],
       [ 2.30788009e-02,  3.73243267e-01,  6.28415961e-01],
       [ 3.60239728e-01, -9.28641521e-01,  2.77348124e+00],
       [-1.36042377e+00, -4.05192079e-04, -1.51775711e+00]])
```

In [323]:
```
m = (n == 'ali') | (n == 'taha')
d[m]
```

Out[323]:
```
array([[-1.94271337e-01,  1.44970384e+00,  4.18066230e-01],
       [ 3.60239728e-01, -9.28641521e-01,  2.77348124e+00],
       [-1.36042377e+00, -4.05192079e-04, -1.51775711e+00]])
```

In [324]:
```
x = np.random.randn(3,4)
x
```

Out[324]:
```
array([[-0.4698512 , -0.88086835,  0.55325894,  0.50988145],
       [ 0.81161967,  0.42887373, -1.33657568,  0.12756088],
       [ 0.67724525, -2.34362601,  1.01806061, -1.12323832]])
```

In [325]:
```
x[x < 0]  = 0
x
```

Out[325]:
```
array([[0.        , 0.        , 0.55325894, 0.50988145],
       [0.81161967, 0.42887373, 0.        , 0.12756088],
       [0.67724525, 0.        , 1.01806061, 0.        ]])
```

Fancy indexing: indexing using integer arrays.

In [326]:
```
arr = np.empty((7,5))
for i in range(7):
    arr[i] = 5*i+1
arr
```

Out[326]:
```
array([[ 1.,  1.,  1.,  1.,  1.],
       [ 6.,  6.,  6.,  6.,  6.],
       [11., 11., 11., 11., 11.],
       [16., 16., 16., 16., 16.],
       [21., 21., 21., 21., 21.],
       [26., 26., 26., 26., 26.],
       [31., 31., 31., 31., 31.]])
```

In [327]:
```
arr[[3, 6, 0]]
```

Out[327]:
```
array([[16., 16., 16., 16., 16.],
       [31., 31., 31., 31., 31.],
       [ 1.,  1.,  1.,  1.,  1.]])
```

In [328]:
```
arr[[-7, -1]]
```

Out[328]:
```
array([[ 1.,  1.,  1.,  1.,  1.],
       [31., 31., 31., 31., 31.]])
```

In [329]:
```
x = np.arange(35).reshape((7,5))
x
```

Out[329]:
```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24],
       [25, 26, 27, 28, 29],
       [30, 31, 32, 33, 34]])
```

In [330]:
```
x[[1, 5, 6, 2] , [0, 3, 1, 2]]
```

Out[330]:
```
array([ 5, 28, 31, 12])
```

In [331]:
```
x
```

Out[331]:
```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24],
       [25, 26, 27, 28, 29],
       [30, 31, 32, 33, 34]])
```

In [332]:
```
x[[2, 6]][:,[0, 3 , 1]]
```

Out[332]:
```
array([[10, 13, 11],
       [30, 33, 31]])
```

In [333]:
```
# Transposing arrays and swapping axes
```

In [334]:
```
arr = np.arange(8).reshape((2,4))
arr
```

Out[334]:
```
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])
```

In [335]:
```
np.transpose(arr)
```

Out[335]:
```
array([[0, 4],
       [1, 5],
       [2, 6],
       [3, 7]])
```

In [336]:
```
arr.T
```

Out[336]:
```
array([[0, 4],
       [1, 5],
       [2, 6],
       [3, 7]])
```

In [337]:
```
z = np.arange(60).reshape((3, 4, 5))
```

In [338]:
```
z
```

Out[338]:
```
array([[[ 0,  1,  2,  3,  4],
        [ 5,  6,  7,  8,  9],
        [10, 11, 12, 13, 14],
        [15, 16, 17, 18, 19]],

       [[20, 21, 22, 23, 24],
        [25, 26, 27, 28, 29],
        [30, 31, 32, 33, 34],
        [35, 36, 37, 38, 39]],

       [[40, 41, 42, 43, 44],
        [45, 46, 47, 48, 49],
        [50, 51, 52, 53, 54],
        [55, 56, 57, 58, 59]]])
```

In [339]:
```
z.swapaxes(0, 1)
```

Out[339]:
```
array([[[ 0,  1,  2,  3,  4],
        [20, 21, 22, 23, 24],
        [40, 41, 42, 43, 44]],

       [[ 5,  6,  7,  8,  9],
        [25, 26, 27, 28, 29],
        [45, 46, 47, 48, 49]],

       [[10, 11, 12, 13, 14],
        [30, 31, 32, 33, 34],
        [50, 51, 52, 53, 54]],

       [[15, 16, 17, 18, 19],
        [35, 36, 37, 38, 39],
        [55, 56, 57, 58, 59]]])
```

In [340]:
```
z.transpose((1, 0, 2))
```

Out[340]:
```
array([[[ 0,  1,  2,  3,  4],
        [20, 21, 22, 23, 24],
        [40, 41, 42, 43, 44]],

       [[ 5,  6,  7,  8,  9],
        [25, 26, 27, 28, 29],
        [45, 46, 47, 48, 49]],

       [[10, 11, 12, 13, 14],
        [30, 31, 32, 33, 34],
        [50, 51, 52, 53, 54]],

       [[15, 16, 17, 18, 19],
        [35, 36, 37, 38, 39],
        [55, 56, 57, 58, 59]]])
```

Universal Function : ufunc

In [341]:
```
arr = np.arange(4)
arr
```

Out[341]:
```
array([0, 1, 2, 3])
```

In [342]:
```
np.sqrt(arr)
```

Out[342]:
```
array([0.        , 1.        , 1.41421356, 1.73205081])
```

In [343]:
```
np.exp(arr)
```

Out[343]:
```
array([ 1.        ,  2.71828183,  7.3890561 , 20.08553692])
```

In [344]:
```
x = [2.6, 8.5, -9]
r, w = np.modf(x)
```

In [345]:
```
r
```

Out[345]:
```
array([ 0.6,  0.5, -0. ])
```

In [346]:
```
w
```

Out[346]:
```
array([ 2.,  8., -9.])
```

In [347]:
```
x = np.random.randn(4)
y = np.random.randn(4)
```

In [348]:
```
x
```

Out[348]:
```
array([ 0.07786941, -0.81174976,  1.70791682,  0.08938819])
```

In [349]:
```
y
```

Out[349]:
```
array([ 0.56662061, -0.0934638 ,  0.70310464,  0.21415909])
```

In [350]:
```
np.maximum(x, y)
```

Out[350]:
```
array([ 0.56662061, -0.0934638 ,  1.70791682,  0.21415909])
```

where

In [351]:
```
arr1 = np.array([1, 5, 8])
arr2 = np.array([4, 7, 12])
cond = np.array([True,False,True])
```

In [352]:
```
r = [(x if c else y)
    for x, y, c in zip(arr1, arr2, cond)]
```

In [353]:
```
list(zip(arr1, arr2, cond))
```

Out[353]:
```
[(1, 4, True), (5, 7, False), (8, 12, True)]
```

In [354]:
```
r
```

Out[354]:
```
[1, 7, 8]
```

In [355]:
```
res = np.where(cond, arr1 , arr2)
res
```

Out[355]:
```
array([1, 7, 8])
```

In [356]:
```
x = np.random.randn(2, 3)
x
```

Out[356]:
```
array([[ 0.79098806,  0.3145874 ,  0.16000977],
       [ 0.56013768, -0.65591675,  0.46978947]])
```

In [357]:
```
x > 0
```

Out[357]:
```
array([[ True,  True,  True],
       [ True, False,  True]])
```

In [358]:
```
np.where(x > 0 , 1 , 0)
```

Out[358]:
```
array([[1, 1, 1],
       [1, 0, 1]])
```

In [359]:
```
x
```

Out[359]:
```
array([[ 0.79098806,  0.3145874 ,  0.16000977],
       [ 0.56013768, -0.65591675,  0.46978947]])
```

In [360]:
```
np.where(x > 0 , 1 , x)
```

Out[360]:
```
array([[ 1.        ,  1.        ,  1.        ],
       [ 1.        , -0.65591675,  1.        ]])
```

In [361]:
```
score = np.array([[7,12,20],[10,15,4]])
score
```

Out[361]:
```
array([[ 7, 12, 20],
       [10, 15,  4]])
```

In [362]:
```
np.where(score>10 , score , 10)
```

Out[362]:
```
array([[10, 12, 20],
       [10, 15, 10]])
```

In [363]:
```
x = [[1,2],[3,4]]
y = [[5,6],[7,8]]
c = [[True,False],[False,True]]
np.where(c, x, y)
```

Out[363]:
```
array([[1, 6],
       [7, 4]])
```

Mathematical and Statistical Methods

In [364]:
```
# amin, amax , mean , averge , nanmean
```

In [365]:
```
arr = np.array([4,2,3,1])
np.amin(arr)
```

Out[365]:
```
1
```

In [366]:
```
arr.min()
```

Out[366]:
```
1
```

In [367]:
```
np.amax(arr)
```

Out[367]: 4

In [368]:
```
np.mean(arr)
```

Out[368]: 2.5

In [369]:
```
np.average(arr,weights=[1,2,3,4])   # (4*1 + 2*2 + 3*3 + 1*4 )/ 10
```

Out[369]: 2.1

In [370]:
```
np.nan
```

Out[370]: nan

In [371]:
```
x = np.array([1, 4, np.nan, 8, np.nan, 7])
np.mean(x)
```

Out[371]: nan

In [372]:
```
np.nanmean(x)   # 20/4
```

Out[372]: 5.0

var , std , median , quantile , percentile

In [373]:
```
x = np.array([3, 5, 9, 8, 1, 4, 12, 17, 6])
```

In [374]:
```
np.var(x)
```

Out[374]: 21.728395061728392

In [375]:
```
np.std(x)
```

Out[375]: 4.661372658534007

In [376]:
```
np.median(x)
```

Out[376]: 6.0

In [377]:
```
np.sort(x)
```

Out[377]: array([1, 3, 4, 5, 6, 8, 9, 12, 17])

In [378]:
```
y = np.array([3, 5, 9, 8, 1, 4, 12, 17])
np.median(y)
```

Out[378]: 6.5

In [379]:
```
np.sort(y)
```

Out[379]: array([1, 3, 4, 5, 8, 9, 12, 17])

In [380]:
```
a = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])
np.quantile(a, 0.25)
```

Out[380]: 3.0

In [381]:
```
np.quantile(a,0.50)
```

Out[381]: 5.0

In [382]:
```
np.quantile(a,0.75)
```

Out[382]: 7.0

In [383]:
```
np.percentile(a,25)
```

Out[383]: 3.0

In [384]:
```
np.percentile(a,50)
```

Out[384]: 5.0

In [385]:
```
np.percentile(a,75)
```

Out[385]: 7.0

In [386]:
```
np.percentile(a,10)
```

Out[386]: 1.8

In [387]:
```
# sum , cumsum
```

In [388]:
```
arr = np.array([1, 2, 3, 4])
```

In [389]:
```
np.sum(arr)
```

Out[389]: 10

In [390]:
```
np.cumsum(arr)
```

Out[390]:
```
array([ 1,  3,  6, 10], dtype=int32)
```

In [391]:
```
x = np.array([[1,2,3],[4,5,6],[7,8,9]])
x
```

Out[391]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [392]:
```
np.sum(x)
```

Out[392]:
```
45
```

In [393]:
```
np.sum(x, axis=0)
```

Out[393]:
```
array([12, 15, 18])
```

In [394]:
```
np.sum(x, axis=1)
```

Out[394]:
```
array([ 6, 15, 24])
```

In [395]:
```
x
```

Out[395]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [396]:
```
np.cumsum(x, axis=0)
```

Out[396]:
```
array([[ 1,  2,  3],
       [ 5,  7,  9],
       [12, 15, 18]], dtype=int32)
```

In [397]:
```
np.cumsum(x, axis=1)
```

Out[397]:
```
array([[ 1,  3,  6],
       [ 4,  9, 15],
       [ 7, 15, 24]], dtype=int32)
```

In [398]:
```
# all , any
```

In [399]:
```
a = np.array([True, True, False])
```

In [400]:
```
a.any()
```

Out[400]: True

In [401]:
```
a.all()
```

Out[401]: False

In [402]:
```
b = [0, 2, -3]
```

In [403]:
```
np.any(b)
```

Out[403]: True

In [404]:
```
np.all(b)
```

Out[404]: False

In [405]:
```
c = [8, 2, -3]
```

In [406]:
```
np.all(c)
```

Out[406]: True

In [407]:
```
# unique
```

In [408]:
```
arr = np.array([3, 4, 7, 4, 2, 1, 3, 5, 4, 4])
np.unique(arr)
```

Out[408]: array([1, 2, 3, 4, 5, 7])

In [409]:
```
a , i = np.unique(arr, return_index=True)
```

In [410]:
```
a
```

Out[410]: array([1, 2, 3, 4, 5, 7])

In [411]:
```
i
```

Out[411]: array([5, 4, 0, 1, 7, 2], dtype=int64)

In [412]:
```
# sort
```

In [413]:
```
data = [('ali',12.5,35) , ('sara',18.75,27),('taha',16.25,27)]
type(data)
```

Out[413]:
```
list
```

In [414]:
```
d = [('name','S10'), ('score',float) , ('age',int)]
```

In [415]:
```
arr = np.array(data, dtype=d)
```

In [416]:
```
np.sort(arr, order='age')
```

Out[416]:
```
array([(b'sara', 18.75, 27), (b'taha', 16.25, 27), (b'ali', 12.5 , 35)],
      dtype=[('name', 'S10'), ('score', '<f8'), ('age', '<i4')])
```

In [417]:
```
np.sort(arr, order='score')
```

Out[417]:
```
array([(b'ali', 12.5 , 35), (b'taha', 16.25, 27), (b'sara', 18.75, 27)],
      dtype=[('name', 'S10'), ('score', '<f8'), ('age', '<i4')])
```

In [418]:
```
#in1d
```

In [419]:
```
x = np.array([7, 1, 4, 2, 5, 7])
y = [3, 4, 7]
np.in1d(x, y)
```

Out[419]:
```
array([ True, False,  True, False, False,  True])
```

In [420]:
```
x = np.array([7, 1, 4, 2, 5, 7])
y = [3, 4, 7]
m = np.in1d(x, y)
x[m]
```

Out[420]:
```
array([7, 4, 7])
```

In [421]:
```
x = np.array([7, 1, 4, 2, 5, 7])
y = [3, 4, 7]
m = np.in1d(x, y, invert=True)
x[m]
```

Out[421]:
```
array([1, 2, 5])
```

In [422]:
```
# save , load
```

In [423]:
```
a = np.array([4, 6, 9])
```

In [424]:
```
np.save('test.npy', a)
```

In [425]:
```
np.load('test.npy')
```

Out[425]:
```
array([4, 6, 9])
```

In [426]:
```
with open('test.npy', 'wb') as f:
    np.save(f, a)
```

In [427]:
```
with open('test.npy', 'rb') as f:
    x = np.load(f)
```

In [428]:
```
x
```

Out[428]:
```
array([4, 6, 9])
```

In [429]:
```
arr1 = np.array([1, 2])
arr2 = np.array([3, 4, 5])
np.savez('test2.npz',x=arr1, y=arr2)
```

In [430]:
```
t = np.load('test2.npz')
t['x']
```

Out[430]:
```
array([1, 2])
```

In [431]:
```
t['y']
```

Out[431]:
```
array([3, 4, 5])
```

In [432]:
```
t.files
```

Out[432]:
```
['x', 'y']
```

random_sample U[a,b) : (b-a)*random_sample(...) + a

In [433]:
```
# U[1,20)
19 * np.random.random_sample(5) + 1
```

Out[433]:
```
array([13.40697458,  6.05157112, 13.96080015, 13.18852091, 14.578737  ])
```

rand : U[0,1)

In [434]:
```
np.random.rand(3)
```

Out[434]:
```
array([0.45101679, 0.27606414, 0.63419546])
```

randint

In [435]:
```
np.random.randint(0,11,7)
```

Out[435]:
```
array([6, 7, 9, 3, 6, 5, 9])
```

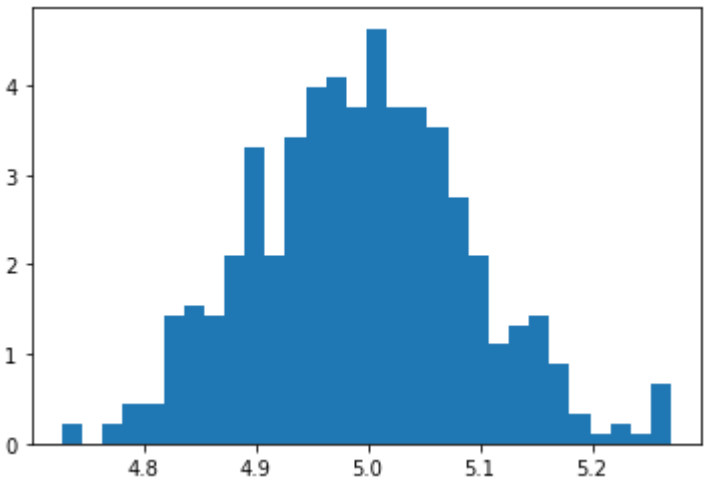
randn : N(mu,sigma^2) : sigma * randn(...) + mu

In [436]:
```
# N(3, 2.5^2)
2.5 * np.random.randn(4) + 3
```

Out[436]:
```
array([ 3.55520993, -1.4553521 ,  4.05033061,  6.83702422])
```

In [437]:
```
mu = 5
sigma = 0.1
s = np.random.normal(mu, sigma, 500)
```

In [438]:
```
import matplotlib.pyplot as plt
plt.hist(s, 30, density=True)
plt.show()
```



In [439]:
```
a = np.random.normal(size=3)
a
```

Out[439]:
```
array([-1.87861205,  0.41757864,  0.50770968])
```

In [440]:
```
np.random.seed(45654)
b = np.random.normal(size=3)
b
```

Out[440]:
```
array([ 0.8355203 ,  0.6241987 , -1.56246633])
```

In [441]:
```
## inner , outer
```

In [442]:
```
a = np.array([1, 2, 3])
b = np.array([5, 6, 0])
```

In [443]:
```
np.inner(a, b)
```

Out[443]:
```
17
```

In [444]:
```
1*5 + 2*6 + 3*0
```

Out[444]:
```
17
```

In [445]:
```
np.outer(a, b)
```

Out[445]:
```
array([[ 5,  6,  0],
       [10, 12,  0],
       [15, 18,  0]])
```

In [446]:
```
x = np.array([[1, 2],
              [3, 4]])

y = np.array([[5, 6],
              [7, 8]])
```

In [447]:
```
np.dot(x, y)
```

Out[447]:
```
array([[19, 22],
       [43, 50]])
```

In [448]:
```
from numpy.linalg import inv
```

In [449]:
```
inv(x)
```

Out[449]:
```
array([[-2. ,  1. ],
       [ 1.5, -0.5]])
```

In [450]:
```
from numpy.linalg import qr
```

In [451]:
```
b = x.T.dot(x)
```

In [452]:
```
i = inv(b)
```

In [453]:
```
b.dot(i)
```

Out[453]:
```
array([[1.00000000e+00, 7.10542736e-15],
       [0.00000000e+00, 1.00000000e+00]])
```

In [454]:
```
q, r= qr(b)
```

In [455]:
```
q
```

Out[455]:
```
array([[-0.58123819, -0.81373347],
       [-0.81373347,  0.58123819]])
```

In [456]:
```
r
```

Out[456]:
```
array([[-17.20465053, -24.41200414],
       [  0.        ,   0.23249528]])
```

دانشگاه شهید مدنی آذربایجان
برنامه نویسی پیشرفته با پایتون
امین گلزاری اسکوئی
۱۴۰۰-۱۴۰۱

Codes and Projects (click here) (https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Advanced-2021) slides and videos (click here) (https://drive.google.com/drive/folders/1Dx3v7fD1QBWL-MNP2hd7iIxaRbeALkkA)