numpy1

November 27, 2023

```
[1]: import numpy as np
```

```
[2]: np.ones((3,3),dtype='i8')
```

```
[2]: array([[1, 1, 1],
            [1, 1, 1],
            [1, 1, 1]], dtype=int64)
```

```
[3]: np.ones((3,3),dtype='f8')
```

```
[3]: array([[1., 1., 1.],
            [1., 1., 1.],
            [1., 1., 1.]])
```

```
[5]: np.ones((3,3))
```

```
[5]: array([[1., 1., 1.],
            [1., 1., 1.],
            [1., 1., 1.]])
```

```
[6]: A=np.eye(3)
```

```
[8]: A
```

```
[8]: array([[1., 0., 0.],
            [0., 1., 0.],
            [0., 0., 1.]])
```

```
[9]: B=np.where(A==0,100,200)
```

```
[10]: B
```

```
[10]: array([[200, 100, 100],
             [100, 200, 100],
             [100, 100, 200]])
```

```
[11]: B[0]
```

```
[11]: array([200, 100, 100])
```

```
[12]: C=B[0:2]
```

```
[13]: C
```

```
[13]: array([[200, 100, 100],
             [100, 200, 100]])
```

```
[15]: C[:,0]
```

```
[15]: array([200, 100])
```

```
[16]: D=C[0,0]
```

```
[17]: D
```

```
[17]: 200
```

```
[18]: D.ndim
```

```
[18]: 0
```

```
[19]: D.shape
```

```
[19]: ()
```

```
[20]: E=C[0:,0:1]
```

```
[21]: E
```

```
[21]: array([[200],
             [100]])
```

```
[22]: E.ndim
```

```
[22]: 2
```

```
[23]: E.shape
```

```
[23]: (2, 1)
```

```
[24]: A1=np.arange(1,10).reshape(3,3)
```

```
[25]: A1
```

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

```
[26]: A2=np.arange(11,20).reshape(3,3)
```

```
[27]: A2
```

```
[27]: array([[11, 12, 13],
             [14, 15, 16],
             [17, 18, 19]])
```

```
[28]: A12=np.concatenate(A1,A2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[28], line 1
----> 1 A12=np.concatenate(A1,A2)

TypeError: only integer scalar arrays can be converted to a scalar index
```

```
[29]: A12=np.concatenate((A1,A2))
```

```
[30]: A12
```

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

```
[31]: A13=np.concatenate((A1,A2),axis=0)
```

```
[32]: A13
```

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

```
[33]: A14=np.concatenate((A1,A2),axis=1)
```

```
[34]: A14
```

[34]:
```
array([[ 1,  2,  3, 11, 12, 13],
       [ 4,  5,  6, 14, 15, 16],
       [ 7,  8,  9, 17, 18, 19]])
```

[35]:
```
A15=np.vstack((A1,A2))
```

[36]:
```
A15
```

[36]:
```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [ 7,  8,  9],
       [11, 12, 13],
       [14, 15, 16],
       [17, 18, 19]])
```

[38]:
```
A16=np.vstack((A1,A2),axis=0)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[38], line 1
----> 1 A16=np.vstack((A1,A2),axis=0)

TypeError: vstack() got an unexpected keyword argument 'axis'
```

[39]:
```
A17=np.vstack(A1,A2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[39], line 1
----> 1 A17=np.vstack(A1,A2)

TypeError: vstack() takes 1 positional argument but 2 were given
```

[40]:
```
A17=np.vstack((A1,A2))
```

[41]:
```
A17
```

[41]:
```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [ 7,  8,  9],
       [11, 12, 13],
       [14, 15, 16],
       [17, 18, 19]])
```

[42]:
```
A18=np.hstack((A1,A2))
```

```
[43]: A18
```

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

```
[44]: B1=A18.T
```

```
[45]: B1
```

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

```
[46]: A18
```

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

```
[47]: A18.T
```

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

```
[48]: A18.swapaxes(0,1)
```

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

```
[49]: A18
```

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

```
[50]: A18.swapaxes(1,0)
```

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

```
[51]: A18
```

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

```
[52]: A18.swapaxes()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[52], line 1
----> 1 A18.swapaxes()

TypeError: swapaxes() takes exactly 2 arguments (0 given)
```

```
[53]: G=np.random.randint(10,110,10)
```

```
[54]: G
```

```
[54]: array([103,  23,  51,  57,  74,  14,  45,  57,  23,  65])
```

```
[55]: G.size
```

```
[55]: 10
```

```
[56]: H=G.reshape(2,-1)
```

```
[57]: H
```

```
[57]: array([[103,  23,  51,  57,  74],
             [ 14,  45,  57,  23,  65]])
```

```
[58]: H.sort()
```

```
[59]: H
```

```
[59]: array([[ 23,  51,  57,  74, 103],
              [ 14,  23,  45,  57,  65]])
```

```
[60]: H.sort(axis=1)
```

```
[61]: H
```

```
[61]: array([[ 23,  51,  57,  74, 103],
              [ 14,  23,  45,  57,  65]])
```

```
[62]: H.sort(axis=0)
```

```
[63]: H
```

```
[63]: array([[ 14,  23,  45,  57,  65],
              [ 23,  51,  57,  74, 103]])
```

```
[66]: H[::-1]
```

```
[66]: array([[ 23,  51,  57,  74, 103],
              [ 14,  23,  45,  57,  65]])
```

```
[68]: H
```

```
[68]: array([[ 14,  23,  45,  57,  65],
              [ 23,  51,  57,  74, 103]])
```

```
[67]: H[:,::-1]
```

```
[67]: array([[ 65,  57,  45,  23,  14],
              [103,  74,  57,  51,  23]])
```

```
[73]: from numpy import linalg
```

```
[74]: A=np.arange(1,10).reshape(3,3)
```

```
[75]: A
```

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

```
[76]: linalg.det(A)
```

```
[76]: 0.0
```

```
[77]: H
```

```
[77]: array([[ 14,  23,  45,  57,  65],
              [ 23,  51,  57,  74, 103]])
```

```
[78]: linalg.det(H)
```

```
---------------------------------------------------------------------------
LinAlgError                               Traceback (most recent call last)
Cell In[78], line 1
----> 1 linalg.det(H)

File␣
 ↪~\AppData\Local\Programs\Python\Python311\Lib\site-packages\numpy\linalg\linalg.
 ↪py:2177, in det(a)
   2175 a = asarray(a)
   2176 _assert_stacked_2d(a)
-> 2177 _assert_stacked_square(a)
   2178 t, result_t = _commonType(a)
   2179 signature = 'D->D' if isComplexType(t) else 'd->d'

File␣
 ↪~\AppData\Local\Programs\Python\Python311\Lib\site-packages\numpy\linalg\linalg.
 ↪py:213, in _assert_stacked_square(*arrays)
    211 m, n = a.shape[-2:]
    212 if m != n:
--> 213     raise LinAlgError('Last 2 dimensions of the array must be square')

LinAlgError: Last 2 dimensions of the array must be square
```

```
[79]: C=np.eye(3,3)
```

```
[80]: C
```

```
[80]: array([[1., 0., 0.],
              [0., 1., 0.],
              [0., 0., 1.]])
```

```
[81]: linalg.det(C)
```

```
[81]: 1.0
```

```
[83]: N=np.random.rand(3,3)
```

```
[84]: N
```

```
[84]: array([[0.24328718, 0.09797536, 0.39609087],
              [0.94465086, 0.98649864, 0.72231883],
              [0.44566882, 0.90513668, 0.15145257]])
```

[85]:

```
linalg.det(N)
```

[85]:

```
0.059341704913366 79
```

[86]:

```
linalg.inv(N)
```

[86]:

```
array([[-8.49974754,  5.79150456, -5.39205365],
       [ 3.01381928, -2.35380632,  3.34396646],
       [ 6.9999078 , -2.97503395,  2.48476115]])
```

[88]:

```
N@linalg.inv(N)
```

[88]:

```
array([[ 1.00000000e+00, -3.14617276e-16, -2.17821238e-17],
       [-1.23217374e-15,  1.00000000e+00, -3.09299323e-16],
       [-1.97494654e-16, -9.59698299e-17,  1.00000000e+00]])
```

[89]:

```
N=np.random.randint(10,size=(3,3))
```

[ ]:

```

```