

Write programs in Python using NumPy library to do the following:

A-> Compute the mean, standard deviation, and variance of a two dimensional random integer array along the second axis.

B-> Get the indices of the sorted elements of a given array. B = [56, 48, 22, 41, 78, 91, 24, 46, 8, 33]

C-> Create a 2-dimensional array of size m x n integer elements, also print the shape, type and data type of the array and then reshape it into nx m array, n and m are user inputs given at the run time.

D-> Test whether the elements of a given array are zero, non-zero and NaN. Record the indices of these elements in three separate arrays.

In [1]:

```
import numpy as np
```

A->

Compute the mean, standard deviation, and variance of a two dimensional random integer array along the second axis.

In [3]:

```
a=np.linspace(10,100,num=15,dtype=int).reshape(3,5)
```

2d ARRAY

In [4]:

```
a
```

Out[4]:

```
array([[ 10,  16,  22,  29,  35],
       [ 42,  48,  55,  61,  67],
       [ 74,  80,  87,  93, 100]])
```

MEAN

In [15]:

```
np.mean(a,axis=0).round(1)
```

Out[15]:

```
array([42. , 48. , 54.7, 61. , 67.3])
```

Standard Deviation

In [21]:

```
np.std(a,axis=0).round(3)
```

Out[21]:

```
array([26.128, 26.128, 26.537, 26.128, 26.537])
```

Variance

In [22]:

```
np.var(a,axis=0).round(3)
```

Out[22]:

```
array([682.667, 682.667, 704.222, 682.667, 704.222])
```

B ->

Get the indices of the sorted elements of a given array. B = [56, 48, 22, 41, 78, 91, 24, 46, 8, 33]

In [36]:

```
b=[56, 48, 22, 41, 78, 91, 24, 46, 8, 33]
```

In [37]:

```
b=np.array(b)
```

In [38]:

```
b
```

Out[38]:

```
array([56, 48, 22, 41, 78, 91, 24, 46, 8, 33])
```

In [54]:

```
b.argsort(axis=-1)
```

Out[54]:

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

In []:

C->

Create a 2-dimensional array of size m x n integer elements, also print the shape, type and data type of the array and then reshape it into nx m array, n and m are user inputs given at the run time.

In [72]:

```
row = input("Enter number of rows: ")
print("rows->",row)
col = input("Enter number of cols: ")
print("cols->",col)
```

```
Enter number of rows: 5
rows-> 5
Enter number of cols: 3
cols-> 3
```

In [74]:

```
row=int(row)
```

In [76]:

```
col=int(col)
```

In [77]:

```
a=np.linspace(10,100,num=15,dtype=int).reshape(row,col)
```

In [78]:

```
a
```

Out[78]:

```
array([[ 10,  16,  22],
       [ 29,  35,  42],
       [ 48,  55,  61],
       [ 67,  74,  80],
       [ 87,  93, 100]])
```

In [79]:

```
#shape
a.shape
```

Out[79]:

```
(5, 3)
```

In [80]:

```
#data type
a.dtype
```

Out[80]:

```
dtype('int64')
```

In [81]:

```
#type of array  
type(a)
```

Out[81]:

numpy.ndarray

In [82]:

```
a.reshape(col,row)
```

Out[82]:

```
array([[ 10,  16,  22,  29,  35],  
       [ 42,  48,  55,  61,  67],  
       [ 74,  80,  87,  93, 100]])
```

In [83]:

```
a.T
```

Out[83]:

```
array([[ 10,  29,  48,  67,  87],  
       [ 16,  35,  55,  74,  93],  
       [ 22,  42,  61,  80, 100]])
```

In []:

In []:

D->

Test whether the elements of a given array are zero, non-zero and NaN. Record the indices of these elements in three separate arrays.

In [96]:

```
0.1/5
```

Out[96]:

0.02

In [120]:

```
arr=np.random.choice([np.nan,0,15], size=25, replace=True,p=[0.4,0.4,0.2]).reshape(5,5)
arr
```

Out[120]:

```
array([[nan, 15., nan, nan, 15.],
       [nan,  0., nan, nan, nan],
       [nan, nan,  0.,  0.,  0.],
       [nan,  0.,  0.,  0., 15.],
       [15., 15., 15., nan, nan]])
```

In [127]:

```
np.isnan(arr)
```

Out[127]:

```
array([[ True, False,  True,  True, False],
       [ True, False,  True,  True,  True],
       [ True,  True, False, False, False],
       [ True, False, False, False, False],
       [False, False, False,  True,  True]])
```

In [123]:

```
np.argwhere(np.isnan(arr))
```

Out[123]:

```
array([[0, 0],
       [0, 2],
       [0, 3],
       [1, 0],
       [1, 2],
       [1, 3],
       [1, 4],
       [2, 0],
       [2, 1],
       [3, 0],
       [4, 3],
       [4, 4]])
```

In [124]:

```
np.argwhere(arr==0)
```

Out[124]:

```
array([[1, 1],
       [2, 2],
       [2, 3],
       [2, 4],
       [3, 1],
       [3, 2],
       [3, 3]])
```

In [133]:

```
np.argwhere((np.isnan(arr)==False) & (arr!=0))
```

Out[133]:

```
array([[0, 1],  
       [0, 4],  
       [3, 4],  
       [4, 0],  
       [4, 1],  
       [4, 2]])
```

In []: