

Data Analysis Tools with Numpy - Exercises

ให้ Import NumPy Library ให้อยู่ในตัวแปร np

```
In [11]: import numpy as np
```

จงสร้าง zeros array ที่มีสมาชิก 10 ตัว

```
In [12]: b = np.zeros([10])  
b
```

```
Out[12]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

จงสร้าง ones array ที่มีสมาชิก 10 ตัว

```
In [13]: a = np.ones([10])  
a
```

```
Out[13]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

จงสร้าง array บรรจุ เลข 5 จำนวน 10 ตัว

```
In [14]: e = np.full([10],5,dtype=int)  
e
```

```
Out[14]: array([5, 5, 5, 5, 5, 5, 5, 5, 5, 5])
```

จงสร้าง array บรรจุจำนวนเต็ม ที่มีค่าตั้งแต่ 10 - 50

```
In [15]: s = np.arange(10,51)  
s
```

```
Out[15]: array([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])
```

จงสร้าง array บรรจุจำนวนเต็มคู่ ที่มีค่าตั้งแต่ 10 - 50

```
In [16]: c = np.arange(10,51,2)
c
```

```
Out[16]: array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
44, 46, 48, 50])
```

จงสร้าง Matrix ขนาด 3x3 บรรจุมหาชิกตัวเลขจำนวนเต็ม ที่มีค่าตั้งแต่ 0 ถึง 8

```
In [17]: a = np.arange(9)
a = a.reshape(3,3)
a
```

```
Out[17]: array([[0, 1, 2],
[3, 4, 5],
[6, 7, 8]])
```

จงสร้าง identity matrix ที่มีขนาด 3x3

```
In [18]: a = np.eye(3)
a
```

```
Out[18]: array([[1., 0., 0.],
[0., 1., 0.],
[0., 0., 1.]])
```

จงใช้ NumPy เพื่อสร้างจำนวนสุ่มตัวเลข (random) ที่มีค่าระหว่าง 0 ถึง 1

```
In [19]: from numpy import random as rd
a = rd.rand()
a
```

```
Out[19]: 0.5312084320727485
```

จงใช้ NumPy เพื่อสร้าง array จำนวนสุ่มตัวเลข (random) 25 ค่า ที่มีค่ากระจายตัวแบบ standard normal distribution

```
In [111]: s = np.random.standard_normal(25)
s
```

```
Out[111]: array([ 0.44168538, -0.95311046, -1.06780191, -0.2346015 , -2.44842455,
-2.07177768, -1.41452556,  0.36124494, -1.16795736, -0.22206701,
-0.31486807,  0.00537868,  0.0289859 , -0.62114026,  1.56892659,
 0.54974886,  0.13708236,  0.83490647,  0.06098535, -1.2430113 ,
-0.18165979, -0.24476827,  1.2092723 ,  1.35543571, -1.46735699])
```

จงสร้าง Matrix ต่อไปนี้

```
array([[ 0.01,  0.02,  0.03,  0.04,  0.05,  0.06,  0.07,  0.08,  0.09,  0.1 ],
       [ 0.11,  0.12,  0.13,  0.14,  0.15,  0.16,  0.17,  0.18,  0.19,  0.2 ],
       [ 0.21,  0.22,  0.23,  0.24,  0.25,  0.26,  0.27,  0.28,  0.29,  0.3 ],
       [ 0.31,  0.32,  0.33,  0.34,  0.35,  0.36,  0.37,  0.38,  0.39,  0.4 ],
       [ 0.41,  0.42,  0.43,  0.44,  0.45,  0.46,  0.47,  0.48,  0.49,  0.5 ],
       [ 0.51,  0.52,  0.53,  0.54,  0.55,  0.56,  0.57,  0.58,  0.59,  0.6 ],
       [ 0.61,  0.62,  0.63,  0.64,  0.65,  0.66,  0.67,  0.68,  0.69,  0.7 ],
       [ 0.71,  0.72,  0.73,  0.74,  0.75,  0.76,  0.77,  0.78,  0.79,  0.8 ],
       [ 0.81,  0.82,  0.83,  0.84,  0.85,  0.86,  0.87,  0.88,  0.89,  0.9 ],
       [ 0.91,  0.92,  0.93,  0.94,  0.95,  0.96,  0.97,  0.98,  0.99,  1. ]])
```

```
In [21]: a = np.linspace(0.01,1,100)
a
```

```
Out[21]: array([0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1 , 0.11,
               0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2 , 0.21, 0.22,
               0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3 , 0.31, 0.32, 0.33,
               0.34, 0.35, 0.36, 0.37, 0.38, 0.39, 0.4 , 0.41, 0.42, 0.43, 0.44,
               0.45, 0.46, 0.47, 0.48, 0.49, 0.5 , 0.51, 0.52, 0.53, 0.54, 0.55,
               0.56, 0.57, 0.58, 0.59, 0.6 , 0.61, 0.62, 0.63, 0.64, 0.65, 0.66,
               0.67, 0.68, 0.69, 0.7 , 0.71, 0.72, 0.73, 0.74, 0.75, 0.76, 0.77,
               0.78, 0.79, 0.8 , 0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88,
               0.89, 0.9 , 0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99,
               1.   ])
```

จงสร้าง array ด้วยคำสั่ง `linspace ()` ที่มีจำนวนสมาชิก 20 ตัว ที่มีค่าเรียงตั้งแต่ 0 ถึง 1

```
In [22]: a = np.linspace(0,1,20)
a
```

```
Out[22]: array([0.          , 0.05263158, 0.10526316, 0.15789474, 0.21052632,
               0.26315789, 0.31578947, 0.36842105, 0.42105263, 0.47368421,
               0.52631579, 0.57894737, 0.63157895, 0.68421053, 0.73684211,
               0.78947368, 0.84210526, 0.89473684, 0.94736842, 1.          ])
```

Numpy Indexing and Selection

ให้นำ matrices (mat) ที่กำหนดให้ ใช้ประยุกต์เพื่อตอบคำถามข้อ a-h

```
In [28]: mat = np.arange(1,26).reshape(5,5)
mat
```

```
Out[28]: array([[ 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]])
```

a.

จงสร้าง Matric ให้มีค่าดังภาพ จาก matrices (mat) ที่กำหนดให้

```
array([[12, 13, 14, 15],
       [17, 18, 19, 20],
       [22, 23, 24, 25]])
```

```
In [60]: mat[-3:5,1:5]
```

```
Out[60]: array([[12, 13, 14, 15],
               [17, 18, 19, 20],
               [22, 23, 24, 25]])
```

b.

จงสร้าง Matric ให้มีค่าดังภาพ จาก matrices (mat) ที่กำหนดให้

```
20
```

```
In [72]: mat[3:4,4:]
```

```
Out[72]: array([[20]])
```

c.

จงสร้าง Matric ให้มีค่าดังภาพ จาก matrices (mat) ที่กำหนดให้

```
array([[ 2],
       [ 7],
       [12]])
```

```
In [81]: mat[:3,1:2]
```

```
Out[81]: array([[ 2],
               [ 7],
               [12]])
```

d.

จงสร้าง Matric ให้มีค่าดังภาพ จาก matrices (mat) ที่กำหนดให้

```
array([21, 22, 23, 24, 25])
```

```
In [83]: mat[4:]
```

```
Out[83]: array([[21, 22, 23, 24, 25]])
```

e.

```
In [86]: mat[3:]
```

```
Out[86]: array([[16, 17, 18, 19, 20],  
               [21, 22, 23, 24, 25]])
```

```
In [23]: mat[3:]
```

```
Out[23]: array([[16, 17, 18, 19, 20],  
               [21, 22, 23, 24, 25]])
```

f. ให้ทำการ sum ทุกค่าใน mat

```
In [91]: np.sum(mat)
```

```
Out[91]: 325
```

g. จงหาค่า standard deviation ของค่าที่อยู่ใน mat

```
In [90]: np.std(mat)
```

```
Out[90]: 7.211102550927978
```

h. ให้ทำการ sum ทุกๆ columns ใน mat

```
In [93]: mat.sum(axis=0)
```

```
Out[93]: array([55, 60, 65, 70, 75])
```

----- ภาวนามยปัญญา ปัญญาที่เกิดจากการลงมือทำ! -----