

# numpy 기초

---

- (numpy 기본) 문제 1. ndarray 다루기 (1)
- (numpy 선형대수) 문제 3. 선형방정식 풀기(해 구하기)

## 다음 데이터를 사용하시오

```
2, 3, 5, 6, 7, 10
```

## ndarray 다루기

---

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

### 1) 데이터 벡터 x를 만드시오

```
In [2]: data_x = np.array([2,3,5,6,7,10])
print(data_x)

[ 2  3  5  6  7 10]
```

### 2) 각 데이터의 제곱으로 구성된 벡터 x2를 만드시오

```
In [3]: x_squared = data_x*2
print(x_squared)

[ 4  6 10 12 14 20]
```

### 3) 각 데이터의 제곱의 합을 구하시오

```
In [4]: sum_squared = x_squared.sum()
print(sum_squared)

66
```

### 4) 각 데이터에서 2를 뺀 값을 구하시오

```
In [5]: minus_2 = data_x - 2  
print(minus_2)
```

```
[0 1 3 4 5 8]
```

## 5) 최대값과 최소값을 구하시오

```
In [6]: x_max = max(data_x)  
print(x_max)
```

```
10
```

## 6) 5보다 큰 값들로만 구성된 데이터 벡터 x\_up을 만드시오

```
In [7]: x_up = data_x[data_x > 5]  
print(x_up)
```

```
[ 6  7 10]
```

## 7) 벡터 x의 길이를 구하시오

```
In [8]: print(data_x.shape)
```

```
(6,)
```

## 8) $x'x$ 를 구하시오 ( $x'$ 는 $x$ 의 전치)

```
In [9]: new_x = data_x[:, np.newaxis]  
  
dot_p = np.dot(new_x.T, new_x)  
print(dot_p)
```

```
[[223]]
```

## 9) $xx'$ 를 구하시오

```
In [10]: dot_p_prime = np.dot(new_x, new_x.T)  
print(dot_p_prime)
```

```
[[ 4  6 10 12 14 20]  
 [ 6  9 15 18 21 30]  
 [10 15 25 30 35 50]  
 [12 18 30 36 42 60]  
 [14 21 35 42 49 70]  
 [20 30 50 60 70 100]]
```

## 10) 벡터 x와 x2를 열결합(column bind)하여 xc에 저장하시오

```
In [11]: x = np.array([1,2,3])  
x2 = np.array([4,5,6])
```

```
In [12]: xc = np.vstack([x,x2])
         print(xc)
```

```
[[1 2 3]
 [4 5 6]]
```

## 11) 벡터 x와 x2를 행결합(row bind)하여 xr에 저장하시오

```
In [13]: xr = np.hstack([x,x2])
         print(xr)
```

```
[1 2 3 4 5 6]
```

## 13) 표준정규분포를 따르는 난수 100개를 가진 배열 x를 만드시오

```
In [14]: x_rand = np.array(np.random.randn(20,5))
         print(x_rand)
```

```
[[ 0.71326504  0.55817042 -0.63353538 -1.35001787  1.53505557 ]
 [ 0.96018057 -0.01842479 -1.79121989  1.36010807  0.08736905 ]
 [-0.27718295  0.69184632  1.41256522 -0.92992509 -0.33175192 ]
 [-1.12490873  0.81877697 -1.64605674  0.07150608 -1.7122869  ]
 [ 0.07975725 -0.66640042 -0.03815047 -0.41708099 -1.47660526 ]
 [-1.63661272  0.80843467 -0.83001601  0.85114314 -0.59648605 ]
 [ 0.76034679 -0.36358512 -1.73287446 -0.58278367 -0.30182337 ]
 [-1.2863477  -0.45607201 -0.59105084 -0.15063118  1.87224124 ]
 [ 0.13899556 -0.77297926 -0.36633314 -1.3192902  -2.07055309 ]
 [ 1.49504343 -0.0532928  0.10838806  0.87148928 -0.43108975 ]
 [ 1.15923789  1.81666972 -0.74014828  1.56106679 -0.99594718 ]
 [-0.11502549 -0.35079373  0.07325027  0.21939142 -0.91951358 ]
 [ 0.57288065  1.0464463  0.12376041 -1.69156995  0.63983133 ]
 [ 0.23048235 -0.13781658 -1.04887919  1.11946474  0.63223717 ]
 [-0.76390733 -1.40855146 -0.56026777  1.28265924  0.76755034 ]
 [-0.54022186  0.51327851 -0.26259679 -0.58054404  1.01087872 ]
 [-1.74154534 -0.13624602  0.16059374  0.67384526  0.44155737 ]
 [ 0.33126197 -0.26792302 -1.69695922 -1.6630509  -0.54608995 ]
 [ 0.32789984  0.79239252 -0.35576294 -2.331994  0.0469796  ]
 [ 1.47034336  0.22302484 -0.38036285  0.60448672 -0.94543764 ]]
```

## 14) x에서 값이 0.2 보다 큰 원소들을 골라내시오

```
In [15]: x_over = x_rand[x_rand > 0.2]
         print(x_over)
```

```
[ 0.71326504  0.55817042  1.53505557  0.96018057  1.36010807  0.69184632
 1.41256522  0.81877697  0.80843467  0.85114314  0.76034679  1.87224124
 1.49504343  0.87148928  1.15923789  1.81666972  1.56106679  0.21939142
 0.57288065  1.0464463  0.63983133  0.23048235  1.11946474  0.63223717
 1.28265924  0.76755034  0.51327851  1.01087872  0.67384526  0.44155737
 0.33126197  0.32789984  0.79239252  1.47034336  0.22302484  0.60448672]
```

# 선형방정식 풀기(해 구하기)

---

## 다음과 같은 방정식이 있다

$$\begin{aligned}3x + 6y - 5z &= 12 \\x - 3y + 2z &= -2 \\5x - y + 4z &= 10\end{aligned}$$

### 1) 위 식을 numpy를 이용해서 $Ax=B$ 형태로 나타내시오 (A와 B만 생성)

```
In [16]: A = np.array([[3,6,-5],
                        [1,-3,2],
                        [5,-1,4]])

b = np.array([[12],
              [-2],
              [10]])

print(A.shape, b.shape)

(3, 3) (3, 1)
```

### 2) A의 역행렬을 구하시오

```
In [17]: A_inv = np.linalg.inv(A)
print(A_inv)

[[ 0.15625  0.296875  0.046875]
 [-0.09375 -0.578125  0.171875]
 [-0.21875 -0.515625  0.234375]]
```

### 3) 방정식의 해 벡터 x를 구하시오

```
In [18]: x = np.dot(A_inv, b)
print(x)

[[ 1.75]
 [ 1.75]
 [ 0.75]]
```