

I. Excel 행렬 연산

II. Py 행렬 연산

1. 행렬의 곱셈

- 행렬의 곱셈에 사용되는 엑셀 함수는 MMULT
 - A3부터 D6까지 A행렬을 입력하고, F3부터 H6까지 B행렬을 입력
 - 행렬 곱셈을 하기 위해 행렬 곱셈의 결과가 구해질 영역(예를 들어, C9부터 E12)을 마우스로 끌어서 연속되게 선택
 - 식 =MMULT(A3:D6,F3:H6)을 입력한 후 Ctrl+Shift+Enter를 동시에 누름

	A	B	C	D	E	F	G	H
1								
2	행렬A					행렬B		
3	3	1	7	5	5	9	7	
4	6	2	9	9	3	6	8	
5	7	3	4	6	8	5	4	
6	1	5	8	4	6	2	7	
7								
8			A*B					
9			=ult(a3:d6,f3:h6)					
10								
11								
12								

	A	B	C	D	E	F	G	H
1								
2	행렬A					행렬B		
3	3	1	7	5	5	9	7	
4	6	2	9	9	3	6	8	
5	7	3	4	6	8	5	4	
6	1	5	8	4	6	2	7	
7								
8			A*B					
9			104	78	92			
10			162	129	157			
11			112	113	131			
12			108	87	107			

2. 전치행렬

- 행과 열이 바뀐 전치행렬을 구하는 엑셀 함수는 TRANSPOSE
- 행렬 B의 전치행렬을 구하기 위해서는 전치행렬이 구해질 영역(예를 들어, G9부터 J11)을 마우스로 끌어서 연속되게 선택
- 식 =TRANSPOSE(F3:H6)를 입력한 후 Ctrl+Shift+Enter를 동시에 누름

	A	B	C	D	E	F	G	H	I	J
1										
2	행렬A					행렬B				
3	3	1	7	5		5	9	7		
4	6	2	9	9		3	6	8		
5	7	3	4	6		8	5	4		
6	1	5	8	4		6	2	7		
7										
8		A*B				행렬B의 전치행렬				
9			104	78	92		5	3	8	6
10			162	129	157		9	6	5	2
11			112	113	131		7	8	4	7
12										

3. 행렬식

- 행렬식(determinant)을 구하는데 사용되는 엑셀 함수는 MDETERM
- 행렬 A의 행렬식을 구하고자 할 경우 아무 셀(예를 들어 A9)에 셀 포인터를 위치
- 식 =MDETERM(A3:D6)을 입력하고 Enter를 누름

	A	B	C	D
1				
2	행렬A			
3	3	1	7	5
4	6	2	9	9
5	7	3	4	6
6	1	5	8	4
7				
8				
9	236			
..				

5. 연립방정식 풀이

- 왼쪽의 연립방정식은 오른쪽의 행렬로 나타낼 수 있음
- 연립방정식의 해는 $X = A^{-1}H$ 와 같이 구할 수 있음

$$\begin{array}{l}
 a_{11}X_1 + a_{12}X_2 + a_{13}X_3 = h_1 \\
 a_{21}X_1 + a_{22}X_2 + a_{23}X_3 = h_2 \\
 a_{31}X_1 + a_{32}X_3 + a_{33}X_3 = h_3
 \end{array}
 \quad
 \begin{array}{c}
 \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}
 \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}
 =
 \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \\
 A \quad X \quad H
 \end{array}$$

- 예 : 다음의 연립방정식 해 구하기
 - A 행렬과 H 행렬을 입력
 - 역행렬의 결과가 구해질 영역(예를 들어, A7부터 C9)을 마우스로 끌어서 연속되게 선택
 - 식 =MINVERSE(A2:C4)을 입력한 후 Ctrl+Shift+Enter를 동시에 누름

$$\begin{array}{l}
 2X + 3Y = 24 \\
 3X - 2Z = 5 \\
 X + 2Y + Z = 17
 \end{array}$$



	A	B	C	D	E	F
1	X	Y	Z		상수	
2		2	3	0		24
3		3	0	-2		5
4		1	2	1		17
5						
6	역행렬				해	
7	-0.57143	0.428571	0.857143		X=	3
8	0.714286	-0.28571	-0.57143		Y=	6
9	-0.85714	0.142857	1.285714		Z=	2
10						

b3-ch3-1.py

```
import numpy as np
# 행렬 만들기 : 4x4 행렬 a 및 4x3 행렬 b를 생성
a = np.matrix([[3,1,7,5], [6,2,9,9], [7,3,4,6], [1,5,8,4]])
a
b = np.matrix([[5,9,7], [3,6,8], [8,5,4], [6,2,7]])
b
print("Data Frame of a is :",f'\n{a}')
print("Data Frame of b is :",f'\n{b}')
# 행렬 a와 b를 곱하여 4x3 행렬을 생성
ab = a @ b
ab
print("Data Frame of a*b is :",f'\n{ab}')
# 행렬 b의 전치행렬과 a를 곱하여 3x4 행렬을 생성
bpa = b.T @ a
bpa
print("Data Frame of b*a is :",f'\n{bpa}')
# 행렬 a의 역행렬을 계산
ainv = np.linalg.inv(a)
ainv
print("Data Frame of inverse a is :",f'\n{ainv}')
# 행렬 a와 a의 역행렬을 곱하여 항등행렬 생성 확인
iden = a @ ainv
iden
print("Data Frame of a*inverse a is :",f'\n{iden}')
```

```
Data Frame of a is :
[[3 1 7 5]
 [6 2 9 9]
 [7 3 4 6]
 [1 5 8 4]]
Data Frame of b is :
[[5 9 7]
 [3 6 8]
 [8 5 4]
 [6 2 7]]
Data Frame of a*b is :
[[104 78 92]
 [162 129 157]
 [112 113 131]
 [108 87 107]]
Data Frame of b*a is :
[[ 95 65 142 124]
 [100 46 153 137]
 [104 70 193 159]]
Data Frame of inverse a is :
[[ 0.56779661 -0.48305085  0.33050847 -0.11864407]
 [-0.33050847  0.04237288  0.07627119  0.20338983]
 [ 0.57627119 -0.3559322  0.05932203 -0.00847458]
 [-0.88135593  0.77966102 -0.29661017  0.04237288]]
Data Frame of a*inverse a is :
[[ 1.00000000e+00  3.33066907e-16  1.11022302e-16 -1.38777878e-17]
 [ 3.33066907e-16  1.00000000e+00  1.11022302e-16 -6.93889390e-17]
 [ 2.22044605e-16  2.22044605e-16  1.00000000e+00 -2.77555756e-17]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

b3-ch3-1.py

앞에서 계속

AX = H의 연립방정식 해 구하기

A = np.matrix([[2,3,0], [3,0,-2], [1,2,1]])

A

print("Data Frame of A is :",f"\n{A}')

H = np.matrix([[24],[5],[17]])

H

print("Data Frame of H is :",f"\n{H}')

Ainv = np.linalg.inv(A)

Ainv

print("Data Frame of inverse A is :",f"\n{Ainv}')

X = Ainv @ H

X

print("Data Frame of X is :",f"\n{X}')

```

Data Frame of A is :
[[ 2  3  0]
 [ 3  0 -2]
 [ 1  2  1]]
Data Frame of H is :
[[24]
 [ 5]
 [17]]
Data Frame of inverse A is :
[[-0.57142857  0.42857143  0.85714286]
 [ 0.71428571 -0.28571429 -0.57142857]
 [-0.85714286  0.14285714  1.28571429]]
Data Frame of X is :
[[3.]
 [6.]
 [2.]]
    
```