Python最終発表

鈴木先生の線形代数 の問題Pythonで解い てみた! 坂田燎平

全部で6題の問題を Pythonを使用して解 きました!

$$(1) \left(\begin{array}{ccc} 2 & -2 \\ 1 & 0 \\ 2 & 1 \end{array}\right) \left(\begin{array}{cccc} -1 & 2 & 1 \\ 2 & 1 & -3 \end{array}\right)$$

$$= \begin{pmatrix} -6 & 2 & 8 \\ -1 & 2 & 1 \\ 0 & 5 & -1 \end{pmatrix} = \begin{pmatrix} 6 \\ -5 \end{pmatrix}$$

$$(1) \begin{pmatrix} 2 & -2 \\ 1 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} -1 & 2 & 1 \\ 2 & 1 & -3 \end{pmatrix}$$

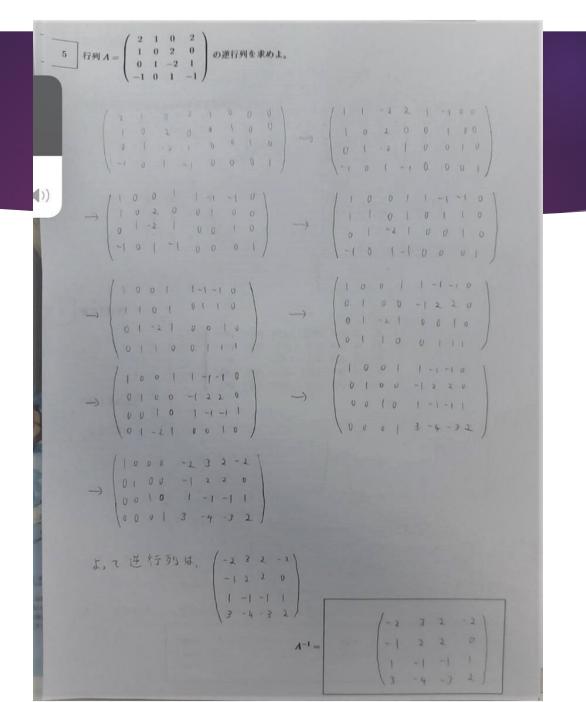
$$(2) \begin{pmatrix} 1 & -2 & 2 & 1 \\ 3 & 2 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ -1 \\ 3 \end{pmatrix}$$

(1)
$$\begin{pmatrix} -6 & 2 & 8 \\ -1 & 2 & 1 \\ 0 & 5 & -1 \end{pmatrix}$$
 (2) $\begin{pmatrix} 6 \\ -5 \end{pmatrix}$

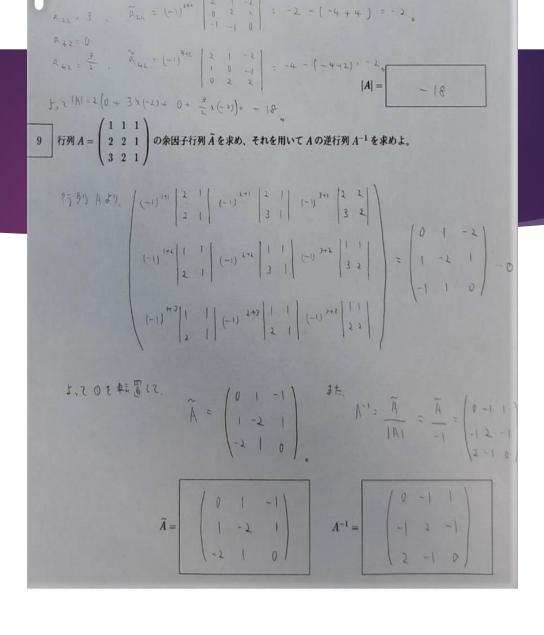
$$\begin{pmatrix} 6 \\ -5 \end{pmatrix}$$

```
In [31]: A=np.array([[2, -2], [1,0],[2, 1]])
In [32]: B=np.array([[-1, 2, 1], [2, 1, -3]])
In [33]: result=np.dot(A, B)
In [34]: result
Out[34]:
array([[-6, 2, 8],
       [-1, 2, 1],
       [0, 5, -1]
In [35]: C=np.array([[1],[-2], [-1], [3]])
In [36]: D=np.array([[1, -2, 2, 1], [3, 2, 1, -1]])
In [37]: result=np.dot(D, C)
In [38]: result
Out[38]:
array([[ 6],
       [-5]])
```

```
..: # シンボルを定義
                           ...: a, b, c, d = sp.symbols('a b c d')
                                       : A = sp.Matrix([
                                                                  [1, 1, 1, 1],
[a, b, c, d],
[a**2, b**2, c**2, d**2],
[a**3, b**3, c**3, d**3]
                 ...: det_A = A.det()
                  ...: condition = sp.simplify(det_A != 0)
                           ...: print("行列式:")
                      ...: sp.pprint(det_A)
                  ...: print("\n正則行列となる必要十分条件:")
                    ...: sp.pprint(condition)
 a · b · c - a · b · d - a · b · c + a · b · d + a · c · d - a · c · d - a · b · c + a · ↔
  ⇔b·d+a·b·c -a·b·d -a·c·d+a·c·d +a·b·c -a·b·d -a·b ↔
        \hookrightarrow \cdot c + a \cdot b \cdot d + a \cdot c \cdot d - a \cdot c \cdot d - b \cdot c \cdot d + b \cdot c \cdot d + b \cdot c \cdot d - b \cdot c \cdot \hookrightarrow c \cdot d + b \cdot c \cdot d + b 
 正則行列となる必要十分条件:
```



sympyというパックージを使用してみました。 numpyより逆行列がわかりやすいです。



```
in [51]: import numpy as np
   ...: # 行列 A を定義
   ...: A = np.array([
           [1, 1, 1],
           [2, 2, 1],
           [3, 2, 1]
   . . . : ])
   ...: # 余因子行列を計算する関数
   ...: def cofactor_matrix(matrix):
           n = matrix.shape[0] # 行列のサイズ (正方行列を仮定)
           cofactors = np.zeros_like(matrix, dtype=float) # 余因子行列
   ...: の初期化
           for i in range(n):
          for j in range(n):
                  # 小行列を取得
                  minor = np.delete(np.delete(matrix, i, axis=0), j, ax
   ...: is=1)
                  # 余因子を計算
                  cofactors[i, j] = ((-1) ** (i + j)) * np.linalg.det(m)
   ...: inor)
           return cofactors
   ...: # 余因子行列を計算
   ...: A_cofactor = cofactor_matrix(A)
   ...: # 結果を表示
   ...: print("行列 A の 余因子行列:")
   ...: print(A_cofactor)
行列 A の余因子行列:
[[ 0. 1. -2.]
[ 1. -2. 1.]
 [-1. 1. 0.]]
```

```
In [50]: import sympy as sp
    ...: # 行列 A を定義
    ...: A = sp.Matrix([
    ...: [1, 1, 1],
    ...: [2, 2, 1],
            [3, 2, 1]
    ...: # 逆行列を計算
    \dots: A_inv = A.inv()
    ...: print("行列 A の逆行列:")
    ...: sp.pprint(A_inv)
    A の逆行列:
```

手で計算するより、 Pythonを使用すれば、簡 単かつ楽に問題を解くこと ができる! Pythonでは記号を扱うの が難しい。