In [1]:

```
import sympy
from sympy import Matrix
import numpy as np
```

In [2]:

```
%matplotlib notebook
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
```

Mathematics for Machine Learning

Session 03: solving systems of linear equations

Gerhard Jäger

October 29, 2024

Python and SymPy

In [3]:

```
import sympy
from sympy import Matrix
```

- creating a vector

In [4]:

```
x = Matrix([1,2,3])
x
```

Out[4]:

$$\begin{bmatrix}1\\2\\3\end{bmatrix}$$

- creating a matrix

In [5]:

```
A = Matrix(
[
    [1,2, 3],
    [4,5,6],
    [7,8,9]
])
A
```

Out[5]:

$$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$$

vector algebra

In [6]:
```
y = Matrix([3,10,4])
y
```

Out[6]:

$$\begin{bmatrix}3\\10\\4\end{bmatrix}$$

In [7]:
```
x+y
```

Out[7]:

$$\begin{bmatrix}4\\12\\7\end{bmatrix}$$

In [8]:
```
3*x
```

Out[8]:

$$\begin{bmatrix}3\\6\\9\end{bmatrix}$$

In [9]:
```
2 * x - 3 * y
```

Out[9]:

$$\begin{bmatrix}-7\\-26\\-6\end{bmatrix}$$

applying a vector to a matrix

In [10]:
```
A * x
```

Out[10]:

$$\begin{bmatrix}14\\32\\50\end{bmatrix}$$

matrix transposition

In [11]:
```
A.T
```

Out[11]:

$$\begin{bmatrix}1 & 4 & 7\\2 & 5 & 8\\3 & 6 & 9\end{bmatrix}$$

matrix multiplication

In [12]:
```
B = Matrix(
[
    [4,1,0],
    [1,0,2],
    [4,5,6]
])
```

```
B
```

Out[12]:

$$\begin{bmatrix}4 & 1 & 0\\1 & 0 & 2\\4 & 5 & 6\end{bmatrix}$$

In [13]:

```
A * B
```

Out[13]:

$$\begin{bmatrix}18 & 16 & 22\\45 & 34 & 46\\72 & 52 & 70\end{bmatrix}$$

identity matrix

In [14]:

```
sympy.eye(3)
```

Out[14]:

$$\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$$

diagonal matrix

In [15]:

```
sympy.diag([1,2,3])
```

Out[15]:

$$\begin{bmatrix}1\\2\\3\end{bmatrix}$$

matrix inverse

In [16]:

```
A = Matrix(
[
    [1, -4, 2],
    [-2, 1, 3],
    [2,6,8]
])
```

In [17]:

```
A.inv()
```

Out[17]:

$$\begin{bmatrix}\frac{5}{63} & -\frac{22}{63} & \frac{1}{9}\\-\frac{11}{63} & -\frac{2}{63} & \frac{1}{18}\\\frac{1}{9} & \frac{1}{9} & \frac{1}{18}\end{bmatrix}$$

In [18]:

```
B = Matrix(
[
    [1, -4, 2],
    [-2, 1, 3],
    [2,6,-10]
])
```

In [19]:

```
try:
```

```
    B.inv()
except ValueError:
    print("B is not invertible")
```

```
B is not invertible
```

Symbolic computation with SymPy

The real strength of Sympy is that it can calculate with variables as well as with numbers.

In [20]:

```
from sympy import symbols
a,b,c,d = symbols('a b c d')
```

In [21]:

```
A = Matrix([
    [a, b],
    [c,d]
])
A
```

Out[21]:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

In [22]:

```
A.inv()
```

Out[22]:

$$\begin{bmatrix} \frac{d}{ad-bc} & -\frac{b}{ad-bc} \\ -\frac{c}{ad-bc} & \frac{a}{ad-bc} \end{bmatrix}$$

Back to solving systems of linear equations

First part: lower elimination

$$\begin{array}{rcrcrcr} 3x & + & 4y & - & z & = & 0 \\ & & y & + & z & = & 4 \\ x & - & y & - & z & = & -5 \end{array}$$

- subtracting $\frac{1}{3}$ times the first equation from the third equation:

$$\begin{array}{rcrcrcr} 3x & + & 4y & - & z & = & 0 \\ & & y & + & z & = & 4 \\ & - & \frac{7}{3}y & - & \frac{2}{3}z & = & -5 \end{array}$$

- adding $\frac{7}{3}$ times the second equation to the third equation:

$$\begin{array}{rcrcrcr} 3x & + & 4y & - & z & = & 0 \\ & & y & + & z & = & 4 \\ & & & & \frac{5}{\ }z & = & \frac{13}{\ } \end{array}$$

Lower elimination is completed if the left-hand side has a **upper triangular** shape.

Second part: upper elimination (a.k.a backward substitution)

In the elimination part, we transformed all coefficients below the diagonal to 0. In the substitution part, we successively transform all coefficients **above** the diagonal to 0.

- subtract $\frac{3}{5}$ times third equation from second equation.

$$\begin{array}{rcrcrcr} 3x & + & 4y & - & z & = & 0 \\ & & y & & & = & \frac{7}{5} \\ & & & & \frac{5}{3}z & = & \frac{13}{3} \end{array}$$

- add $\frac{3}{5}$ times third equation to first equation.

$$\begin{array}{rcrcrcr} 3x & + & 4y & & & = & \frac{13}{5} \\ & & y & & & = & \frac{7}{5} \\ & & & & \frac{5}{3}z & = & \frac{13}{3} \end{array}$$

- subtract 4 times second equation from first equation.

$$\begin{array}{rcrcrcr} 3x & & & & & = & -3 \\ & & y & & & = & \frac{7}{5} \\ & & & & \frac{5}{3}z & = & \frac{13}{3} \end{array}$$

Third step: dividing by pivots

The diagonal coefficients are called the **pivots**. In the last step, we transform them into 1.

- divide first equation by 3

$$\begin{array}{rcrcrcr} x & & & & & = & -1 \\ & & y & & & = & \frac{7}{5} \\ & & & & \frac{5}{3}z & = & \frac{13}{3} \end{array}$$

- divide third equation by $\frac{5}{3}$

$$\begin{array}{rcrcrcr} x & & & & & = & -1 \\ & & y & & & = & \frac{7}{5} \end{array}$$

$$z = \frac{13}{5}$$

checking result

$$\begin{array}{rcrcrcr}
3x & + & 4y & - & z & = & 0 \\
 & & y & + & z & = & 4 \\
x & - & y & - & z & = & -5
\end{array}$$

becomes

$$\begin{array}{rcrcrcr}
-3 & + & 4\frac{7}{5} & - & \frac{13}{5} & = & 0 \\
 & & \frac{7}{5} & + & \frac{13}{5} & = & 4 \\
x & - & \frac{7}{5} & - & \frac{13}{5} & = & -5
\end{array}$$

In [23]:

```python
A = np.array([
    [3,4,-1],
    [0, 1, 1],
    [1, -1, -1]
])
print(A)
```

```
[[ 3  4 -1]
 [ 0  1  1]
 [ 1 -1 -1]]
```

In [24]:

```python
v = np.array([0, 4, -5])
print(v)
```

```
[ 0  4 -5]
```

In [25]:

```python
np.linalg.inv(A) @ v
```

Out[25]:

```
array([-1. ,  1.4,  2.6])
```

Stripping down the superfluous notation, this is a sequence of **augmented matrices**. The augmented matrix is the matrix of coefficients on the left-hand side, plus the right-hand side as additional column.

- system of equations in matrix notation

$$A = \begin{bmatrix} 3 & 4 & -1 \\ 0 & 1 & 1 \\ 1 & -1 & -1 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} 0 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} 4 \\ -5 \end{bmatrix}$$

$$A\mathbf{x} = \mathbf{b}$$

- augmented matrix

$$\left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 1 & 4 \\ 1 & -1 & -1 & -5 \end{array}\right]$$

lower elimination: transforming below-diagonal entries to 0

- subtracting $\frac{1}{3}$ times first row from third row

$$\left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 1 & 4 \\ 0 & -\frac{7}{3} & -\frac{2}{3} & -5 \end{array}\right]$$

- adding $\frac{7}{3}$ times second row to third row

$$\left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 1 & 4 \\ 0 & 0 & \frac{5}{3} & \frac{13}{3} \end{array}\right]$$

upper elimination: transforming above-diagonal entries to 0

- subtract $\frac{3}{5}$ times third row from second row

$$\left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 0 & \frac{7}{5} \\ 0 & 0 & \frac{5}{3} & \frac{13}{3} \end{array}\right]$$

- add $\frac{3}{5}$ times third row to first row

$$\left[\begin{array}{ccc|c} 3 & 4 & 0 & \frac{13}{5} \\ 0 & 1 & 0 & \frac{7}{5} \\ 0 & 0 & \frac{5}{3} & \frac{13}{3} \end{array}\right]$$

- subtract 4 times second row from first row

$$\left[\begin{array}{ccc|c} 3 & 0 & 0 & -3 \end{array}\right]$$

$$\left[\begin{array}{ccc|c} 0 & 1 & 0 & \frac{7}{5} \\ 0 & 0 & \frac{5}{3} & \frac{13}{3} \end{array}\right]$$

dividing by pivots

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & \frac{7}{5} \\ 0 & 0 & 1 & \frac{13}{5} \end{array}\right]$$

All these steps can be formalized as multiplication with some matrix.

- subtracting $\frac{1}{3}$ times first row from third row

$$E_{3,1}\left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 1 & 4 \\ 1 & -1 & -1 & -5 \end{array}\right] = \left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 1 & 4 \\ 0 & -\frac{7}{3} & \frac{2}{3} & -5 \end{array}\right]$$

$$E_{3,1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{bmatrix}$$

- adding $\frac{7}{3}$ times second row to third row

$$E_{3,2}\left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 1 & 4 \\ 0 & -\frac{7}{3} & -\frac{2}{3} & -5 \end{array}\right] = \left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 1 & 4 \\ 0 & 0 & \frac{5}{3} & \frac{13}{3} \end{array}\right]$$

$$E_{3,2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{7}{3} & 1 \end{bmatrix}$$

- subtract $\frac{3}{5}$ times third row from second row

$$E_{2,3}\left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 1 & 4 \\ 0 & 0 & \frac{5}{3} & \frac{13}{3} \end{array}\right] = \left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 0 & \frac{7}{5} \\ 0 & 0 & \frac{5}{3} & \frac{13}{3} \end{array}\right]$$

$$E_{2,3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\frac{3}{5} \\ 0 & 0 & 1 \end{bmatrix}$$

- add $\frac{3}{5}$ times third row to first row

$$E_{1,3}\left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 0 & \frac{7}{5} \\ 0 & 0 & \frac{5}{3} & \frac{13}{3} \end{array}\right] = \left[\begin{array}{ccc|c} 3 & 4 & 0 & \frac{13}{5} \\ 0 & 1 & 0 & \frac{7}{5} \\ 0 & 0 & \frac{5}{3} & \frac{13}{3} \end{array}\right]$$

$$E_{1,3} = \begin{bmatrix} 1 & 0 & \frac{3}{5} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- subtract 4 times second row from first row

$$E_{1,2}\left[\begin{array}{ccc|c} 3 & 4 & 0 & \frac{13}{5} \\ 0 & 1 & 0 & \frac{7}{5} \\ 0 & 0 & \frac{5}{3} & \frac{13}{3} \end{array}\right] = \left[\begin{array}{ccc|c} 3 & 0 & 0 & -3 \\ 0 & 1 & 0 & \frac{7}{5} \\ 0 & 0 & \frac{5}{3} & \frac{13}{3} \end{array}\right]$$

$$E_{1,2} = \begin{bmatrix} 1 & -4 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- divide each row by its pivot

$$D\left[\begin{array}{ccc|c} 3 & 0 & 0 & -3 \\ 0 & 1 & 0 & \frac{7}{5} \\ 0 & 0 & \frac{5}{3} & \frac{13}{3} \end{array}\right] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & \frac{7}{5} \\ 0 & 0 & 1 & \frac{13}{5} \end{array}\right]$$

$$D = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{3}{5} \end{bmatrix}$$

Let us reflect what we did. Taking everyting together, we have

$$DE_{1,2}E_{1,3}E_{2,3}E_{3,2}E_{3,1}A = \mathbf{I}$$
$$DE_{1,2}E_{1,3}E_{2,3}E_{3,2}E_{3,1}\mathbf{b} = \mathbf{x}$$

Therefore

$$DE_{1,2}E_{1,3}E_{2,3}E_{3,2}E_{3,1} = A^{-1}$$

Let's check this with SymPy.

In [26]:
```python
from sympy import Rational, diag
```

In [27]:
```python
A = Matrix([
    [3,4,-1],
    [0,1,1],
    [1,-1,-1]
])
A
```

Out[27]:
$$\begin{bmatrix} 3 & 4 & -1 \\ 0 & 1 & 1 \\ 1 & -1 & -1 \end{bmatrix}$$

In [28]:
```python
b = Matrix([0, 4, -5])
b
```

Out[28]:
$$\begin{bmatrix} 0 \\ 4 \\ -5 \end{bmatrix}$$

In [29]:
```python
E31 = Matrix(
[
    [1, 0, 0],
    [0, 1, 0],
    [-Rational(1,3), 0, 1]
])
E31
```

Out[29]:
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{bmatrix}$$

In [30]:
```python
E32 = Matrix([
    [1,0,0],
    [0,1,0],
    [0,Rational(7,3),1]
])
E32
```

Out[30]:
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{7}{3} & 1 \end{bmatrix}$$

In [31]:
```python
E23 = Matrix([
    [1,0,0],
    [0,1,-Rational(3,5)],
    [0,0,1]
])
E23
```

Out[31]:
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\frac{3}{5} \\ 0 & 0 & 1 \end{bmatrix}$$

In [32]:

```
E13 = Matrix([
    [1,0,Rational(3,5)],
    [0,1,0],
    [0,0,1]
])
E13
```

Out[32]:

$$\begin{bmatrix}1 & 0 & \frac{3}{5}\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$$

In [33]:

```
E12 = Matrix([
    [1,-4,0],
    [0,1,0],
    [0,0,1]
])
E12
```

Out[33]:

$$\begin{bmatrix}1 & -4 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$$

In [34]:

```
D = diag(Rational(1,3), 1, Rational(3,5))
D
```

Out[34]:

$$\begin{bmatrix}\frac{1}{3} & 0 & 0\\0 & 1 & 0\\0 & 0 & \frac{3}{5}\end{bmatrix}$$

In [35]:

```
D * E12 * E13 * E23 * E32 * E31 * A
```

Out[35]:

$$\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$$

In [36]:

```
D * E12 * E13 * E23 * E32 * E31 * b
```

Out[36]:

$$\begin{bmatrix}-1\\\frac{7}{5}\\\frac{13}{5}\end{bmatrix}$$

In [37]:

```
D * E12 * E13 * E23 * E32 * E31
```

Out[37]:

$$\begin{bmatrix}0 & 1 & 1\\\frac{1}{5} & -\frac{2}{5} & -\frac{3}{5}\\-\frac{1}{5} & \frac{7}{5} & \frac{3}{5}\end{bmatrix}$$

In [38]:

```
A.inv()
```

Out[38]:

$$\begin{bmatrix} 0 & 1 & 1 \\ \frac{1}{5} & -\frac{2}{5} & -\frac{3}{5} \\ -\frac{1}{5} & \frac{7}{5} & \frac{3}{5} \end{bmatrix}$$

So we can compute the inverse of a square matrix by multiplying out the elimination matrices.

There is also a shorter way to the inverse matrix: augment A with the identity matrix!

$$\begin{aligned} A\begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3\end{bmatrix} &= \mathbf{I} \\ &:: \\ DE_{1,2}E_{1,3}E_{2,3}E_{3,2}E_{3,1}A\begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3\end{bmatrix} &= DE_{1,2}E_{1,3}E_{2,3}E_{3,2}E_{3,1}\mathbf{I} \\ A^{-1}A\begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3\end{bmatrix} &= A^{-1}\mathbf{I} \\ \begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3\end{bmatrix} &= A^{-1} \end{aligned}$$

- augmented matrix:

$$\left[\begin{array}{ccc|ccc} 3 & 4 & -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 1 \end{array}\right]$$

- left-multiplication with $E_{3,1}$:

$$\left[\begin{array}{ccc|ccc} 3 & 4 & -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & -\frac{7}{3} & -\frac{2}{3} & -\frac{1}{3} & 0 & 1 \end{array}\right]$$

- left-multiplication with $E_{3,2}$:

$$\left[\begin{array}{ccc|ccc} 3 & 4 & -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & \frac{5}{3} & -\frac{1}{3} & \frac{7}{3} & 1 \end{array}\right]$$

- left-multiplication with $E_{2,3}$:

$$\left[\begin{array}{ccc|ccc} 3 & 4 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & \frac{1}{5} & -\frac{2}{5} & -\frac{3}{5} \\ 0 & 0 & \frac{5}{3} & -\frac{1}{3} & \frac{7}{3} & 1 \end{array}\right]$$

- left-multiplication with $E_{1,3}$:

$$\left[\begin{array}{ccc|ccc} 3 & 4 & 0 & \frac{4}{5} & \frac{7}{5} & \frac{3}{5} \\ 0 & 1 & 0 & \frac{1}{5} & -\frac{2}{5} & -\frac{3}{5} \end{array}\right]$$

$$\left[\begin{array}{ccc|ccc} 0 & 0 & \frac{5}{3} & -\frac{1}{3} & \frac{7}{3} & 1 \end{array}\right]$$

- left-multiplication with $E_{1,2}$:

$$\left[\begin{array}{ccc|ccc} 3 & 0 & 0 & 0 & 3 & 3 \\ 0 & 1 & 0 & \frac{1}{5} & -\frac{2}{5} & -\frac{3}{5} \\ 0 & 0 & \frac{5}{3} & -\frac{1}{3} & \frac{7}{3} & 1 \end{array}\right]$$

- left-multiplication with D:

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & \frac{1}{5} & -\frac{2}{5} & -\frac{3}{5} \\ 0 & 0 & 1 & -\frac{1}{5} & \frac{7}{5} & \frac{3}{5} \end{array}\right]$$

$$A^{-1} = \begin{bmatrix} 0 & 1 & 1 \\ \frac{1}{5} & -\frac{2}{5} & -\frac{3}{5} \\ -\frac{1}{5} & \frac{7}{5} & \frac{3}{5} \end{bmatrix}$$

Algorithm to find the inverse of a square matrix A (preliminary)

(@ represents matrix multiplication.)

```
Let n be the number of rows of A
X := A
Y := I
for i in 1:(n-1)
    for j in (i+1):n
        l := X[j,i]/X[i,i]
        Eji := I
        Eji[j,i] := −l
        X := Eji @ X
        Y := Eji @ Y
      end
end

for i in n:2
```

```
        for j in (i-1):1
                u := X[j,i]/X[i,i]
                Eji := I
                Eji[j,i] := -u
                X := Eji @ X
                Y := Eji @ Y
        end
    end

    D := I

    for i in 1:n
        D[i,i] = 1/X[i,i]
    end

    Y := D @ Y

    return Y
```

Permutation

The algorithm fails if a 0 occurs at the diagonal somewhere along the way. Sometimes, this is benign, sometimes not.

Here is a benign example:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

This matrix is invertible, according to SymPy:

In [39]:

```
A = Matrix([
    [1,1,1],
    [1,1,-1],
    [1,-1,1]
])
A
```

Out[39]:

$$\begin{bmatrix}1 & 1 & 1\\1 & 1 & -1\\1 & -1 & 1\end{bmatrix}$$

In [40]:

```
A.inv()
```

Out[40]:

$$\begin{bmatrix}0 & \frac{1}{2} & \frac{1}{2}\\\frac{1}{2} & 0 & -\frac{1}{2}\\\frac{1}{2} & -\frac{1}{2} & 0\end{bmatrix}$$

Still, elimination fails:

$$\left[\begin{array}{ccc|ccc}1 & 1 & 1 & 1 & 0 & 0\\1 & 1 & -1 & 0 & 1 & 0\\1 & -1 & 1 & 0 & 0 & 1\end{array}\right]$$

- subtracting first row from second:

$$\left[\begin{array}{ccc|ccc}1 & 1 & 1 & 1 & 0 & 0\\0 & 0 & -2 & -1 & 1 & 0\\1 & -1 & 1 & 0 & 0 & 1\end{array}\right]$$

- subtracting first row from third:

$$\left[\begin{array}{ccc|ccc}1 & 1 & 1 & 1 & 0 & 0\\0 & 0 & -2 & -1 & 1 & 0\\0 & -2 & 0 & -1 & 0 & 1\end{array}\right]$$

Now, according to the algorithm, we have to subtract l times the second row from the third, with

$$l = \frac{-2}{0}$$

This is not possible. We can recover from this though, by swapping the second and third row:

$$\left[\begin{array}{ccc|ccc}1 & 1 & 1 & 1 & 0 & 0\\0 & -2 & 0 & -1 & 0 & 1\\0 & 0 & -2 & -1 & 1 & 0\end{array}\right]$$

The rest is straightforward:

- add $\frac{1}{2}$ times third row to first row:

$$\left[\begin{array}{ccc|ccc}1 & 1 & 0 & \frac{1}{2} & \frac{1}{2} & 0\\0 & -2 & 0 & -1 & 0 & 1\\0 & 0 & -2 & -1 & 1 & 0\end{array}\right]$$

- add $\frac{1}{2}$ times second row to first row:

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & -2 & 0 & -1 & 0 & 1 \\ 0 & 0 & -2 & -1 & 1 & 0 \end{array}\right]$$

- divide each row by its pivot

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 & \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{2} & -\frac{1}{2} & 0 \end{array}\right]$$

The step of exchanging the second and third row can also be done via matrix multiplication:

$$P_{2,3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Here we had a configuration where, during lower elimination, a 0 was in $X_{i,i}$ and a non-zero entry $X_{j,i}$ was below it.

In such a configuration, we can save the day by multiplying everything with $P_{i,j}$ from the left.

If both $X_{i,i} = 0$ and $X_{j,i} = 0$, we just do nothing and proceed to the next row.

This cannot happen during upper elimination, because it does not affect the diagonal.

Algorithm to find the inverse of a square matrix A (final version)

(@ represents matrix multiplication.)

```
Let n be the number of rows of A
X := A
Y := I
for i in 1:(n-1)
    for j in (i+1):n
        if X[j,i] != 0
            if X[i,i] == 0
```

```
                X := Pij @ X
                Y := Pij @ Y
            end
            l := X[j,i]/X[i,i]
            Eji := I
            Eji[j,i] := −l
            X := Eji @ X
            Y := Eji @ Y
        end
    end
end

for i in n:2
    for j in (i-1):1
          u := X[j,i]/X[i,i]
          Eji := I
          Eji[j,i] := −u
          X := Eji @ X
          Y := Eji @ Y
    end
end

D := I

for i in 1:n
    D[i,i] = 1/X[i,i]
end

Y := D @ Y

return Y
```

Systems of linear equation without solution

$$A = \begin{bmatrix} 3 & 4 & -1 \\ 0 & 1 & 1 \\ 0 & -1 & -1 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} 0 \\ 4

$$\mathbf{b} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$$

elimination step

$$\left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 1 & 4 \\ 0 & 0 & 0 & 9 \end{array}\right]$$

Here we already see that this system cannot have a solution, because the third row has no solution.

Systems of linear equation with infinitely many solution

$$A = \begin{bmatrix} 3 & 4 & -1 \\ 0 & 1 & 1 \\ 0 & -1 & -1 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} 0 \\ 4 \\ -4 \end{bmatrix}$$

augmented matrix

$$\left[\begin{array}{ccc|c} 3 & 4 & -1 & 0 \\ 0 & 1 & 1 & 4 \\ 0 & -1 & -1 & -4 \end{array}\right]$$

elimination step

$$\left[\begin{array}{ccc|c} 3 & 0 & -5 & -16 \\ 0 & 1 & 1 & 4 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

dividing by the pivots

$$\left[\begin{array}{ccc|c} 1 & 0 & -\frac{5}{3} & -\frac{16}{3} \\ 0 & 1 & 1 & 4 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

The set of solutions obtained this way is

$$x = \frac{5}{\ }z - \frac{16}{\ }$$

$$y = 4 - z$$
$$z \in \mathbb{R}$$

Permutations during elimination

Consider the following matrix:

In [41]:
```
A = Matrix([
    [1,4,5],
    [4,16,6],
    [5,6,3]
])
A
```

Out[41]:
$$\begin{bmatrix} 1 & 4 & 5 \\ 4 & 16 & 6 \\ 5 & 6 & 3 \end{bmatrix}$$

We want to find the inverse.

$$\left[\begin{array}{ccc|ccc} 1 & 4 & 5 & 1 & 0 & 0 \\ 4 & 16 & 6 & 0 & 1 & 0 \\ 5 & 6 & 3 & 0 & 0 & 1 \end{array}\right]$$

$$E_1 = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \left[\begin{array}{ccc|ccc} 1 & 4 & 5 & 1 & 0 & 0 \\ 0 & 0 & -14 & -4 & 1 & 0 \\ 5 & 6 & 3 & 0 & 0 & 1 \end{array}\right]$$

Since we have a 0 above a non-0, we have to exchange rows.

$$P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \left[\begin{array}{ccc|ccc} 1 & 4 & 5 & 1 & 0 & 0 \\ 5 & 6 & 3 & 0 & 0 & 1 \\ 0 & 0 & -14 & -4 & 1 & 0 \end{array}\right]$$

Now we can continue with elimination.

$$E_2 = \begin{bmatrix} 1 & 0 & 0 \\ -5 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \left[\begin{array}{ccc|ccc} 1 & 4 & 5 & 1 & 0 & 0 \\ 0 & -14 & -22 & -5 & 0 & 1 \\ 0 & 0 & -14 & -4 & 1 & 0 \end{array}\right]$$

$$E_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\frac{11}{7} \\ 0 & 0 & 1 \end{bmatrix} \quad \left[\begin{array}{ccc|ccc} 1 & 4 & 5 & 1 & 0 & 0 \\ 0 & -14 & 0 & \frac{9}{7} & -\frac{11}{7} & 1 \\ 0 & 0 & -14 & -4 & 1 & 0 \end{array}\right]$$

$$\begin{bmatrix} 1 & 0 & \frac{5}{14} \end{bmatrix} \quad \left[\begin{array}{ccc|ccc} 1 & 4 & 0 & -\frac{3}{7} & \frac{5}{14} & 0 \end{array}\right]$$

$$E_4 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \left[\begin{array}{ccc|ccc} 0 & -14 & 0 & \frac{9}{7} & -\frac{11}{7} & 1 \\ 0 & 0 & -14 & -4 & 1 & 0 \end{array}\right]$$

Alltogether we have