Linear algebra – Part 1

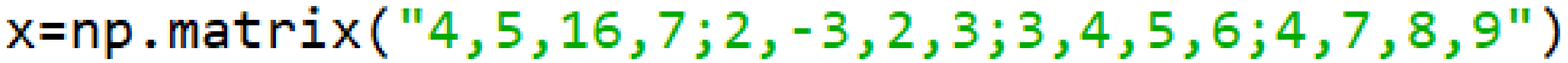
GITAA
Transforming careers

In this lecture

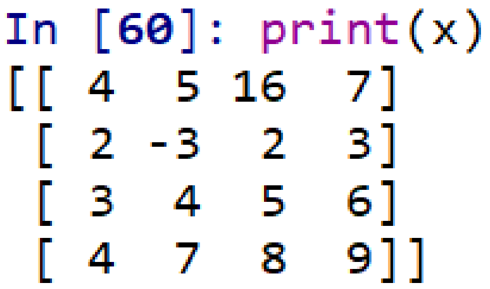
- Linear algebra operations in python
    - Determinant of matrix
    - Rank of matrix
    - Inverse of matrix
    - Solving system of equations

Determinant of matrix

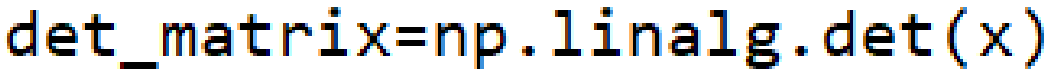
- **numpy.linalg.det()**- returns determinant of the matrix
- Syntax: **numpy.linalg.det(matrix)**



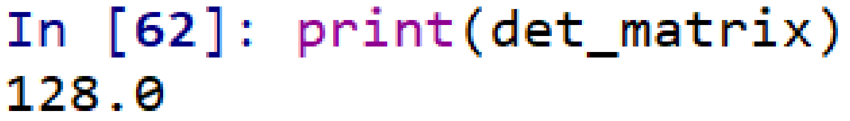
```
x=np.matrix("4,5,16,7;2,-3,2,3;3,4,5,6;4,7,8,9")
```



```
In [60]: print(x)
[[ 4  5 16  7]
 [ 2 -3  2  3]
 [ 3  4  5  6]
 [ 4  7  8  9]]
```



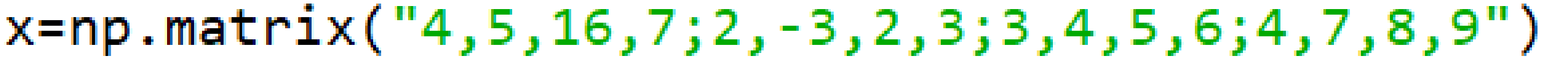
```
det_matrix=np.linalg.det(x)
```



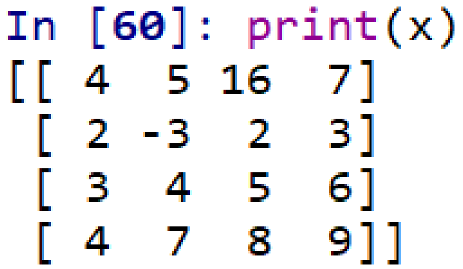
```
In [62]: print(det_matrix)
128.0
```

Rank of matrix

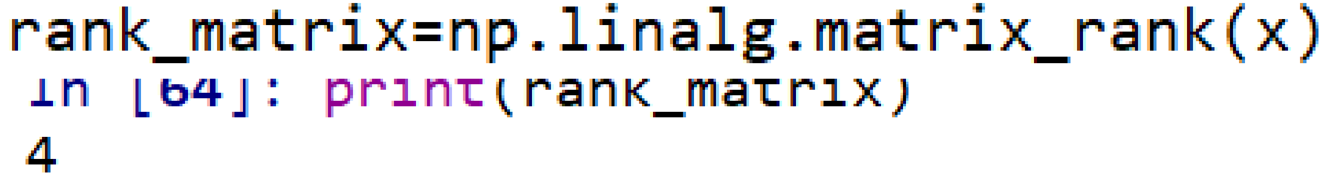
- **numpy.linalg.matrix_rank()**- returns rank of the matrix
- Syntax: **numpy.linalg.matrix_rank(matrix)**
- Consider the matrix ***x***



```
x=np.matrix("4,5,16,7;2,-3,2,3;3,4,5,6;4,7,8,9")
```



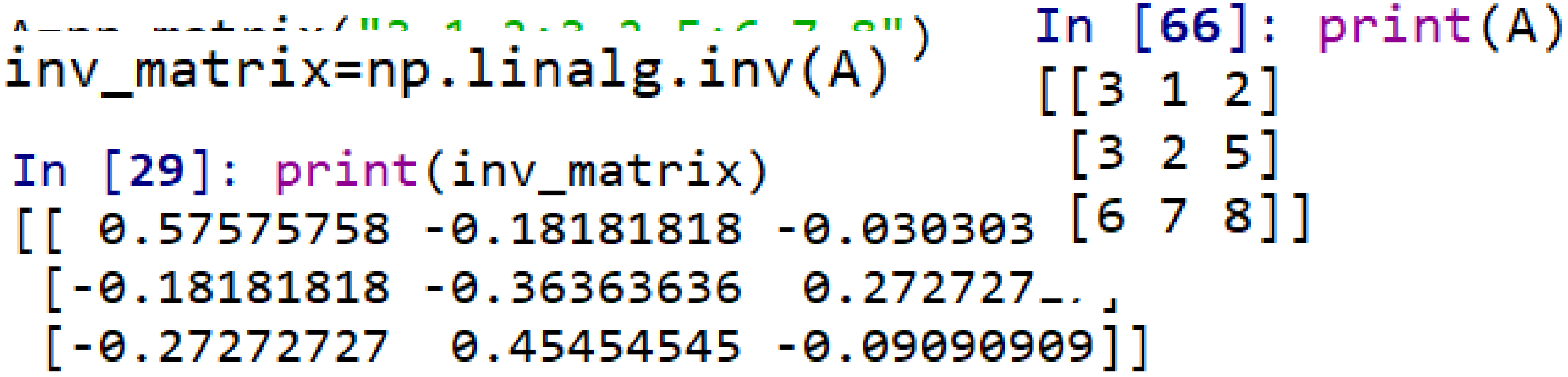
```
In [60]: print(x)
[[ 4  5 16  7]
 [ 2 -3  2  3]
 [ 3  4  5  6]
 [ 4  7  8  9]]
```



```
rank_matrix=np.linalg.matrix_rank(x)
 In [64]: print(rank_matrix)
 4
```

Inverse of a matrix

- **numpy.linalg.inv()**- returns the multiplicative inverse of a matrix
- Syntax: **numpy.linalg.inv(matrix)**
- Consider the matrix ***A***



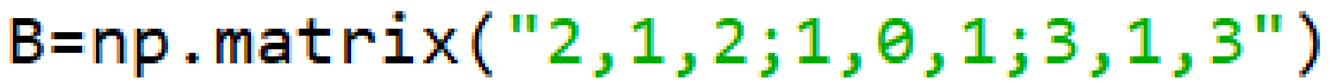
```
A=np.matrix("3 1 2;3 2 5;6 7 8")
inv_matrix=np.linalg.inv(A)
```

```
In [29]: print(inv_matrix)
[[ 0.57575758 -0.18181818 -0.030303
 [-0.18181818 -0.36363636  0.272727
 [-0.27272727  0.45454545 -0.09090909]]
```

```
In [66]: print(A)
[[3 1 2]
 [3 2 5]
 [6 7 8]]
```

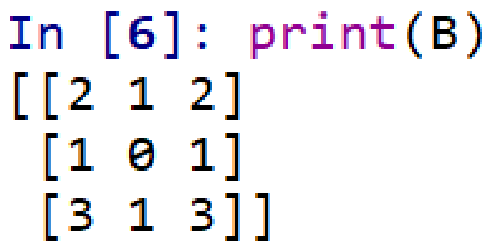
Inverse of a matrix

- Create a another matrix ***B***



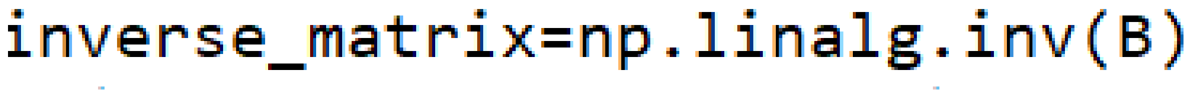
```
B=np.matrix("2,1,2;1,0,1;3,1,3")
```

- Print matrix ***B***

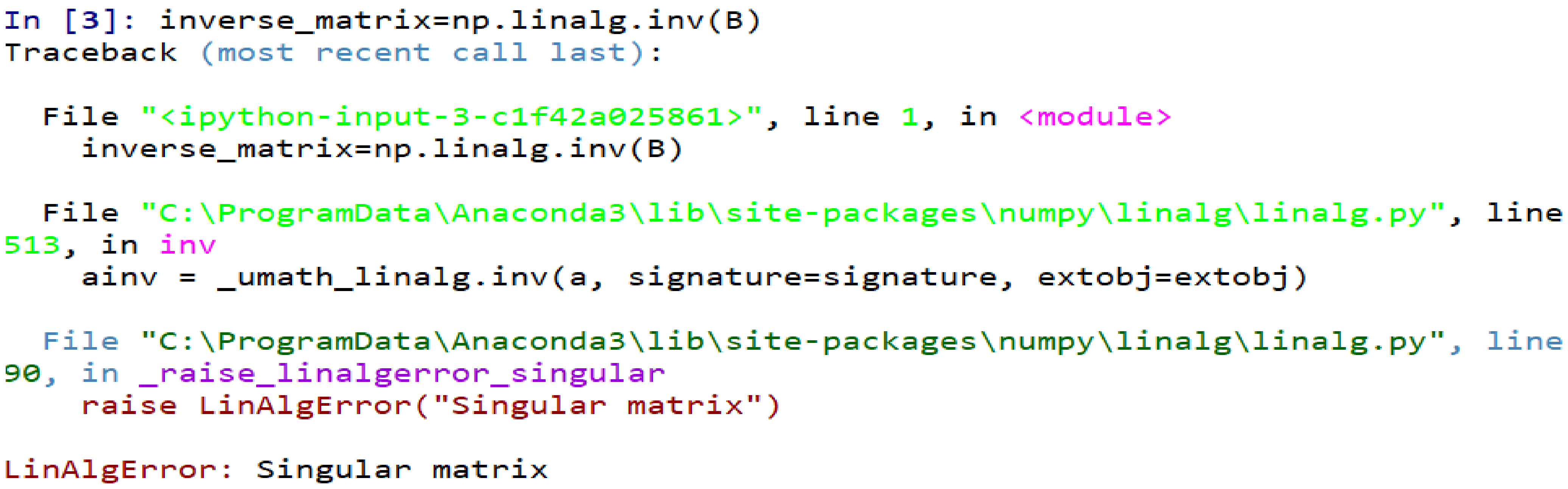


```
In [6]: print(B)
[[2 1 2]
 [1 0 1]
 [3 1 3]]
```

Inverse of a matrix



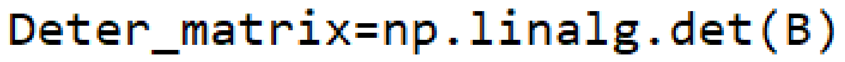
```
inverse_matrix=np.linalg.inv(B)
```



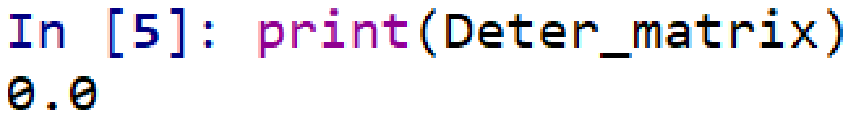
```
In [3]: inverse_matrix=np.linalg.inv(B)
Traceback (most recent call last):

  File "<ipython-input-3-c1f42a025861>", line 1, in <module>
    inverse_matrix=np.linalg.inv(B)

  File "C:\ProgramData\Anaconda3\lib\site-packages\numpy\linalg\linalg.py", line
513, in inv
    ainv = _umath_linalg.inv(a, signature=signature, extobj=extobj)

  File "C:\ProgramData\Anaconda3\lib\site-packages\numpy\linalg\linalg.py", line
90, in _raise_linalgerror_singular
    raise LinAlgError("Singular matrix")

LinAlgError: Singular matrix
```

Inverse of a matrix

- Find the determinant of matrix ***B***



```
Deter_matrix=np.linalg.det(B)
```



```
In [5]: print(Deter_matrix)
0.0
```

- The matrix ***B*** is singular matrix i.e. determinant is zero

System of linear equations

- Consider a system of linear equations

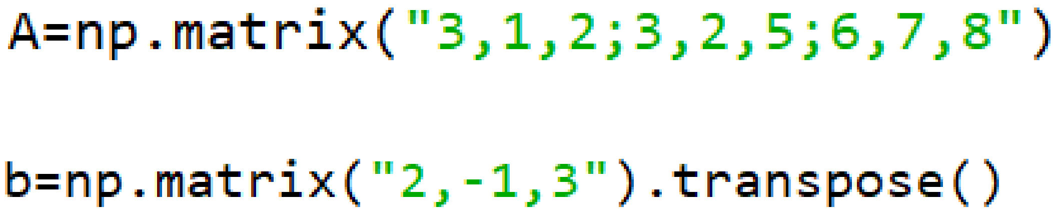
$$\begin{aligned} 3x + y + 2z &= 2 \\ 3x + 2y + 5z &= -1 \\ 6x + 7y + 8z &= 3 \end{aligned}$$

- Now we can write the equations in the form of Ax=b

$$\begin{aligned} 3x + y + 2z \;\; &= 2 \\ 3x + 2y + 5z &= -1 \\ 6x + 7y + 8z &= 3 \end{aligned} \quad \Longrightarrow \quad \underset{\textbf{A}}{\begin{pmatrix} 3 & 1 & 2 \\ 3 & 2 & 5 \\ 6 & 7 & 8 \end{pmatrix}} \underset{\textbf{x}}{\begin{pmatrix} x \\ y \\ z \end{pmatrix}} = \underset{\textbf{b}}{\begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix}}$$

System of linear equations

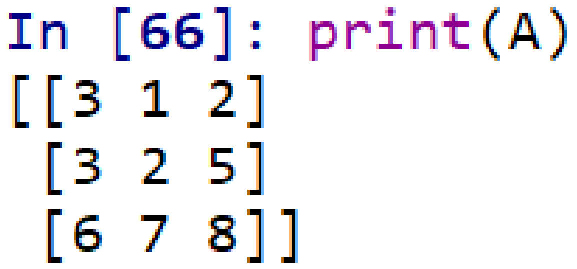
- **numpy.linalg.solve()**- return the solution to the system Ax=b
- Syntax: **numpy.linalg.solve(matrix_A,matrix_b)**
- Create matrix ***A*** and ***b***



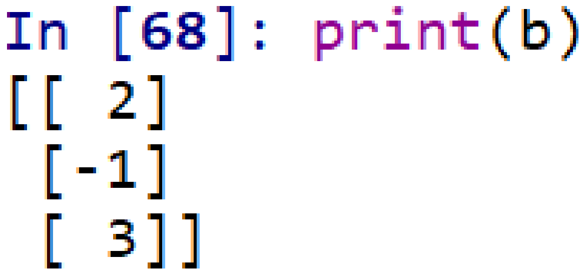
```
A=np.matrix("3,1,2;3,2,5;6,7,8")

b=np.matrix("2,-1,3").transpose()
```

System of linear equations

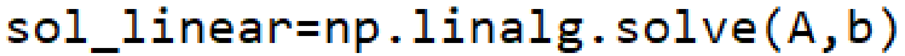
- Print matrix ***A*** and ***b***



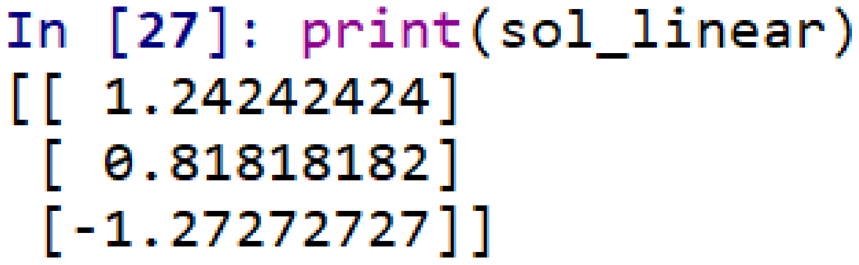
```
In [66]: print(A)
[[3 1 2]
 [3 2 5]
 [6 7 8]]
```



```
In [68]: print(b)
[[ 2]
 [-1]
 [ 3]]
```



```
sol_linear=np.linalg.solve(A,b)
```



```
In [27]: print(sol_linear)
[[ 1.24242424]
 [ 0.81818182]
 [-1.27272727]]
```

Summary

- Determinant of matrix
- Rank of matrix
- Inverse of matrix
- Solving system of equations

THANK YOU