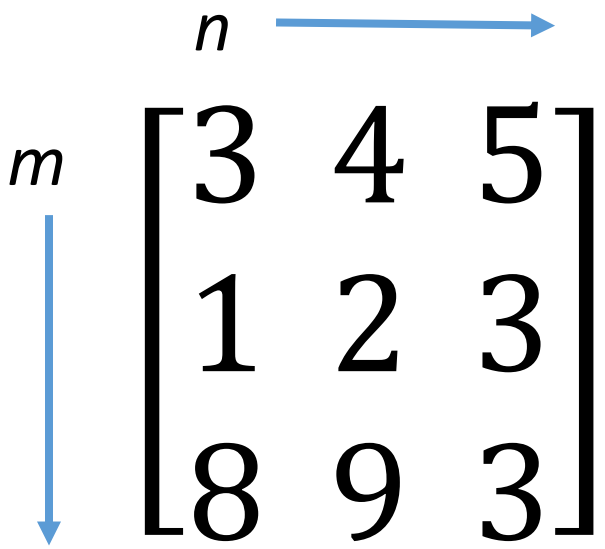
Matrix Operations with Python and Numpy




```
A = np.array([[3.0, 4.0, 5.0],
              [1.0, 2.0, 3.0],
              [8.0, 9.0, 3.0]])
```

Create Arrays in Python Numpy

```
A = np.array([[3.0, 4.0, 5.0],
              [1.0, 2.0, 3.0],
              [8.0, 9.0, 3.0]])
```

Create array A with values. 3 x 3 array with float datatype.

```
A = np.zeros((3, 3), np.float32)
```

Create array A with zeros. 3 x 3 array with float datatype.

```
A = np.zeros(10, np.int16)
```

Create array A with zeros. 1 Dimensional array with length of 10. Integer 16 bit depth datatype.

Element wise Addition

$$\begin{bmatrix} 3 & 4 & 5 \\ 1 & 2 & 3 \\ 8 & 9 & 3 \end{bmatrix} + \begin{bmatrix} 2 & 5 & 9 \\ 8 & 0 & 4 \\ 12 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 5 & 9 & 14 \\ 9 & 2 & 7 \\ 20 & 9 & 6 \end{bmatrix}$$

$$A + B = C$$

```
C = A + B # Element wise addition
```

Adding Each Element of Matrix (Sum of all elements)

$$A = \begin{bmatrix} 3 & 4 & 5 \\ 1 & 2 & 3 \\ 8 & 9 & 3 \end{bmatrix}$$

$$\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{(i,j)} = 38$$

```
total = sum(sum(A)) #Sum all elements
```

Element wise subtraction

$$\begin{bmatrix} 3 & 4 & 5 \\ 1 & 2 & 3 \\ 8 & 9 & 3 \end{bmatrix} - \begin{bmatrix} 2 & 5 & 9 \\ 8 & 0 & 4 \\ 12 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -4 \\ -7 & 2 & -1 \\ -4 & 9 & 6 \end{bmatrix}$$

$$A - B = C$$

```
C = A - B # Element wise subtraction
```

Element wise multiplication

$$\begin{bmatrix} 3 & 4 & 5 \\ 1 & 2 & 3 \\ 8 & 9 & 3 \end{bmatrix} * \begin{bmatrix} 2 & 5 & 9 \\ 8 & 0 & 4 \\ 12 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 6 & 20 & 45 \\ 8 & 0 & 12 \\ 96 & 0 & 9 \end{bmatrix}$$

$$A * B = C$$

```
C = A * B # Element wise multiplication
```

Scalar Multiplication

$$\begin{bmatrix} 3 & 4 & 5 \\ 1 & 2 & 3 \\ 8 & 9 & 3 \end{bmatrix} * 3 = \begin{bmatrix} 9 & 12 & 15 \\ 3 & 6 & 9 \\ 24 & 27 & 9 \end{bmatrix}$$

$$A * 3 = C$$

```
C = A * 3 # Scalar wise multiplication
```

Dot Product (Multiplication)

$$(AB)_{ij} = \sum_{k=0}^{m-1} A_{ik} B_{kj}$$

$$\begin{bmatrix} 3 & 4 & 5 \\ 1 & 2 & 3 \\ 8 & 9 & 3 \end{bmatrix} \text{x} \begin{bmatrix} 2 & 5 & 9 \\ 8 & 0 & 4 \\ 12 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 98 & 15 & 58 \\ 54 & 5 & 26 \\ 124 & 40 & 117 \end{bmatrix}$$

$$A \text{x} B = C \quad \text{or} \quad AB = C$$

```
C = np.dot(A, B) # Multiplication
```

Transpose Matrix

$$\begin{bmatrix} 3 & 4 & 5 \\ 1 & 2 & 3 \\ 8 & 9 & 3 \end{bmatrix}^T = \begin{bmatrix} 3 & 1 & 8 \\ 4 & 2 & 9 \\ 5 & 3 & 3 \end{bmatrix}$$

$$D = A^T$$

```
D = np.transpose(A) # Transpose
```

Inverse Matrix

$$\begin{bmatrix} 3 & 4 & 5 \\ 1 & 2 & 3 \\ 8 & 9 & 3 \end{bmatrix}^{-1} \approx \begin{bmatrix} 1.5 & -2.35 & -0.14 \\ -1.5 & 2.21 & 0.28 \\ 0.5 & -0.36 & -0.14 \end{bmatrix}$$

$$D = A^{-1}$$

```
D = np.linalg.inv(A) # Invert
```

Change datatype of Array

```
# Create Float Matrix
A = np.array([3.0, 4.0, 5.0])

# Convert to Int
B = A.astype(np.int)

#Convert to Float
C = A.astype(np.float)
```

Really Great Trick! Solving Systems of Equations

$$7x + 5y - 3z = 16$$

$$3x - 5y + 2z = -8$$

$$5x + 3y - 7z = 0$$

Solve for *x*, *y*, and *z*.

Systems of Equations: Convert to Matrix / Linear Algebra Expression

$$7x + 5y - 3z = 16$$
$$3x - 5y + 2z = -8$$
$$5x + 3y - 7z = 0$$

$$\begin{bmatrix} 7 & 5 & -3 \\ 3 & -5 & 2 \\ 5 & 3 & -7 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 16 \\ -8 \\ 0 \end{bmatrix}$$

$$A \quad \mathrm{x} \quad X \quad = \quad B$$

$$AX \quad = \quad B$$

Solve for X with Linear Algebra

$$\begin{bmatrix} 7 & 5 & -3 \\ 3 & -5 & 2 \\ 5 & 3 & -7 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 16 \\ -8 \\ 0 \end{bmatrix}$$

$$7x + 5y - 3z = 16$$
$$3x - 5y + 2z = -8$$
$$5x + 3y - 7z = 0$$

$$AX = B$$

$$X = A^{-1}B$$

Implement in Python

$$7x + 5y - 3z = 16$$
$$3x - 5y + 2z = -8$$
$$5x + 3y - 7z = 0$$

```
# Import Numpy
import numpy as np


# Setup Array - This works


A = np.array([[7., 5., -3.],
              [3., -5., 2.],
              [5., 3., -7.]])

B = np.array([[16.],
              [-8.],
              [0.]])


# Solve AX = B for X   (X = A^-1 . B)
X = np.dot(np.linalg.inv(A), B)
```

$$\begin{bmatrix} 7 & 5 & -3 \\ 3 & -5 & 2 \\ 5 & 3 & -7 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 16 \\ -8 \\ 0 \end{bmatrix}$$

$$AX = B$$

$$X = A^{-1}B$$

```
>>> X
array([[ 1.],
       [ 3.],
       [ 2.]])
```

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

$$x = 1, y = 3, z = 2$$