In [1]:
```python
import numpy as np
```

In [2]:
```python
a=np.array([[3,1,4],[9,4,5],[9,4,6]])
b=np.array([3,6,9])
```

In [3]:
```python
d=np.linalg.det(a)
d
```

Out[3]:
```
2.9999999999999996
```

Determinant of a!=0 Hence, Inverse Exist. Rank(A)=3 Rank(AB)=3, Hence has a Solution.

In [4]:
```python
x=np.linalg.solve(a,b)
x
```

Out[4]:
```
array([-9., 18.,  3.])
```

In [5]:
```python
a=np.array([[2,4,6,8],[6,6,8,8]])
a
```

Out[5]:
```
array([[2, 4, 6, 8],
       [6, 6, 8, 8]])
```

In [6]:
```python
a[0]=a[0]-a[0].mean()
a[1]=a[1]-a[1].mean()
#centered data in each row in matrix A
a
```

Out[6]:
```
array([[-3, -1,  1,  3],
       [-1, -1,  1,  1]])
```

In [7]:
```python
# Co-Variance Matrix
C=np.matmul(a,a.T)
C
```

Out[7]:
```
array([[20,  8],
       [ 8,  4]])
```

In [8]:
```python
Values,Vectors=np.linalg.eig(C)
print(f'Values: {Values}\nVectors: {Vectors}')
```

```
Values: [23.3137085  0.6862915]
Vectors: [[ 0.92387953 -0.38268343]
 [ 0.38268343  0.92387953]]
```

In [9]:
```python
# First Principle component is of Large value of eigen Value
First_PC=Vectors[:,0]
First_PC
```

Out[9]:
```
array([0.92387953, 0.38268343])
```

In [10]:
```python
# Reduced Data Matrix
np.matmul(First_PC,a)
```

```
Out[10]: array([-3.15432203, -1.30656296,  1.30656296,  3.15432203])
```