In [1]:
```
#Finding the inverse of a matrix
import numpy as np
from scipy import linalg
matrix = np.array([[10,6],[2,7]])
matrix
```

Out[1]:
```
array([[10,  6],
       [ 2,  7]])
```

In [2]:
```
type(matrix)
```

Out[2]:
```
numpy.ndarray
```

In [3]:
```
linalg.inv(matrix)
```

Out[3]:
```
array([[ 0.12068966, -0.10344828],
       [-0.03448276,  0.17241379]])
```

In [4]:
```
#Finding the determinant of a matrix
import numpy as np
from scipy import linalg
matrix = np.array([[4,9],[3,5]])
matrix
```

Out[4]:
```
array([[4, 9],
       [3, 5]])
```

In [5]:
```
linalg.det(matrix)
```

Out[5]:
```
-6.999999999999999
```

In [6]:
```
#The Singular-Value Decomposition, or SVD for short, is a matrix decomposition meth
```

In [7]:
```
#applying SVD
import numpy as np
from scipy import linalg
```

In [8]:
```
#creating a matrix
numSvdArr = np.array([[3,5,1],[9,5,7]])
```

In [9]:
```
#Finding the shape of the matrix
numSvdArr.shape       #m=2,n=3
```

Out[9]:
```
(2, 3)
```

In [11]:
```
linalg.svd(numSvdArr)
```

Out[11]:
```
(array([[-0.37879831,  0.92547925],
        [-0.92547925, -0.37879831]]),
 array([13.38464336,  3.29413449]),
 array([[-0.7072066 , -0.4872291 , -0.51231496],
        [-0.19208294,  0.82977932, -0.52399467],
        [-0.68041382,  0.27216553,  0.68041382]]))
```

In [12]:
```
#In out[27] the first array shows the square matrix which is 2*2(given numSvdArr is
#which is a diagonal matrix and has elements equal to the rank of the matrix(as num
#third array is is V^T matrix which is an 3*3 matrix
```

In [ ]:
```
```