The Inverse of a Matrix

A $m \times m$ matrix A (also called a square matrix) has an inverse if A times another matrix B results in the identity matrix I also of shape $m \times m$. This matrix B is called the inverse of A and is denoted as A^{-1}. This relationship is formally written as

$$AA^{-1} = A^{-1}A = I$$

However, not all matrices have an inverse. A matrix with an inverse is called a *nonsingular* or *invertible* matrix, while those without an inverse are known as *singular* or *degenerate*.

Note A square matrix is a matrix that has the same number of rows and columns.

Let's use NumPy to get the inverse of a matrix. Some linear algebra modules are found in a sub-module of NumPy called **linalg**.

```
A = np.array([[15, 29, 24],
              [ 5, 23, 26],
              [30, 14, 44]])
# find the inverse of A
np.linalg.inv(A)
'Output':
array([[ 0.05848375, -0.08483755,  0.01823105],
       [ 0.05054152, -0.00541516, -0.02436823],
       [-0.05595668,  0.05956679,  0.01805054]])
```

NumPy also implements the *Moore-Penrose pseudo inverse*, which gives an inverse derivation for degenerate matrices. Here, we use the **pinv** method to find the inverses of invertible matrices.

```
# using pinv()
np.linalg.pinv(A)
'Output':
array([[ 0.05848375, -0.08483755,  0.01823105],
       [ 0.05054152, -0.00541516, -0.02436823],
       [-0.05595668,  0.05956679,  0.01805054]])
```

Reshaping

A NumPy array can be restructured to take on a different shape. Let's convert a 1-D array to a $m \times n$ matrix.

```
# make 20 elements evenly spaced between 0 and 5
a = np.linspace(0,5,20)
a
'Output':
array([ 0.        ,  0.26315789,  0.52631579,  0.78947368,  1.05263158,
        1.31578947,  1.57894737,  1.84210526,  2.10526316,  2.36842105,
        2.63157895,  2.89473684,  3.15789474,  3.42105263,  3.68421053,
        3.94736842,  4.21052632,  4.47368421,  4.73684211,  5.        ])
# observe that a is a 1-D array
a.shape
'Output': (20,)
# reshape into a 5 x 4 matrix
A = a.reshape(5, 4)
A
'Output':
array([[ 0.        ,  0.26315789,  0.52631579,  0.78947368],
       [ 1.05263158,  1.31578947,  1.57894737,  1.84210526],
       [ 2.10526316,  2.36842105,  2.63157895,  2.89473684],
       [ 3.15789474,  3.42105263,  3.68421053,  3.94736842],
       [ 4.21052632,  4.47368421,  4.73684211,  5.        ]])
# The vector a has been reshaped into a 5 by 4 matrix A
A.shape
'Output': (5, 4)
```

Reshape vs. Resize Method

NumPy has the **np.reshape** and **np.resize** methods. The reshape method returns an ndarray with a modified shape without changing the original array, whereas the resize method changes the original array. Let's see an example.