Elimination Matrices I: The Basics

In [1]:

```
#keep
import numpy as np
```

In [2]:

```
#keep
n = 4
```

Let's create an elimination matrix as M:

In [24]:

```
M = np.eye(n)
M[1,0] = 2
M
```

Out[24]:

```
array([[ 1.,  0.,  0.,  0.],
       [ 2.,  1.,  0.,  0.],
       [ 0.,  0.,  1.,  0.],
       [ 0.,  0.,  0.,  1.]])
```

Here's a matrix A. See if M has the desired effect on A:

In [19]:

```
#keep
np.random.seed(5)
A = np.random.randn(n, n).round(1)
A
```

Out[19]:

```
array([[ 0.4, -0.3,  2.4, -0.3],
       [ 0.1,  1.6, -0.9, -0.6],
       [ 0.2, -0.3, -1.2, -0.2],
       [-0.4,  0.6, -1.7, -0.7]])
```

In [20]:

```
#keep
M.dot(A)
```

Out[20]:

```
array([[ 0.4, -0.3,  2.4, -0.3],
       [ 0.9,  1. ,  3.9, -1.2],
       [ 0.2, -0.3, -1.2, -0.2],
       [-0.4,  0.6, -1.7, -0.7]])
```

Next, see if you can build the inverse of M:

In [25]:

```
Minv = np.eye(n)
Minv[1,0] = -2
Minv
```

Out[25]:

```
array([[ 1.,  0.,  0.,  0.],
       [-2.,  1.,  0.,  0.],
       [ 0.,  0.,  1.,  0.],
       [ 0.,  0.,  0.,  1.]])
```

In [26]:

```
#keep
M.dot(Minv)
```

Out[26]:

```
array([[ 1.,  0.,  0.,  0.],
       [ 0.,  1.,  0.,  0.],
       [ 0.,  0.,  1.,  0.],
       [ 0.,  0.,  0.,  1.]])
```