Rank-1 Approximation

In [1]:

```
#keep
import numpy as np
import numpy.linalg as la
import matplotlib.pyplot as pt
%matplotlib inline
```

In [2]:

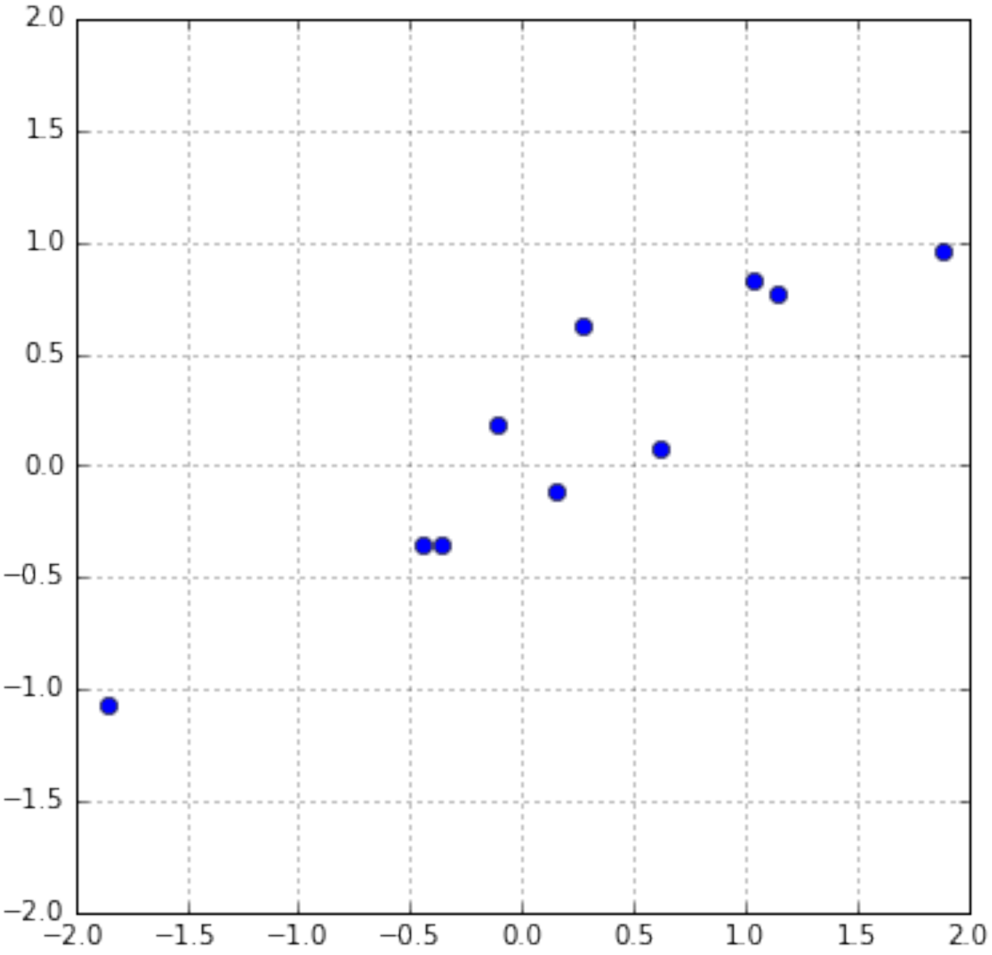
```
#keep
np.random.seed(17)
n = 10
X = np.random.randn(2, n)

X[1] = 0.7*X[0] + 0.2 * X[1]

# uncomment this for a different data set
#X[0] = 0.1 * X[0]
```

In [3]:

```
#keep
pt.figure(figsize=(6,6))
pt.gca().set_aspect("equal")
pt.xlim([-2, 2])
pt.ylim([-2, 2])
pt.grid()
pt.plot(X[0], X[1], "o")
```

```
Out[3]:
```

```
[<matplotlib.lines.Line2D at 0x10d350ac8>]
```



Now compute the SVD. Use `numpy.linalg.svd(..., full_matrices=False)`.

```
In [4]:
```

```
U, sigma, VT = la.svd(X, full_matrices=False)
```

Now find the vectors `u` and `v`:

```
In [5]:
```

```
u = U[:, 0]
v = VT[0]
```

Now find `X1`:

```
In [6]:
```

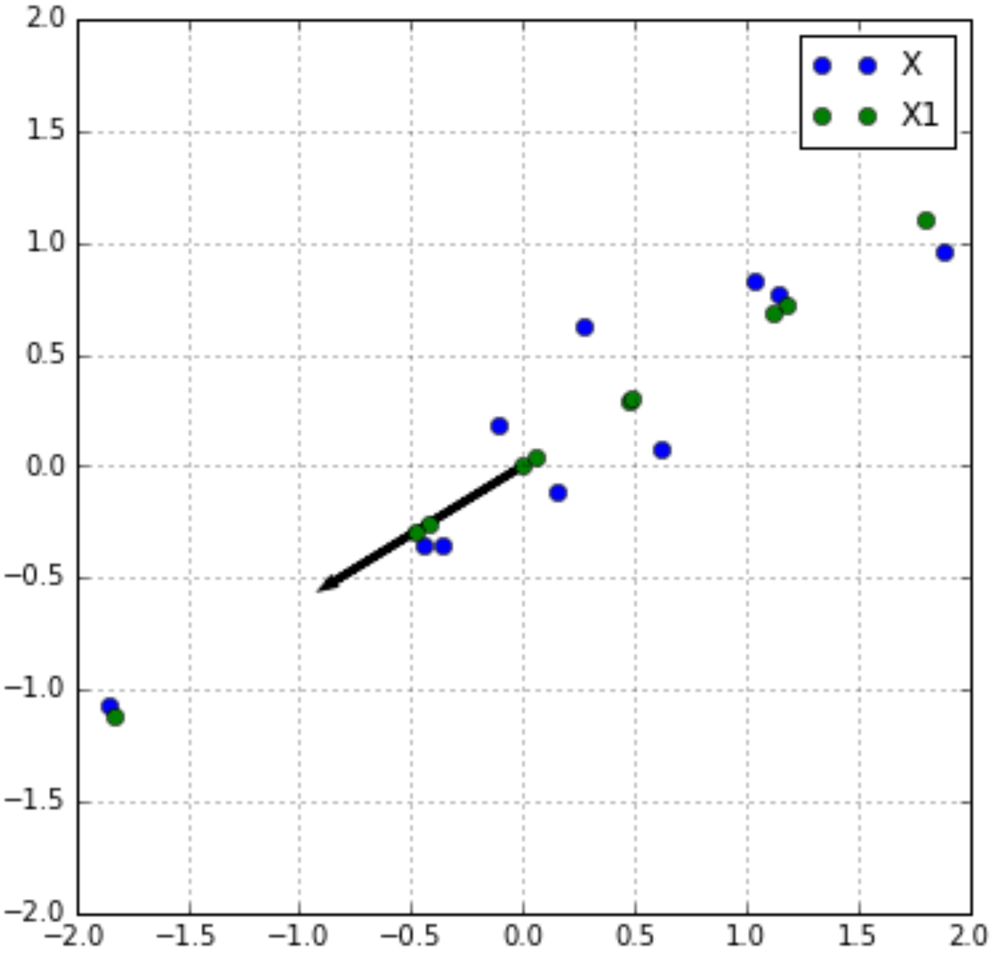
```
X1 = np.outer(u, v) * sigma[0]
```

In [7]:

```
#keep

pt.figure(figsize=(6,6))

pt.arrow(0, 0, u[0], u[1], lw=3)

pt.gca().set_aspect("equal")
pt.xlim([-2, 2])
pt.ylim([-2, 2])
pt.grid()

pt.plot(X[0], X[1], "ob", label="X")
pt.plot(X1[0], X1[1], "og", label="X1")
pt.legend(loc="best")
```

Out[7]:

```
<matplotlib.legend.Legend at 0x10d501c18>
```



X
X1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0