analyseFXXP.EX

Paul Balzer

February 5, 2014

Part I

Python Pandas for Financial Stuff

In [16]:
```
import datetime

import pandas as pd
import pandas.io.data
from pandas import Series, DataFrame
pd.__version__

import matplotlib.pyplot as plt
import numpy as np

%pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

1 Stoxx-Europe-600-Index

Thanks to this: http://nbviewer.ipython.org/github/twiecki/financial-analysis-python-tutorial/blob/master/1.%20Pandas%20Basics.ipynb

In [17]:
```
sc='FXXP.EX'
stoxx = pd.io.data.get_data_yahoo(sc,
                                  start=datetime.datetime(2013, 1, 1))
stoxx.head(10)
```

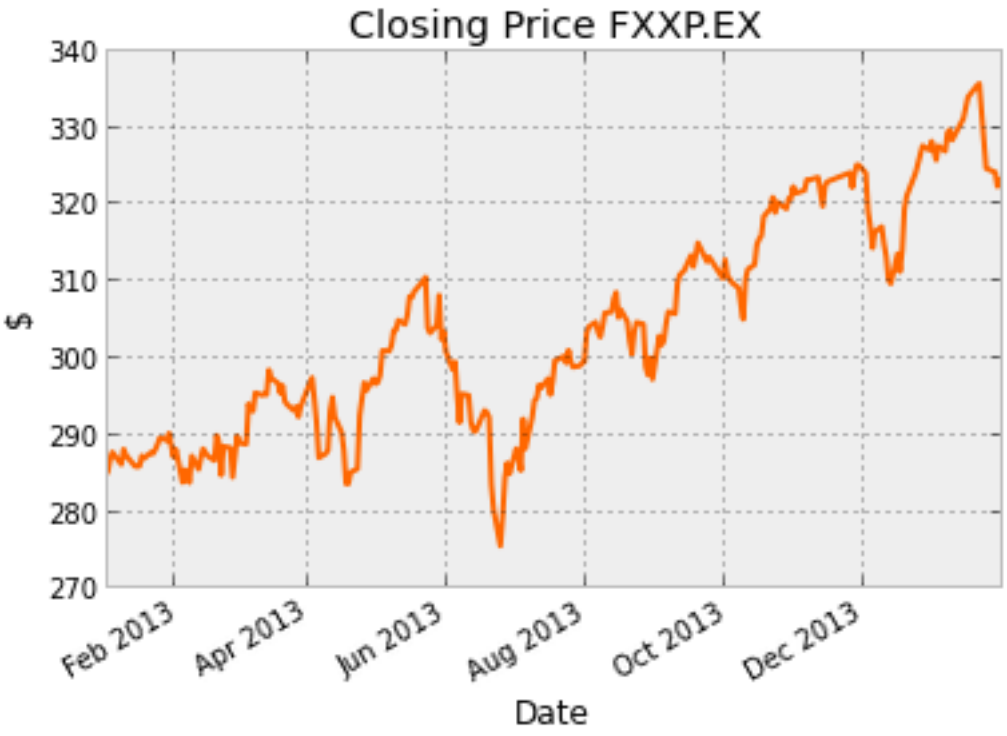
Out [17]:

Date	Open	High	Low	Close	Volume	Adj Close
2013-01-02	285.33	285.33	285.33	285.33	0	285.33
2013-01-03	286.83	286.83	286.83	286.83	0	286.83
2013-01-04	287.83	287.83	287.83	287.83	0	287.83
2013-01-07	286.63	286.63	286.63	286.63	0	286.63
2013-01-08	286.25	286.25	286.25	286.25	0	286.25
2013-01-09	288.22	288.22	288.22	288.22	0	288.22
2013-01-10	287.44	287.44	287.44	287.44	0	287.44
2013-01-11	287.08	287.08	287.08	287.08	0	287.08

```
2013-01-14  286.01  286.01  286.01  286.01       0     286.01
2013-01-15  285.97  285.97  285.97  285.97       0     285.97
```

2 Schlusspreis

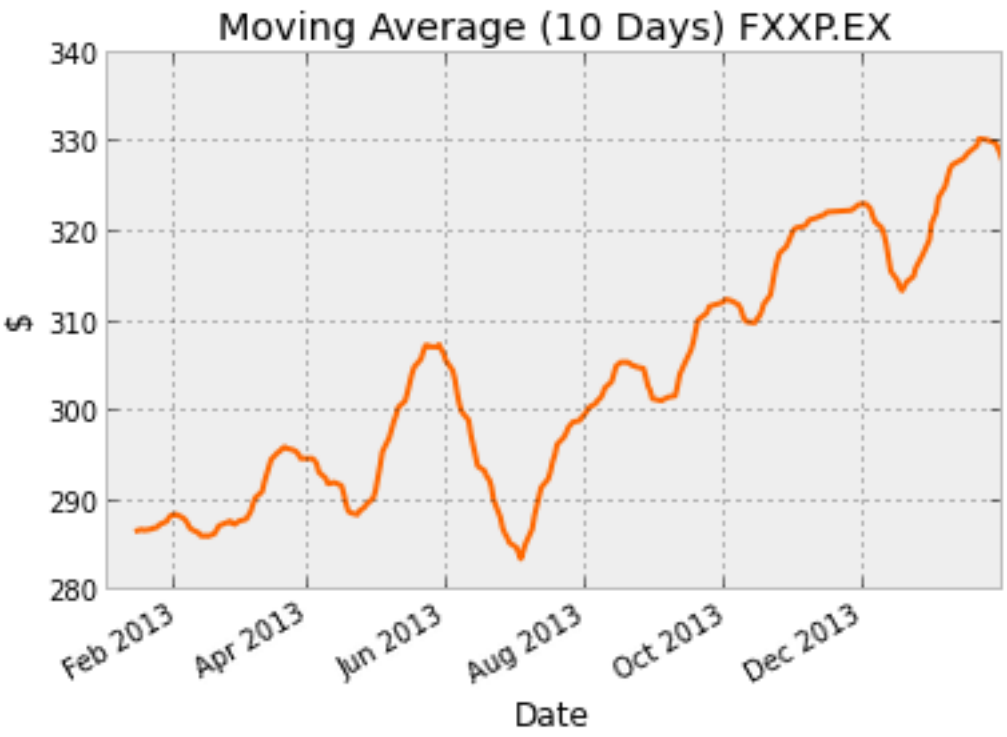
In [18]:
```
stoxx['Close'].plot();
plt.ylabel('\$')
plt.title('Closing Price %s' % sc);
```



3 Financial Stuff

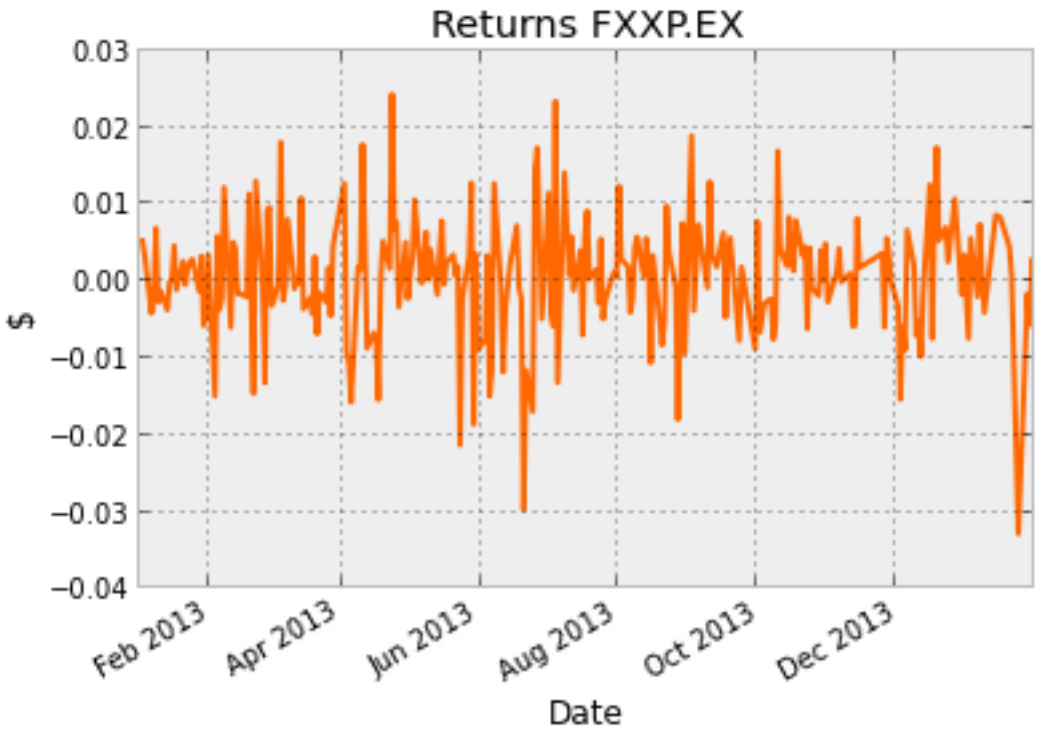
3.1 Moving Average

In [19]:
```
close_px = stoxx['Adj Close']
mad = 10
mavg = pd.rolling_mean(close_px, mad)
mavg.plot();
plt.ylabel('\$')
plt.title('Moving Average (%i Days) %s' % (mad, sc));
```



3.2 Returns

In [20]:
```
stoxx['rets'] = close_px.pct_change()
stoxx.rets.plot();
plt.title('Returns %s' % sc);
plt.ylabel('\$');
```

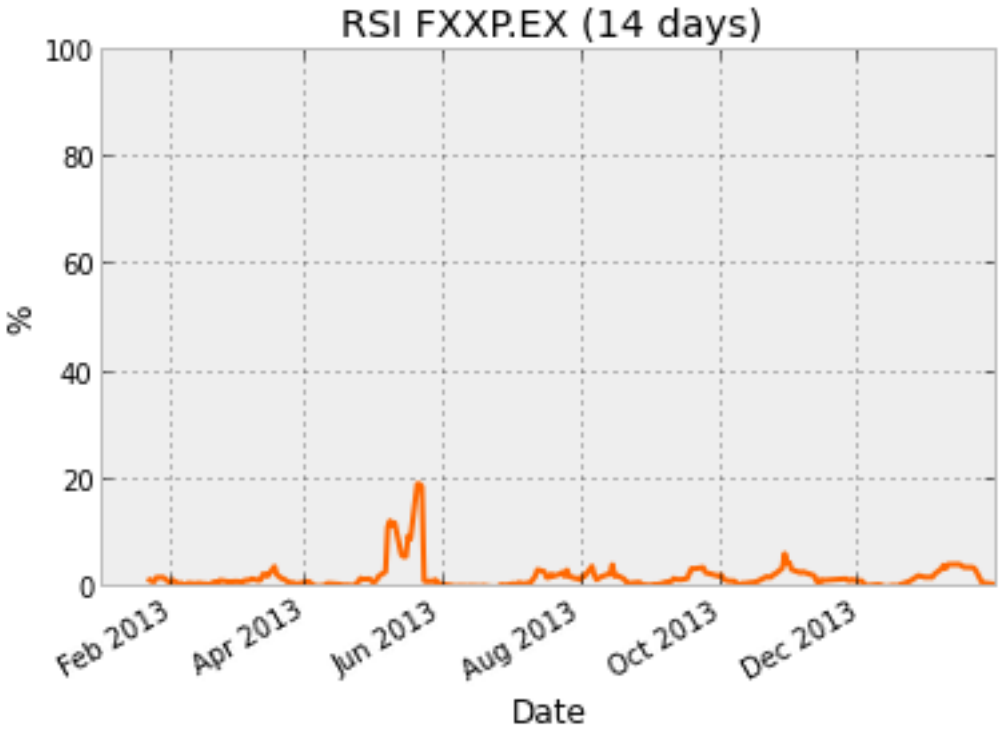


3.3 Relative Strength Index

Source: http://stackoverflow.com/a/20527056

In [21]:
```
delta = stoxx['Close'].diff()
dUp, dDown = delta.copy( ), delta.copy( )
dUp[ dUp < 0 ] = 0
dDown[ dDown > 0 ] = 0


n=14
RolUp = pd.rolling_mean( dUp, n )
RolDown = pd.rolling_mean( dDown, n).abs()

RS = RolUp / RolDown

RS.plot();
plt.title('RSI %s (%i days)' % (sc, n));
plt.ylim([0,100]);
```

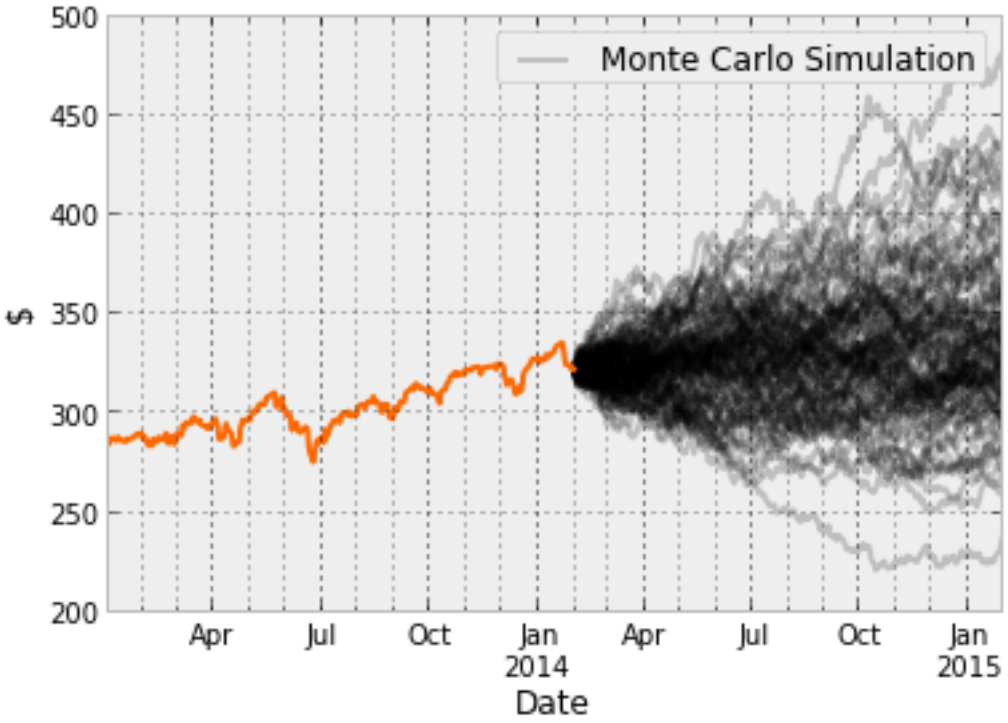
```
plt.ylabel('%');
```



3.4 Monte Carlo Simulation

In [22]:
```
SO=stoxx['Close'][-1] # letzter Preis
vol=np.std(stoxx['rets'])*np.sqrt(252) # Historical Volatility
r=0.025 # Constant Short Rate

K = SO*1.1 # 10% OTM Call Option
T = 1.0 # Maturity 1 Year

M=364
dt=T/M # Time Steps
I = 100 # Simulation Paths
```

In [23]:
```
S=np.zeros((M+1,I))
S[0,:]=SO
for t in range(1, M+1):
    ran = np.random.standard_normal(I)
    S[t,:]=S[t-1,:] * np.exp((r-vol**2/2)*dt + vol*np.sqrt(dt)*ran)


MC=pd.DataFrame(data=S, index=pd.date_range(start=stoxx.index[-1], periods

ax=MC.plot(alpha=0.2, color='k');
stoxx['Close'].plot(ax=ax);
plt.legend(['Monte Carlo Simulation']);
plt.ylabel('\$');
```



3.5 Option Valuation

In [24]:
```
VO=np.exp(-r*T)*np.sum(np.max(S[-1]-K,0))/I
print('Call Value %8.3f' % VO)
```

```
Call Value    1.254
```

Part II

Vergleich

In [25]:
```
df = pd.io.data.get_data_yahoo(['AAPL', 'FXXP.EX', 'GOOG', 'FDAX.EX'],
                               start=datetime.datetime(2013, 1, 1))['Adj C
df.head()
```

Out [25]:

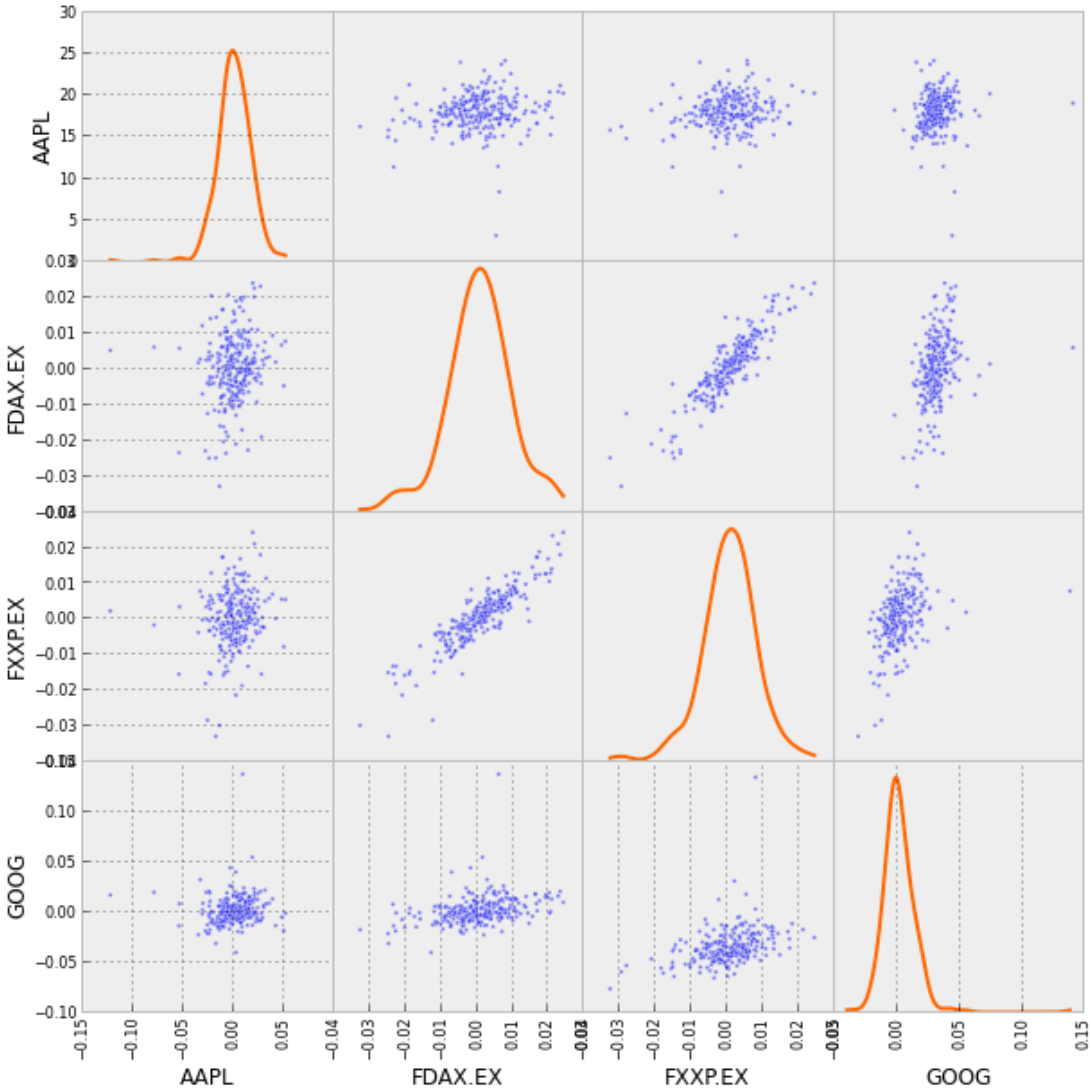
Date	AAPL	FDAX.EX	FXXP.EX	GOOG
2013-01-02	535.58	7778.78	285.33	723.25
2013-01-03	528.82	7756.44	286.83	723.67
2013-01-04	514.09	7776.37	287.83	737.97
2013-01-07	511.06	7732.66	286.63	734.75
2013-01-08	512.44	7695.83	286.25	733.30

3.6 Returns

In [26]:
```
rets = df.pct_change()
```

In [27]:
```
fig=plt.figure(figsize=(12,12));
pd.scatter_matrix(rets, diagonal='kde', figsize=(10, 10));
```

```
<matplotlib.figure.Figure at 0x109689ad0>
```



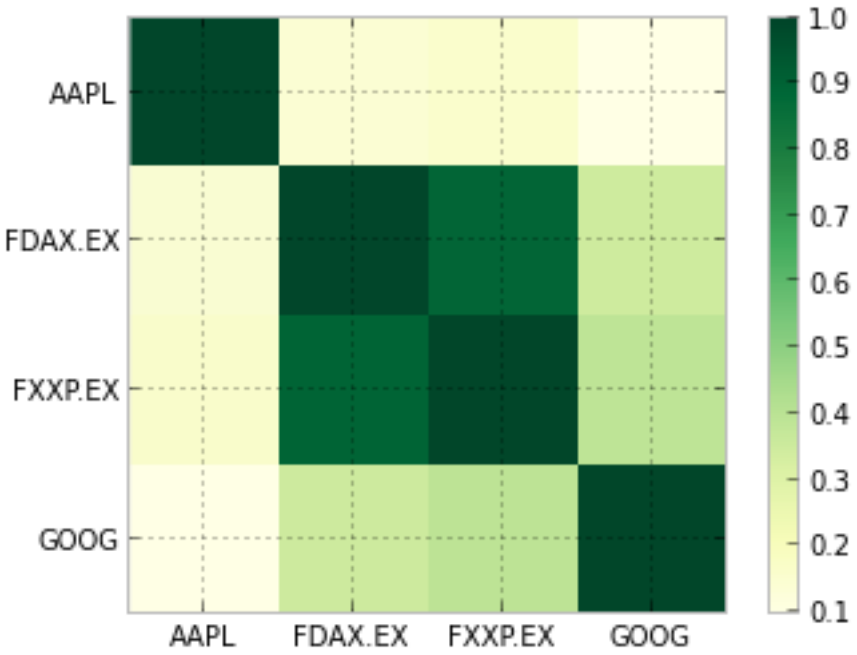
3.7 Korrelation der Returns

In [28]:
```
corr = rets.corr()
corr
```

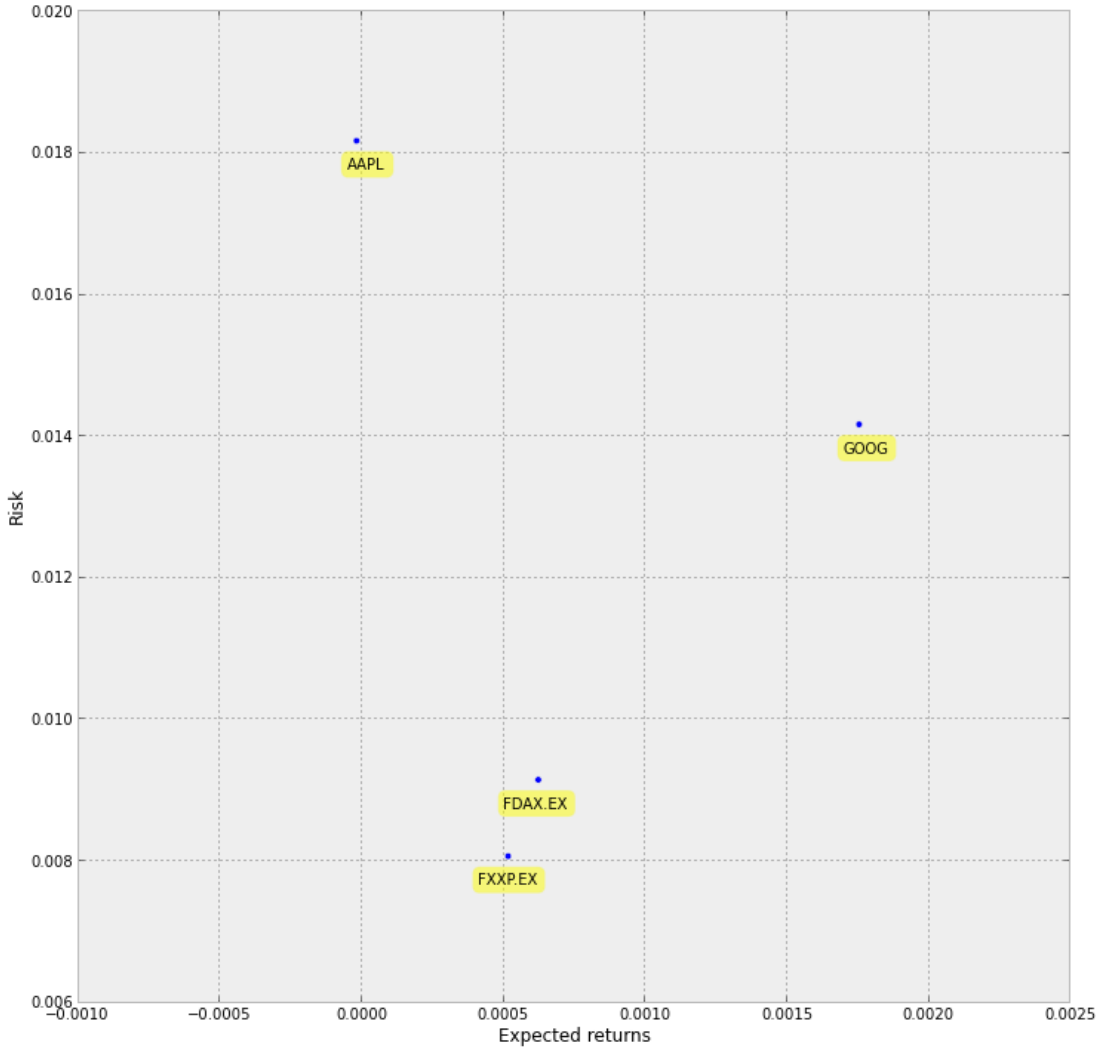
Out [28]:

	AAPL	FDAX.EX	FXXP.EX	GOOG
AAPL	1.000000	0.143314	0.160806	0.096369
FDAX.EX	0.143314	1.000000	0.896059	0.350308
FXXP.EX	0.160806	0.896059	1.000000	0.391265
GOOG	0.096369	0.350308	0.391265	1.000000

In [29]:
```
plt.imshow(corr, cmap='YlGn', interpolation='none')
plt.colorbar()
plt.xticks(range(len(corr)), corr.columns)
plt.yticks(range(len(corr)), corr.columns);
```



In [30]:
```
fig=plt.figure(figsize=(12,12))
plt.scatter(rets.mean(), rets.std())
plt.xlabel('Expected returns')
plt.ylabel('Risk')
for label, x, y in zip(rets.columns, rets.mean(), rets.std()):
    plt.annotate(
        label,
        xy = (x, y), xytext = (20, -20),
        textcoords = 'offset points', ha = 'right', va = 'bottom',
        bbox = dict(boxstyle = 'round,pad=0.5', fc = 'yellow', alpha = 0.5
        arrowprops = dict(arrowstyle = '->', connectionstyle = 'arc3,rad=(
```


0.020
0.018
0.016
0.014
0.012
0.010
0.008
0.006
Risk
AAPL
GOOG
FDAX.EX
FXXP.EX
−0.0010
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
Expected returns