In [ ]:
```
from scipy import stats
import pandas as pd
import pandas_datareader as web
import datetime as dt
import matplotlib.pyplot as plt
%matplotlib inline
```

Choosing the data

Date

In [ ]:
```
start = dt.datetime(2013,1,1) #YYYY,MM,DD
end = dt.datetime(2016,1,1)
```

Stocks

Requirements: Market data (S&P 500) and stock data (Facebook)

In [ ]:
```
df_spy = web.DataReader('SPY', 'yahoo', start, end)  #tick, source, start date, end date
df_fb = web.DataReader('FB','yahoo',start,end)
```

In [ ]:
```
df_spy.head()
```

Out[ ]:

Date	High	Low	Open	Close	Volume	Adj Close
2012-12-31	142.559998	139.539993	139.660004	142.410004	243935200.0	119.823318
2013-01-02	146.149994	144.729996	145.110001	146.059998	192059000.0	122.894394
2013-01-03	146.369995	145.339996	145.990005	145.729996	144761800.0	122.616669
2013-01-04	146.610001	145.669998	145.970001	146.369995	116817700.0	123.155205
2013-01-07	146.110001	145.429993	145.850006	145.970001	110002500.0	122.818619

In [ ]:
```
df_fb.head()
```

Out[ ]:

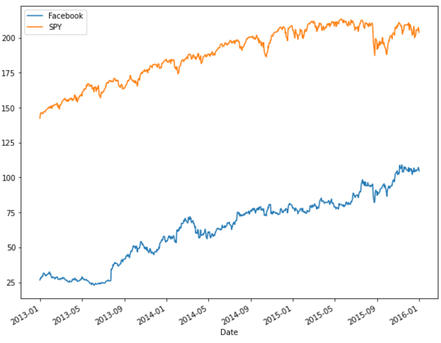
Date	High	Low	Open	Close	Volume	Adj Close
2012-12-31	26.990000	26.110001	26.200001	26.620001	60374500	26.620001
2013-01-02	28.180000	27.420000	27.440001	28.000000	69846400	28.000000
2013-01-03	28.469999	27.590000	27.879999	27.770000	63140600	27.770000
2013-01-04	28.930000	27.830000	28.010000	28.760000	72715400	28.760000
2013-01-07	29.790001	28.650000	28.690001	29.420000	83781800	29.420000

Create a graph to visually show the relationship between the market data and the stocks

Requirement: The data between the market and the stocks chosen must have some relationship

In [ ]:
```
df_fb['Close'].plot(label = 'Facebook', figsize=(10,8))
df_spy['Close'].plot(label = 'SPY')
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x1298b5840>
```



Creating a new columns of Cumulative returns. This is done by taking the ratio between the current entry to the rest of the entries.

$CS = \frac{pricealltimet}{currentprice}$

In [ ]:
```
df_fb['Cumu'] = df_fb['Close']/df_fb['Close'].iloc[0]
df_spy['Cumu'] = df_spy['Close']/df_spy['Close'].iloc[0]
```

In [ ]:
```

```

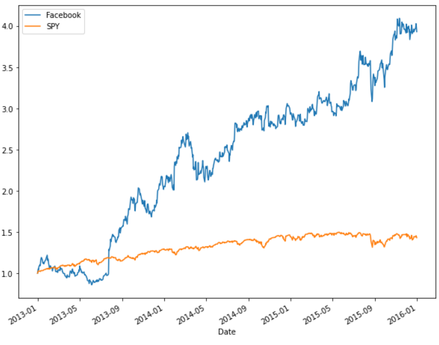
Out[ ]:

Date	High	Low	Open	Close	Volume	Adj Close	Cumu
2012-12-31	26.990000	26.110001	26.200001	26.620001	60374500	26.620001	1.000000
2013-01-02	28.180000	27.420000	27.440001	28.000000	69846400	28.000000	1.051841
2013-01-03	28.469999	27.590000	27.879999	27.770000	63140600	27.770000	1.043201
2013-01-04	28.930000	27.830000	28.010000	28.760000	72715400	28.760000	1.080391
2013-01-07	29.790001	28.650000	28.690001	29.420000	83781800	29.420000	1.105184
...	...	...	...	...	...	...	...
2015-12-24	105.320000	104.500000	104.739998	105.019997	6501800	105.019997	3.945154
2015-12-28	105.980003	104.529999	105.019997	105.930000	13069700	105.930000	3.979339
2015-12-29	107.739998	106.250000	106.419998	107.260002	17179900	107.260002	4.029301
2015-12-30	107.250000	106.059998	107.000000	106.220001	13115000	106.220001	3.990233
2015-12-31	106.169998	104.620003	106.000000	104.660004	18391100	104.660004	3.931630

757 rows × 7 columns

In [ ]:
```
df_fb['Cumu'].plot(label = 'Facebook', figsize=(10,8))
df_spy['Cumu'].plot(label = 'SPY')
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x1299c3eb0>
```



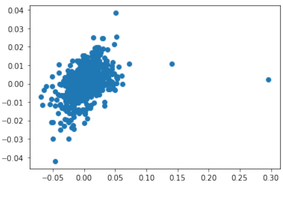
Obtaining the Daily Returns of both assets

$\ln \frac{S_i}{S_{i-t}}$ is the formula

In [ ]:
```
df_fb['daily_ret'] = df_fb['Close'].pct_change(1) #pct.change(1) DataFrame.pct_change(periods=1, fill_method='pad', limit=None, freq=None)
df_spy['daily_ret'] = df_spy['Close'].pct_change(1)
```

In [ ]:
```
plt.scatter(df_fb['daily_ret'],df_spy['daily_ret'])
```

Out[ ]:
```
<matplotlib.collections.PathCollection at 0x1299c9540>
```



Applying linear regression to obtain the parameters α and β

In [ ]:
```
LR = stats.linregress(df_fb['daily_ret'].iloc[1:],df_spy['daily_ret'].iloc[1:]) #scipy.stats.linregress(x, y=None, alternative='two-sided')
beta,alpha,r_val,p_val,std_err = LR #slope, intercept, pearson R to get R^2 just square r-value, p_valu, standard error
```

In [ ]:
```

```