In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

df = pd.read_csv('data_stocks.csv')

df.head()
```

Out[1]:

	DATE	SP500	NASDAQ.AAL	NASDAQ.AAPL	NASDAQ.ADBE	NASDAQ.ADI	NASDAQ.ADP	NASDAQ
0	1491226200	2363.6101	42.3300	143.6800	129.6300	82.040	102.2300	8
1	1491226260	2364.1001	42.3600	143.7000	130.3200	82.080	102.1400	8
2	1491226320	2362.6799	42.3100	143.6901	130.2250	82.030	102.2125	8
3	1491226380	2364.3101	42.3700	143.6400	130.0729	82.000	102.1400	8
4	1491226440	2364.8501	42.5378	143.6600	129.8800	82.035	102.0600	8

5 rows × 502 columns

In [2]:

```
df.shape
```

Out[2]:

```
(41266, 502)
```

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41266 entries, 0 to 41265
Columns: 502 entries, DATE to NYSE.ZTS
dtypes: float64(501), int64(1)
memory usage: 158.0 MB
```

In [4]:

```
df.describe()
```

Out[4]:

	DATE	SP500	NASDAQ.AAL	NASDAQ.AAPL	NASDAQ.ADBE	NASDAQ.ADI	NASDAQ.ADP
count	4.126600e+04	41266.000000	41266.000000	41266.000000	41266.00000	41266.000000	41266.000000
mean	1.497749e+09	2421.537882	47.708346	150.453566	141.31793	79.446873	103.480398
std	3.822211e+06	39.557135	3.259377	6.236826	6.91674	2.000283	4.424244
min	1.491226e+09	2329.139900	40.830000	140.160000	128.24000	74.800000	95.870000
25%	1.494432e+09	2390.860100	44.945400	144.640000	135.19500	78.030000	101.300000
50%	1.497638e+09	2430.149900	48.360000	149.945000	142.26000	79.410000	102.440000
75%	1.501090e+09	2448.820100	50.180000	155.065000	147.10000	80.580000	104.660000
max	1.504210e+09	2490.649900	54.475000	164.510000	155.33000	90.440000	121.770000

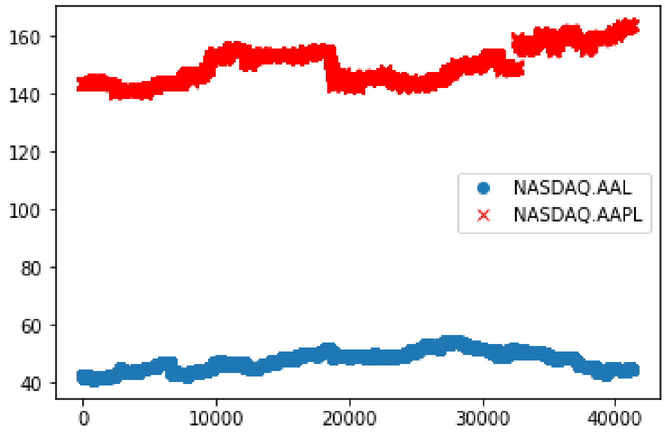
8 rows × 502 columns

In [5]:
```
df.isnull().sum()
```

```
NASDAQ.AMD      0
NASDAQ.AMGN     0
NASDAQ.AMZN     0
NASDAQ.ATVI     0
NASDAQ.AVGO     0
NASDAQ.BBBY     0
NASDAQ.BIIB     0
NASDAQ.CA       0
NASDAQ.CBOE     0
NASDAQ.CELG     0
NASDAQ.CERN     0
NASDAQ.CHRW     0
NASDAQ.CHTR     0
NASDAQ.CINF     0
NASDAQ.CMCSA    0
NASDAQ.CME      0
NASDAQ.COST    0
NASDAQ.CSCO     0
NASDAQ.CSX      0
               ..
```

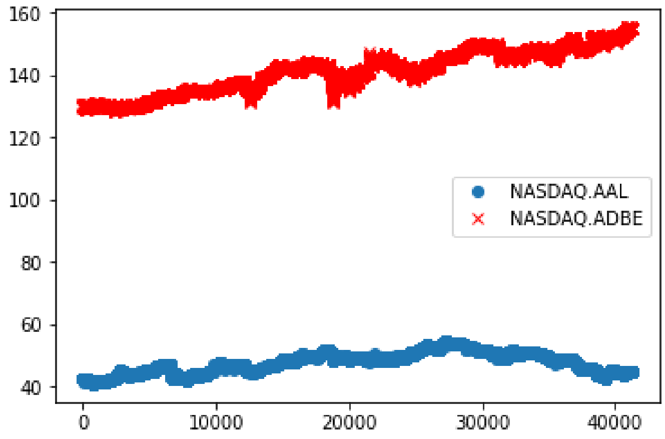
In [6]:
```
df[['NASDAQ.AAL','NASDAQ.AAPL']].plot(style=['o','rx'])
```

Out[6]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x239f2b38c88>
```



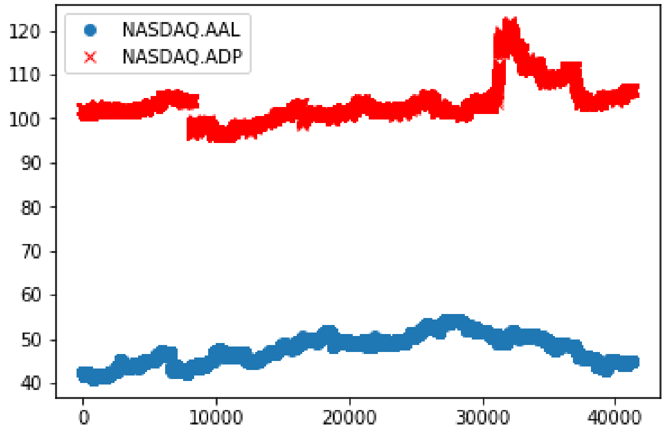
In [7]:
```
df[['NASDAQ.AAL','NASDAQ.ADBE']].plot(style=['o','rx'])
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x239f2b38e80>
```



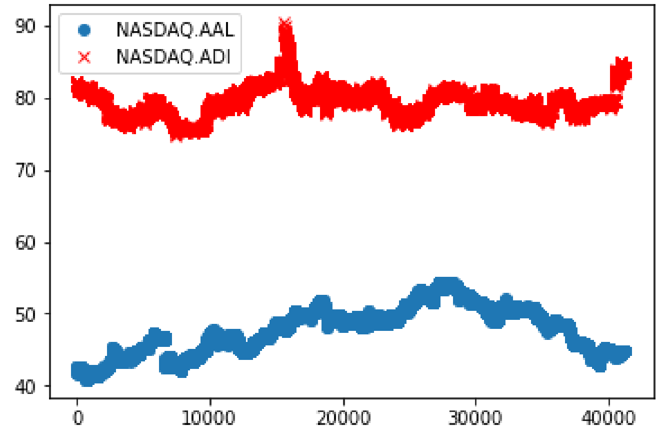
```
In [8]: df[['NASDAQ.AAL','NASDAQ.ADP']].plot(style=['o','rx'])
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x239f2c50da0>
```



```
In [9]: df[['NASDAQ.AAL','NASDAQ.ADI']].plot(style=['o','rx'])
```

```
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x239f2c36668>
```



```
In [10]: df.drop(['DATE','SP500'], axis=1, inplace=True)
```

```
In [11]: df.head()
```

Out[11]:

	NASDAQ.AAL	NASDAQ.AAPL	NASDAQ.ADBE	NASDAQ.ADI	NASDAQ.ADP	NASDAQ.ADSK	NASDAQ.AKAM
0	42.3300	143.6800	129.6300	82.040	102.2300	85.2200	59.760
1	42.3600	143.7000	130.3200	82.080	102.1400	85.6500	59.840
2	42.3100	143.6901	130.2250	82.030	102.2125	85.5100	59.795
3	42.3700	143.6400	130.0729	82.000	102.1400	85.4872	59.620
4	42.5378	143.6600	129.8800	82.035	102.0600	85.7001	59.620

5 rows × 500 columns

Original dataframe is transposed to a new dataframe in order to categorize 500 different stocks into 3 different clusters on the basis of similar data

patterns.

In [12]:

```
data_stocks_transposed = df.T
data_stocks_transposed.head()
```

Out[12]:

	0	1	2	3	4	5	6	7	8	9	...	
NASDAQ.AAL	42.33	42.36	42.3100	42.3700	42.5378	42.5399	42.470	42.47	42.390	42.330	...	
NASDAQ.AAPL	143.68	143.70	143.6901	143.6400	143.6600	143.7800	143.864	143.81	143.815	143.800	...	1
NASDAQ.ADBE	129.63	130.32	130.2250	130.0729	129.8800	130.0700	130.180	130.14	130.100	130.210	...	1
NASDAQ.ADI	82.04	82.08	82.0300	82.0000	82.0350	82.0400	82.120	82.19	82.230	82.165	...	
NASDAQ.ADP	102.23	102.14	102.2125	102.1400	102.0600	102.0400	102.330	102.37	102.380	102.330	...	1

5 rows × 41266 columns

In [13]:

```
from sklearn.cluster import KMeans
num_clusters = 3 # 3 clusters
model = KMeans(n_clusters = num_clusters)
model.fit(data_stocks_transposed.values)
y_kmeans = model.predict(data_stocks_transposed.values)
```

In [14]:

```
data_stocks_transposed['Cluster'] = y_kmeans
```

In [15]:

```
data_stocks_transposed['Cluster'].value_counts()
```

Out[15]:

```
0    417
1     78
2      5
Name: Cluster, dtype: int64
```

The below three cells indicates which all stocks are moving together and which are different from each other.¶

In [16]:

```
data_stocks_transposed.loc[data_stocks_transposed['Cluster']==0].head()
```

Out[16]:

	0	1	2	3	4	5	6	7	8	9	...	
NASDAQ.AAL	42.33	42.36	42.3100	42.3700	42.5378	42.5399	42.470	42.47	42.39	42.330	...	4
NASDAQ.ADI	82.04	82.08	82.0300	82.0000	82.0350	82.0400	82.120	82.19	82.23	82.165	...	8
NASDAQ.ADP	102.23	102.14	102.2125	102.1400	102.0600	102.0400	102.330	102.37	102.38	102.330	...	10
NASDAQ.ADSK	85.22	85.65	85.5100	85.4872	85.7001	85.9200	85.912	85.82	85.88	85.860	...	1
NASDAQ.AKAM	59.76	59.84	59.7950	59.6200	59.6200	59.6100	59.540	59.41	59.43	59.260	...	4

5 rows × 41267 columns

In [17]:

```
data_stocks_transposed.loc[data_stocks_transposed['Cluster']==1].head()
```

Out[17]:

	0	1	2	3	4	5	6	7	8	9	...
NASDAQ.AAPL	143.68	143.70	143.6901	143.6400	143.66	143.7800	143.864	143.8100	143.815	143.8000	...
NASDAQ.ADBE	129.63	130.32	130.2250	130.0729	129.88	130.0700	130.180	130.1400	130.100	130.2100	...
NASDAQ.AMGN	164.63	164.68	164.9050	164.7600	164.85	164.9225	164.820	164.7269	164.705	164.7900	...
NASDAQ.AVGO	219.11	219.98	219.3900	219.3000	219.18	219.4900	219.720	219.8700	219.820	219.2701	...
NASDAQ.BIIB	274.08	273.99	274.2750	273.5900	273.54	274.1300	273.870	274.0500	273.870	274.3800	...

5 rows × 41267 columns

In [18]:

```
data_stocks_transposed.loc[data_stocks_transposed['Cluster']==2].head()
```

Out[18]:

	0	1	2	3	4	5	6	7	8
NASDAQ.AMZN	888.55	887.1173	887.5110	886.2700	886.5800	886.8000	887.310	888.5997	889.80
NASDAQ.GOOG	829.41	833.1000	835.9184	833.6200	835.3200	835.1400	835.300	835.8000	836.32
NASDAQ.GOOGL	847.83	852.1700	854.2800	852.7800	854.7500	853.9700	854.250	854.8250	855.28
NASDAQ.ISRG	766.47	766.5400	767.0300	766.5853	766.5853	766.5853	766.610	767.2650	767.98
NASDAQ.PCLN	1776.26	1779.4606	1777.5500	1776.4301	1774.0800	1775.0000	1776.585	1777.0100	1776.03

5 rows × 41267 columns