`pandas.plotting` subpackage

Pandas provides some extra plotting functions for a few select plot types.

About the Data

In this notebook, we will be working with Facebook's stock price throughout 2018 (obtained using the `stock_analysis` package).

Setup

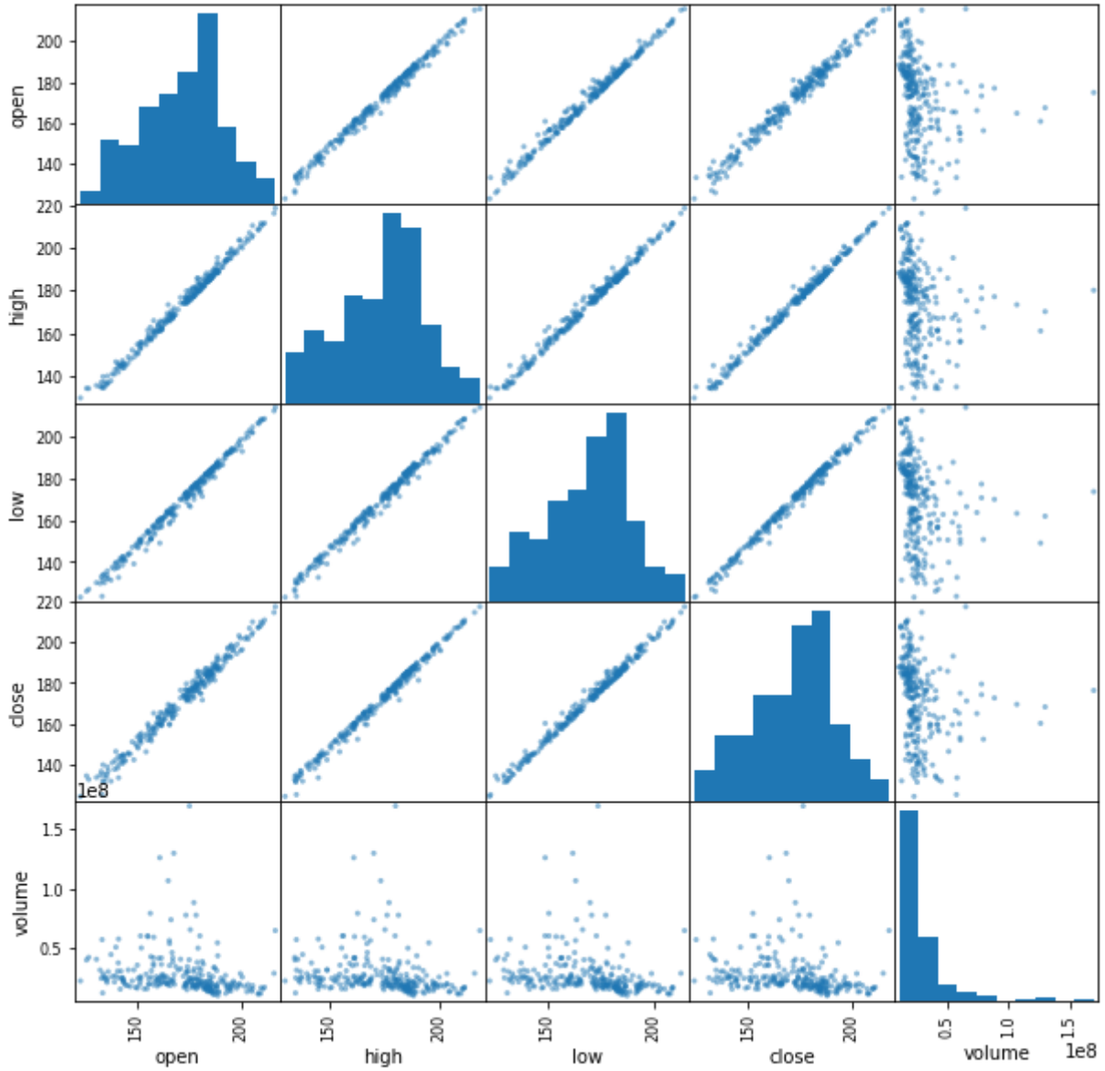
In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

fb = pd.read_csv(
    'data/fb_stock_prices_2018.csv', index_col='date', parse_dates=True
)
```

Scatter matrix

In [2]:
```
from pandas.plotting import scatter_matrix
scatter_matrix(fb, figsize=(10, 10))
```

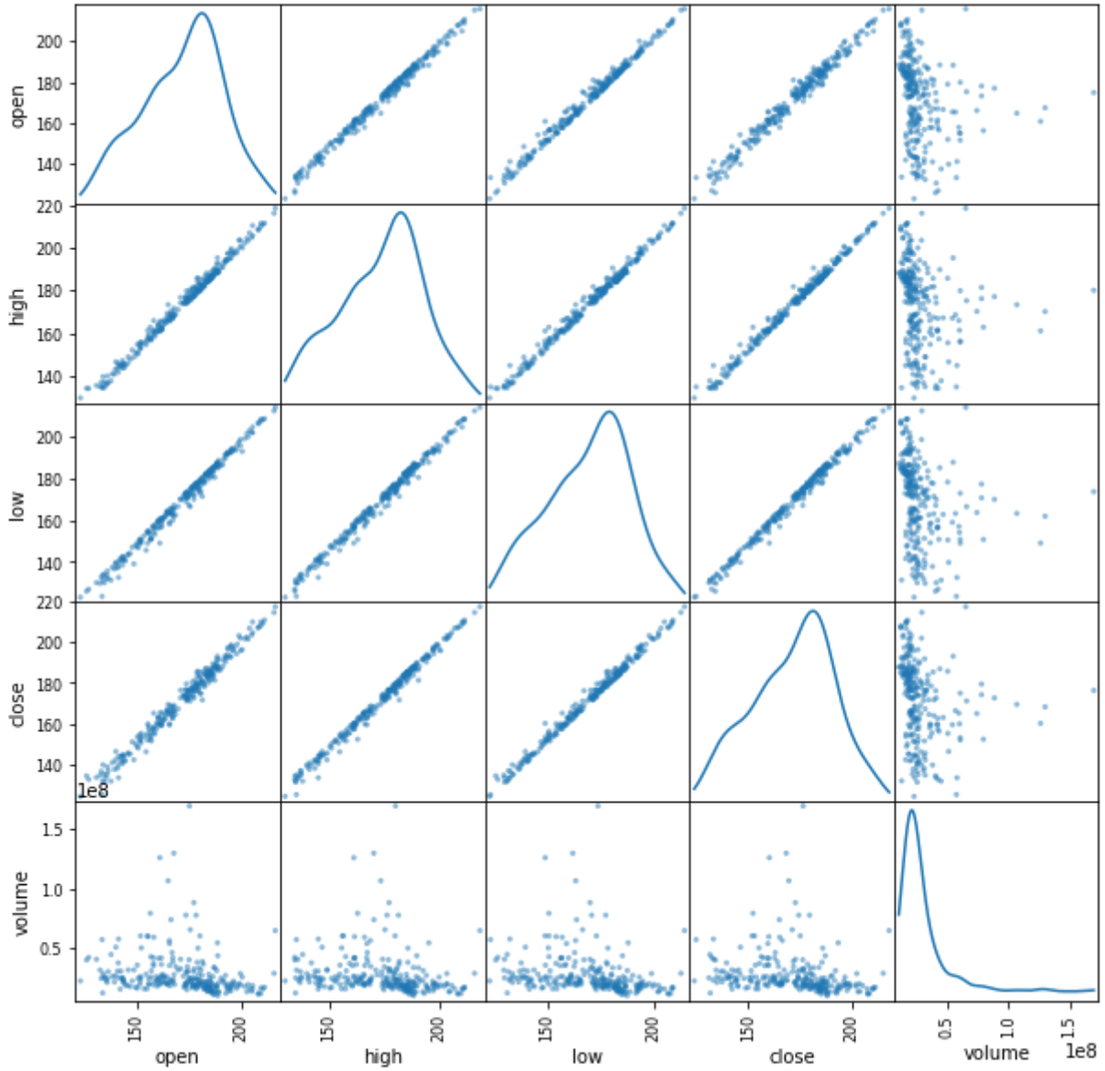
Out[2]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x1132E210>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1135DA90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1237CF30>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12396FD0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x123B1AF0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x123CBB90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x123E6C50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12400D10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x124092B0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12423430>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x12451F50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1246BFD0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1248E670>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x124A9730>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x124C57F0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x124E08B0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x124FB970>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12516A30>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12530AF0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1254CBB0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x12565C70>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1257ED30>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1259ADF0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x125B4EB0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x125D1F70>]],
      dtype=object)
```



Changing the diagonal from histograms to KDE:

In [3]:
```
scatter_matrix(fb, figsize=(10, 10), diagonal='kde')
```

Out[3]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x12796090>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x129D3E10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12867E10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12882E90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1289AF50>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x128B7FD0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x128DB6F0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x128F51B0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x128F5730>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x129118B0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x12948970>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12964A30>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1297DAF0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12999BB0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x14103C70>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x1411ED30>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x14139DF0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x14154EB0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1416EF70>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1418AAB0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x141A6B70>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x141C2C30>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x141DCCF0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x141F7DB0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x14213E70>]],
      dtype=object)
```

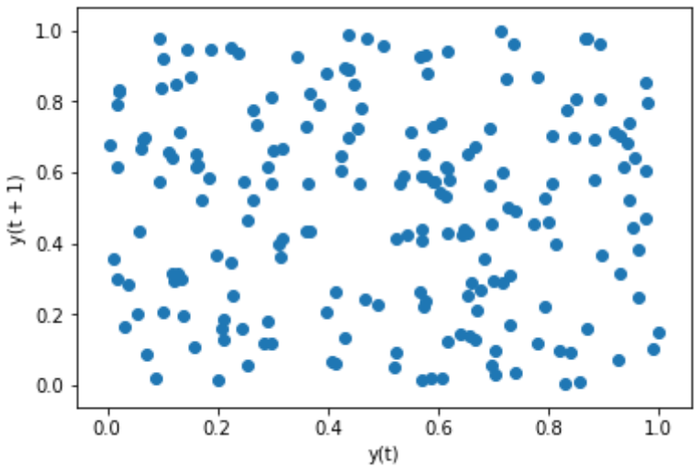


Lag plot

Lag plots let us see how the variable correlations with past observations of itself. Random data has no pattern:

In [4]:
```
from pandas.plotting import lag_plot
np.random.seed(0) # make this repeatable
lag_plot(pd.Series(np.random.random(size=200)))
```

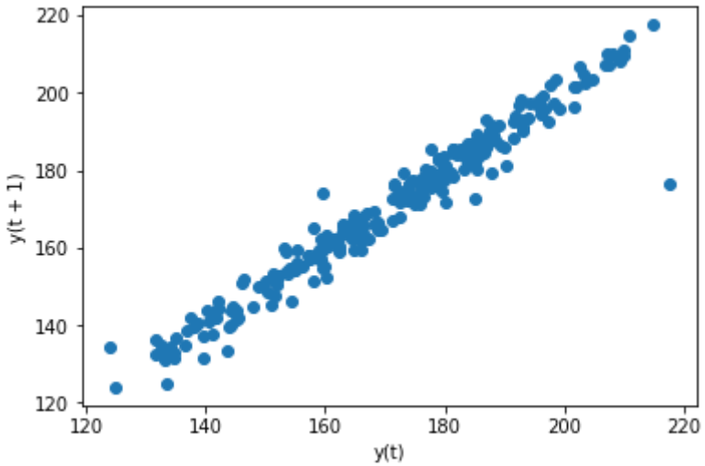
Out[4]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x15410770>
```



Data with some level of correlation to itself (autocorrelation) may have patterns. Stock prices are highly auto-correlated:

In [5]:
```
lag_plot(fb.close)
```

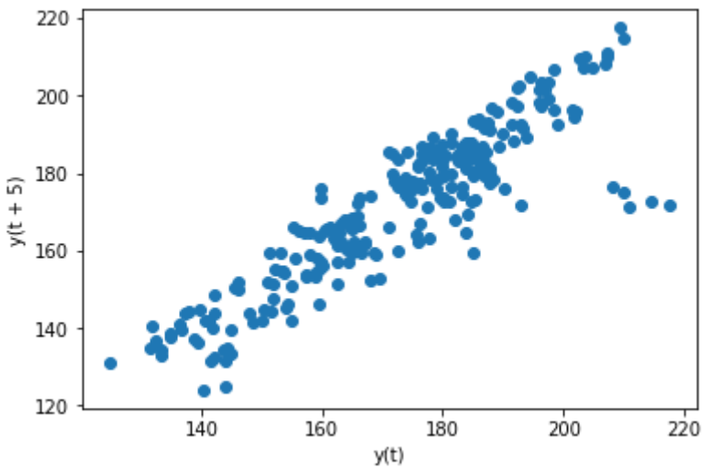
Out[5]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x15217f70>
```



The default lag is 1, but we can alter this with the `lag` parameter. Let's look at a 5 day lag (a week of trading activity):

In [6]:
```
lag_plot(fb.close, lag=5)
```

Out[6]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x15255810>
```

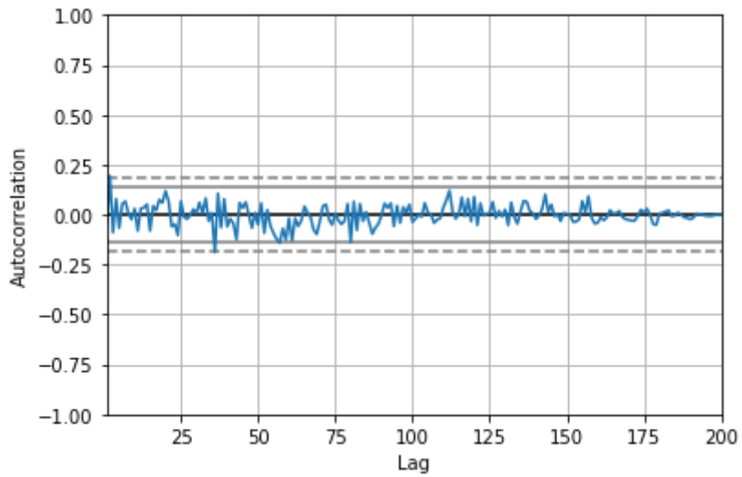


Autocorrelation plots

We can use the autocorrelation plot to see if this relationship may be meaningful or just noise. Random data will not have any significant autocorrelation (it stays within the bounds below):

In [7]:
```
from pandas.plotting import autocorrelation_plot
np.random.seed(0) # make this repeatable
autocorrelation_plot(pd.Series(np.random.random(size=200)))
```

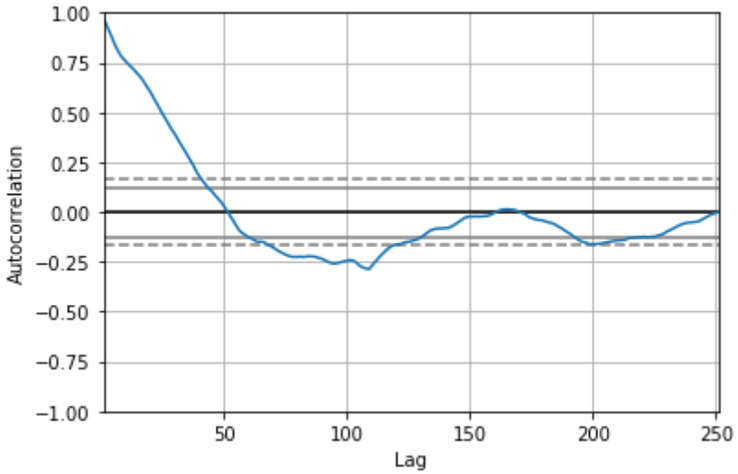
Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x152939f0>
```



Stock data, on the other hand, does have significant autocorrelation:

In [8]:
```
autocorrelation_plot(fb.close)
```

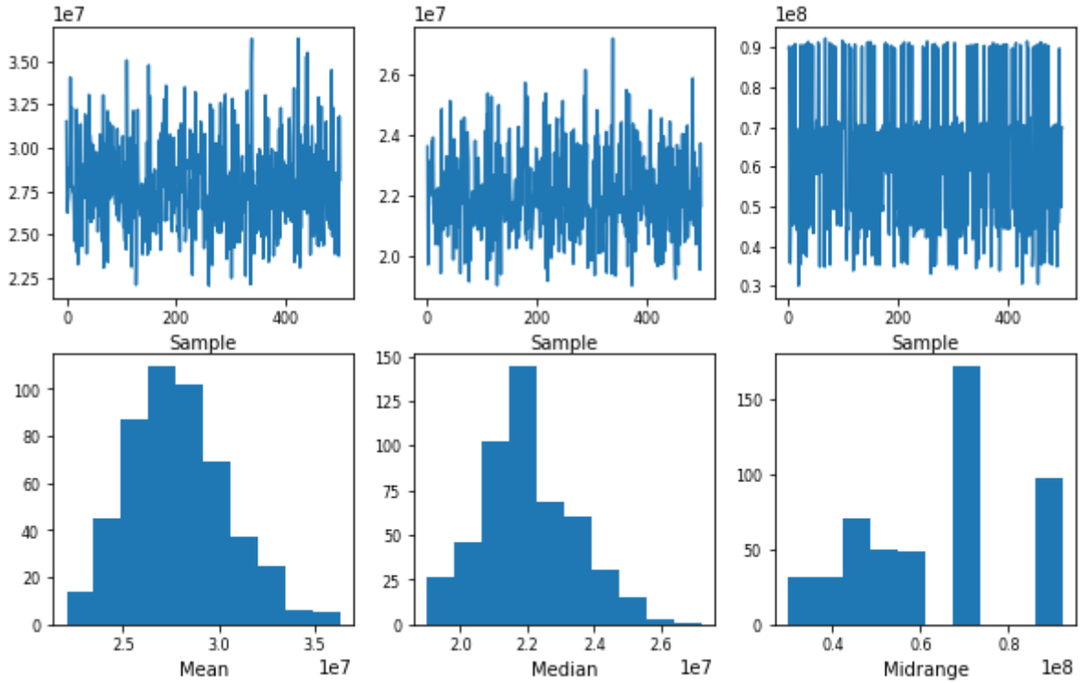
Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x152e5f50>
```



Bootstrap plot

This plot helps us understand the uncertainty in our summary statistics:

In [9]:
```
from pandas.plotting import bootstrap_plot
fig = bootstrap_plot(fb.volume, fig=plt.figure(figsize=(10, 6)))
```


1e7
3.50
3.25
3.00
2.75
2.50
2.25
0
200
400
Sample
1e7
2.6
2.4
2.2
2.0
0
200
400
Sample
1e8
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
200
400
Sample
100
80
60
40
20
0
2.5
3.0
3.5
Mean
1e7
150
125
100
75
50
25
0
2.0
2.2
2.4
2.6
Median
1e7
150
100
50
0
0.4
0.6
0.8
Midrange
1e8