# Pandas Playground- Michael Lanier

December 7, 2016

```
In [51]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         from pandas.tools.plotting import scatter_matrix
         from pandas.tools.plotting import andrews_curves
         from pandas.tools.plotting import autocorrelation_plot
         from pandas.tools.plotting import lag_plot
```

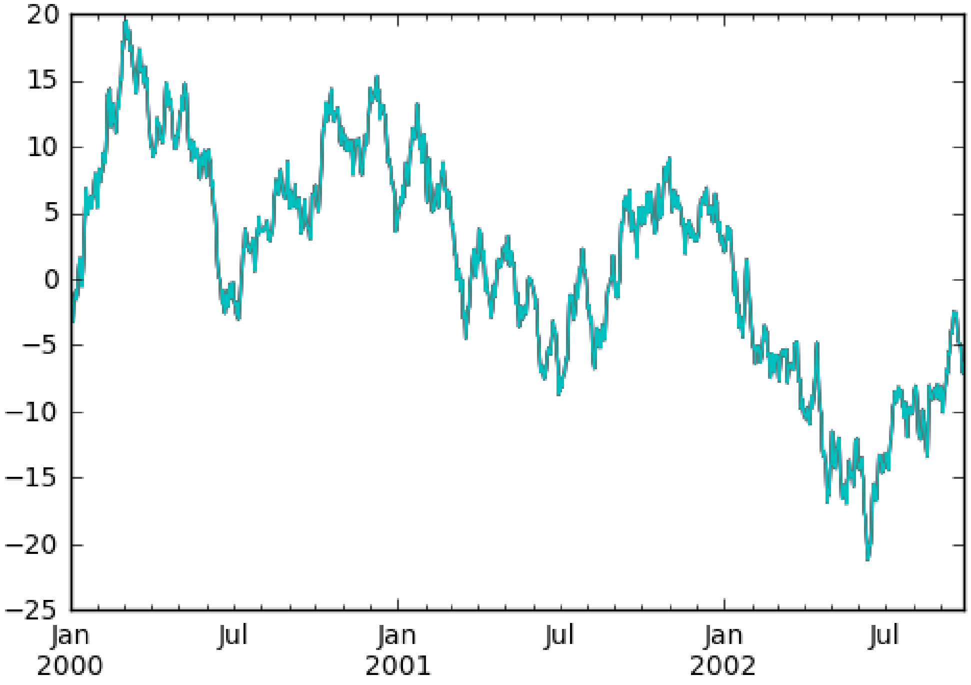
```
In [2]: ts = pd.Series(np.random.randn(1000),
                       index=pd.date_range('1/1/2000', periods=1000))
```

```
In [3]: ts = ts.cumsum()
```

```
In [8]: ts.plot()
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3cf7b4a3c8>
```

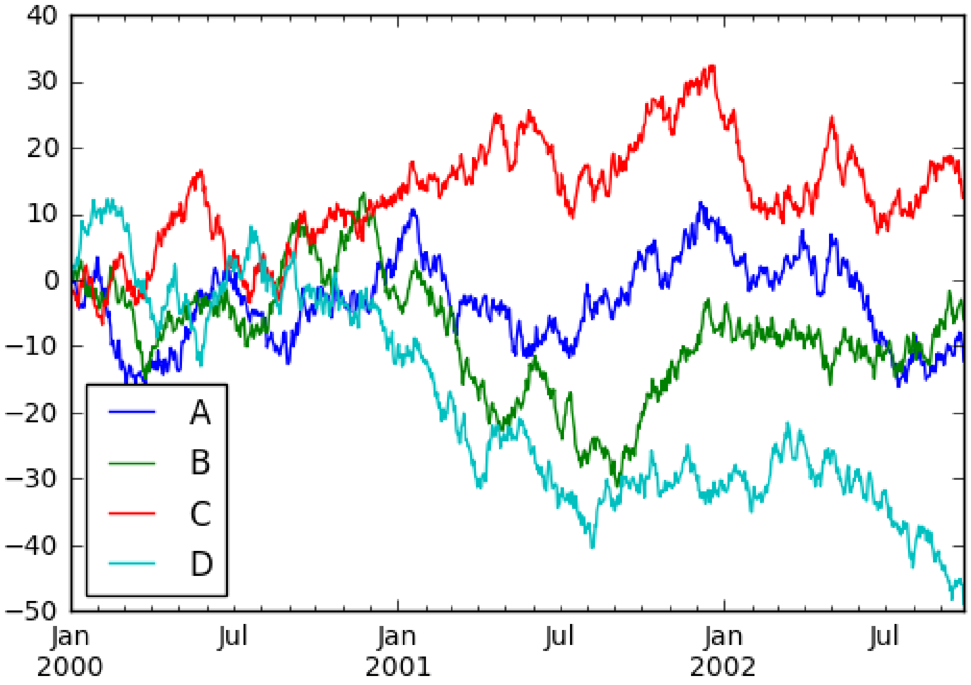
```
In [11]: plt.show()
```



```
In [12]:  df = pd.DataFrame(np.random.randn(1000, 4),
                            index=ts.index, columns=list('ABCD'))

          df = df.cumsum()

          plt.figure(); df.plot();
```

```
In [13]: plt.show()
```

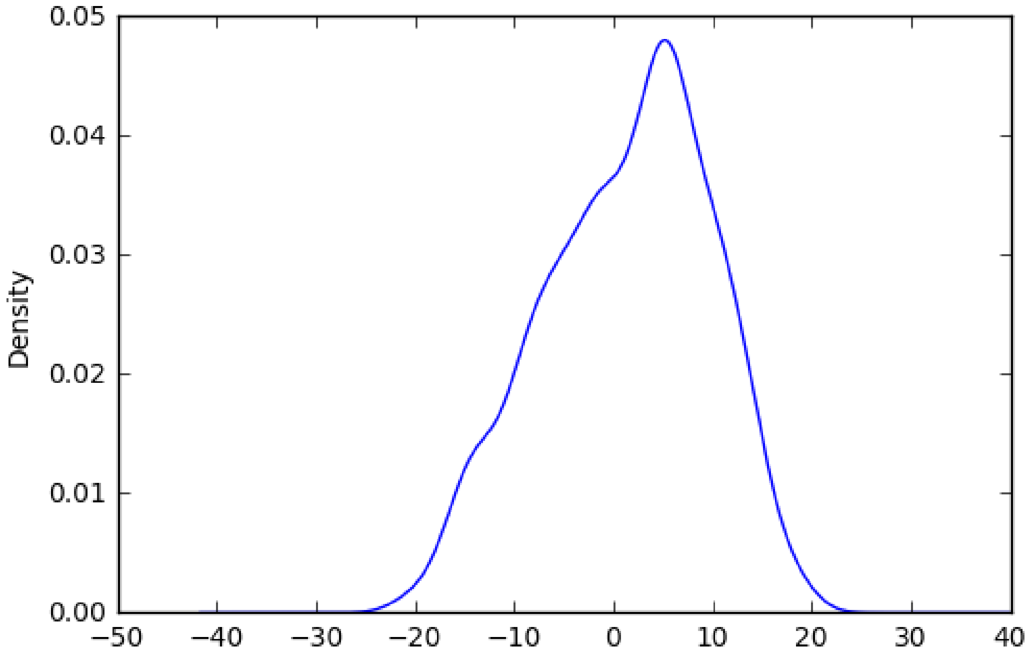
```
<matplotlib.figure.Figure at 0x7f3cc1c54d68>
```



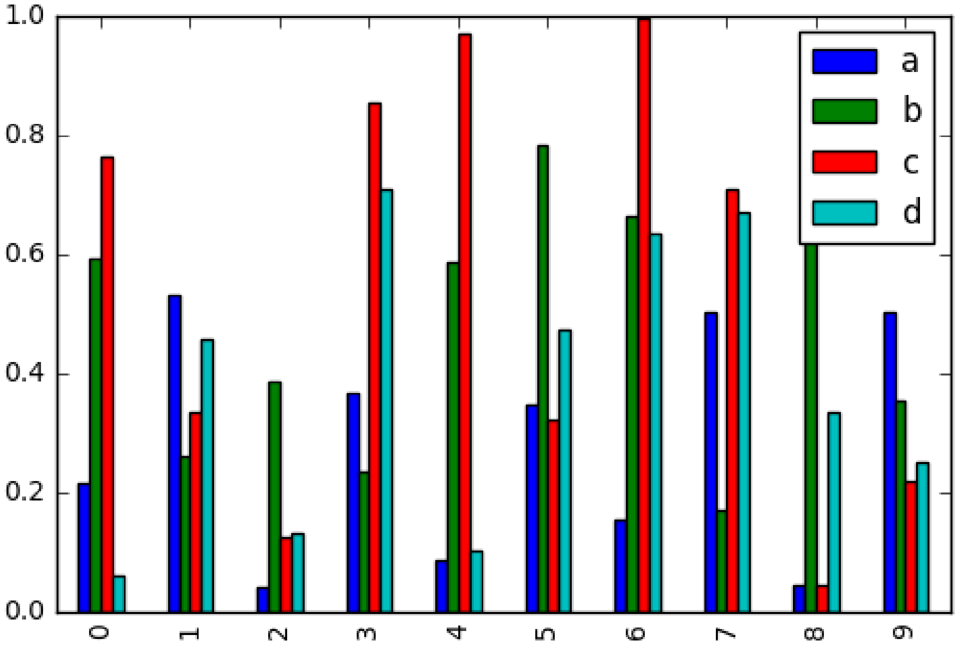
```
In [14]: ts.describe()
```

```
Out[14]: count     1000.000000
         mean         1.344488
         std          8.369861
         min        -21.226486
         25%         -4.847585
         50%          2.471992
         75%          7.331394
         max         19.502199
         dtype: float64
```

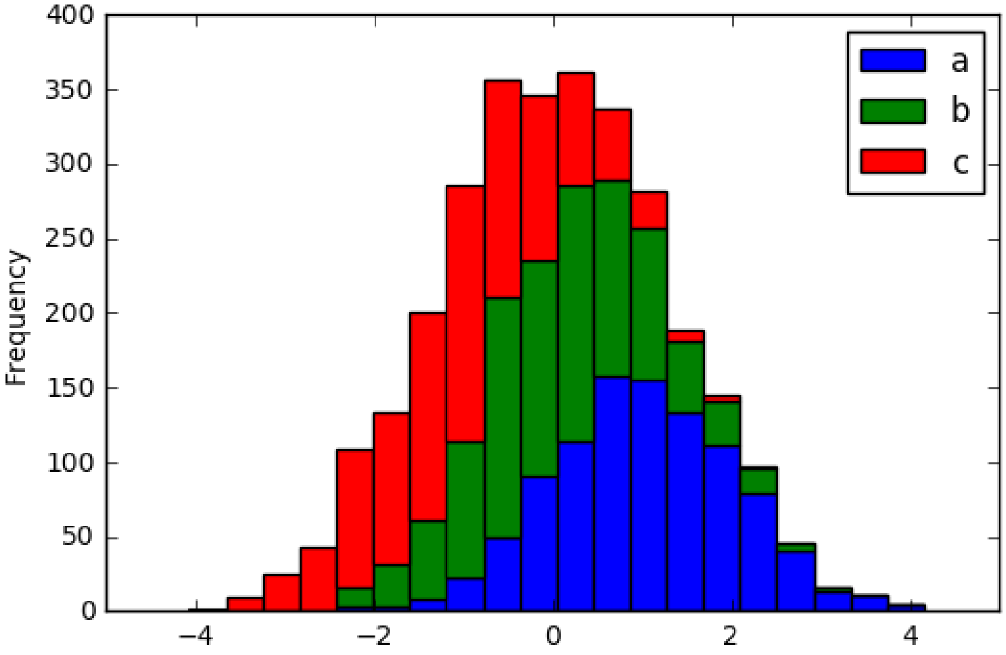
```
In [20]: ts.plot(kind='density')

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3cc58d5710>

In [21]: plt.show()
```



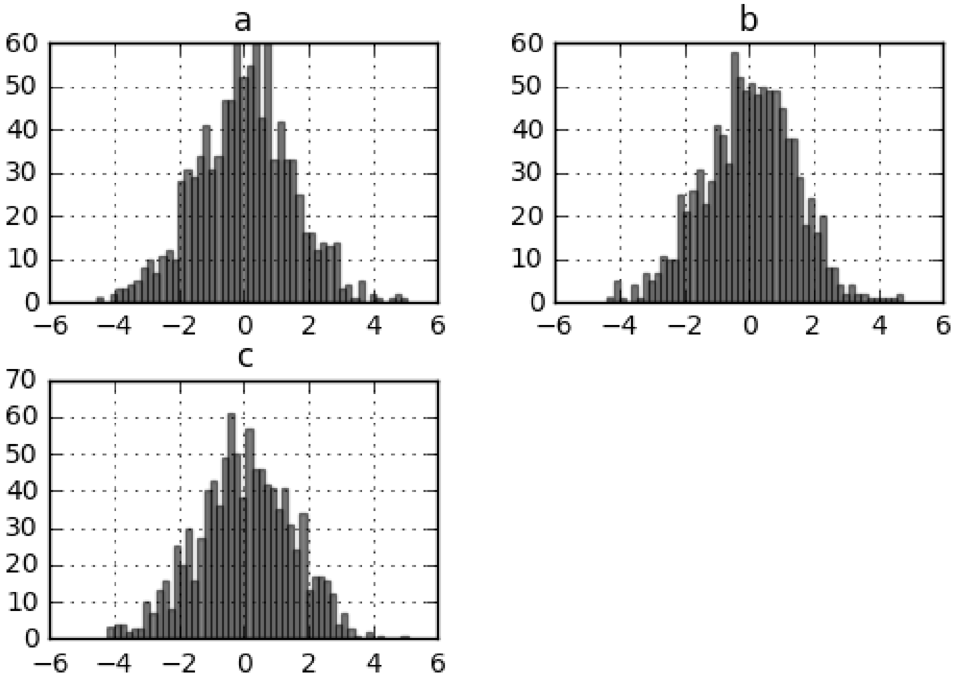
```
In [22]: df2 = pd.DataFrame(np.random.rand(10, 4), columns=['a', 'b', 'c', 'd'])

         df2.plot.bar();
         plt.show()
```

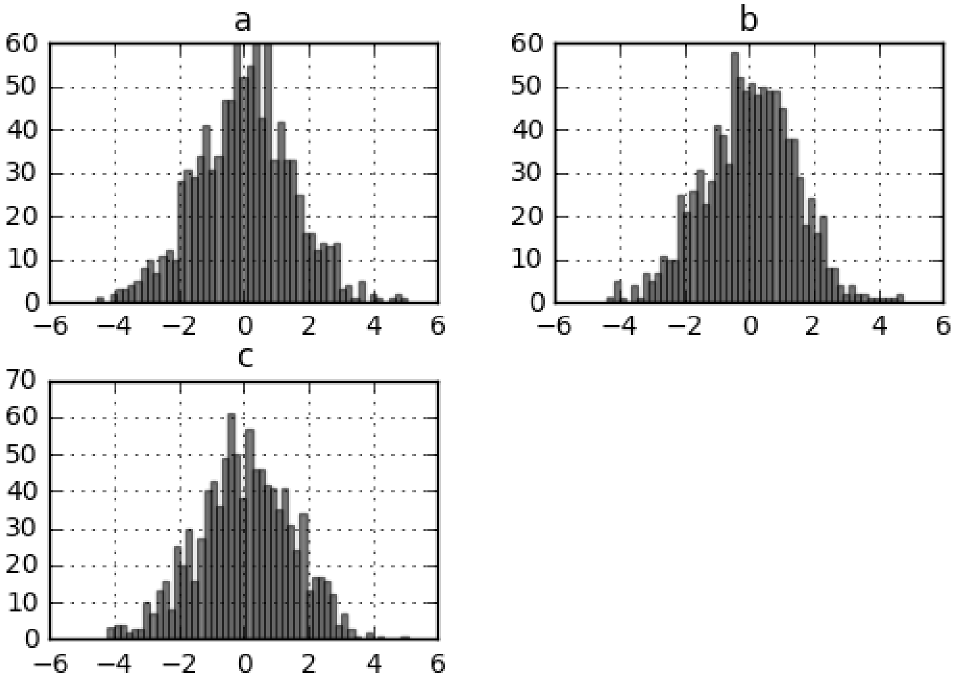


```
In [24]: df4 = pd.DataFrame({'a': np.random.randn(1000) + 1,
                             'b': np.random.randn(1000),
                             'c': np.random.randn(1000) - 1},
                              columns=['a', 'b', 'c'])

In [25]: df4.plot.hist(stacked=True, bins=20)
         plt.show()
```

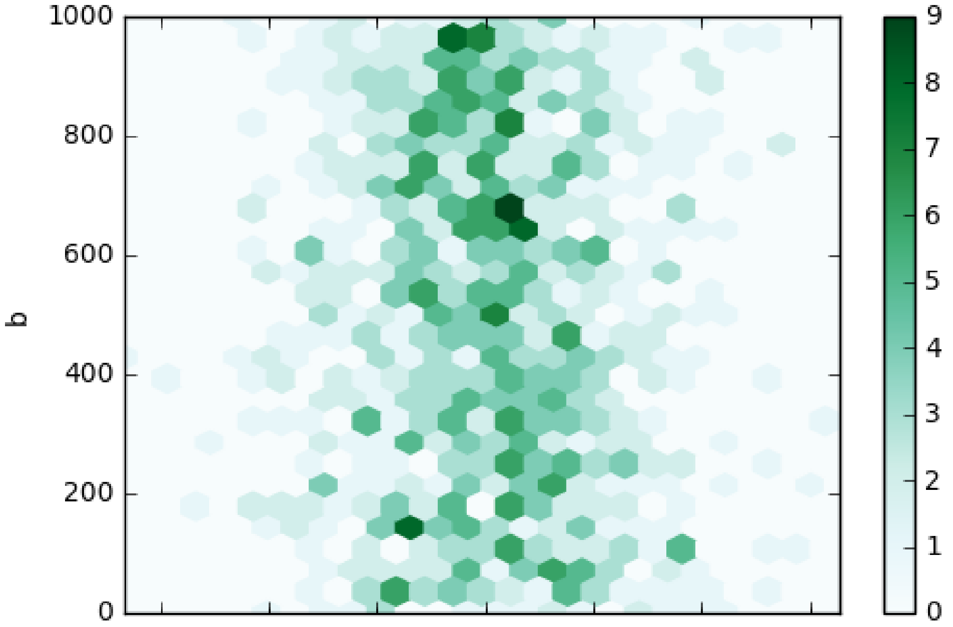


```
In [27]: df4.diff().hist(color='k', alpha=0.5, bins=50)
         plt.show()
```

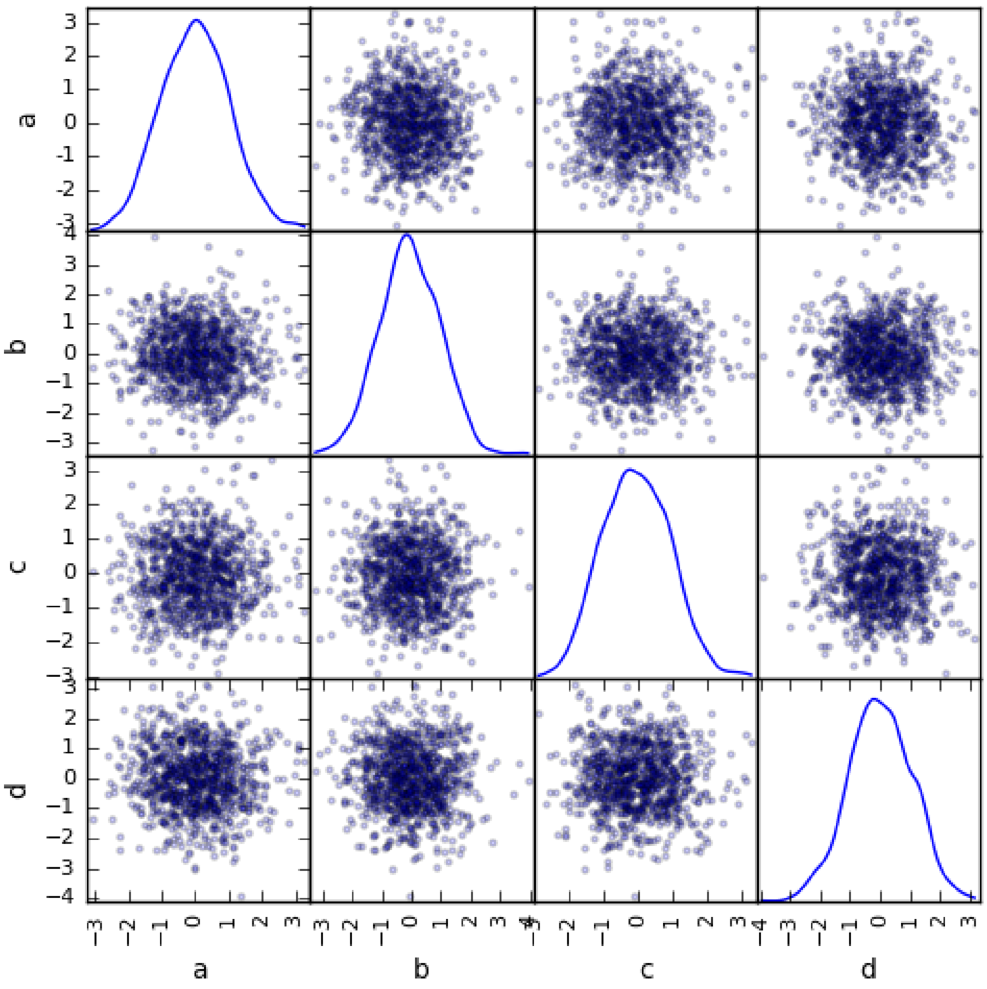




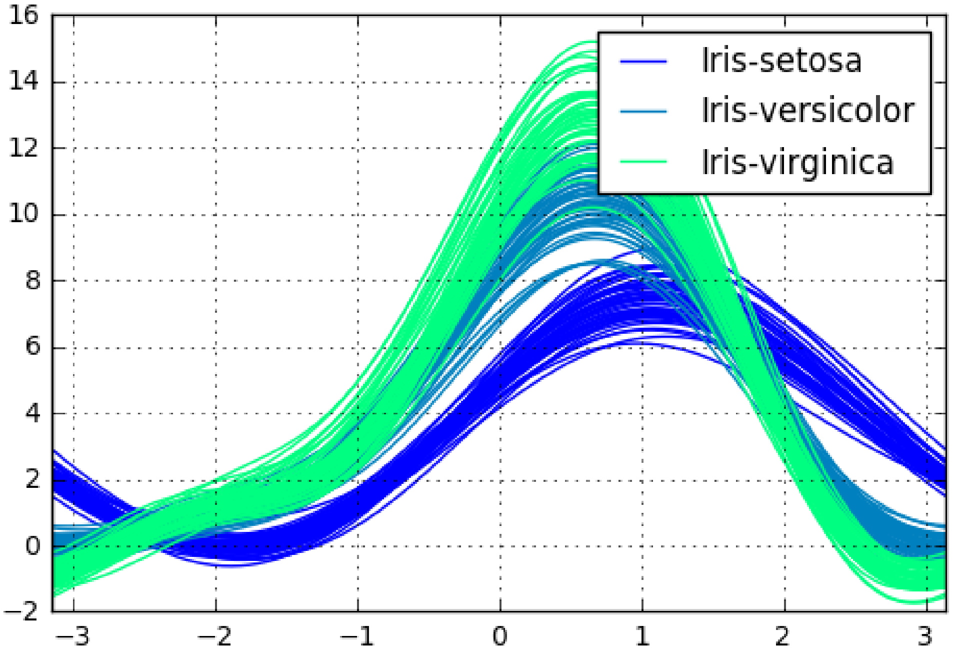
```
In [29]: plt.show
         df = pd.DataFrame(np.random.randn(1000, 2), columns=['a', 'b'])
         df['b'] = df['b'] + np.arange(1000)
         df.plot.hexbin(x='a', y='b', gridsize=25)
         plt.show()
         plt.savefig("temp.png")
```



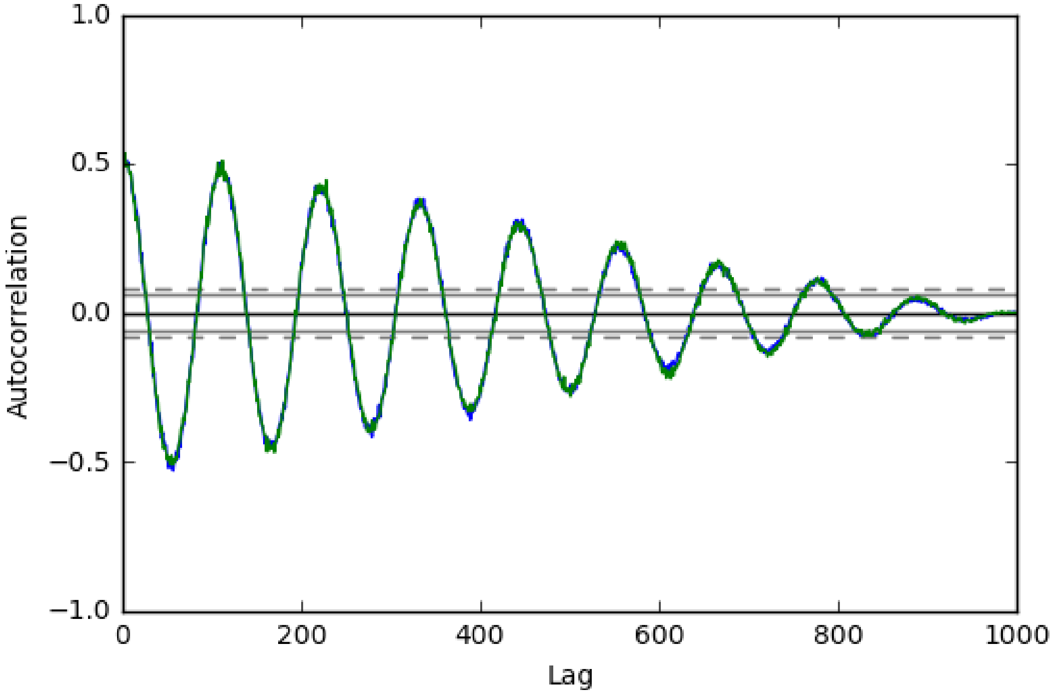
```
In [33]: df = pd.DataFrame(np.random.randn(1000, 4), columns=['a', 'b', 'c', 'd'])
         scatter_matrix(df, alpha=0.2, figsize=(6, 6), diagonal='kde')
         plt.show()
```



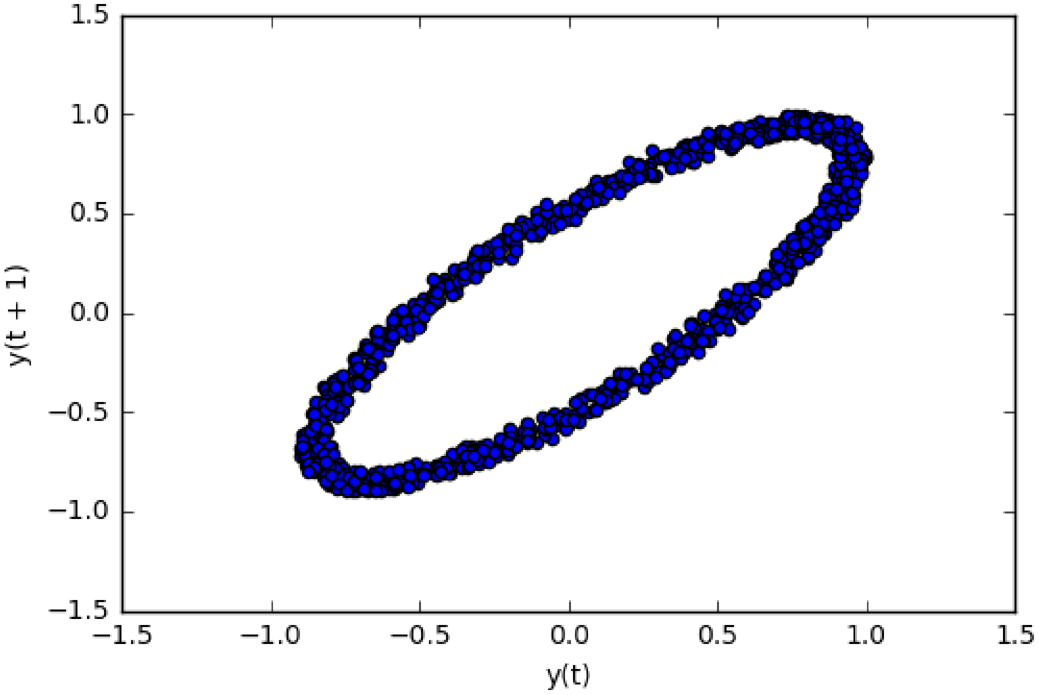
```
In [34]: #Andrews curves allow one to plot multivariate data as a large number of c
         #that are created using the attributes of samples as coefficients for Four
         #By coloring these curves differently for each class it is possible to vis
         #clustering. Curves belonging to samples of the same class will usually be
         #together and form larger structures.

In [53]: data = pd.read_csv('https://raw.githubusercontent.com/pandas-dev/pandas/ma
         plt.figure()
         andrews_curves(data, 'Name',colormap='winter')
         plt.show()
```



```
In [50]: data = pd.Series(0.7 * np.random.rand(1000) +
            0.3 * np.sin(np.linspace(-9 * np.pi, 9 * np.pi, num=1000)))

         autocorrelation_plot(data)
         plt.show()
```



```
In [52]: plt.figure()
         data = pd.Series(0.1 * np.random.rand(1000) +
             0.9 * np.sin(np.linspace(-99 * np.pi, 99 * np.pi, num=1000)))
         lag_plot(data)
         plt.show()
```



```
In [ ]:
```