Hands-on Activity 9.3 Pandas Plotting Subpackage.ipynb - Colaboratory 3/31/24, 4:16 AM

on on one volume les

close

low

### pandas.plotting subpackage

```
Setup
```

%matplotlib inline import matplotlib.pyplot as plt import numpy as np import pandas as pd fb = pd.read\_csv( 'fb\_stock\_prices\_2018.csv', index\_col='date', parse\_dates=True

#### from pandas.plotting import scatter\_matrix scatter\_matrix(fb, figsize=(10, 10))

```
array([[<Axes: xlabel='open', ylabel='open'>,
        <Axes: xlabel='high', ylabel='open'>,
        <Axes: xlabel='low', ylabel='open'>,
        <Axes: xlabel='close', ylabel='open'>,
        <Axes: xlabel='volume', ylabel='open'>],
      [<Axes: xlabel='open', ylabel='high'>,
        <Axes: xlabel='high', ylabel='high'>,
        <Axes: xlabel='low', ylabel='high'>,
        <Axes: xlabel='close', ylabel='high'>,
        <Axes: xlabel='volume', ylabel='high'>],
      [<Axes: xlabel='open', ylabel='low'>,
        <Axes: xlabel='high', ylabel='low'>,
        <Axes: xlabel='low', ylabel='low'>,
        <Axes: xlabel='close', ylabel='low'>,
        <Axes: xlabel='volume', ylabel='low'>],
      [<Axes: xlabel='open', ylabel='close'>,
        <Axes: xlabel='high', ylabel='close'>,
        <Axes: xlabel='low', ylabel='close'>,
        <Axes: xlabel='close', ylabel='close'>,
        <Axes: xlabel='volume', ylabel='close'>],
      [<Axes: xlabel='open', ylabel='volume'>,
        <Axes: xlabel='high', ylabel='volume'>,
        <Axes: xlabel='low', ylabel='volume'>,
        <Axes: xlabel='close', ylabel='volume'>,
        <Axes: xlabel='volume', ylabel='volume'>]], dtype=object)
 <del>0</del> 160 -
   140 -
   220
   200 -
   200 -
   180 -
   200 -
 흥
160 -
   1.5
```

#### Changing the diagonal from histograms to KDE:

open

scatter\_matrix(fb, figsize=(10, 10), diagonal='kde')

array([[<Axes: xlabel='open', ylabel='open'>, <Axes: xlabel='high', ylabel='open'>, <Axes: xlabel='low', ylabel='open'>, <Axes: xlabel='close', ylabel='open'>, <Axes: xlabel='volume', ylabel='open'>], [<Axes: xlabel='open', ylabel='high'>, <Axes: xlabel='high', ylabel='high'>, <Axes: xlabel='low', ylabel='high'>, <Axes: xlabel='close', ylabel='high'>, <Axes: xlabel='volume', ylabel='high'>], [<Axes: xlabel='open', ylabel='low'>, <Axes: xlabel='high', ylabel='low'>, <Axes: xlabel='low', ylabel='low'>, <Axes: xlabel='close', ylabel='low'>, <Axes: xlabel='volume', ylabel='low'>], [<Axes: xlabel='open', ylabel='close'>, <Axes: xlabel='high', ylabel='close'>, <Axes: xlabel='low', ylabel='close'>, <Axes: xlabel='close', ylabel='close'>, <Axes: xlabel='volume', ylabel='close'>], [<Axes: xlabel='open', ylabel='volume'>, <Axes: xlabel='high', ylabel='volume'>, <Axes: xlabel='low', ylabel='volume'>, <Axes: xlabel='close', ylabel='volume'>,

<Axes: xlabel='volume', ylabel='volume'>]], dtype=object) \_ 180 -140 220 200 dgir 180 200 -180 yolume 1e8 close

# Lag plot

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

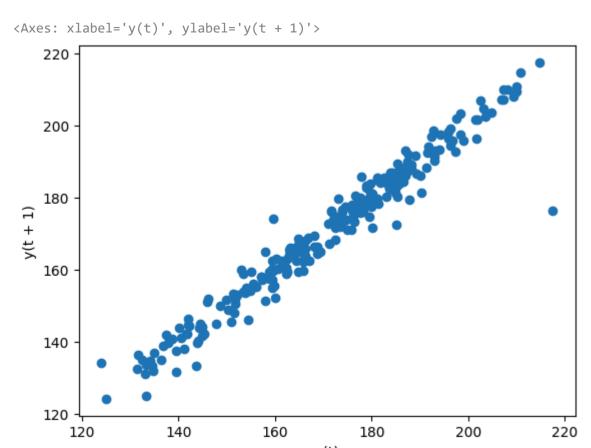
from pandas.plotting import lag\_plot np.random.seed(0) # make this repeatable lag\_plot(pd.Series(np.random.random(size=200)))

<Axes: xlabel='y(t)', ylabel='y(t + 1)'>

y(t)

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

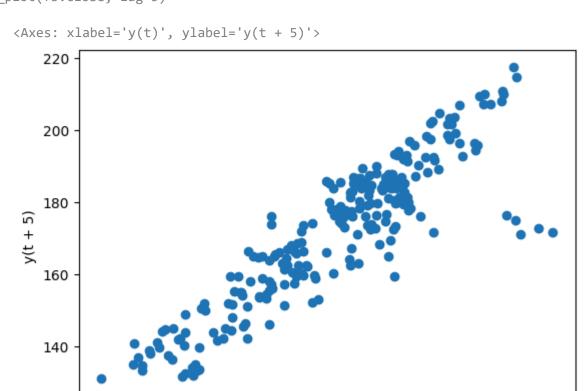
lag\_plot(fb.close)



3/31/24, 4:16 AM

Hands-on Activity 9.3 Pandas Plotting Subpackage.ipynb - Colaboratory

#### lag\_plot(fb.close, lag=5)



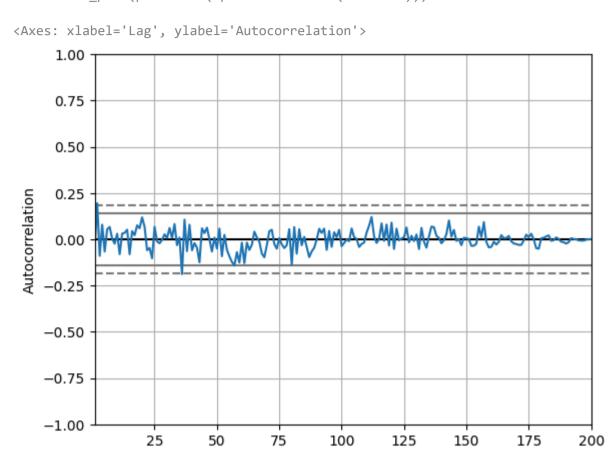
The default lag is 1, but we can alter this with the parameter. Let's look at a 5 day lag (a week of trading activity):

200

220

#### Autocorrelation plots

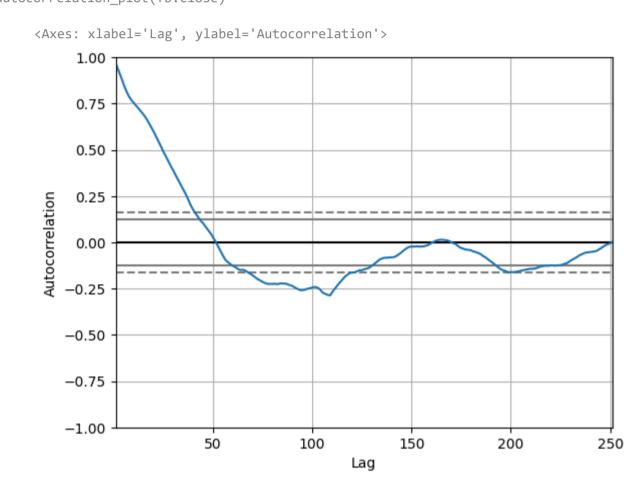
from pandas.plotting import autocorrelation\_plot
np.random.seed(0) # make this repeatable
autocorrelation\_plot(pd.Series(np.random.random(size=200)))



Lag

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

#### autocorrelation\_plot(fb.close)



### Bootstrap plot

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

from pandas.plotting import bootstrap\_plot
fig = bootstrap\_plot(fb.volume, fig=plt.figure(figsize=(10, 6)))

