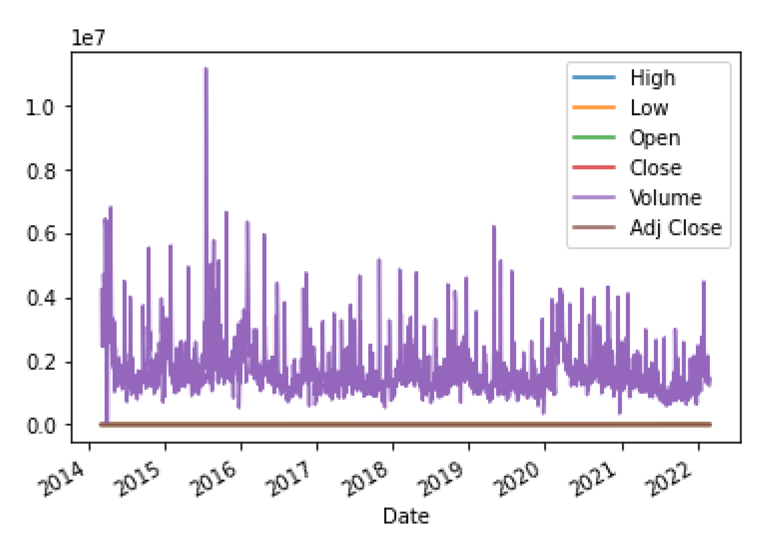
In [14]:
```python
from pandas_datareader import data
import pandas as pd

start_date = '2014-03-01'
end_date = '2022-03-01'

google_data = data.DataReader('GOOG','yahoo', start_date,end_date)

print(google_data)
```

```
                   High          Low         Open        Close     Volume  \
Date
2014-03-03   601.664795   593.844116   601.121826   599.099426  4225379.0
2014-03-04   605.729553   602.491699   605.231445   605.186584  2946403.0
2014-03-05   609.256348   603.443176   605.256348   606.855347  2479860.0
2014-03-06   610.785645   607.024719   608.857849   607.527832  2545706.0
2014-03-07   611.204041   603.458069   611.109436   605.126831  3041558.0
...                 ...          ...          ...          ...        ...
2022-02-23  2634.979980  2550.070068  2621.570068  2551.699951  1321600.0
2022-02-24  2660.739990  2495.290039  2500.000000  2653.469971  2158300.0
2022-02-25  2707.780029  2635.300049  2670.510010  2690.389893  1311800.0
2022-02-28  2712.810059  2656.504883  2665.689941  2697.820068  1483800.0
2022-03-01  2722.219971  2667.570068  2689.600098  2683.360107  1232000.0

              Adj Close
Date
2014-03-03   599.099426
2014-03-04   605.186584
2014-03-05   606.855347
2014-03-06   607.527832
2014-03-07   605.126831
...                 ...
2022-02-23  2551.699951
2022-02-24  2653.469971
2022-02-25  2690.389893
2022-02-28  2697.820068
2022-03-01  2683.360107

[2015 rows x 6 columns]
```

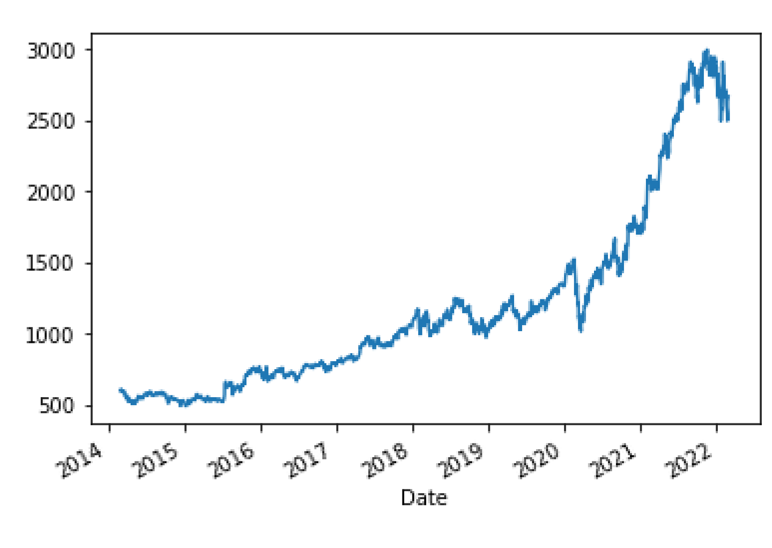
In [5]:
```python
google_data.plot()
```

Out[5]: <AxesSubplot:xlabel='Date'>



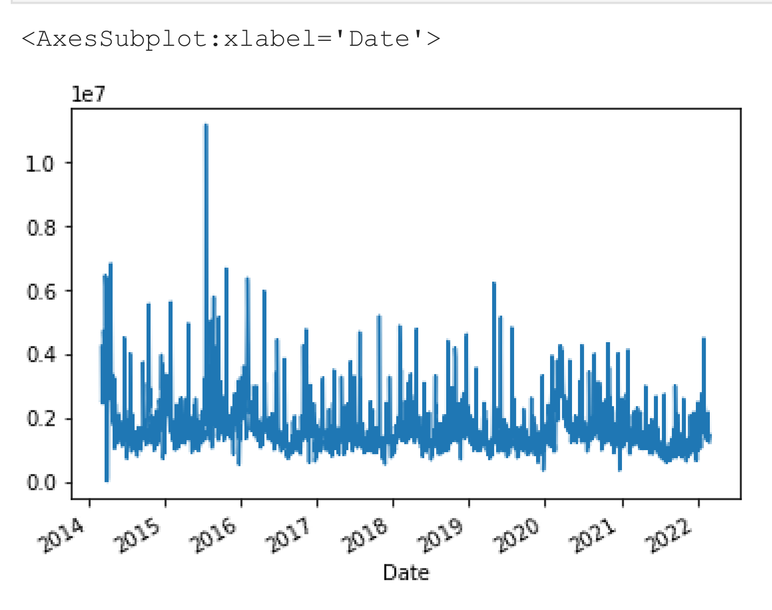
In [6]:
```python
google_data['Low'].plot()
```

Out[6]: <AxesSubplot:xlabel='Date'>



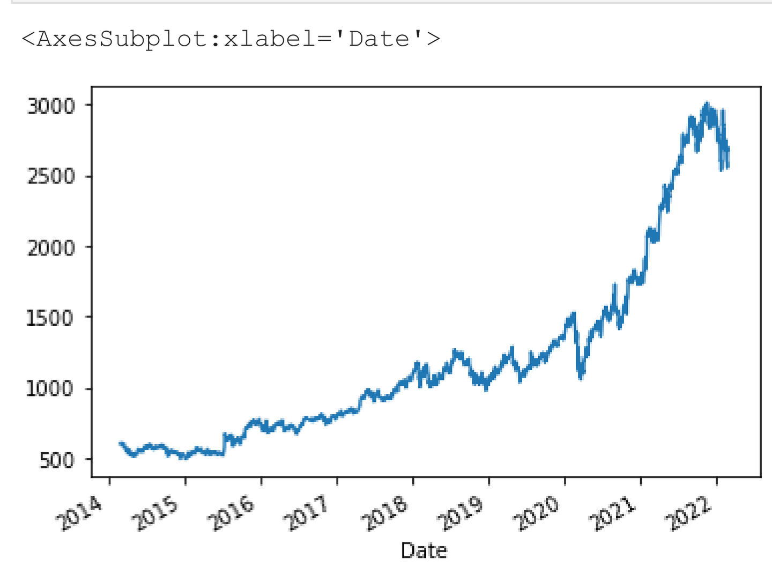
In [8]:
```python
google_data['Volume'].plot()
```

Out[8]: <AxesSubplot:xlabel='Date'>



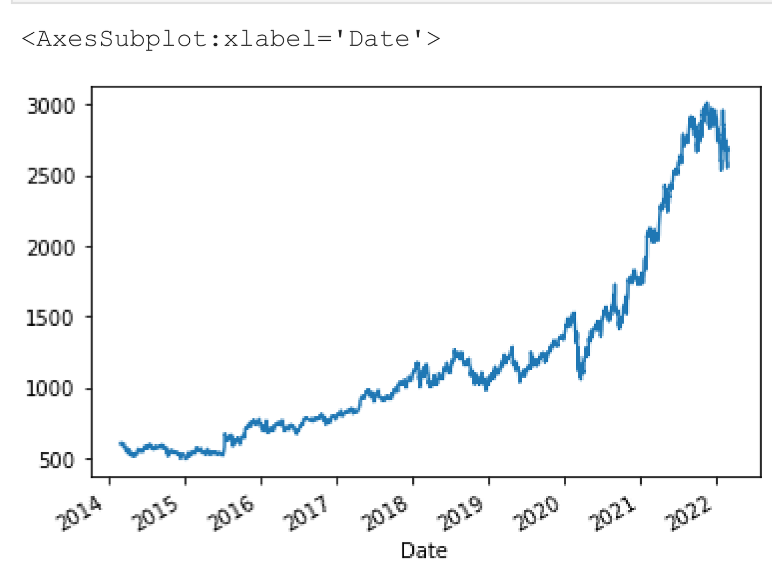
In [9]:
```python
google_data['Close'].plot()
```

Out[9]: <AxesSubplot:xlabel='Date'>



In [12]:
```python
google_data['Adj Close'].plot()
```

Out[12]: <AxesSubplot:xlabel='Date'>



In [17]:
```python
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
```

In [18]:
```python
print(google_data)
```

```
                   High          Low         Open        Close     Volume    Adj Close
Date
2014-03-03   601.664795   593.844116   601.121826   599.099426  4225379.0   599.099426
2014-03-04   605.729553   602.491699   605.231445   605.186584  2946403.0   605.186584
2014-03-05   609.256348   603.443176   605.256348   606.855347  2479860.0   606.855347
2014-03-06   610.785645   607.024719   608.857849   607.527832  2545706.0   607.527832
2014-03-07   611.204041   603.458069   611.109436   605.126831  3041558.0   605.126831
...                 ...          ...          ...          ...        ...          ...
2022-02-23  2634.979980  2550.070068  2621.570068  2551.699951  1321600.0  2551.699951
2022-02-24  2660.739990  2495.290039  2500.000000  2653.469971  2158300.0  2653.469971
2022-02-25  2707.780029  2635.300049  2670.510010  2690.389893  1311800.0  2690.389893
2022-02-28  2712.810059  2656.504883  2665.689941  2697.820068  1483800.0  2697.820068
2022-03-01  2722.219971  2667.570068  2689.600098  2683.360107  1232000.0  2683.360107

[2015 rows x 6 columns]
```

In [64]:
```python
google_data_signal = pd.DataFrame(index = google_data.index)
google_data_signal['price'] = google_data['Adj Close']
```

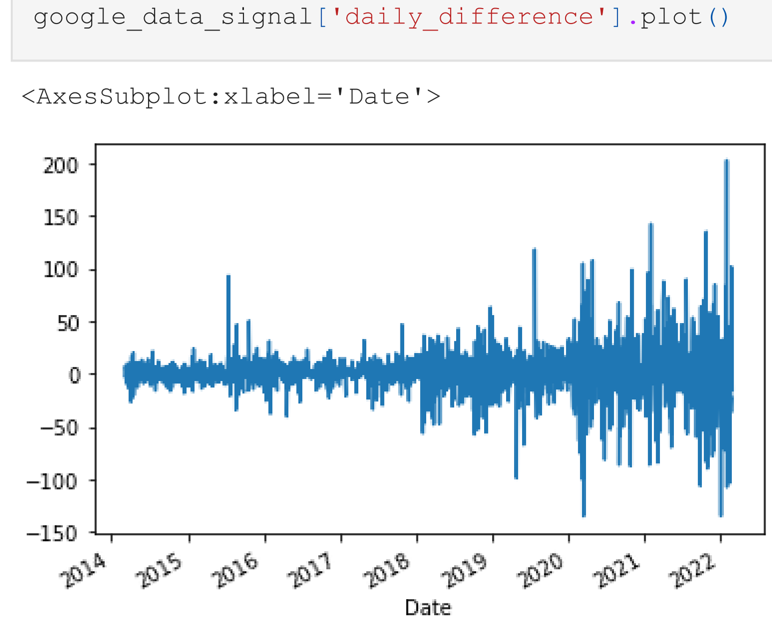
> Based on our trading strategiy, we nee to have a column, **daily_difference, to store the difference betwen two consecutive days in order to create this column , we will use the** diff function from the data frame object :

In [65]:
```python
google_data_signal['daily_difference'] = google_data_signal['price'].diff()

print(google_data_signal.head())
```

```
                 price  daily_difference
Date
2014-03-03  599.099426               NaN
2014-03-04  605.186584          6.087158
2014-03-05  606.855347          1.668762
2014-03-06  607.527832          0.672485
2014-03-07  605.126831         -2.401001
```

In [66]:
```python
google_data_signal['daily_difference'].plot()
```

Out[66]: <AxesSubplot:xlabel='Date'>



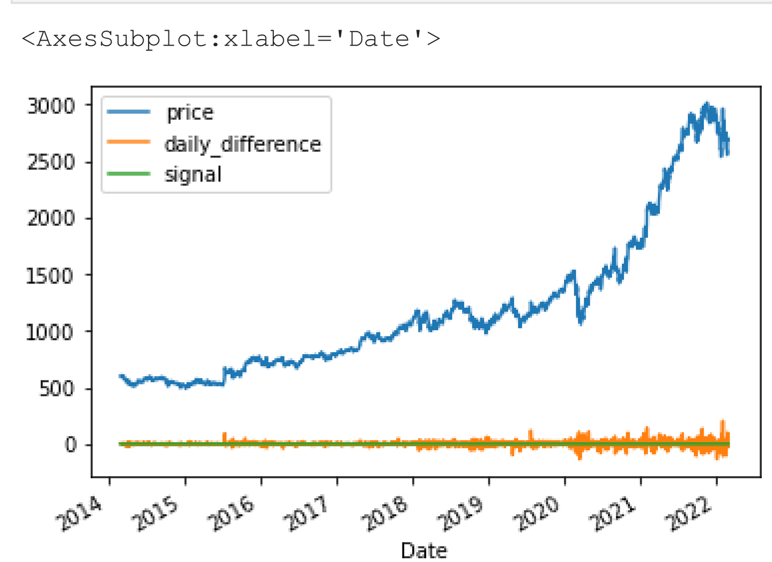
> We will create the signal based on the values of te column, **daily_difference if the value is positve, we will give the value 1, othewise, the value will remain 0:

In [67]:
```python
import numpy as np

google_data_signal['signal'] = 0.0
google_data_signal['signal'] = np.where(google_data_signal['daily_difference'] > 0 , 

print(google_data_signal['signal'])
```

```
Date
2014-03-03    0.0
2014-03-04    1.0
2014-03-05    1.0
2014-03-06    1.0
2014-03-07    0.0
             ...
2022-02-23    0.0
2022-02-24    1.0
2022-02-25    1.0
2022-02-28    1.0
2022-03-01    0.0
Name: signal, Length: 2015, dtype: float64
```

In [68]:
```python
google_data_signal.plot()
```

Out[68]: <AxesSubplot:xlabel='Date'>



> Reading the column signal, we have 0 when we need to buy, and we have 1 when we need to sell Since we don't want to constantly buy if the market keeps movings down, or constantly sell when the market is moving up, we will limit the number of orders by restricting ourselves to the nmbers of poditions on the market, Te position is your inventory of stocks or assets that you have on the market; for instance, if you buy one google share, this means you have a position of one share on the market. if you sell this share you will not have any positions on the market

> To symplify our exemple and limit the position on the market, it will be impossible to buy or sell more than one time consecutively. therefore, we will apply diff() to the column signal

In [69]:
```python
google_data_signal['positions'] = google_data_signal['signal'].diff()

print(google_data_signal['positions'])
```

```
Date
2014-03-03    NaN
2014-03-04    1.0
2014-03-05    0.0
2014-03-06    0.0
2014-03-07   -1.0
             ...
2022-02-23    0.0
2022-02-24    1.0
2022-02-25    0.0
2022-02-28    0.0
2022-03-01   -1.0
Name: positions, Length: 2015, dtype: float64
```

Singnal Visualisation

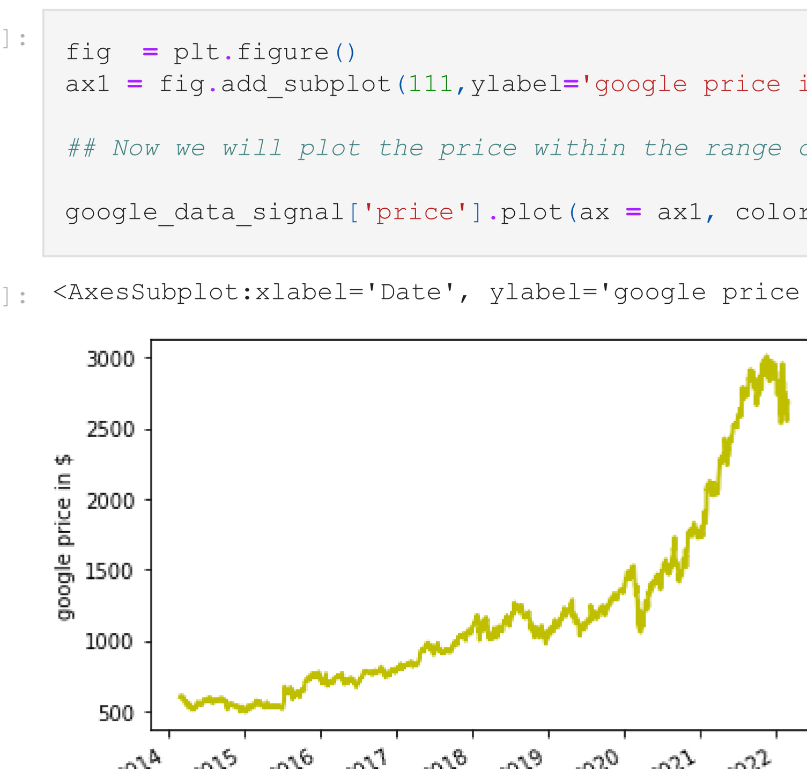
> We will start by importing matplotlib

In [70]:
```python
import matplotlib.pyplot as plt
```

> Next, we will define a figure that will contain our chart:

In [71]:
```python
fig  = plt.figure()
ax1 = fig.add_subplot(111,ylabel='google price in $')

## Now we will plot the price within the range of days we initaily chose :

google_data_signal['price'].plot(ax = ax1, color='y', lw = 2.)
```

Out[71]: <AxesSubplot:xlabel='Date', ylabel='google price in $'>



> next we will draw an up arrow when we buy on google share :

In [72]:
```python
ax1.plot(google_data_signal.loc[google_data_signal.positions == 1.0].index,
        google_data_signal.price[google_data_signal.positions == 1.0],
        '^',markersize = 4, color = 'y')
plt.show()
```

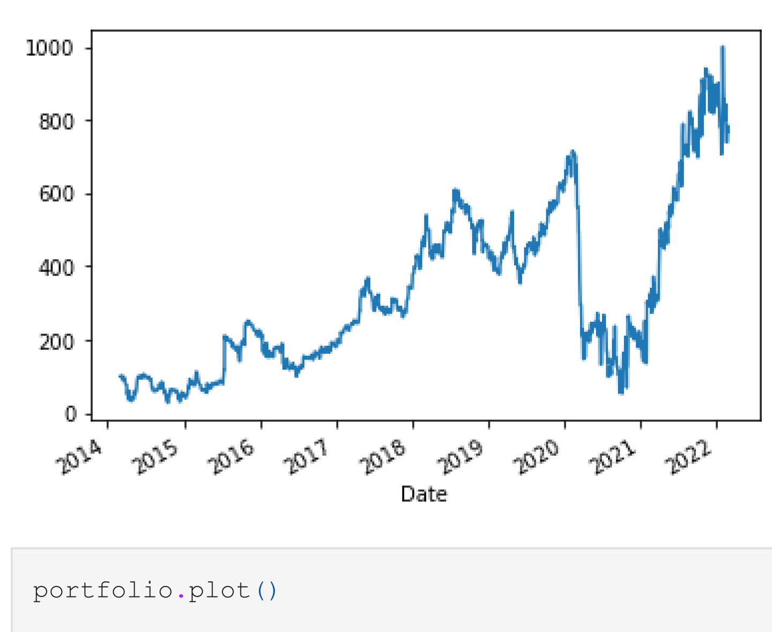
In [73]:
```python
#next we will draw a down arrow when we sell google share

ax1.plot(google_data_signal.loc[google_data_signal.positions == -1.0].index,
        google_data_signal.price[google_data_signal.positions == -1.0],
        'v',markersize = 4, color = 'm')
```

Out[73]: [<matplotlib.lines.Line2D at 0x2503b28f220>]

In [76]:
```python
initial_capital = float (100.0)
positions = pd.DataFrame(index = google_data_signal.index).fillna(0.0)
portfolio = pd.DataFrame(index = google_data_signal.index).fillna(0.0)
positions['GOOG'] = google_data_signal['signal']

portfolio['positions'] = (positions.multiply(google_data_signal['price'],axis=0))

portfolio['cash'] = initial_capital - (positions.diff().multiply(google_data_signal['
```

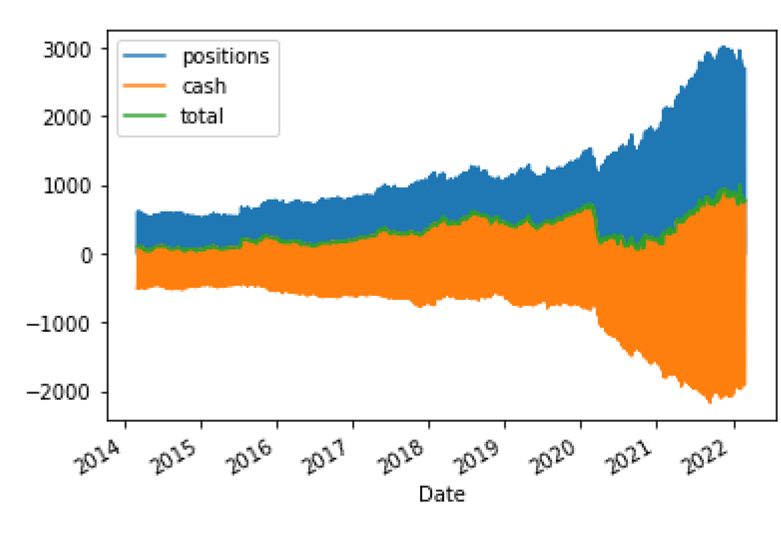
In [78]:
```python
portfolio['total'] = portfolio['positions'] + portfolio['cash']

portfolio['total'].plot()
```

Out[78]: <AxesSubplot:xlabel='Date'>



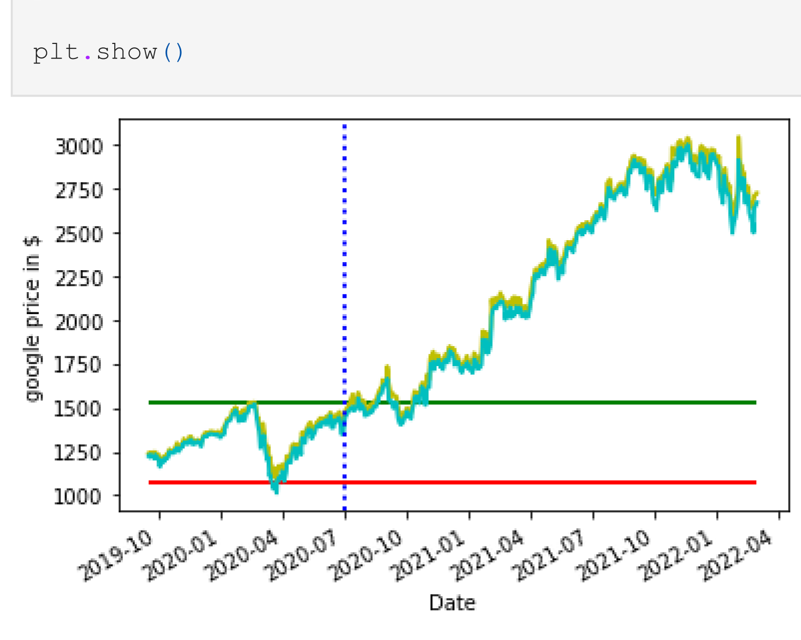
In [79]:
```python
portfolio.plot()
```

Out[79]: <AxesSubplot:xlabel='Date'>



Trading signal Generation and Strategies

In [84]:
```python
SRC_DATA_FILENAME = 'google_data.pkl'

try:
    google_data2 = pd.read_pickle(SRC_DATA_FILENAME)
except FileNotFoundError:
    google_data2 = data.DataReader('GOOG', 'yahoo', start_date,end_date)
    google_data2.to_pickle(SRC_DATA_FILENAME)

google_data = google_data2.tail(620)
lows = google_data['Low']
highs = google_data['High']

fig = plt.figure()

ax1 = fig.add_subplot(111,ylabel='google price in $')
highs.plot(ax=ax1,color='y',lw=2.)
lows.plot(ax=ax1,color='c',lw=2.)
plt.hlines(highs.head(200).max(),lows.index.values[0],lows.index.values[-1],linewidth=
plt.hlines(highs.head(200).min(),lows.index.values[0],lows.index.values[-1],linewidth=
plt.axvline(linewidth=2,color='b',x=lows.index.values[200],linestyle=':')

plt.show()
```



In [ ]: