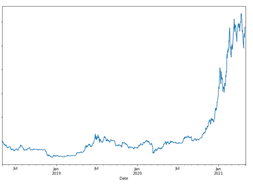
In [12]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
data_B = pd.read_csv('BTC-USD.csv', index_col='Date', parse_dates=True)
data_B.head()
```

Out[12]:

| Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|
| **2018-05-03** | 9233.968727 | 9796.330078 | 9188.150391 | 9743.860352 | 9743.860352 | 1.020709e+10 |
| **2018-05-04** | 9695.500000 | 9779.208199 | 9585.959961 | 9700.759766 | 9700.759766 | 8.217830e+09 |
| **2018-05-05** | 9700.280273 | 9964.500000 | 9686.135117 | 9858.150391 | 9858.150391 | 7.651940e+09 |
| **2018-05-06** | 9845.309570 | 9940.139648 | 9465.250000 | 9654.799805 | 9654.799805 | 7.222280e+09 |
| **2018-05-07** | 9645.669922 | 9665.849609 | 9231.530273 | 9373.009766 | 9373.009766 | 7.394020e+09 |

In [13]:
```
data_B['Close'].plot(figsize=(12, 8))
```

Out[13]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xd9df938a60>
```

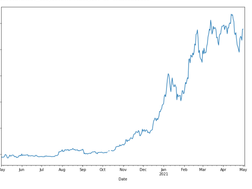



In [14]:
```
data_B.index
```

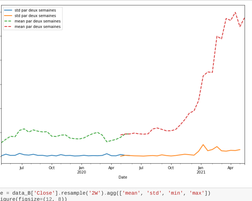
Out[14]:
```
DatetimeIndex(['2018-05-03', '2018-05-04', '2018-05-05', '2018-05-06',
               '2018-05-07', '2018-05-08', '2018-05-09', '2018-05-10',
               '2018-05-11', '2018-05-12',
               ...
               '2021-04-24', '2021-04-25', '2021-04-26', '2021-04-27',
               '2021-04-28', '2021-04-29', '2021-04-30', '2021-05-01',
               '2021-05-02', '2021-05-03'],
              dtype='datetime64[ns]', name='Date', length=1097, freq=None)
```

In [5]:
```
data_B['2020-05':'2021-05']['Close'].plot(figsize=(12, 8))
```

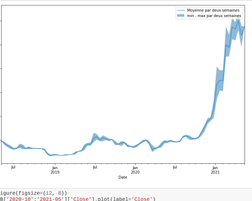
Out[5]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xd9eba69580>
```



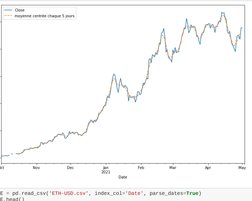

In [15]:
```
plt.figure(figsize=(12, 8))
data_B['2019-05':'2020-05']['Close'].resample('2W').std().plot(label='std par deux semaines'
, lw=2, alpha=1)
data_B['2019-05':'2021-05']['Close'].resample('2W').std().plot(label='std par deux semaines'
, lw=2, alpha=1)
data_B['2019-05':'2020-05']['Close'].resample('2W').mean().plot(label='mean par deux semaine
s', lw=2,ls='--', alpha=1)
data_B['2020-05':'2021-05']['Close'].resample('2W').mean().plot(label='mean par deux semaine
s', lw=2,ls='--', alpha=1)
plt.legend()
plt.show()
```




In [16]:
```
sample = data_B['Close'].resample('2W').agg(['mean', 'std', 'min', 'max'])
plt.figure(figsize=(12, 8))
sample['mean'].plot(label='Moyenne par deux semaines', lw=1)
plt.fill_between(sample.index, sample['min'], sample['max'], alpha=0.5, label='min - max par
deux semaines')
plt.legend()
plt.show()
```




In [17]:
```
plt.figure(figsize=(12, 8))
data_B['2020-10':'2021-05']['Close'].plot(label='Close')
data_B['2020-10':'2021-05']['Close'].rolling(window=5, center=True).mean().plot(label='moyen
ne centrée chaque 5 jours', ls='--')
plt.legend()
plt.show()
```




In [18]:
```
data_E = pd.read_csv('ETH-USD.csv', index_col='Date', parse_dates=True)
data_E.head()
```

Out[18]:

| Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|
| **2018-05-03** | 162.275165 | 170.068741 | 161.090627 | 167.952408 | 167.952408 | 7.299411e+09 |
| **2018-05-04** | 167.897222 | 170.645935 | 161.791428 | 164.026581 | 164.026581 | 6.656100e+09 |
| **2018-05-05** | 164.015259 | 165.399979 | 159.750653 | 163.450699 | 163.450699 | 5.938416e+09 |
| **2018-05-06** | 163.337982 | 175.760101 | 159.986190 | 172.653214 | 172.653214 | 7.540097e+09 |
| **2018-05-07** | 172.427277 | 180.394409 | 169.694717 | 169.798660 | 169.798660 | 8.411140e+09 |

In [25]:
```
data_BE = pd.merge(data_B, data_E,on='Date', how='outer', suffixes=('-BTC', '-ETH') )
data_BE.head()
data_BE.shape
```

Out[25]:
```
(1097, 12)
```

In [26]:
```
data_BE = pd.merge(data_B, data_E,on='Date', how='inner', suffixes=('-BTC', '-ETH') )
data_BE.shape
```

Out[26]:
```
(732, 12)
```

In [34]:
```
data_BE[['Close-BTC', 'Close-ETH']].plot(subplots=True, figsize=(15, 10))
plt.show()
```

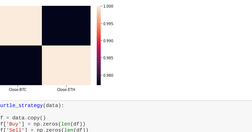


In [37]:
```
corr_BTC_ETH = data_BE[['Close-BTC', 'Close-ETH']].corr()
print(corr_BTC_ETH)
```

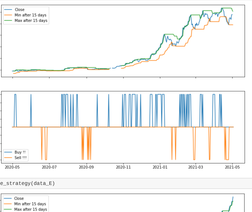
```
           Close-BTC  Close-ETH
Close-BTC   1.000000   0.977838
Close-ETH   0.977838   1.000000
```

In [36]:
```
sns.heatmap(corr_BTC_ETH)
```

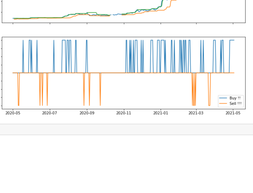
Out[36]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xd9ec05dca0>
```




In [42]:
```
def turtle_strategy(data):

    df = data.copy()
    df['Buy'] = np.zeros(len(df))
    df['Sell'] = np.zeros(len(df))
    df['rollingmin'] = df['Close'].shift(1).rolling(window=15).min()
    df['rollingmax'] = df['Close'].shift(1).rolling(window=15).min()
    df.loc[df['rollingmin'] > df['Close'], 'Sell'] = -1
    df.loc[df['rollingmax'] < df['Close'], 'Buy'] = 1

    fig, ax = plt.subplots(2, figsize=(12, 8), sharex=True, )
    ax[0].plot(df['2020-05':'2021-05']['Close'])
    ax[0].plot(df['2020-05':'2021-05']['rollingmin'])
    ax[0].plot(df['2020-05':'2021-05']['rollingmax'])
    ax[0].plot(df['2020-05':'2021-05']['Close'], '^', 'Max after 15 days'])
    ax[1].plot(df['2020-05':'2021-05']['Buy'])
    ax[1].plot(df['2020-05':'2021-05']['Sell'])
    ax[1].legend(['Buy !!','Sell !!!'])
    plt.show()

turtle_strategy(data_B)
```




In [43]:
```
turtle_strategy(data_E)
```




In [ ]:
```
```