Time Series Analysis of Litecoin Cryptocurrency in Python

Shpaner, Leonid - March 11. 2022

The following time series analysis is an update to the pre-existing analysis conducted in R, but this time done in Python. The data is sourced from Yahoo Finance.

[1]:
```
import pandas as pd
import numpy as np
import yfinance as yf

import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
ltc = yf.download('LTC-USD', start ='2011-01-01',
                              end='2022-03-10',
                              progress=False)
```

[3]:
```
ltc.head()
```

[3]:

Date	Open	High	Low	Close	Adj Close	Volume
2014-09-17	5.08589	5.17077	4.96595	5.05855	5.05855	3071840
2014-09-18	5.06543	5.06543	4.57996	4.68523	4.68523	4569260
2014-09-19	4.68729	4.75582	4.25435	4.32777	4.32777	3917450
2014-09-20	4.32920	4.61608	4.20219	4.28644	4.28644	5490660
2014-09-21	4.26307	4.30013	4.15499	4.24592	4.24592	2931220

Simple Returns

Simple returns are aggregated over assets, which is a "weighted sum of the returns of the individual assests in the portfolio" (Lewinson, 2020).

$$R_t = \frac{(P_t - P_{t-1})}{P_{t-1}} = \frac{P_t}{P_{t-1}} - 1$$

Log Returns

Log returns are aggregated over time.

$$r_t = log\left(\frac{P_t}{P_{t-1}}\right) = log(P_t) - log(P_t - 1)$$

[4]:
```
ltc.rename(columns = {'Adj Close': 'adj_close'}, inplace=True)
ltc['simple_rtn'] = ltc.adj_close.pct_change()
ltc['log_rtn'] = np.log(ltc.adj_close/ltc.adj_close.shift(1))
```

[5]:
```
ltc.head()
```

[5]:

Date	Open	High	Low	Close	adj_close	Volume	\
2014-09-17	5.08589	5.17077	4.96595	5.05855	5.05855	3071840	
2014-09-18	5.06543	5.06543	4.57996	4.68523	4.68523	4569260	

```
2014-09-19  4.68729  4.75582  4.25435  4.32777    4.32777  3917450
2014-09-20  4.32920  4.61608  4.20219  4.28644    4.28644  5490660
2014-09-21  4.26307  4.30013  4.15499  4.24592    4.24592  2931220

            simple_rtn   log_rtn
Date
2014-09-17         NaN       NaN
2014-09-18   -0.073800 -0.076665
2014-09-19   -0.076295 -0.079363
2014-09-20   -0.009550 -0.009596
2014-09-21   -0.009453 -0.009498
```

[6]:
```
# ltc.reset_index(inplace=True)
```

[7]:
```
ltc.head()
# ltc['Date'] = pd.to_datetime(ltc['Date'])
```

[7]:
```
               Open     High      Low    Close  adj_close   Volume  \
Date
2014-09-17  5.08589  5.17077  4.96595  5.05855    5.05855  3071840
2014-09-18  5.06543  5.06543  4.57996  4.68523    4.68523  4569260
2014-09-19  4.68729  4.75582  4.25435  4.32777    4.32777  3917450
2014-09-20  4.32920  4.61608  4.20219  4.28644    4.28644  5490660
2014-09-21  4.26307  4.30013  4.15499  4.24592    4.24592  2931220

            simple_rtn   log_rtn
Date
2014-09-17         NaN       NaN
2014-09-18   -0.073800 -0.076665
2014-09-19   -0.076295 -0.079363
2014-09-20   -0.009550 -0.009596
2014-09-21   -0.009453 -0.009498
```

[8]:
```
print('Number of Rows:', ltc.shape[0])
print('Number of Columns:', ltc.shape[1], '\n')

data_types = ltc.dtypes
data_types = pd.DataFrame(data_types)
data_types = data_types.assign(Null_Values =
                               ltc.isnull().sum())
data_types.reset_index(inplace = True)
data_types.rename(columns={0:'Data Type',
                           'index': 'Column/Variable',
                           'Null_Values': "# of Nulls"})
```

```
Number of Rows: 2732
Number of Columns: 8
```

[8]:
```
  Column/Variable Data Type  # of Nulls
0            Open   float64           0
1            High   float64           0
```

```
2             Low   float64           0
3           Close   float64           0
4       adj_close   float64           0
5          Volume     int64           0
6      simple_rtn   float64           1
7         log_rtn   float64           1
```

Below, litecoin's historical prices (2015 - Present), simple returns, and log returns, respectively, are shown.

[9]:
```
fig, ax = plt.subplots(3,1,figsize=(24,22), sharex=True)
ltc.adj_close.plot(ax=ax[0], color = 'black')
ax[0].set(title = 'Litecoin Time Series',
          ylabel = 'Price ($)')
ltc.simple_rtn.plot(ax=ax[1], color = 'black')
ax[1].set(xlabel = 'Date',
          ylabel = 'Simple Returns (%)')

ltc.log_rtn.plot(ax=ax[2], color = 'black')
ax[2].set(xlabel = 'Date',
          ylabel = 'Log Returns (%)')
plt.show()
```