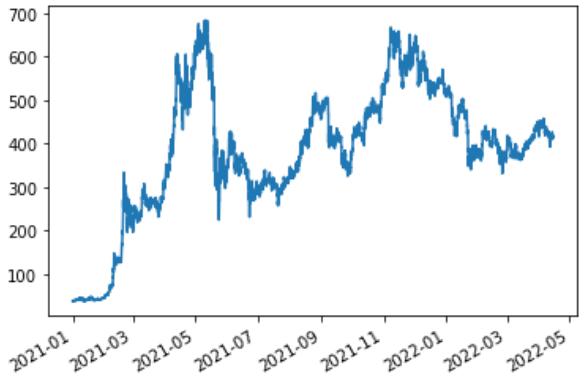
Rafay Zeeshan

Constructing Cointegrated Cryptocurrency Portfolios for Statistical Arbitrage

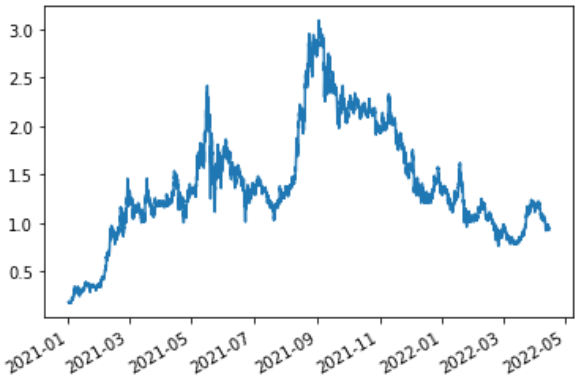
This report comprises the code outputs and figures for a python code carried out to create a cointegrated portfolio using four cryptocurrencies in order to possibly reap profit from an arbitrage opportunity.

(1) The four cryptocurrencies selected for formulating the cointegrated portfolio are Solana (SOL), Cardano (ADA), Ripple (XRP), and Binance Coin (BNB). The hourly data for the past year is acquired from Binance.com and a linear regression is run over the closing price data for the four cryptocurrencies. The graphs for the closing prices of each cryptocurrency show a random walk.

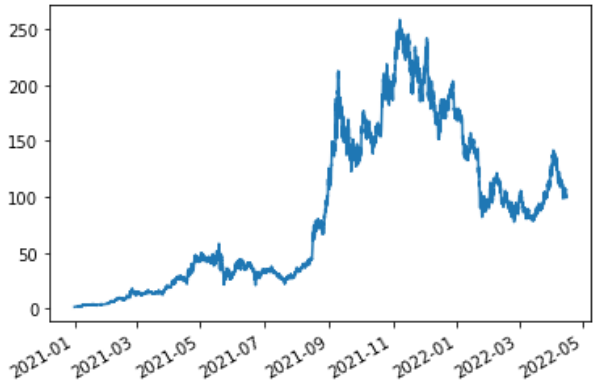
BNB:



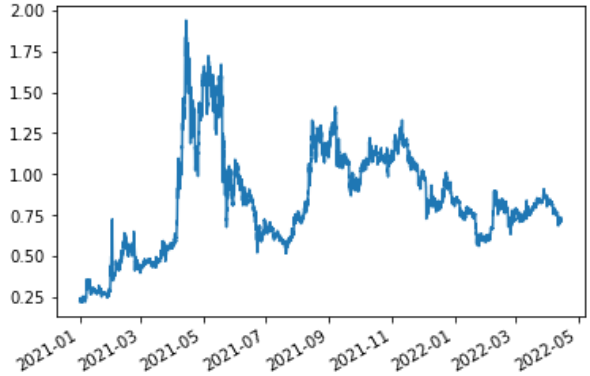
ADA:



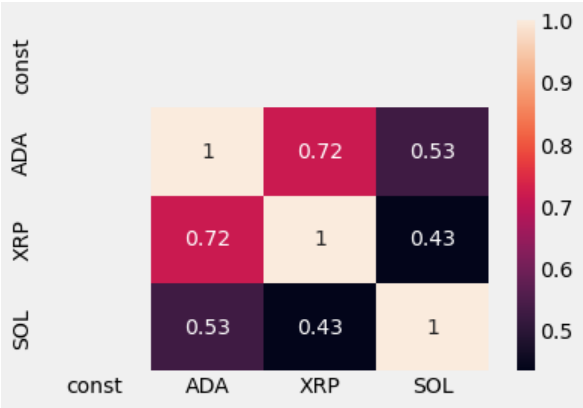
SOL:



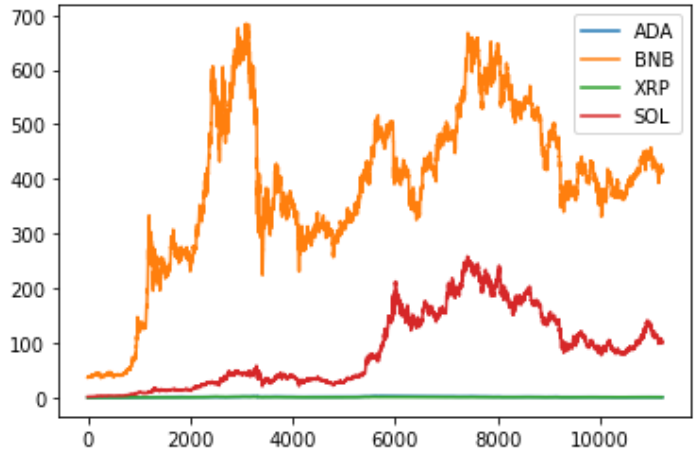
XRP:



There is no observable colinearity amongst the four coins, so the linear regression could be run on the data.



The following graph plots the percentage change in the closing prices of the four cryptocurrencies in a combined data frame with respect to time.



The timeseries were divided by placing 90% of the data for training the model. The OLS regression statistics give a high R-squared value of 0.834 and p-values of 0.00 % for all independent variables, thus making the analysis statistically significant.

OLS Regression Results

Dep. Variable:	BNB	R-squared:	0.834
Model:	OLS	Adj. R-squared:	0.834
Method:	Least Squares	F-statistic:	1.694e+04
Date:	Fri, 15 Apr 2022	Prob (F-statistic):	0.00
Time:	01:12:11	Log-Likelihood:	-56417.
No. Observations:	10090	AIC:	1.128e+05
Df Residuals:	10086	BIC:	1.129e+05
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	55.2707	1.877	29.447	0.000	51.591	58.950
ADA	-41.9226	1.724	-24.318	0.000	-45.302	-38.543
XRP	359.1464	2.861	125.537	0.000	353.539	364.754
SOL	0.9958	0.011	93.575	0.000	0.975	1.017

Omnibus:	1359.828	Durbin-Watson:	0.005
Prob(Omnibus):	0.000	Jarque-Bera (JB):	772.964
Skew:	-0.541	Prob(JB):	1.42e-168
Kurtosis:	2.183	Cond. No.	551.

This regression gives the following equation:

BNB_t = 55.2707 - 41.9226*ADA_t + 359.1464*XRP_t + 0.9958*SOL_t + e

The spread is then given by the following equation:

SPREAD_t = BNB_t + 41.9226*ADA_t - 359.1464*XRP_t - 0.9958*SOL_t

This equation depicts the portfolio to be used for trading and calculating the profit/loss.

To prove the stationarity of the residual terms, the augmented Dickey-Fuller (ADF) test, the Phillips–Peron (PP) test, and the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test was performed. Through each of the ADF, PP, and KPSS tests, we initially fail to confirm that the residuals are stationary. For ADF and PP, the test failed to reject the null hypothesis as the p-values were higher than 5% wherewith the null hypothesis was *Ho: Presence of unit root in observable price series.*

For the KPSS test again, the p-values for significant, hence rejecting the null hypothesis. The null hypothesis for KPSS is *Ho: Unit root does not exist in observable price series*. This showed that through both the tests, closing prices for all four coins in the dataset are non-stationary. However, the ADF and KPSS tests were then run on the closing price differences for each coin. After one round of differencing, all the four coins' closing prices indicated stationarity.

These test results show the stationarity of the coin prices after differencing, indicating stationarity:

```
1.SOL
KPSS Test Results
Test Statistic            0.178939
p-value                   0.100000
Lags Used                13.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
dtype: float64

2.XRP
KPSS Test Results
Test Statistic           0.124272
p-value                  0.100000
Lags Used                5.000000
Critical Value (10%)     0.347000
Critical Value (5%)      0.463000
Critical Value (2.5%)    0.574000
Critical Value (1%)      0.739000
dtype: float64

3.ADA
KPSS Test Results
Test Statistic            0.273069
p-value                   0.100000
Lags Used                11.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
dtype: float64

4.BNB
KPSS Test Results
Test Statistic           0.17813
p-value                  0.10000
```

```
1.SOL
ADF Statistic: -18.404594
p-value: 0.000000
Residuals are stationary
Critical values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
None

2.XRP
ADF Statistic: -18.234756
p-value: 0.000000
Residuals are stationary
Critical values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
None

3.ADA
ADF Statistic: -23.090715
p-value: 0.000000
Residuals are stationary
Critical values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
None

4.BNB
ADF Statistic: -17.435529
p-value: 0.000000
Residuals are stationary
```

Then was made the combined dataframe, dfcombined, for the four coins' prices and the price differences. The test data frame was created with data from the initial test dataframe on which the linear regression was performed. Using the Spread equation from the initial OLS regression model, the Spread column was added to the dataframe.

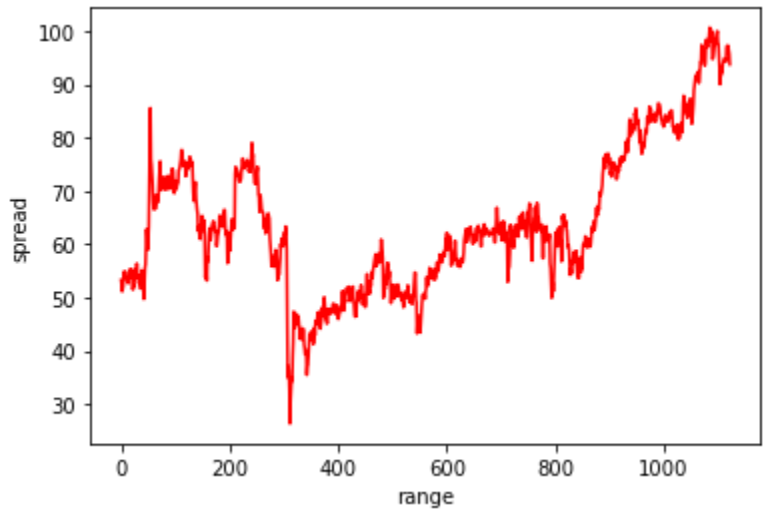
Combined dataframe with the Spread:

	const	ADA	XRP	SOL	BNB	Spread
10091	1.0	0.863	0.7333	87.01	366.2	52.375485
10092	1.0	0.863	0.7345	87.15	365.5	51.105013
10093	1.0	0.869	0.7360	87.74	366.8	51.530499
10094	1.0	0.881	0.7443	89.00	371.1	52.098065
10095	1.0	0.878	0.7386	88.83	371.7	54.788829
...	...	...	...	...	...	...
11208	1.0	0.928	0.7145	99.57	414.0	97.147632
11209	1.0	0.932	0.7176	100.18	414.9	96.494517
11210	1.0	0.939	0.7226	100.74	416.2	95.734692
11211	1.0	0.929	0.7237	100.54	414.4	93.318922
11212	1.0	0.932	0.7279	100.72	415.0	92.356988

Combined dataframe with the Spread and the price differences:

	ADA	BNB	XRP	SOL	SOL_diff	XRP_diff	ADA_diff	BNB_diff
11208	0.928	414.0	0.7145	99.57	0.54	0.0046	0.003	1.9
11209	0.932	414.9	0.7176	100.18	0.61	0.0031	0.004	0.9
11210	0.939	416.2	0.7226	100.74	0.56	0.0050	0.007	1.3
11211	0.929	414.4	0.7237	100.54	-0.20	0.0011	-0.010	-1.8
11212	0.932	415.0	0.7279	100.72	0.18	0.0042	0.003	0.6

The spread vs. range graph is shown as:

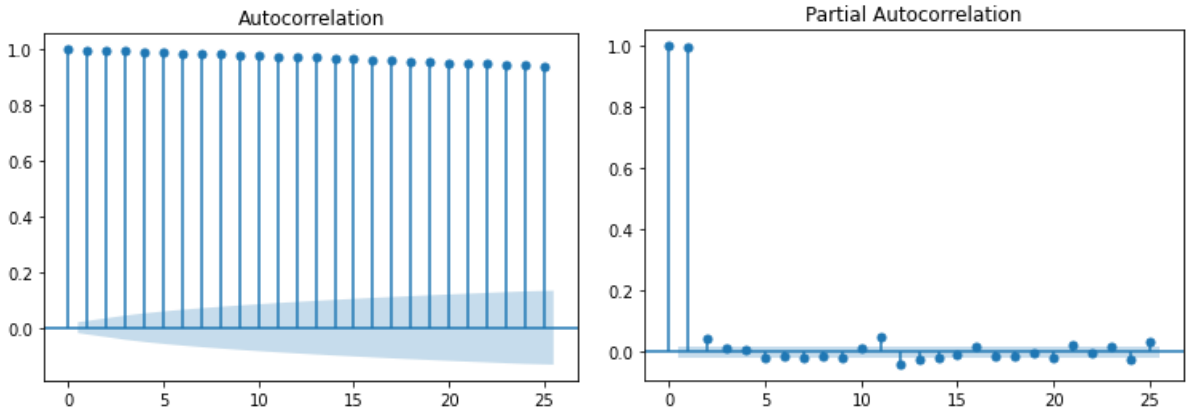


(2) Prediction

The prediction for the values of coin prices was made using the Autoregressive Integrated Moving Average (ARIMA) model. The ARIMA model was used to factor out the trend so the pricing forecast could be made accurately. The parameters used were:

1. number of autoregressive terms (AR order) = 1
2. number of nonseasonal differences (differencing order) =1
3. number of moving-average terms (MA order) = 0

The autocorrelation and partial autocorrelation were observed to determine how to define the subsequent ARIMA model.



ARIMA Model Results

Dep. Variable:	D.Spread	No. Observations:	10067
Model:	ARIMA(1, 1, 0)	Log Likelihood	-29201.500
Method:	css-mle	S.D. of innovations	4.401
Date:	Fri, 15 Apr 2022	AIC	58408.999
Time:	02:30:48	BIC	58430.650
Sample:	1	HQIC	58416.326

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0084	0.042	0.200	0.842	-0.074	0.091
ar.L1.D.Spread	-0.0448	0.010	-4.504	0.000	-0.064	-0.025

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	-22.2989	+0.0000j	22.2989	0.5000

Strategy

The strategy used for the calculating the arbitrage profit is using upper and lower bands given by μ +/- c*σ, around the mean and standard deviation of the Spread, and using the trading strategy of short-selling when the Spread at a time t is below the lower specified limit, and using the long position for the Spread increasing beyond the limit, i.e.

$$\text{Short position for: } S_t < \mu - c * \sigma$$
$$\text{Long position for: } S_t > \mu + c * \sigma$$

After using different values of "c" for code runs, it was observed that a "c*σ" value of **0.75σ** gave the highest profit margin i.e. **209,895.20955233 USD** on an initial investment of 100,000 USD, giving an **ROI of 209.9%** over the given period.