Implementation of Pairs Trading Strategy for Crypto Currency Market

Alisher Mirmanov

September 2025

1 Introduction

This research serves as a re-implementation of a popular statistical arbitrage trading technique called pairs trading strategy on the cryptocurency market. Mostly it follows the trading strategy described in [1] and [2].Like in [2], we look at crypto spots, however, exclude the USDT-marginated futures from the analysis for now[1].

Methodologically, our approach follows the standard pipeline used in equity pairs trading and adapted by [1, 2]: identify candidate pairs; test for a stable long-run relationship (e.g., via Engle–Granger residual stationarity); construct a spread and standardize it to obtain a trading signal; and execute threshold-based entries and exits while enforcing risk controls. While the focus is on a faithful reproduction of the original design, we document each implementation choice—data resampling and execution assumptions—so that any deviations from [1, 2] are transparent.

The central questions we ask are twofold. Firs, do the gains reported for equities and prior crypto studies persist on recent Binance spot data? Second, does the choice between arithmetic (non-logged) and logarithmic compound. By structuring the study around a single-pair allocation per period, we make these sensitivities visible in a way that multi-pair or dollar-neutral baskets can obscure.

2 Methodology

Before going through each step in the strategy, let us introduce the main steps of the method. Put simply, the main idea for the pairs trading strategy, is to find historically cointegrated pairs of assets (heuristically, closely correlated), and find arbitrage opportunities, i.e. find windows where the pairs veer off into different directions, to short one security and long the other.

For this project, we use the assumptions that one asset is linearly dependent on another, in other words, for the prices of two securities P^1 and P^2, the spread is defined to be:

$$S_t = P_t^1 - \beta P_t^2 + \alpha,$$

[1]In my case, even at the initial backtest, with only maker/taker + funding fees, and the levergae of 2x, the position was liquidated, so a more sophisticated approach might be needed, for instance Copula Based pairs trading also described in [2]

where α and β are the regression coefficients, and S_t are the residuals. The first step in the process is to identify the parameters α, β by regressing directly using the whole training period. This way, by constructing all possibilities of different pairs of assets[2]. Finally, one runs an Augmented Dickey-Fuller test to check is the series S_t is stationary, i.e. it passes the unit root test, using $1\% p-$value threshold. If the series is stationary, then the specific pair is called co integrated. The whole procedure is known commonly as the Engle-Granger cointegration test (EG test).

After the set of cointegratied pairs is collected, one needs to choosee a specific pair to trade during the period. To do that, one models the spread S_t as a Ornstein-Uhlebeck process. Using the mean reverting properties of OU one can defined a so called "half-life" of the series, that's related to the speed of mean reversion of the series, i.e. how fast it turns back to its long term mean. The pair is then chosen on the basis of the least half-life.

During the training period, robust trade signals must be constructed for a given pair. We use the popular Z-score to establish such signals.

2.1 Engle–Granger

After specifying the cointegration regression and computing the in-sample residuals S_t, the remaining task is to determine whether these residuals are stationary. Following [1], we consider both linear and non-linear cointegration diagnostics (e.g., the Augmented Dickey–Fuller and KSS tests). However, consistent with the evidence reported in [1], there is no significant advanteage of using non-linear cointegration tests in the crypto market, as such, in this study, we focus on the ADF test.

For context, recall that if several price series are non-stationary of the same integration order $I(d)$,[3] then a linear combination that is $I(0)$ is said to be *cointegrated*. The ADF test can be motivated from an autoregressive representation. Let (S_t) be an AR(p) process, given by

$$S_t \;=\; c \;+\; \phi_1 S_{t-1} \;+\; \cdots \;+\; \phi_p S_{t-p} \;+\; \varepsilon_t,$$

(ε_t) is the white noise.Without loss of ngenerality set $c = 0$. Rearranging the formula one achieves:

$$(1 - \phi_1 B - ... - \phi_p B^p) S_t = \epsilon_t,$$

where B is a backshift operator[4]. Stationarity requires that the polynomial $\Phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p$ have all roots outside the unit circle. A unit root corresponds to a factor $(1 - B)$ in $\Phi(B)$; rearranging yields the familiar ADF regression,

$$\Delta S_t \;=\; \alpha_0 \;+\; \rho\, S_{t-1} \;+\; \sum_{i=1}^{p-1} \gamma_i\, \Delta S_{t-i} \;+\; u_t,$$

where the inclusion of lagged differences ΔS_{t-i} controls for serial correlation in u_t. Testing $H_0 : \rho = 0$ is the unit-root test. As such, given two price series P_t^1 and P_t^2 on the formation window, estimate

$$P_t^1 \;=\; \alpha \;+\; \beta\, P_t^2 \;+\; \widehat{\varepsilon}_t \quad \text{(OLS on the training window)},$$

[2]Note that the pairs do not commute here, since regressing one asset against each other is not the same as doing it vice versa, since the resultant spreads might not both be stationary (or not), see more in Stationarity.

[3]Cointegration of order d means the series become stationary after differencing d times.

[4]$B^n S_t = S_{t-n}$

and then run the ADF regression on the residuals

$$\Delta\widehat{\varepsilon}_t \;=\; \alpha_0 \;+\; \rho\,\widehat{\varepsilon}_{t-1} \;+\; \sum_{i=1}^{p}\gamma_i\,\Delta\widehat{\varepsilon}_{t-i} \;+\; u_t.$$

The null $H_0 : \rho = 0$ states that $\widehat{\varepsilon}_t$ retains a unit root; rejection at conventional levels of 1% indicates that $\widehat{\varepsilon}_t$ is $I(0)$ and, hence, that (P^1, P^2) are cointegrated.

2.2 Modeling as Ornstein-Uhlebeck process

After all the cointegrated pairs are chosen, the spread pair is modeled using the Ohrenstein-Uhlenbeck Process, defined by:

$$dS_t = \theta(\mu - S_t)\,dt + \sigma\,dW_t,$$

where S_0 is known, and both $\theta, \mu > 0$, μ is the long term spread mean, θ is the mean reversion speed, and σ is the spread's instantaneous volatility. The general solution to OU process of this form is given by:

$$S_t = S_0 e^{-\theta t} + \mu(1 - e^{-\theta t}) + \sigma \int_0^t e^{-\theta(t-u)} dW_u,$$

with the expected value of the process at time t given by:

$$\mathbb{E}[S_t|s_0] = s_0 e^{-\theta t} + \mu(1 - e^{-\theta t})$$

Then we define the stocks half-life $t_{1/2}$, i.e. expected time for S_t to reach to reach average value between S_0 and μ

$$\mathbb{E}[S_{t_{1/2}}|s_0] - \mu = \frac{s_0 - \mu}{2}$$

that is, by combining the two equations:

$$\mathbb{E}[S_{t_{1/2}}|s_0] - \mu = s_0 e^{-\theta t_{1/2}} + \mu(1 - e^{-\theta t_{1/2}}) - \mu = e^{-\theta t_{1/2}}(s_0 - \mu)$$

$$e^{-\theta t_{1/2}}(s_0 - \mu) = \frac{s_0 - \mu}{2} \implies t_{1/2} = \frac{\ln 2}{\theta}.$$

It remains to calibrate the parameter θ using MLE/OLS, i.e. the "speed of mean-reversion". To estimate theta, one first obtains the finite-difference formula of the OU process:

$$\begin{aligned} S_t - S_{t-1} &= \theta(\mu - S_{t-1})\Delta t - \sigma\sqrt{\Delta t} Z_t \\ &= \theta\mu\Delta t - \theta S_{t-1}\Delta t - \sigma\sqrt{\Delta t} Z_t \end{aligned}$$

then by constructing the regression problem by setting $S_t - S_{t-1} := y = a + bx + \epsilon$, the OLS gives

$$\hat{\theta} = \frac{-\hat{b}}{\Delta t},$$

which we use to calculate the $t_{1/2}$.

2.3 Signal Generation via Z-Score

Once the cointegrated pair with the lowest half-life is chosen, commencing the trading period, it remains to generate the robust signals. Let S_t denote the spread constructed during training, $S_t = P_t^1 - \beta P_t^2 + \alpha$, with (α, β) fixed from the training OLS. In order to make the spread scalable, normalize the spread by calculating Z-score as follows

$$Z_t = \frac{S_t - \bar{S}_t}{s_t}.$$

where for a given lookback window N,$\bar{S}_t$ and $\bar{\sigma}_t$ are the rolling mean and standard deviation given by

$$\bar{S}_t = \frac{1}{N}\sum_{i=0}^{N-1} S_{t-i}, \qquad \bar{\sigma}_t = \sqrt{\frac{1}{N-1}\sum_{i=0}^{N-1}\left(S_{t-i} - \bar{S}_t\right)^2}.$$

2.3.1 Choise of optimal lookback window

A robust rule is to tie N to the estimated OU half-life $t_{1/2}$. Follwing [1], define the exponential moving average for a series recursively as

$$Z_t = \begin{cases} S_1, & t = 1 \\ \lambda \cdot S_t + (1-\lambda)\cdot Z_{t-1}, & t > 1 \end{cases}$$

where $\lambda \in [0, 1]$ gives degree of weighting decrease, with higher λ discounting older observations faster.Then, one can expand

$$\text{EMA}_{\text{Now}} = \lambda\left[s_1 + (1-\lambda)s_2 + (1-\lambda)^2 s_3 + (1-\lambda)^3 s_4 + \cdots\right],$$

where s_1 is the spread value at the moment, s_2 is the spread value of one unit of measurement before, etc. Stopping the series after t terms and defining over the whole EMA, one has

$$\frac{\lambda(1-\lambda)^t\left[1 + (1-\lambda) + (1-\lambda)^2 + \cdots\right]}{\lambda\left[1 + (1-\lambda) + (1-\lambda)^2 + \cdots\right]} = (1-\lambda)^t$$

To have half of the weights, we set the above fraction to 0.5, and obtain

$$t_{1/2} = \frac{\ln(0.5)}{\ln(1-\lambda)} \implies t_{1/2} \approx \frac{\ln(0.5)}{-\lambda} = \frac{\ln(2)}{\lambda},$$

Since $\lambda \to 0$ as $N \to \infty$, we know $\ln(1-\lambda)$ approaches $-\lambda$ as $N \to \infty$. Then using the formula for half life one has $\lambda = \hat{\theta}$. Then the lookback window is given by

$$\lambda_{EMA} = \frac{2}{N_{SMA}+1}, \quad \Rightarrow \quad N_{SMA} = \frac{2}{\lambda_{EMA}} - 1$$

2.3.2 Trading rules

$$\text{Enter long at } t \text{ if } Z_{t-1} \le -2 \text{ and } Z_t > -2, \tag{1}$$

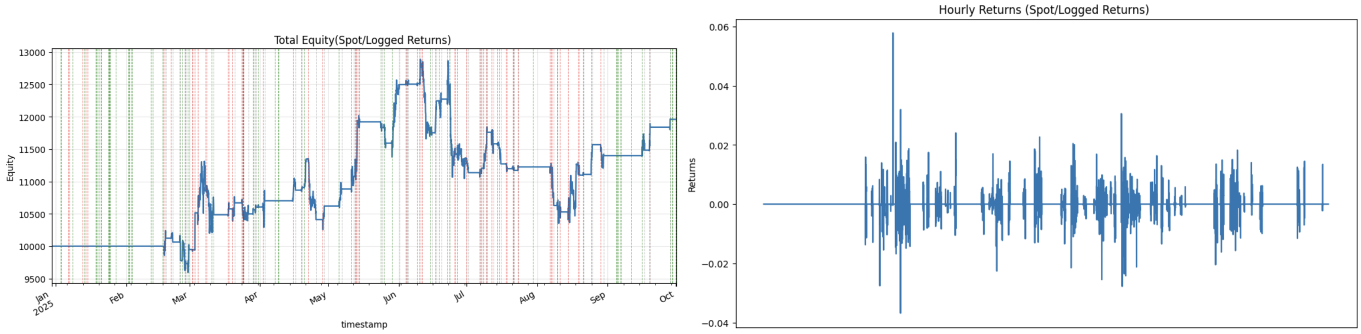
$$\text{Exit long at } t \text{ if } Z_{t-1} \ge -1 \text{ and } Z_t < -1, \tag{2}$$

$$\text{Enter short at } t \text{ if } Z_{t-1} \ge +2 \text{ and } Z_t < +2, \tag{3}$$

$$\text{Exit short at } t \text{ if } Z_{t-1} \le +1 \text{ and } Z_t > +1. \tag{4}$$

3 Data Source and Trade Implementation

For the back testing, we use the Binance API to pull the hourly close data for the period of 01/10/2024 – 01/10/2025. For this investigation. As mentioned previously, for now, only the spot data is analysed. Note, that since spots are



(a) Logged Equity

(b) Logged Returns

Figure 1: Equity and returns (log scale).

quoted in USDT, the P&L calculation is also done in USDT, with all the other calculations. In this scenario, we only consider the maker and taker fees and min notional constraint[5]. As such, this model might be overly simplistic, since we don't account for bid/ask spreads (making shorting somewhat unrealistic since binance allows spot shorting via margin trading) and only use the close hourly data, and don't consider partial fills, which altogether might result in exaggerated returns during the backtest.

The trading is done under the following scenario: the first three months are used to establish the model, i.e. establish the stationary pairs, and choose the best pair, with calculated look-back window and speed of mean reversion. The best pair is the then traded for a following week using the Z-score that is calculated each hour.

4 Empirical Results

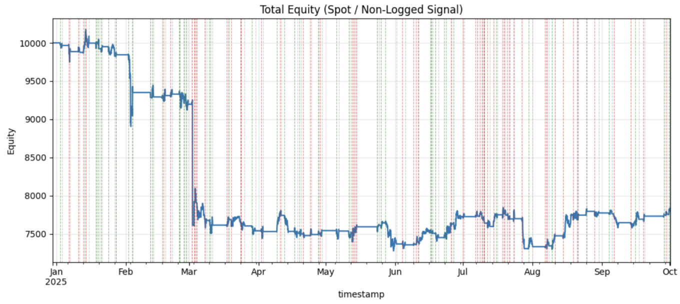
To assess the overall performance of the strategy, we use the following metrics: final equity for the whole period of 52 weeks, average hourly return, annual return for the whole period. To analyse the volatility of the strategy we use 2 metrics: Sharpe ratio and the maximum drawdown. Note that in our calculation, since the data is hour binned, to annualise the Sharpe we use the factor of $\sqrt{24 * 365}$, and not the traditional $\sqrt{24 * 252}$, since binance allows to trade each day of the year.

The figures below demonstrate the obtained results during the backtesting.

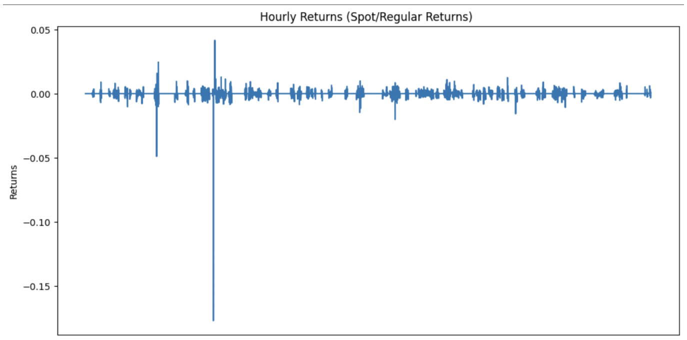
	Regular	Log
Starting Equity	10,000.00	10,000.00
Final Equity	7,808.82	11,959.04
Annual return (%)	-21.91	19.59
Sharpe	-1.09	1.05
Max Drawdown (%)	-28.4	-19.66

The hourly returns and equity curve for the logged values The hourly returns and equity curve for the non-logged values

[5]Note that, potentially, in further analysis, for the futures, one also needs to consider funding fees (every 8 hours) into the P&L calculation, liquidation, and quantity and price constraints in the model



(a) Non-Logged Equity



(b) Non-Logged Returns

Figure 2: Equity and returns.

The results above are off the results provided in [1], and as such the model did not perform as well as expected. The log-price variant, however, delivers a positive outcome over the test period, finishing at 11,959.04 (an annual return of 19.59%) with a Sharpe of 1.05 and a maximum drawdown of 19.66%, which is still considered a successful and profitable investment strategy. In contrast, the non-logged (regular price) is unprofitable, ending at 7,808.82 (annual return 21.91%) with a Sharpe of 1.09 and a deeper maximum drawdown of 28.4% (whilch clearly cost the majority of the loss over the period, from which it didn't recover). Overall, this suggests that the log specification provides more robust risk-adjusted performance and better downside control in this backtest. That said, again, these results should be interpreted cautiously. The implementation follows the general approach in [1] and [2], but the backtest remains idealized: it does not model slippage, partial fills, or constraints and costs associated with shorting on spot markets (borrow availability and borrow fees).

5 Conclusion

We re-implemented an Engle–Granger–OU pairs strategy on Binance USDT spot markets following the work in [1]. Over 01/10/2024–01/10/2025, the **log** variant was profitable (final equity 11,959.04; Sharpe 1.05; max drawdown –19.66%), while the **regular** variant was unprofitable (final equity 7,808.82; Sharpe –1.09; max drawdown –28.4%). These results suggest that residual mean-reversion may remain exploitable in crypto, though performance is sensitive to the compounding/price specification.

That said, effect size and significance must be judged against realistic frictions and model risk. Future work should (i) incorporate slippage/partial fills and liquidity screens, (ii) correct for multiple testing when selecting weekly pairs.

As for further model complications and improvements, one could also consider wider thresholds for signals, to incorporate the fat tails nature of the crypto market. An additional promising extension is to adapt signals to regimes. Two complementary approaches:

- **Markov switching:** Fit a two-state HMM to S_t or Z_t (mean-reverting vs. trending). Use tighter/wider entry bands—or skip trades—in the trending state; use standard bands in the mean-reverting state.
- **Eigenvalue filter:** Track the largest eigenvalue $\lambda_1(t)$ of a rolling correlation matrix of returns. High λ_1 (market

moving together) $\Rightarrow$ raise thresholds or cut size; low λ_1 $\Rightarrow$ trade normally.

Together with EG–OU selection, these filters reduce false entries in trend/high-correlation periods while preserving alpha in calm, mean-reverting regimes.

References

[1] Masood Tadi and Irina Kortchemski. Evaluation of dynamic cointegration-based pairs trading strategy in the cryptocurrency market. *Studies in Economics and Finance*, 38(5):1054–1075, July 2021.

[2] Masood Tadi and Jiří Witzany. Copula-based trading of cointegrated cryptocurrency pairs, 2023.