MAN456- Business Analysis

Homework - 2 and 3

Ahmet Sayan
21903426

Preliminary Information:

Numbers in the report and model differ slightly from each other because of the dataset I used when I started to write this report, but the code has some different hours of data as well. I didn't know the hours I wrote this report, causing slight deviations in the numbers in the code.

Question 1:

a. I created **get_data_binance** function using our previous Binance examples to get the crypto data. At first try, I imported all the crypto examples that exist in Binance to decide which ones I would choose to continue. I checked their covariance with BTC and chose the coins with high covariance. I used SOL, BNB, TRX, NEXO, PAXG, BCH cryptocurrencies to use in regression and imported data between *2023-04-24 - 2025–04-23* (last 2 years of data).

b. I split my data with a 0.8 train ratio. I didn't split randomly to model integrity instead, I am using data from *2024-11-28* to *2025-04-23* for the test dataset.

c. I used SOL, BNB, TRX, NEXO, PAXG, BCH cryptocurrencies for explanatory variables and BTC as the dependent variable. By fitting the OLS model, I got the result below.

Model:	OLS	Adj. R-squared:	0.983
Dependent Variable:	BTCUSDT	AIC:	257087.8065
Date:	2025-04-23 17:14	BIC:	257140.6422
No. Observations:	14016	Log-Likelihood:	-1.2854e+05
Df Model:	6	F-statistic:	1.391e+05
Df Residuals:	14009	Prob (F-statistic):	0.00
R-squared:	0.983	Scale:	5.4117e+06

R-squared values are perfectly good. The model explains the variance almost perfectly in the training set. Also probability of the F-statistic is lower than 0.01, meaning our explanatory variables are effective in explaining the Dependent variable. I used 0.01 for significance to remain consistent with the article.

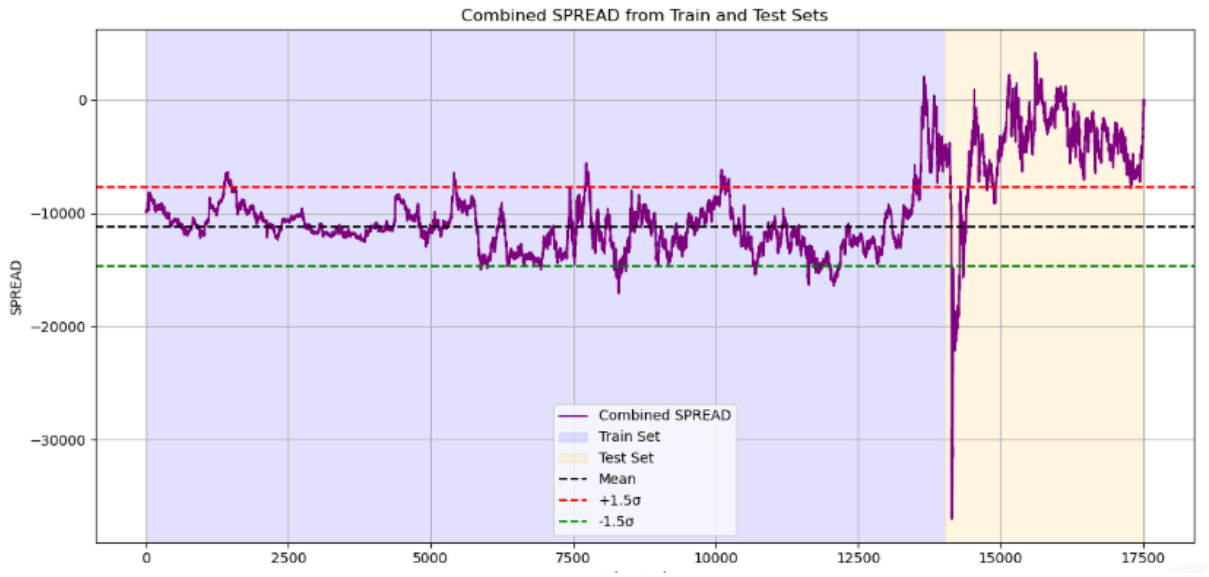
	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
const	-11178.3987	549.8636	-20.3294	0.0000	-12256.2046	-10100.5928
SOLUSDT	99.2061	1.5302	64.8308	0.0000	96.2067	102.2056
BNBUSDT	3.6178	0.6521	5.5481	0.0000	2.3396	4.8959
TRXUSDT	131483.5825	1934.5457	67.9661	0.0000	127691.6150	135275.5500
NEXOUSDT	15360.4128	283.2720	54.2250	0.0000	14805.1619	15915.6637
PAXGUSDT	6.4644	0.3153	20.5003	0.0000	5.8463	7.0824
BCHUSDT	17.8177	0.3580	49.7762	0.0000	17.1161	18.5193

Also, every independent variable has a p-value lower than zero, showing they are significant for the model.

Even though the model performs very well in the training set, it's not that good for explaining the test set. The crypto coins started to diverge after 2024-11. The reason for this divergence might be the US elections or interest rate changes from central banks. Therefore, test scores and prediction power are not good for this model

```
Regression Performance on Test Set:
MAE  : 7641.8076
MSE  : 65776381.1624
RMSE : 8110.2639
R²   : -0.1346
```

Also, you can observe the divergence from the spread differences at the beginning and the end (my test set).



d. For this part, I have applied the tests ADF, PP, KPSS from the article for each Crypto, the First Difference of Cryptos and the constant.

Test	BTC	SOL	BNB	TRX	NEXO	PAXG	BCH
ADF	0.99	0.95	0.92	0.99	0.84	0.97	0.57
PP	0.99	0.94	0.91	0.99	0.81	0.96	0.51
KPSS	0.01	0.01	0.01	0.01	0.01	0.01	0.01

Test	△BTC	△SOL	△BNB	△TRX	△NEXO	△PAXG	△BCH
ADF	0.000	0.00	0.00	0.00	0.00	0.00	0.00
PP	0.000	0.000	0.000	0.000	0.000	0.000	0.000
KPSS	0.10	0.10	0.10	0.10	0.10	0.10	0.10

Test	ADF	PP	KPSS
Residuals	0.0015	0.0013	0.01

All cryptocurrencies are non-stationary, but they become stationary after the first derivative, similar to the article. Additionally, I used the same test to prove my residuals are stationary, which also aligns with the article's model.

Question 2

First, I calculated SPREAD with the coefficients I found from the OSL model.

$$SPREAD_t = BTC_t - 99.21SOL_t - 3.62\times BNB_t - 131,483.58\times TRX_t - 15,360.41\times NEXO_t - 6.46\times PAXG_t - 17.82\times BCH_t$$

SPREAD over time looks like the line graph above for the training set. I will use this SPREAD to create a trading strategy. Strategy is taking a long position if SPREAD is below the Sigma value and taking a short position if SPREAD is above Sigma. If a position is already open and the SPREAD crosses the opposite threshold, the position is closed and reversed. What we are doing here is selling cryptos when they are high and expecting prices to decrease. When it decreases, we buy at back lower price. I have tried this strategy with 5 different sigma values. I started with 0 money for this testing.

Sigma	Total Value at the End	Number of Trades
0.50	70565.84	29
0.75	64076.83	19
1.00	45830.36	11
1.25	59289.53	11
1.50	43257.69	7

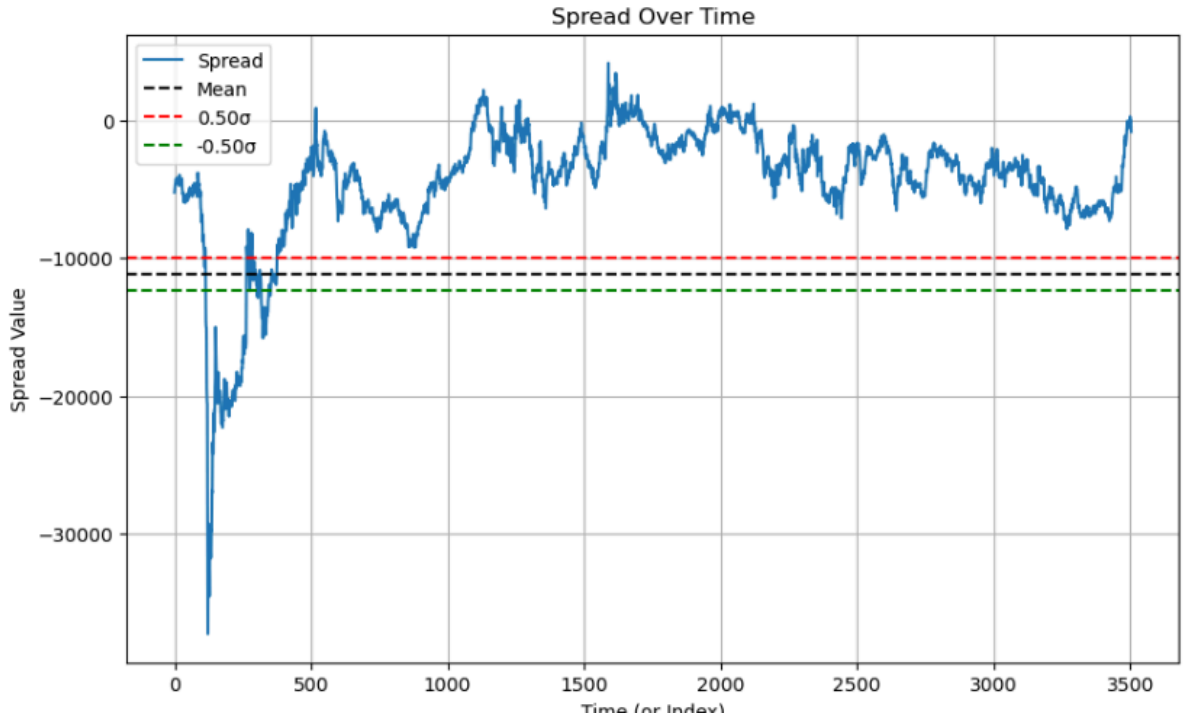
The strategy I previously mentioned will be applied with a sigma value of 0.50, as it yielded the highest Total Value at the end. Detailed statistics for each sigma value are presented in the table below.

Sigma	0.50	0.75	1.00	1.25	1.50
Net P&L	70565.85	64076.84	45830.37	59289.53	43257.70
Num Trades	29	19	11	11	7
Avg Trade P&L	2433.31	3372.47	4166.40	5389.96	6179.67
Max Profit	3264.92	4388.46	5362.68	6670.69	8025.96
Max Loss	-5593.00	-4905.46	-4414.65	-3319.38	-2676.29
Gross Profit	76158.85	68982.30	50245.02	62608.91	45933.99
Gross Loss	-5593.00	-4905.46	-4414.65	-3319.38	-2676.29
Profit Factor	13.62	14.06	11.38	18.86	17.16
% Win	96.55	94.74	90.91	90.91	85.71
% Loss	3.45	5.26	9.09	9.09	14.29
Max Drawdown	5593.00	4905.46	4414.65	3319.38	2676.29
Profit/Drawdown	12.62	13.06	10.38	17.86	16.16

Using the strategy in the test dataset

By using this strategy, I have found:

Sigma	Total Value	Number of Trades
0.50	19057.94	7

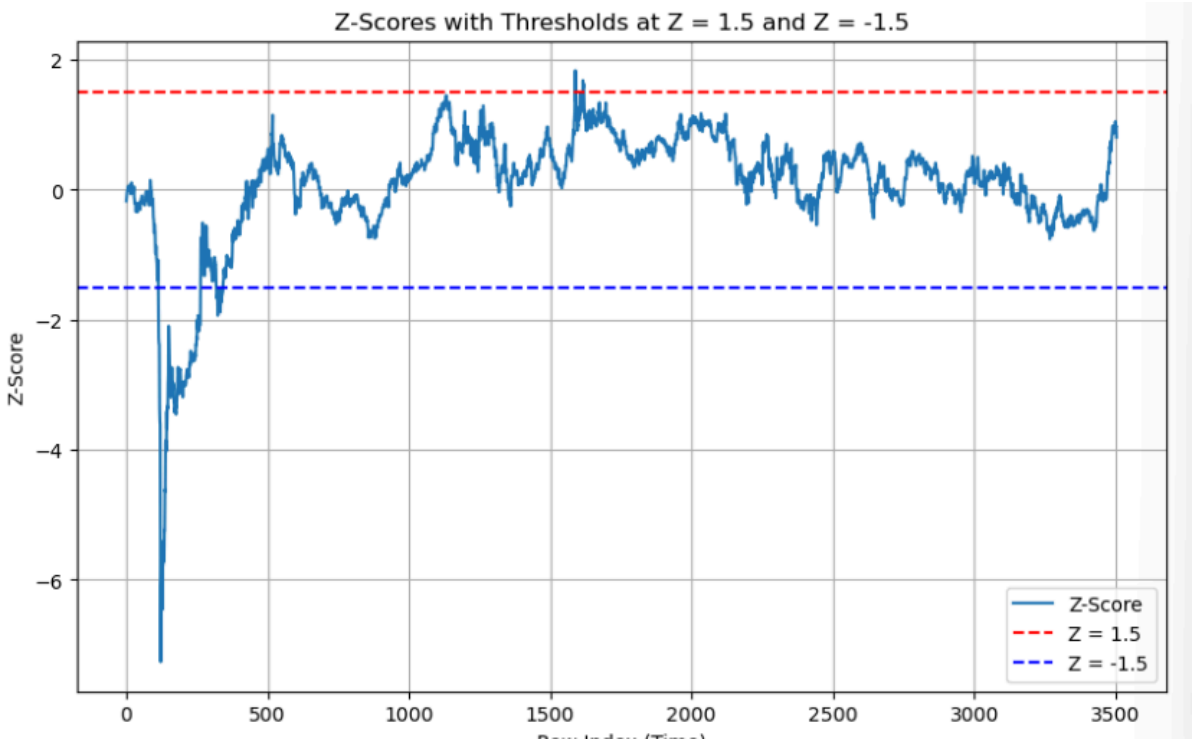


As I mentioned before, I observe high divergence at the test dataset time; therefore, this approach is not effective as I observed in the train dataset, but it's still profitable. However, it's possible to improve this model. Currently, sigma lines do not effectively capture the sudden divergence. But before the divergence, there are seven trades helping us to profit. Therefore, I will present my idea to make this model in the next bonus part.

Bonus Part

I previously observed divergence among cryptocurrencies, which caused problems during the testing phase. Therefore, I decided to develop a new model that updates the spread each time new data from the test set is observed, recalculating the spread dynamically. As a result, the boundaries now adjust based on new data instead of remaining fixed.

You can see the graph below showing the results after applying this updated model.



I successfully brought the Z-scores back within the boundaries whenever they exceeded the sigma level during the testing phase previously. However, to apply the strategy I need SPREAD to hit the boundaries; therefore, I tried the sigma values again to see which one is most profitable.

Sigma	Total Value at the End	Number of Trades
0.25	22679.74	25
0.50	3322.32	13
1.00	-2346.86	5
1.25	4481.9	5
1.50	9785.81	5

The new strategy performs best with a 0.25 sigma value. This model works fine with low sigma values and adapt to the new data better than previous one. I believe this model works better with May's data as well. Therefore, I would prefer you to use this model for the bonus section.