UCLA Anderson School of Management

MGMTMFE 431 Final Project

Statistical Arbitrage Pair Trading Using Cointegration, Mean Reversion, and Bayesian Optimization

A Full Pipeline: Data → Pair Selection → Mean Reversion Modeling → Backtesting → Optimization

Team 3: *Hiu Chun (Sunny) Chan, Saurabh Kulkarni, Shaoxiong Yuan, Weiyi Wang*

Instructor: *Prof. Mikhail Chernov*

Presentation Date: *01st December 2025*

Meet Our Team

Hiu Chun (Sunny) Chan

Saurabh Kulkarni

Shaoxiong Yuan

Weiyi Wang

Presentation Overview

- Main Execution Framework
- Data and Universes
- Pair Selection Methodology
- Mean Reversion Modeling (Spread, Z-score, κ, Half-life)
- Strategy Design (Entry/Exit Logic)
- Backtest Engine
- Bayesian Optimization (Hyperparameter Tuning)
- Results and Evaluation
- Conclusion and Future Extensions

Main Execution Framework

- Get the S&P 500 constituent list for the past ten years, from January 1, 2016 to the present
- Identify the stocks that appear consistently in the years 2016 to 2020
- Sort the stable constituent list by market capitalization and select the top fifty to keep the pair universe manageable
- Generate all possible pairs from these fifty stocks (fifty choose two combinations)
- Run cointegration and mean reversion analysis on all candidate pairs using data from 2016 to 2020
- Select the best performing pair based on cointegration strength and mean reversion statistics (COST and NEE)
- Run out of sample backtesting with data from 2021 to the present to evaluate robustness

Universe & Data: *Data Description and Universe Selection*

- **Universe:** S&P 500 constituents (past 10 years)
- **Data used:** daily closing prices from Refinitiv API
- **Time periods:**
  - Pair selection window: **2016–2020**
  - Out-of-sample test window: **2021–2025**
- **Data cleaning:**
  - Aligned dates
  - Removed missing data

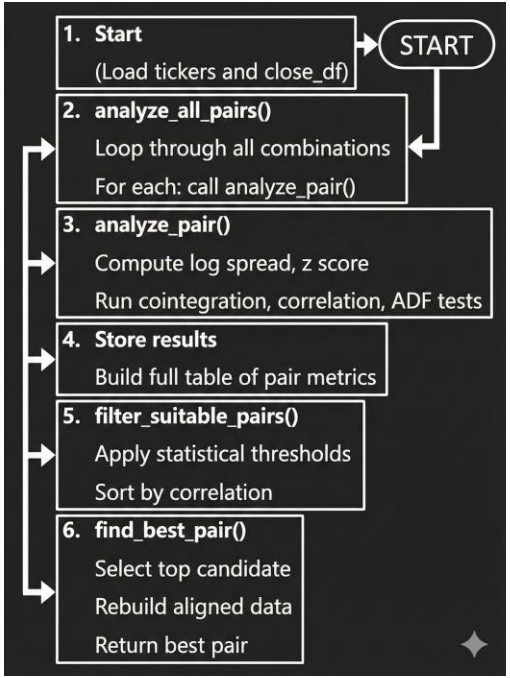
```python
try:
    stocklist = pd.read_pickle('spx_constituents.pkl')
except:
    rd.open_session()
    stocklist = rd.get_data(
            universe = ['.SPX'],
            fields = [f'TR.IndexConstituentRIC(SDate={year}-01-01)'
    )
    rd.close_session()
    stocklist = stocklist.iloc[:, 1:]
    stocklist.columns = range(startyear, endyear + 1)
    stocklist.to_pickle('spx_constituents.pkl')
display(stocklist)
```

	Instrument	Date	Price Close	Total Return	Bid Price	Ask Price	Volume	Company Market Capitalization
0	A.N	2016-01-04	40.69	-0.563581	40.68	40.69	1167911	13516033514.1
1	A.N	2016-01-05	40.55	-0.563581	40.55	40.56	664634	13469529589.5
2	A.N	2016-01-06	40.73	-1.800254	40.72	40.73	687280	13529320349.700001
3	A.N	2016-01-07	39.0	2.34375	39.0	39.01	922848	12954664710.0
4	A.N	2016-01-08	38.59	1.553719	38.58	38.59	1445601	12818474645.1
...	...	...	...	...	...	...	...	...
1510970	ZTS.N	2025-11-21	122.06	-2.833479	122.05	122.11	1275064	53791013700.839996
1510971	ZTS.N	2025-11-24	122.87	-1.460481	122.85	122.86	2222244	54147975204.18
1510972	ZTS.N	2025-11-25	127.89	-3.0404	127.83	127.84	1104654	56360255138.459999
1510973	ZTS.N	2025-11-26	127.69	0.020829	127.65	127.66	840080	56272116495.660004
1510974	ZTS.N	2025-11-28	128.18	1.565475	128.19	128.29	576786	56488056170.519997

1510975 rows × 8 columns

Pair Generation: *Generating All Candidate Pairs*

- Generate all combinations:
- $n = 50 \Rightarrow \binom{50}{2} = \frac{50 \times 49}{2} = 1225 \text{ pairs}$
- For each pair:
  - compute log-spread
  - evaluate statistical properties



Spread Construction: *Log Spread Calculation*

- $\text{log–spread}(t) = \ln\left(P_{1,t}\right) - \ln\left(P_{2,t}\right)$
- Why log-spread?
  - scale invariance
  - Linearizable
  - matches OU assumptions
- Spread properties used:
  - Mean
  - Standard Deviation
  - Z-score

Statistical Test: *Filters for Pair Selection*

- **Correlation Test**
  - Remove unrelated pairs
- **Engle-Granger Cointegration Test**
  - Cointegration p-value < 0.05
- **ADF Test (Augmented Dickey-Fuller)**
  - Check if spread is stationary
  - ADF p-value < 0.05
- **Z-score Stability**
  - Mean near zero
  - Std Dev stable

```
suitable_pairs
```

	pair	cointegration_pvalue	correlation	adf_pvalue	mean_zscore	std_zscore
1194	COST.OQ-NEE.N	0.000033	0.981836	0.000037	1.354490e-16	1.0
386	V.N-ABT.N	0.017675	0.979157	0.008362	4.966462e-16	1.0
717	HD.N-ACN.N	0.012285	0.974604	0.001419	-2.821854e-16	1.0
1125	TMO.N-COST.OQ	0.042744	0.973373	0.030303	0.000000e+00	1.0
1063	ABT.N-COST.OQ	0.014473	0.972814	0.014070	-1.354490e-15	1.0
178	GOOG.OQ-ACN.N	0.024877	0.968758	0.011432	2.116390e-15	1.0
481	WMT.N-NEE.N	0.006351	0.968013	0.024297	0.000000e+00	1.0
157	GOOG.OQ-HD.N	0.004690	0.966925	0.000105	1.061017e-15	1.0
223	GOOGL.OQ-ACN.N	0.030324	0.966247	0.012596	-1.535088e-15	1.0
996	CRM.N-ABT.N	0.017416	0.966218	0.010661	1.805986e-16	1.0
1000	CRM.N-TMO.N	0.012775	0.966185	0.005151	-3.386224e-16	1.0
202	GOOGL.OQ-HD.N	0.006665	0.964882	0.000151	-2.827497e-15	1.0
1168	ACN.N-TXN.OQ	0.037947	0.964510	0.017933	-2.596105e-16	1.0
188	GOOG.OQ-TXN.OQ	0.020956	0.963145	0.003289	-5.643707e-17	1.0
233	GOOGL.OQ-TXN.OQ	0.023452	0.961440	0.005975	-9.029932e-17	1.0
1167	ACN.N-NEE.N	0.046583	0.960400	0.016647	1.580238e-15	1.0
934	NKE.N-TMO.N	0.015294	0.960028	0.039051	4.063469e-16	1.0
472	WMT.N-ACN.N	0.002515	0.957184	0.000379	-2.043022e-15	1.0
451	WMT.N-HD.N	0.004850	0.952266	0.001446	-2.900865e-15	1.0
154	GOOG.OQ-UNH.N	0.012083	0.949880	0.003247	8.126938e-16	1.0
187	GOOG.OQ-NEE.N	0.018269	0.949382	0.018934	1.805986e-16	1.0
199	GOOGL.OQ-UNH.N	0.013197	0.948438	0.005775	7.675442e-16	1.0
150	GOOG.OQ-WMT.N	0.006755	0.943948	0.001518	5.700144e-16	1.0
195	GOOGL.OQ-WMT.N	0.008571	0.940099	0.001471	5.869455e-16	1.0

Pair Chosen: *Selected Best Pair Based on 2016–2020 Data*

- **Final Best Pair:** COST.OQ-NEE.N
- **Metrics:**
  - Correlation ≈ 0.98
  - Cointegration p-value ≈ 3.2e–5
  - ADF p-value ≈ 3.7e–5
  - z-score std = 1

```
suitable_pairs
```

	pair	cointegration_pvalue	correlation	adf_pvalue	mean_zscore	std_zscore
1194	COST.OQ-NEE.N	0.000033	0.981836	0.000037	1.354490e-16	1.0
386	V.N-ABT.N	0.017675	0.979157	0.008362	4.966462e-16	1.0
717	HD.N-ACN.N	0.012285	0.974604	0.001419	-2.821854e-16	1.0
1125	TMO.N-COST.OQ	0.042744	0.973373	0.030303	0.000000e+00	1.0
1063	ABT.N-COST.OQ	0.014473	0.972814	0.014070	-1.354490e-15	1.0

Estimating Mean-Reversion Strength: Kappa and Half-Life

Regress Δspread on lagged spread:

$$\Delta X_t = \alpha + \beta X_{t-1} + \varepsilon_t$$

Then

$$\kappa = -\beta$$

$$\text{Half-Life} = \frac{\ln(2)}{\kappa}$$

Interpretations:

- Smaller half-life → spreads revert quickly
- Signals how long we should hold a position

Strategy Overview: *Pair Trading Strategy Logic*

- **Entry Condition:**
  - If z-score < –threshold → Long spread
  - If z-score > +threshold → Short spread

- **Exit Conditions:**
  - Spread reverts to mean
  - Max holding time = half-life × holding_time_factor
  - Stop-loss triggered

- **Position Sizing:**
  - Dollar-neutral: equal capital on each leg

Backtesting Framework: *Backtrader Engine*

- **Created** `BacktestEngine` **class to:**
  - add OHLCV data
  - inject parameters
  - compute metrics
  - plot equity & drawdown
  - store trade logs
- **Metrics computed:**
  - Annualized Sharpe
  - Max Drawdown
  - Total Return
  - Win Rate
  - Daily statistics

Hyperparameter Tuning with Bayesian

- **Goal:** maximize Sharpe ratio
- **Parameters tuned:**
  - Lookback window
  - Entry z-threshold
  - Stop-loss factor
  - Holding-time factor
- **Why Bayesian?**
  - More sample-efficient than grid search
  - Handles non-linear objective surfaces
  - Reduces runtime

Optimization Results: *Best Parameters for COST-NEE Pair*

- **Goal:** maximize Sharpe ratio
- **Best Parameters:**
  - Lookback = 28
  - Entry threshold ≈ 1.50
  - Stop-loss factor ≈ 2.98
  - Holding time factor ≈ 1.61
  - Half-life ≈ 20.511 days
  - Best Sharpe: 1.58

```
**************************************************
Stock Pair for Pair Trading:
Stock 1: COST.OQ
Stock 2: NEE.N
Kappa: 0.0338
Half-Life: 20.51
**************************************************
Optimization progress: 10.00%
Optimization progress: 20.00%
Optimization progress: 30.00%
Optimization progress: 40.00%
Optimization progress: 50.00%
Optimization progress: 60.00%
Optimization progress: 70.00%
Optimization progress: 80.00%
Optimization progress: 90.00%
Optimization progress: 100.00%
**************************************************

Best parameters found:
lookback: 28
entry_threshold: 1.5011681487615216
stoploss_factor: 2.9883173389368265
holding_time_factor: 1.6174815096277166
half_life: 20.51113498530895
Best Sharpe Ratio: 1.5805
**************************************************
**************************************************
```

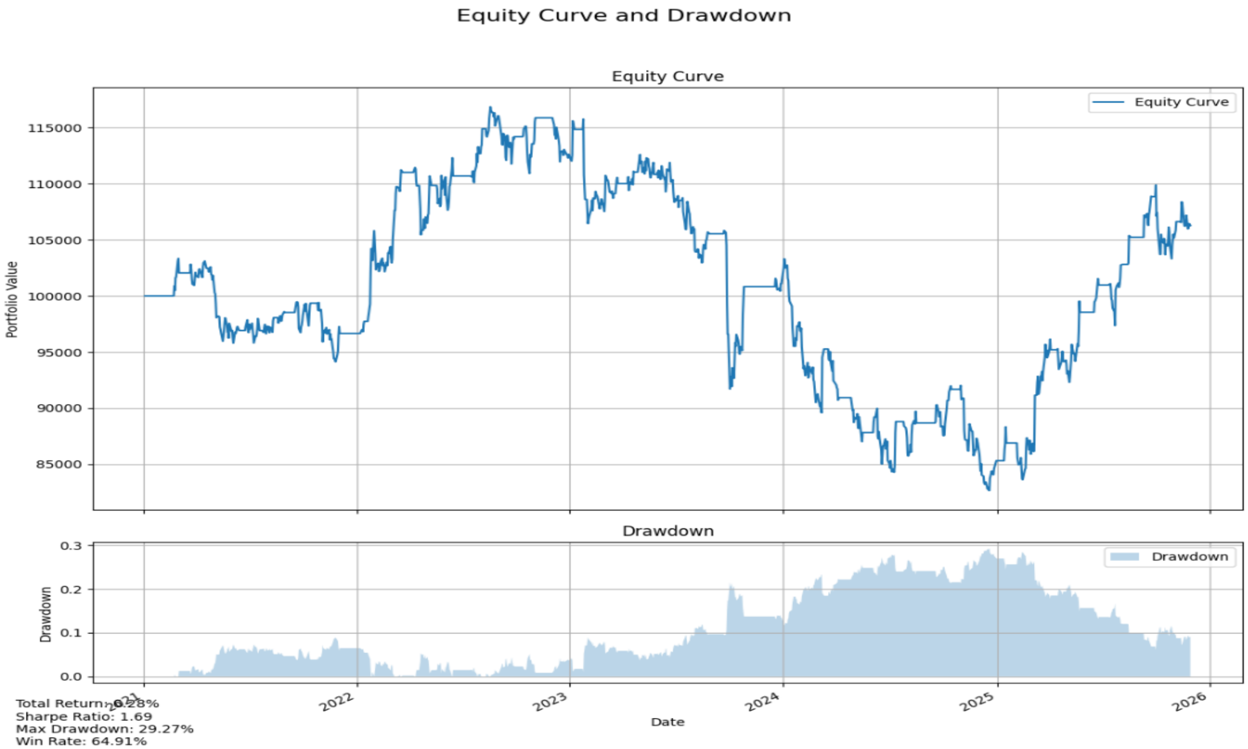
```
best_pair
```


```
{'stock1': 'COST.OQ',
 'stock2': 'NEE.N',
 'correlation': 0.9818364490056924,
 'cointegration_pvalue': 3.265657524541857e-05,
 'adf_pvalue': 3.7401307474117534e-05,
 'mean_zscore': 1.3544897266276732e-16,
 'std_zscore': 0.9999999999999999,
 'log_spread': Date
 2016-01-04    1.730378
 2016-01-05    1.723115
 2016-01-06    1.717075
 2016-01-07    1.689495
 2016-01-08    1.667357
                 ...
 2020-12-24    1.559039
 2020-12-28    1.569878
 2020-12-29    1.576463
 2020-12-30    1.573424
 2020-12-31    1.563424
 Length: 1259, dtype: Float64,
 'z_score': Date
 2016-01-04    2.828359
 2016-01-05    2.715355
 2016-01-06     2.62137
 2016-01-07    2.192276
 2016-01-08    1.847836
                 ...
 2020-12-24    0.162545
 2020-12-28    0.331184
 2020-12-29    0.433645
 2020-12-30     0.38636
 2020-12-31    0.230774
 Length: 1259, dtype: Float64}
```

Final Backtest Results:

Backtest Results:

Metric	Value
Initial Capital	100000
Final Portfolio Value	106277
Total Return (%)	6.28
Sharpe Ratio	1.69
Max Drawdown (%)	29.27
Total Trades	57
Winning Trades	37
Losing Trades	20
Win Rate (%)	64.91
Mean Daily Return (%)	0.09
Std Dev of Daily Return (%)	0.81



Trade History:

Entry Date	Exit Date	Days Held	PnL	PnL %	Entry Price 1	Entry Price 2	Exit Price 1	Exit Price 2	Exit Type	Exit Reason
2021-02-19	2021-03-02	7	3343.78	3.34%	346.88	78.1	321.16	75.54	EXIT	Spread Reversion
2021-03-19	2021-04-16	19	1062.75	1.04%	321.6	70.84	362.48	80.94	EXIT	Spread Reversion
2021-04-21	2021-05-06	11	-4406.06	-4.30%	365.77	77.97	374.25	74.01	STOP-LOSS	Stop-Loss Hit
2021-05-07	2021-06-11	24	-906.65	-0.92%	375.77	74.53	373.34	73.49	EXIT	Spread Reversion
2021-06-23	2021-07-16	16	1071.69	1.11%	383.25	73.35	401.25	77.92	EXIT	Spread Reversion
2021-07-22	2021-08-11	14	880.03	0.91%	408.26	76.12	434.42	82.37	EXIT	Spread Reversion
2021-08-18	2021-08-30	8	530.7	0.54%	436.29	84.42	445.79	83.95	EXIT	Spread Reversion
2021-09-16	2021-10-13	19	650.68	0.66%	453.01	84	435.4	80.19	EXIT	Spread Reversion
2021-10-25	2021-11-22	20	-4337.39	-4.37%	479.2	84.29	527.65	87.36	STOP-LOSS	Stop-Loss Hit
2021-11-23	2021-12-02	6	2845.19	3.01%	533.14	87.31	513.83	88.62	EXIT	Spread Reversion
2022-01-06	2022-01-13	5	829.2	0.86%	537.58	85.77	505.39	84.9	EXIT	Spread Reversion
2022-01-19	2022-01-25	4	4426.98	4.53%	479.26	82.59	466.71	75.1	EXIT	Spread Reversion
2022-01-26	2022-03-04	26	898.03	0.86%	472.72	72.64	513.82	80.21	EXIT	Spread Reversion
2022-03-07	2022-03-18	9	3594.53	3.34%	516.77	84.18	548.87	82.37	EXIT	Spread Reversion
2022-04-07	2022-04-21	9	-5558.14	-5.01%	594.53	87.08	578.58	76.18	STOP-LOSS	Stop-Loss Hit
2022-04-22	2022-05-06	10	4837.61	4.57%	558.99	73.95	492.17	72.47	EXIT	Spread Reversion
2022-05-17	2022-06-14	19	2465.27	2.24%	479.56	72.02	444.69	71.48	EXIT	Spread Reversion
2022-07-14	2022-08-03	14	2759.54	2.49%	500.56	80.13	534.85	86.46	EXIT	Spread Reversion
2022-08-09	2022-09-27	34	-936.59	-0.82%	523.91	88.92	467.66	81.08	EXIT	Max Holding Time
2022-10-11	2022-11-02	16	1341.68	1.17%	461.52	76.03	472.76	77.28	EXIT	Spread Reversion
2022-12-01	2023-01-06	24	-275.45	-0.24%	492.66	84.83	472.13	83.65	EXIT	Spread Reversion
2023-01-20	2023-01-26	4	-5074.23	-4.42%	469.43	81.82	487.22	76.4	STOP-LOSS	Stop-Loss Hit
2023-01-27	2023-03-03	24	2137.51	1.97%	492.1	75.58	464.69	73.83	EXIT	Spread Reversion
2023-03-06	2023-03-23	13	251.36	0.23%	472.34	74.4	476.91	73.81	EXIT	Spread Reversion

Backtesting Other Pairs

- Built generic function `run_pair_trading()`
- Randomly selects new pairs from suitable list
- Runs full pipeline:
  - kappa/half-life
  - Bayesian optimization
  - Backtest
- Example best-performing alternative pair:
  - ACN.N-TXN.OQ
  - Returns: 16.5%
  - Sharpe: 1.66
  - Win Rate: 90%
  - DD: 8%

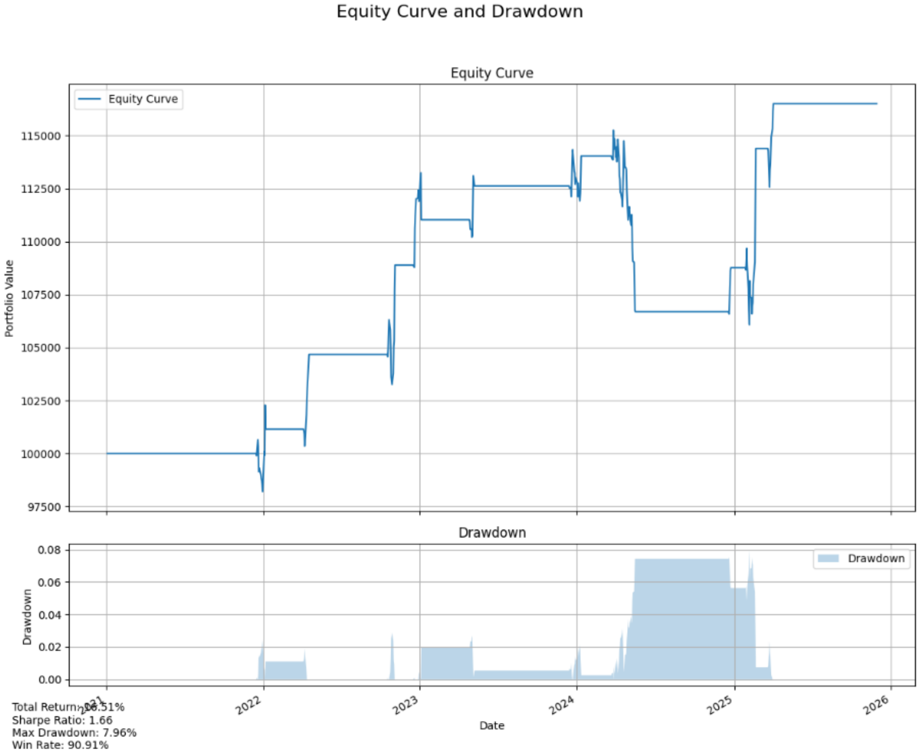
```
**************************************************
Stock Pair for Pair Trading:
Stock 1: ACN.N
Stock 2: TXN.OQ
Kappa: 0.0108
Half-Life: 64.37
**************************************************
Optimization progress: 2.00%
Optimization progress: 4.00%
Optimization progress: 6.00%
Optimization progress: 8.00%
Optimization progress: 10.00%
Optimization progress: 12.00%
Optimization progress: 14.00%
Optimization progress: 94.00%
Optimization progress: 96.00%
Optimization progress: 98.00%
Optimization progress: 100.00%
**************************************************

Best parameters found:
lookback: 21
entry_threshold: 2.741391854702842
stoploss_factor: 1.821012225329017
holding_time_factor: 1.8555487192017546
half_life: 64.36527148096044
Best Sharpe Ratio: 1.6551
**************************************************
**************************************************
```

Backtesting Results

Backtest Results:

Metric	Value
Initial Capital	100000
Final Portfolio Value	116512
Total Return (%)	16.51
Sharpe Ratio	1.66
Max Drawdown (%)	7.96
Total Trades	11
Winning Trades	10
Losing Trades	1
Win Rate (%)	90.91
Mean Daily Return (%)	0.04
Std Dev of Daily Return (%)	0.34



Key Findings : *Insights & Interpretations*

Observations:

- Pair selection crucial: strong cointegration → stable PnL
- Mean-reverting pairs outperform trending ones
- Bayesian optimization significantly improves Sharpe
- Some pairs (ACN–TXN) far outperform others
- Strategy robust to parameter variation

Limitations & Model Risks

- Assumes stable cointegration, but may break under regime change
- Transaction costs & slippage not deeply modeled
- Mean-reversion assumptions may fail
- Overnight gaps influence spread behavior
- OU process is a simplification

Future Improvements

- Expand to multi pair trading
- Use Johansen cointegration test
- Add volatility-adjusted position sizing
- Try dynamic z-thresholds (rolling optimization)
- Incorporate machine learning for spread forecasting
- Multi-pair portfolio optimization
- Incorporate slippage models

Conclusion

- Built a complete stat-arb pipeline:
  - Pair selection
  - OU-based mean reversion modeling
  - Backtesting
  - Bayesian optimization
- Combined econometrics and empirically validated profitability & robustness with rigorous out of sample testing

Thank You

Modeling Mean Reversion:

Ornstein–Uhlenbeck Process for Spread Modeling

Spread modeled by OU process:

$$dX_t = \kappa(\mu - X_t)\,dt + \sigma\,dW_t$$

Where:

- κ (kappa) = speed of mean reversion
- μ = long-term mean
- half-life = time to revert halfway to μ

Why OU?

- Standard model in statistical arbitrage
- Provides mathematically consistent mean-reversion
- Gives interpretable metrics: speed & stability

Motivation: *Why Pair Trading & Statistical Arbitrage?*

- Market-neutral strategy → profit from relative mispricing
- Lower market risk vs. directional strategies
- Rich mathematical foundation (cointegration, OU process)
- Cointegration tests and mean reversion modeling
- Bayesian optimization for parameter tuning
- Python based implementation using Backtrader, Statsmodels, and skopt