

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

SCHOOL OF
DATA SCIENCE
數據科學學院

Alpha Quest Algorithmic Trading Engine

Group 5 Members: ZHANG Feifan 223040185
ZHENG Xiaotong 223040208
LI Xutao 223040183
DAI Jingxian 223040201

Contents

01 Data Preprocess & Infrastructure
Data Source, Class and Data Reading

02 Strategy-Algorithmic trading
Factor construction and order function

03 TCA Analysis
Cost Decomposition and Execution Quality

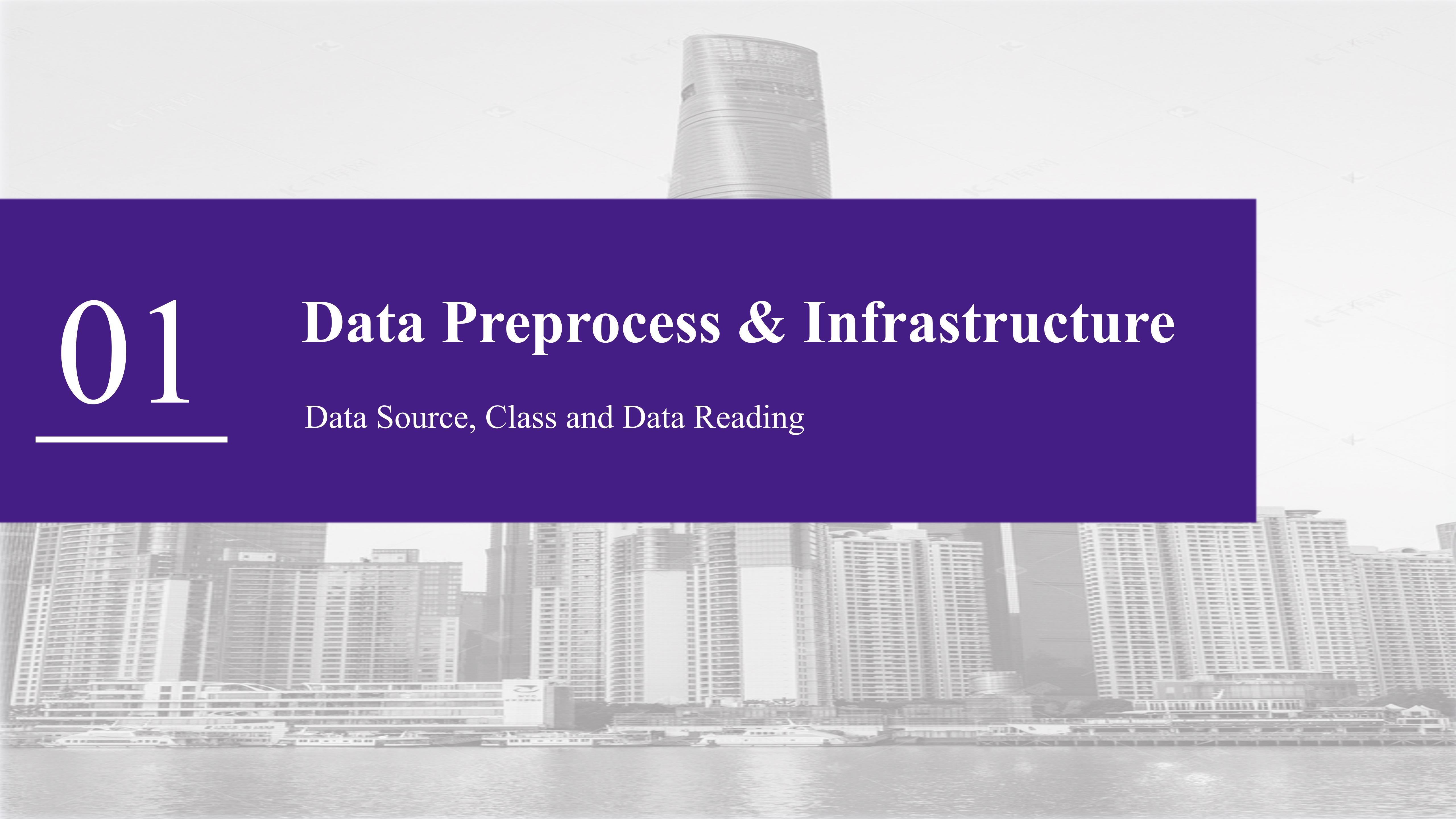
04 GUI System
Current Holdings & Trade Log & TCA Analysis

05 Results
Return & Sharpe & MDD

01

Data Preprocess & Infrastructure

Data Source, Class and Data Reading



Data Foundation - Data Source

Minute-level trading data of CSI 300 Index component stocks:

Trading_Date; Trading_Time; Open; High; Low; Close; Volume; Amount

Function:

- a) Providing Trading Decision Basis
- b) Tracking Index Performance

Security_Id	Trading_Date	Trading_Time	Open	High	Low	Close	Volume	Amount
000001.SZ	20240102	20240102093000000	9.39	9.39	9.39	9.39	6301	5,916,310.00
000001.SZ	20240102	20240102093100000	9.4	9.42	9.36	9.37	43232	40,598,438.00
000001.SZ	20240102	20240102093200000	9.37	9.38	9.37	9.38	7131	6,686,130.00

Weight of Constituents in the CSI 300 Index:

Con_Code; Weight

Function:

- a) Guiding Capital Allocation

Index_Code	Con_Code	Trade_Date	Weight
000300.SH	600519.SH	20240102	6.1717
000300.SH	601318.SH	20240102	2.4772

Infrastructure - RunParams Class & Portfolio Class

RunParams Class

Stores information about the parameters required for the algorithm to run

- Start date: 20240101
- End date: 20240412
- Run type: Backtest
- Commission rate: 0.0003
- Algo type: Vwap; Twap
- Order type: volume; price

```
1. class RunParams:  
2.     def __init__(self) -> None:  
3.         self.start_date = None # 起始日期  
4.         self.end_date = None # 终止日期  
5.         self.run_type = None # 运行类型 (回测/模拟)  
6.         self.commission_rate = None # 佣金费率  
7.         self.algo_type = None # 算法类型  
8.         self.algo_params = None # 算法参数  
9.         self.order_type = None # 订单类型
```

Portfolio Class

The role of this class in the algorithm is to manage and track information about the portfolio

- Current position
- Available cash
- Total value
- Returns
- Starting cash
- Position value

```
1. class Portfolio:  
2.     def __init__(self, starting_cash=5_000_000):  
3.         self.positions = {} # 记录当前持仓  
4.         self.available_cash = starting_cash # 记录当前可用资金  
5.         self.total_value = starting_cash # 记录当前总资产  
6.         self.returns = 0 # 记录当前收益  
7.         self.starting_cash = starting_cash # 记录初始资金  
8.         self.position_value = 0 # 记录当前持仓市值
```

Infrastructure - Context Class

Context Class

Provide a context in which we can store and track various information about the algorithm as it runs, including parameters, portfolios, current time, stock pools, benchmark indexes and transaction cost analysis etc.

- **Data record variable:** current_dt; previous_dt; Universe; benchmark
- **Universe:** The stock pool represents all the constituent stocks of CSI 300
- **Benchmark:** Representing the CSI 300 Index
- **Data output variable:** trade_book; target_book; position_book; total_value_book; tca_book

```
1. class Context:  
2.     def __init__(self, run_params, portfolio):  
3.         self.run_params = run_params # RunParams类  
4.         self.portfolio = portfolio # Portfolio类  
5.         self.current_dt = run_params.start_date # 记录当前时间  
6.         self.previous_dt = run_params.start_date # 记录上一个交易日  
7.         self.universe = None # 股票池  
8.         self.benchmark = None # 基准指数  
9.         self.trade_book = {} # 交易记录本  
10.        self.target_book = pd.DataFrame(columns=self.universe)  
11.        self.position_book = None # 交易持仓  
12.        # 新建一个pd用于存放total_value_book  
13.        self.total_value_book = pd.DataFrame(columns=['total_value'])  
14.        self.tca_book = {}
```

Infrastructure - Data Reading

Retrieve tick data for the corresponding stock code.

- Set the user's token to access financial data API.
- Retrieve real-time data from the TS library port based on the stock code.
- Data processing: Combine DATE and TIME into one datetime column.

Read minute-level stock data from the database.

- Retrieve data according to the database path
- Data format conversion
 - Convert strings and the time to datetime format.
- Set the time column as the index.
- Filter out data within the specified time range.

02

Strategy-Algorithmic trading

Factor construction and order function



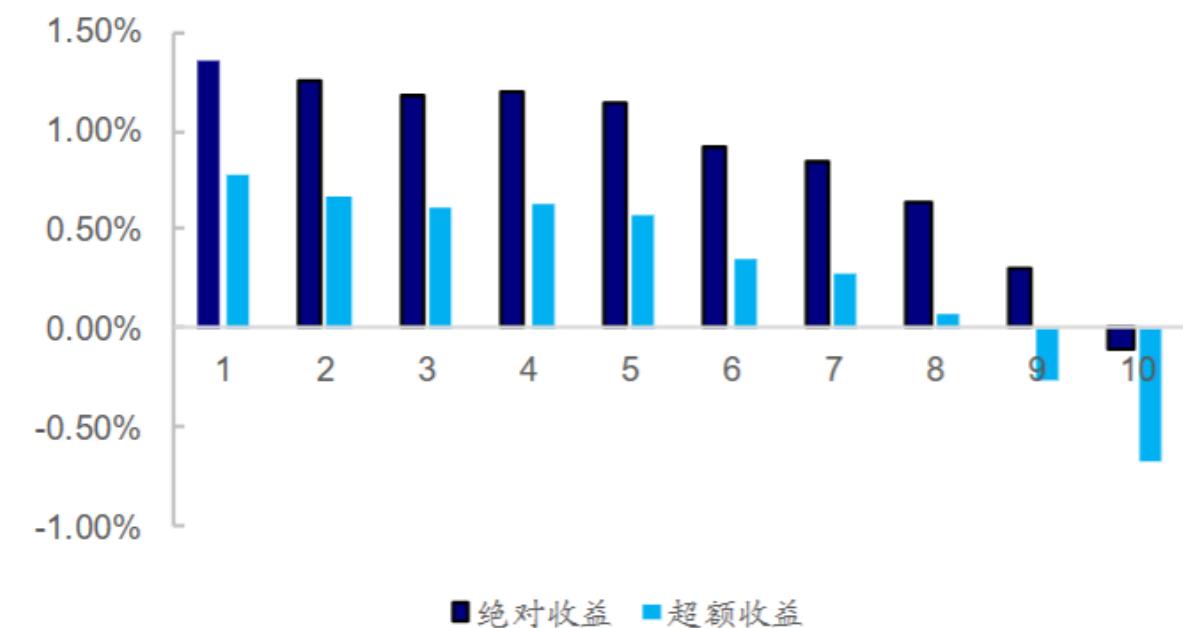
Strategy - Factor construction

Factor construction

Amaya et al. (2011) found a connection between intra-day high-order moments and future returns of stocks.¹

- Utilizing intra-day tick-by-tick data of stocks, they constructed a high-frequency realized variance indicator
- r_{ij} represents the intra-day 1-minute logarithmic return series of stock i ($j = 1, \dots, N+$), with factor values as the moving average of indicators over the past 20 days.

图2 股票高频偏度因子分组收益



资料来源: Wind, 海通证券研究所

- High frequency realized variance

$$RVar_i = \sum_{j=1}^N r_{ij}^2$$

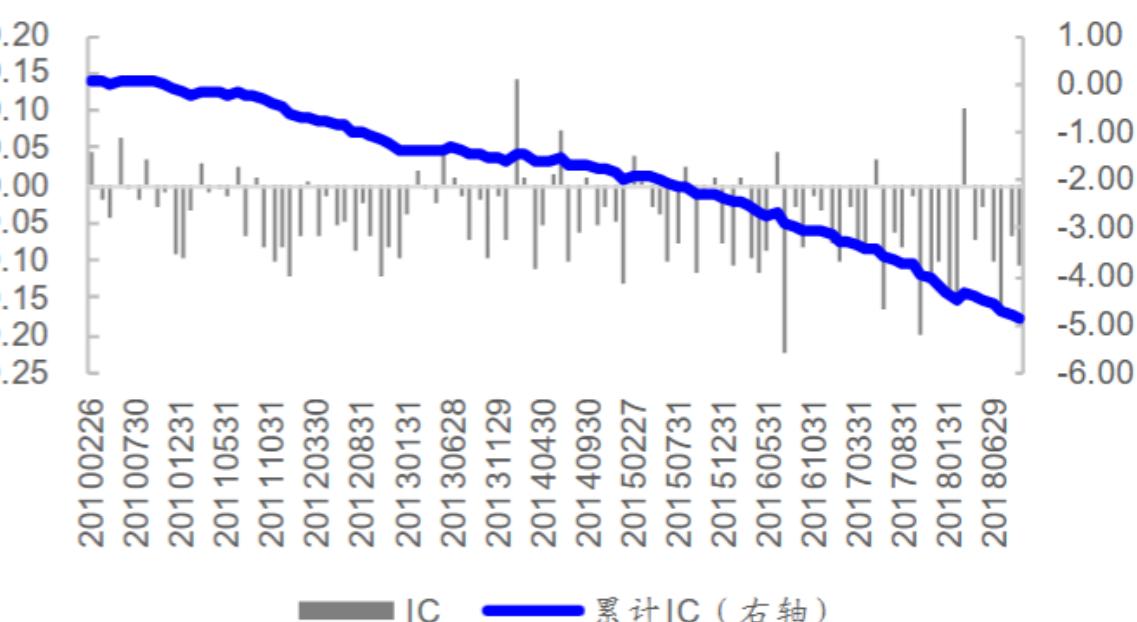
- High frequency realized skewness

$$RSkew_i = \frac{\sqrt{N} \sum_{j=1}^N r_{ij}^3}{RVar_i^{\frac{3}{2}}}$$

- High frequency realized kurtosis

$$RKurtosis_i = \frac{N \sum_{j=1}^N r_{ij}^4}{RVar_i^2}$$

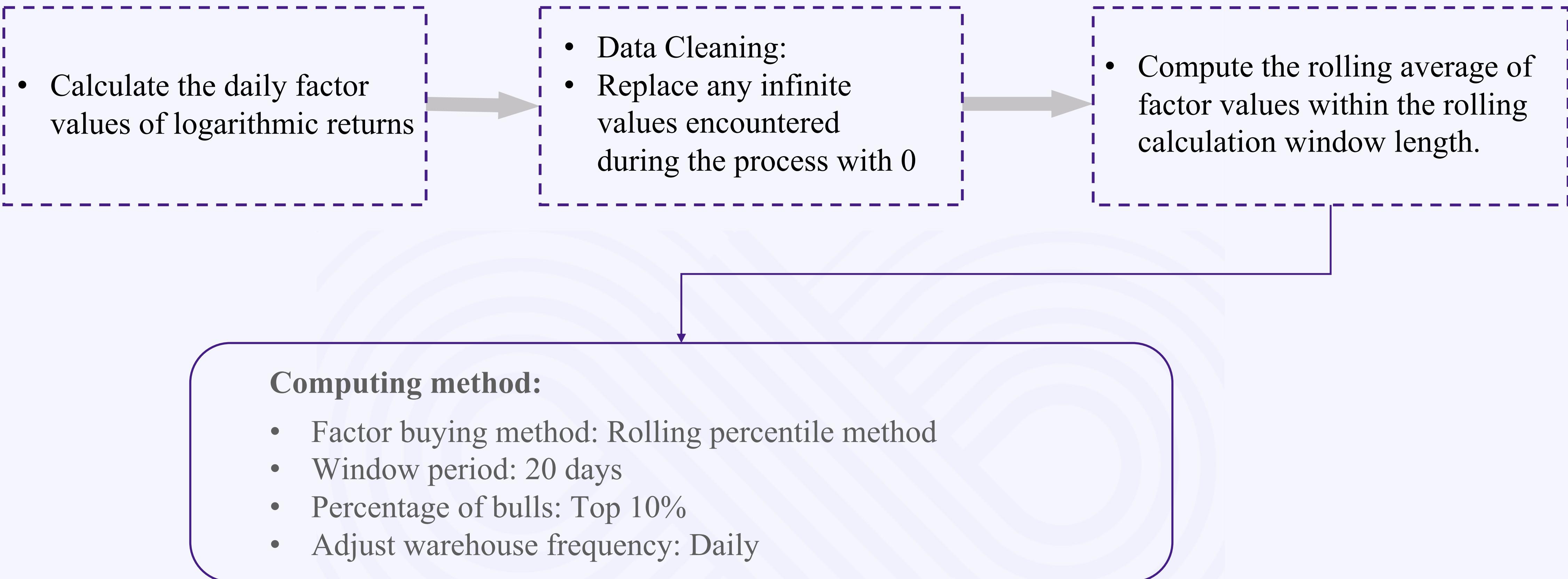
图3 股票高频偏度因子 IC



资料来源: Wind, 海通证券研究所

Strategy - Factor construction

Factor value calculation



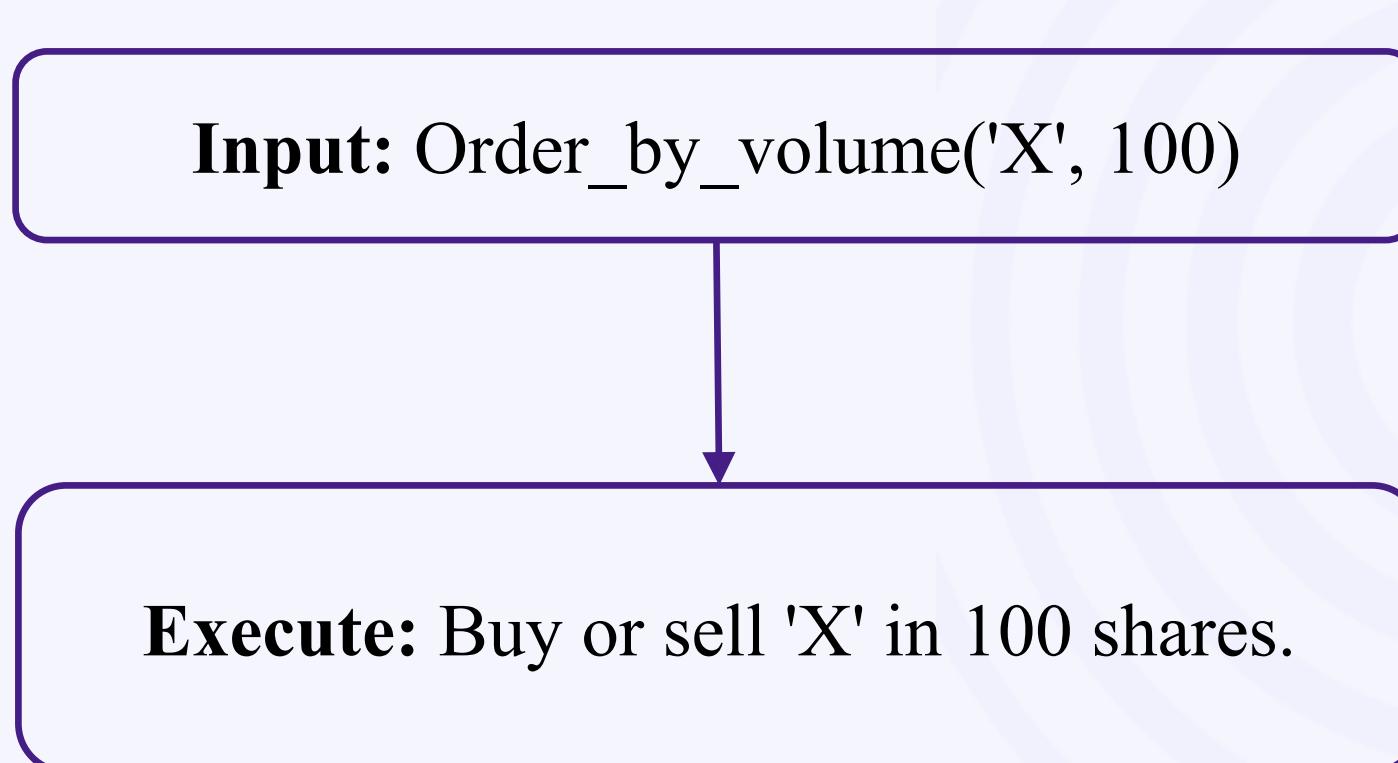
Strategy - Customize the order function

1

Order directly by share count

- *Order_by_volume(stock_code, volume)*

- Purpose: Based on a given ticker symbol and number of shares to at the current market price. perform a buy or sell operation

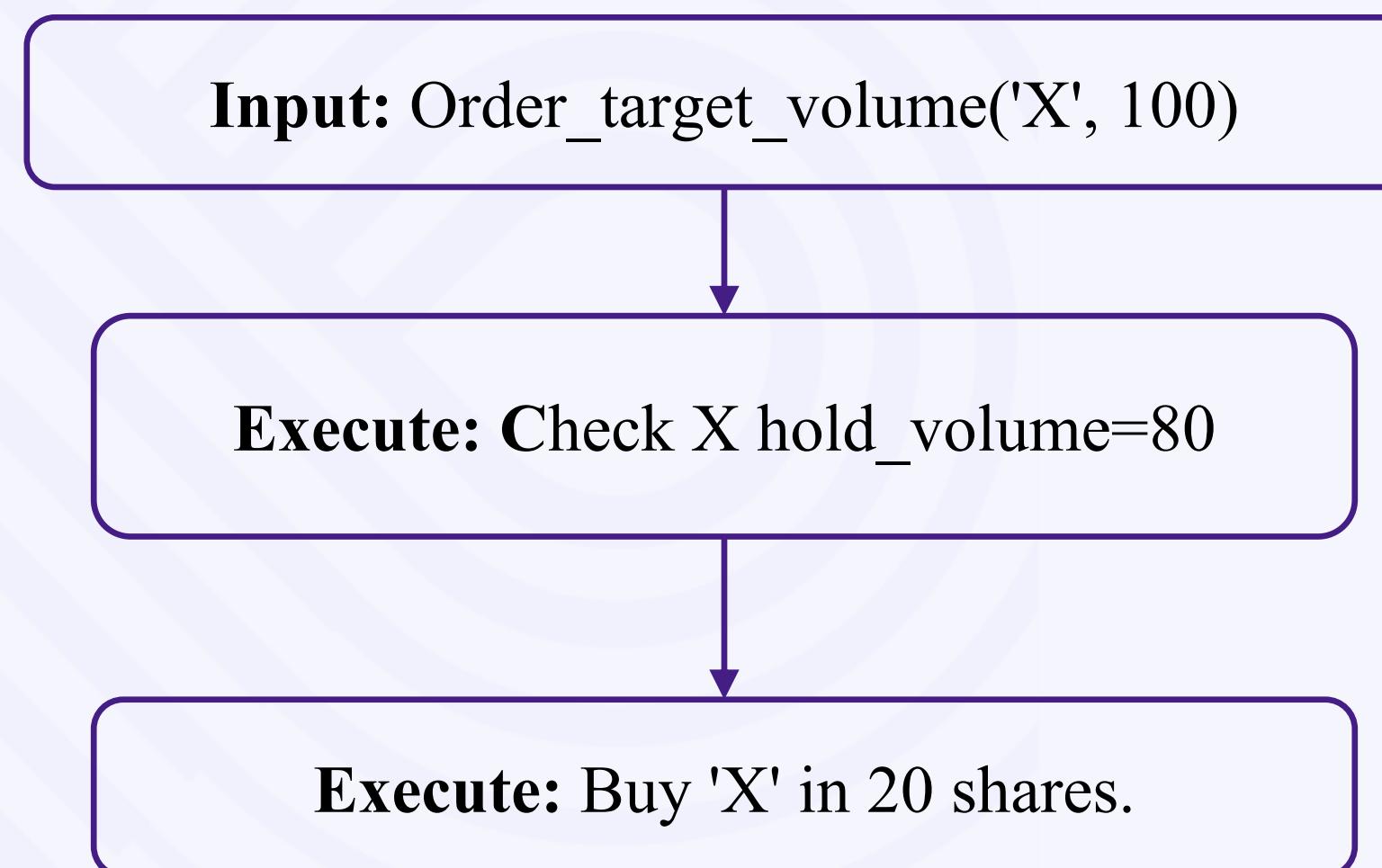


2

Order with the desired position

- *Order_target_volume(stock_code, volume)*

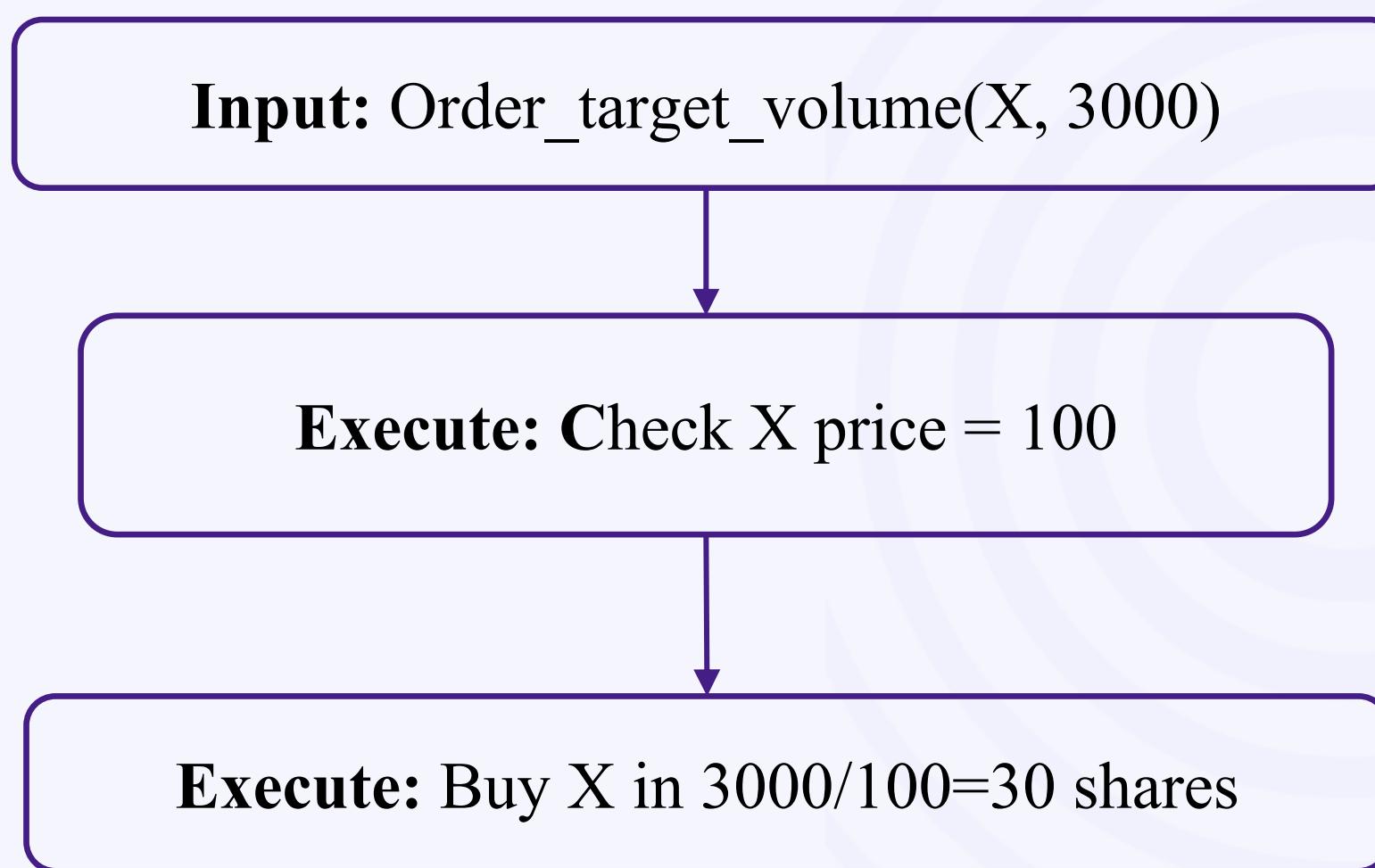
- Purpose: calculates the quantity of stocks to buy or sell in order to adjust the current position to match the target position, based on the provided symbol and target position quantity.



Strategy - Place order function

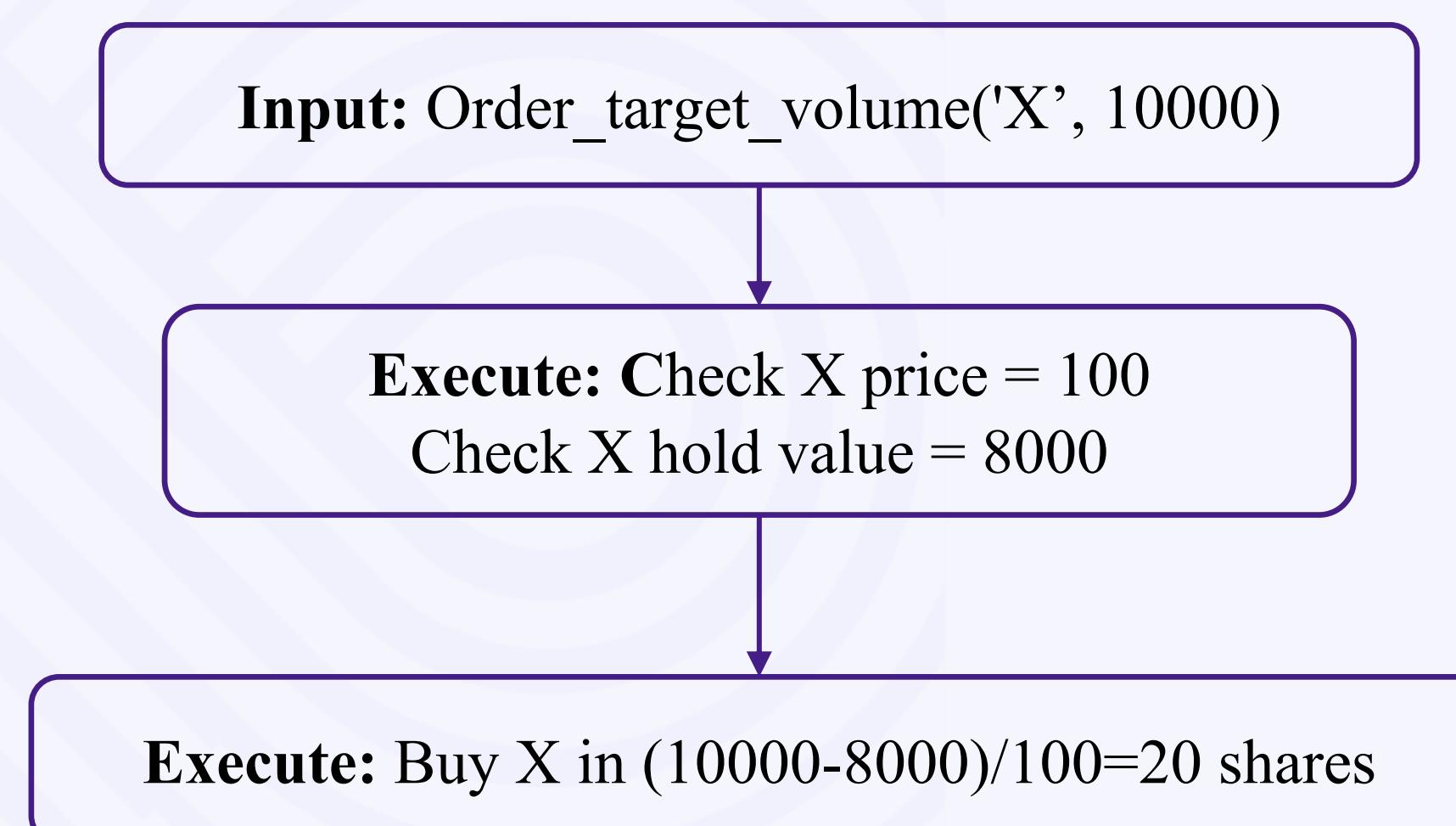
3 Order by value/price

- ***Order_by_value(stock_code, value)***
- Purpose: Buy or sell a stock based on the given value and the current stock price, ensuring that the order amount is consistent with the given value.



4 Order by (Target - Current value) /price

- ***Order_target_value(stock_code, value)***
- Purpose: Make sure that the position value is consistent with the target value by placing an order to buy or sell the stock based on the target value and the current position.



Strategy - Algorithmic trading order function

TWAP (Time Weighted Average Price)

- It divides the total trading volume into multiple batches and executes orders over a specified period of time to achieve trading at the average price.

- **Parameter:**

- The number of split copies: twap_param=16

- The number of minutes between splits: twap_gap=15 mins

3 stocks (X, Y, Z) with target trading volumes of 10,000 shares, 20,000 shares, and 15,000 shares, respectively

1. Calculate the target trading volume per trade:
 $10,000 / 16 = 625$ shares
 $20,000 / 16 = 1250$ shares
 $15,000 / 16 = 937.5$ shares.

2. Gradually execute these target trading volumes

Process

1. Calculate the target trading volume for each batch based on the total trading volume and TWAP parameters.



2. Iterate through the trading process, calculating the order volume, updating the traded volume, and recalculating the order quantity.



3. Update the time or sleep for a specified period.



4. Recalculate the target trading volume in each iteration to ensure even distribution of trading volume over the entire TWAP period.

Strategy - Algorithmic trading order function

VWAP (Volume Weighted Average Price)

- It splits the total trading volume into multiple time slots and executes orders based on the volume-weighted average price of each time slot.

- **Parameter:**

The number of split copies: vwap_param=16

The number of minutes between splits: vwap_gap=15 mins

3 stocks (X, Y, Z) with target trading volumes of 10,000 shares, 20,000 shares, and 15,000 shares, respectively

1. calculate the average proportion of trading volume for each time slot (0.05, 0.04, 0.06)

2. Get the target transaction volume
 $10,000 * 0.05 = 500$ shares
 $20,000 * 0.04 = 800$ shares
 $15,000 * 0.06 = 900$ shares.

Process

1. Computes VWAP for each time slot using past trading volume data over a specified number of previous trading days.



2. Iterates through trades, determining target trading volume per time slot based on VWAP and total volume, then executes orders.



3. Updates the time or sleeps for a specified period between each order.

03

TCA Analysis

Cost Decomposition and Execution Quality



Cost Decomposition & Execution Quality

$$vwap\ average\ price_{min} = \frac{open_{min} + high_{min} + low_{min} + close_{min}}{4}$$

$$vwap\ average\ price_{day} = \frac{\sum_1^N vwap\ average\ price_{min} * volume_{min}}{\sum_1^n volume_{day}}$$

$$traded\ average\ price_{day} = \frac{\sum_1^N traded\ price_{min} * volume_{min}}{traded\ volume_{day}}$$

$$traded\ volume_{day} = \sum_1^n traded\ volume_{min}$$

traded average price_{day} ? vwap average price_{day}

$$commission\ fee = \sum_1^n |traded\ price_{min} * traded\ volume_{min}| * commission\ rate$$

$$trade\ related\ cost = (traded\ average\ price_{day} - open\ price_{day}) * traded\ volume_{day}$$

$$opportunity\ cost = (close\ price_{day} - open\ price_{day}) * (target\ volume - traded\ volume)$$

04

GUI System

Current Holdings & Trade Log & TCA Analysis



Current Holdings

算法交易系统

股票交易 交易日志输出 tca分析

当前持仓信息

股票代码	持仓数量
603806.SH	0
002709.SZ	117800
002459.SZ	0
601059.SH	0
600732.SH	0
601138.SH	136400
002129.SZ	0
000617.SZ	0
600023.SH	0
003816.SZ	705700
600515.SH	882400
601618.SH	885500
601818.SH	946200
000069.SZ	1114800
600219.SH	820300
601916.SH	968200
600606.SH	1584100
601868.SH	1353800
600010.SH	1767400
601800.SH	0
688363.SH	0
000977.SZ	78100
300661.SZ	0

投资组合信息

总资产: 109344018.82
可用资金: 21411312.82
持仓市值: 87932706.00
初始资金: 100000000.00

开始策略 停止策略

股票代码: 搜索日期: 搜索 重置

Trade Log

算法交易系统

股票交易 交易日志输出 tca分析

2024-02-02 11:15:00 - 002459.SZ - Price: 16.89 - Volume: -200
2024-02-02 11:15:00 - 601868.SH - Price: 2.1 - Volume: 1600
2024-02-02 11:15:00 - 603806.SH - Price: 21.83 - Volume: 100
2024-02-02 11:15:00 - 688041.SH - Price: 67.93 - Volume: -100
2024-02-02 11:15:00 - 000069.SZ - Price: 2.88 - Volume: 700
2024-02-02 11:15:00 - 600732.SH - Price: 14.13 - Volume: -200
2024-02-02 11:15:00 - 300308.SZ - Price: 111.33 - Volume: -100
2024-02-02 11:15:00 - 600010.SH - Price: 1.43 - Volume: 500
2024-02-02 11:15:00 - 601916.SH - Price: 2.7 - Volume: 1100
2024-02-02 11:30:00 - 601868.SH - Price: 2.11 - Volume: 100
2024-02-02 11:30:00 - 601618.SH - Price: 3.18 - Volume: 100
2024-02-02 11:30:00 - 600023.SH - Price: 5.47 - Volume: -1600
2024-02-02 11:30:00 - 688363.SH - Price: 54.81 - Volume: 100
2024-02-02 11:30:00 - 003816.SZ - Price: 3.72 - Volume: 100
2024-02-02 11:30:00 - 601818.SH - Price: 3.16 - Volume: 100
2024-02-02 11:30:00 - 688599.SH - Price: 22.1 - Volume: 300
2024-02-02 11:30:00 - 600515.SH - Price: 3.5 - Volume: 100
2024-02-02 11:30:00 - 601916.SH - Price: 2.69 - Volume: 100
2024-02-02 11:30:00 - 000617.SZ - Price: 5.68 - Volume: 100
2024-02-02 11:30:00 - 002709.SZ - Price: 17.73 - Volume: -300
2024-02-02 13:15:00 - 000617.SZ - Price: 5.68 - Volume: 1100
2024-02-02 13:15:00 - 601916.SH - Price: 2.69 - Volume: 900
2024-02-02 13:15:00 - 002459.SZ - Price: 16.74 - Volume: -100
2024-02-02 13:15:00 - 601800.SH - Price: 8.54 - Volume: -100
2024-02-02 13:15:00 - 601059.SH - Price: 14.36 - Volume: 100
2024-02-02 13:15:00 - 605117.SH - Price: 69.4 - Volume: -100
2024-02-02 13:15:00 - 601868.SH - Price: 2.1 - Volume: 1000
2024-02-02 13:15:00 - 688363.SH - Price: 54.16 - Volume: 1900
2024-02-02 13:15:00 - 003816.SZ - Price: 3.72 - Volume: 1500
2024-02-02 13:15:00 - 600606.SH - Price: 2.08 - Volume: 400
2024-02-02 13:15:00 - 601818.SH - Price: 3.13 - Volume: 700
2024-02-02 13:15:00 - 688041.SH - Price: 67.51 - Volume: -100
2024-02-02 13:15:00 - 002709.SZ - Price: 17.54 - Volume: -5900
2024-02-02 13:15:00 - 600732.SH - Price: 14.04 - Volume: -300

股票代码: 搜索日期: 搜索 重置

TCA Analysis

算法交易系统

股票交易 | 交易日志输出 | tca分析

股票代码	日期	平均价格	加权平均价格	目标交易量	实际交易量	交易相关成本	今开	今收	机会成本	佣金费用
300896.SZ	2024-01-31	275.69	274.52	9456.26	9800.00	-61795.00	282.00	270.88	3822.34	810.54
300033.SZ	2024-01-31	118.77	117.81	21877.65	22700.00	-70713.00	121.89	115.04	5633.10	808.86
688256.SH	2024-01-31	114.05	113.17	22988.51	23800.00	-46446.00	116.00	110.00	4868.97	814.31
300751.SZ	2024-01-31	105.11	104.55	24943.10	25900.00	-46686.00	106.91	102.50	4219.94	816.68
300308.SZ	2024-01-31	102.52	103.46	26661.33	27600.00	68977.00	100.02	102.80	-2609.49	848.86
605117.SH	2024-01-31	69.39	68.85	37479.50	38800.00	-68302.00	71.15	67.57	4727.38	807.70
688041.SH	2024-01-31	66.28	66.65	41215.87	42800.00	67536.00	64.70	66.27	-2487.09	851.01
300763.SZ	2024-01-31	60.15	60.32	44540.95	46100.00	12683.00	59.87	59.70	265.04	831.81
603392.SH	2024-01-31	54.46	54.69	48857.95	50600.00	-6000.00	54.58	53.99	1027.81	826.72
688599.SH	2024-01-31	23.30	23.40	116601.08	121100.00	51601.00	22.87	23.10	-1034.75	846.35
300759.SZ	2024-01-31	21.63	21.49	119635.11	124200.00	-81605.00	22.29	21.10	5432.22	806.04
603806.SH	2024-01-31	22.16	22.38	121765.60	126400.00	32925.00	21.90	22.73	-3846.55	840.33
002709.SZ	2024-01-31	19.02	19.10	140424.78	145800.00	4280.00	18.99	18.68	1666.32	831.91
002459.SZ	2024-01-31	17.63	17.54	149981.25	155500.00	-23686.00	17.78	17.03	4139.06	822.33
601059.SH	2024-01-31	15.30	15.40	177777.78	184700.00	54577.00	15.00	15.22	-1522.89	847.52
600732.SH	2024-01-31	14.12	14.15	193236.71	200300.00	64052.00	13.80	13.80	-0.00	848.46
601138.SH	2024-01-31	13.30	13.32	200501.25	207800.00	599.00	13.30	13.31	-72.99	829.30
002129.SZ	2024-01-31	12.57	12.63	214018.19	221600.00	24105.00	12.46	12.30	1213.09	835.57
000617.SZ	2024-01-31	6.14	6.18	442233.28	459900.00	50134.00	6.03	6.15	-2120.01	847.00
600023.SH	2024-01-31	5.40	5.42	504095.78	524200.00	55582.00	5.29	5.38	-1809.38	848.58
003816.SZ	2024-01-31	3.75	3.77	734618.92	764500.00	92608.00	3.63	3.82	-5677.41	860.32
600515.SH	2024-01-31	3.50	3.49	761904.76	792900.00	-3546.00	3.50	3.49	309.95	831.48
601618.SH	2024-01-31	3.25	3.25	815494.39	848300.00	-15969.00	3.27	3.24	984.17	827.39
601818.SH	2024-01-31	3.14	3.14	851970.18	885900.00	12522.00	3.13	3.15	-678.60	835.62
000000.00	2024-01-31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

股票代码: 搜索日期: 搜索 重置

TCA Analysis

算法交易系统

股票交易 | 交易日志输出 | tca分析

股票代码	日期	平均价格	加权平均价格	目标交易量	实际交易量	交易相关成本	今开	今收	机会成本	佣金费用
600023.SH	2024-01-31	5.40	5.42	504095.78	524200.00	55582.00	5.29	5.38	-1809.38	848.58

股票代码: 600023.SH 搜索日期: 2024-01-31

05

Results

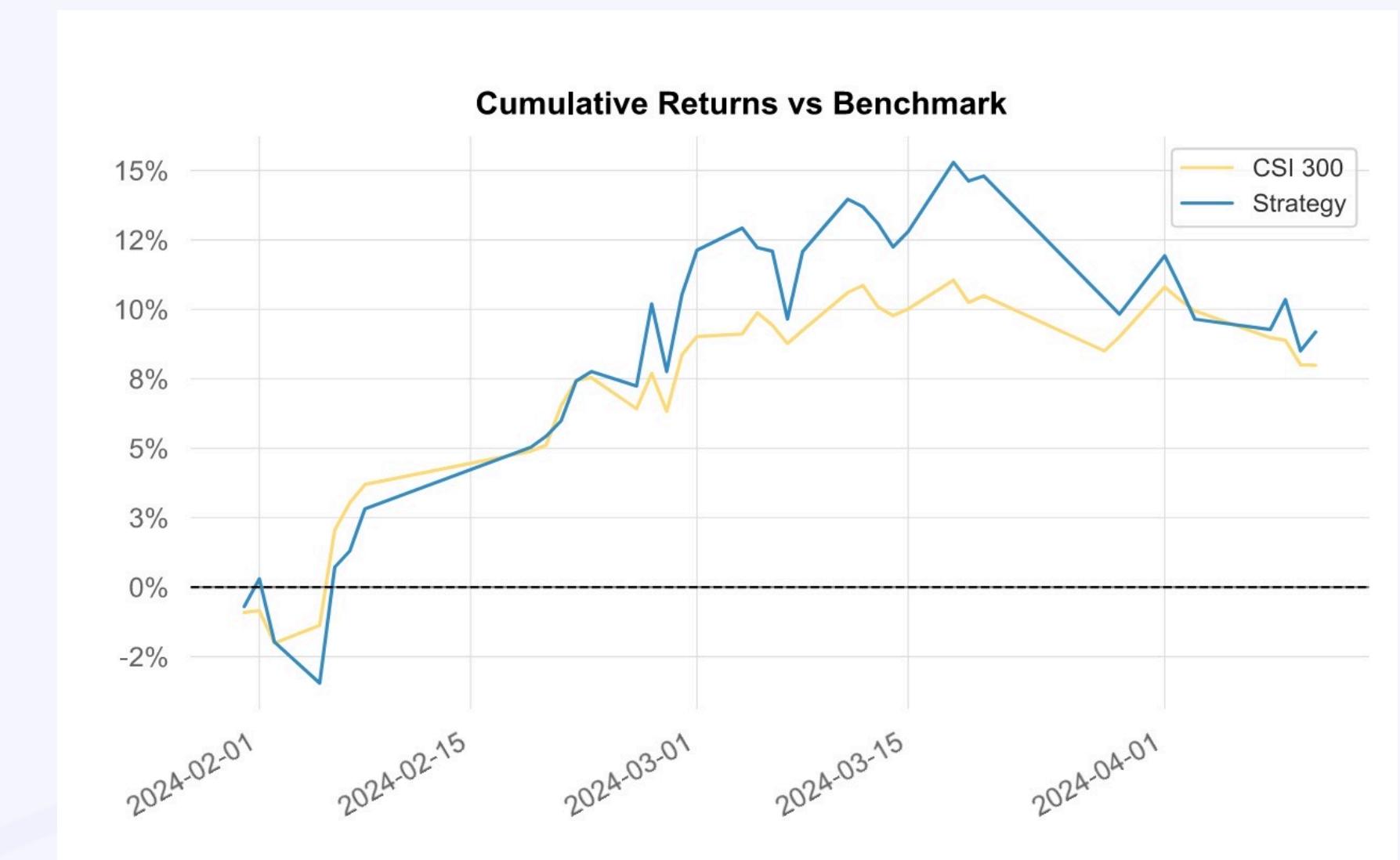
Return & Sharpe & MDD



Strategy Effect

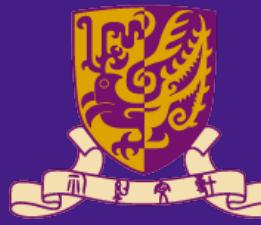
Key Indicator

- Using the aforementioned algorithmic trading model, the strategy has a better effect than the CSI 300 index and obtains excess returns
- In terms of stability, it is also better than the index, which can be seen by indicators such as volatility



Key performance Metrics

Metrics	Strategy
Cumulative Return	9.19%
Sharpe	2.32
Max Drawdown	-5.89%
Volatility(ann.)	25.9%



香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

SCHOOL OF
DATA SCIENCE
數據科學學院



Thanks for Listening