PHBS 北京大学汇丰商学院 商界军校

Software Engineering Project

Pair Strategy

Final Presentation
2021/11/11

@PHBS 331

Group 2

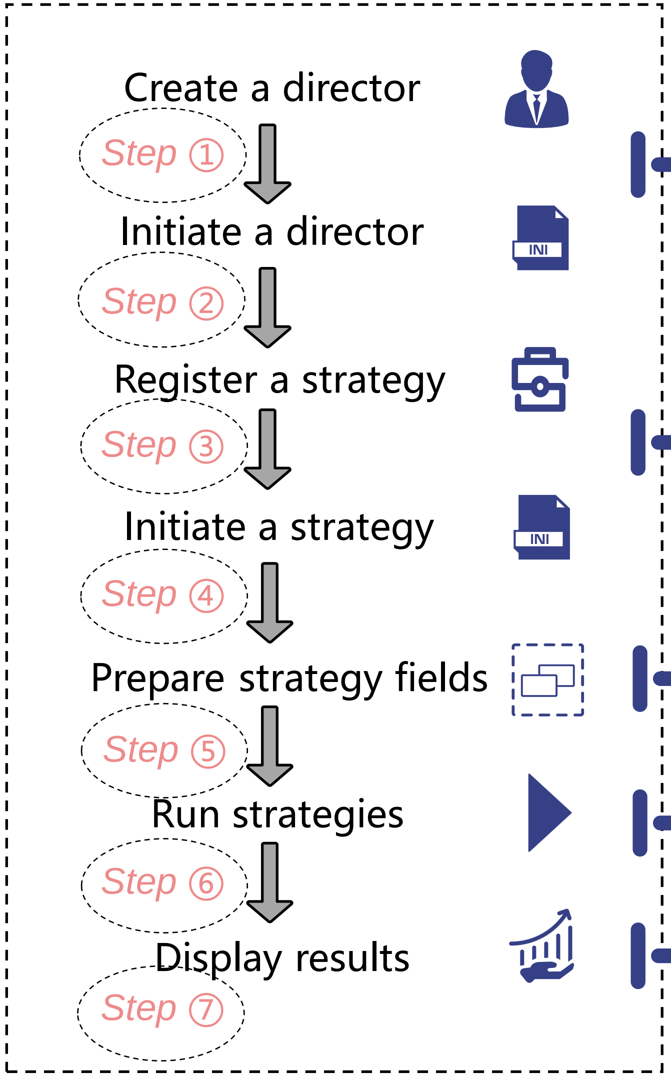
- 2001212374 庞立文
- 2001212376 漆岱峰
- 2001212377 强音
- 2001212388 汪子杰
- 2001212409 叶梦婕

Contents

- Overview
- Details of *Signal Class*
- Details of *Strategy Class*
- Performance & Sensitivity Analysis
- Highlights & Bug Reporting

1. Overview

Overview: Form Test Perspective



Create a director

Step ①

Initiate a director

Step ②

Register a strategy

Step ③

Initiate a strategy

Step ④

Prepare strategy fields

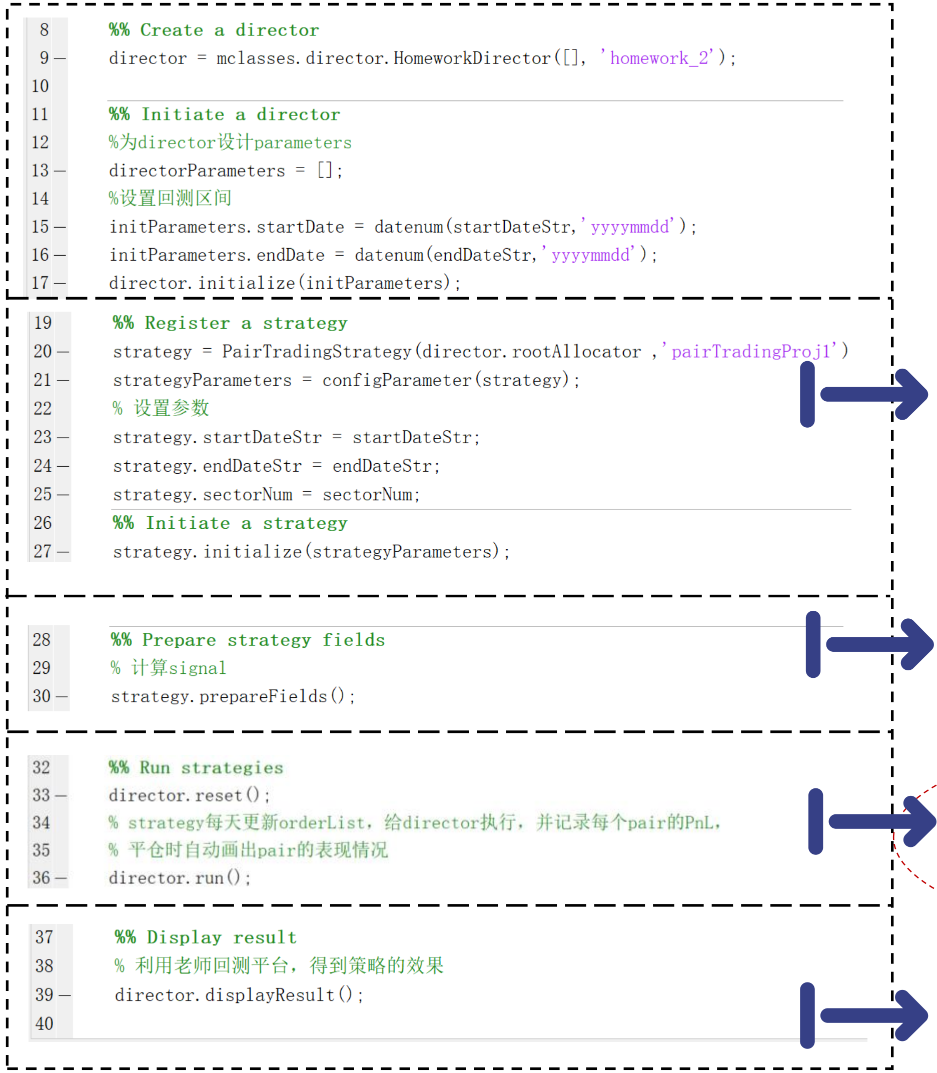
Step ⑤

Run strategies

Step ⑥

Display results

Step ⑦



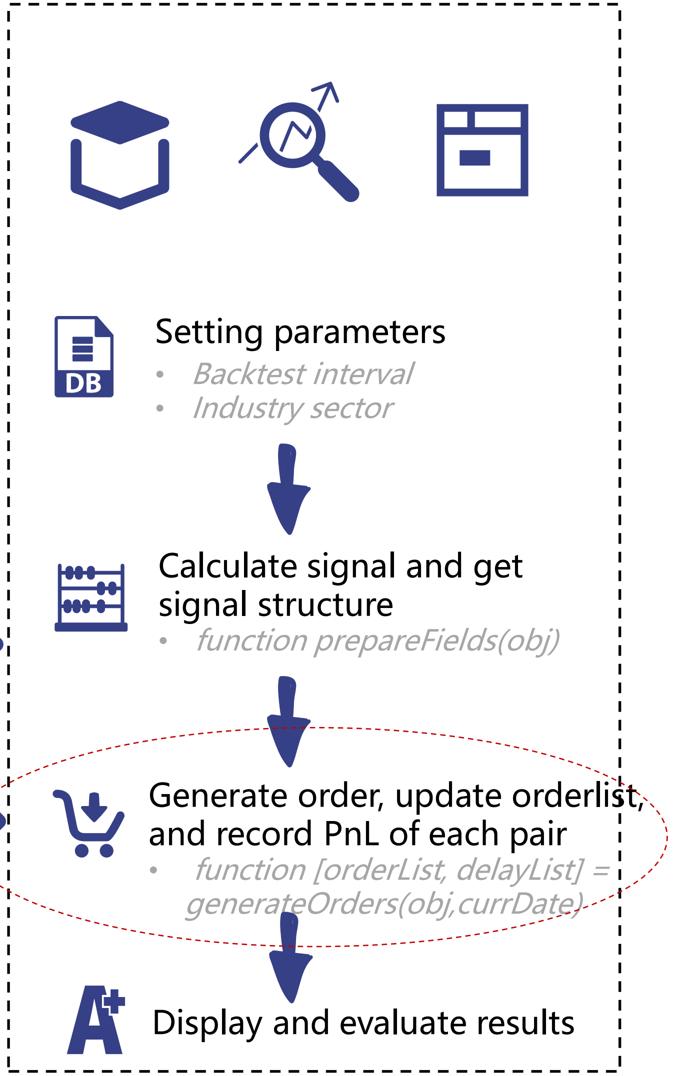
```
%% Create a director
director = mclasses.director.HomeworkDirector([], 'homework_2');

%% Initiate a director
%为director设计parameters
directorParameters = [];
%设置回测区间
initParameters.startDate = datenum(startDateStr,'yyyymmdd');
initParameters.endDate = datenum(endDateStr,'yyyymmdd');
director.initialize(initParameters);
```

```
%% Register a strategy
strategy = PairTradingStrategy(director.rootAllocator ,'pairTradingProj1')
strategyParameters = configParameter(strategy);
% 设置参数
strategy.startDateStr = startDateStr;
strategy.endDateStr = endDateStr;
strategy.sectorNum = sectorNum;
%% Initiate a strategy
strategy.initialize(strategyParameters);
```

```
%% Prepare strategy fields
% 计算signal
strategy.prepareFields();
```

```
%% Run strategies
director.reset();
% strategy每天更新orderList，给director执行，并记录每个pair的PnL，
% 平仓时自动画出pair的表现情况
director.run();
```

```
%% Display result
% 利用老师回测平台，得到策略的效果
director.displayResult();
```



Setting parameters

- *Backtest interval*
- *Industry sector*

Calculate signal and get signal structure

- *function prepareFields(obj)*

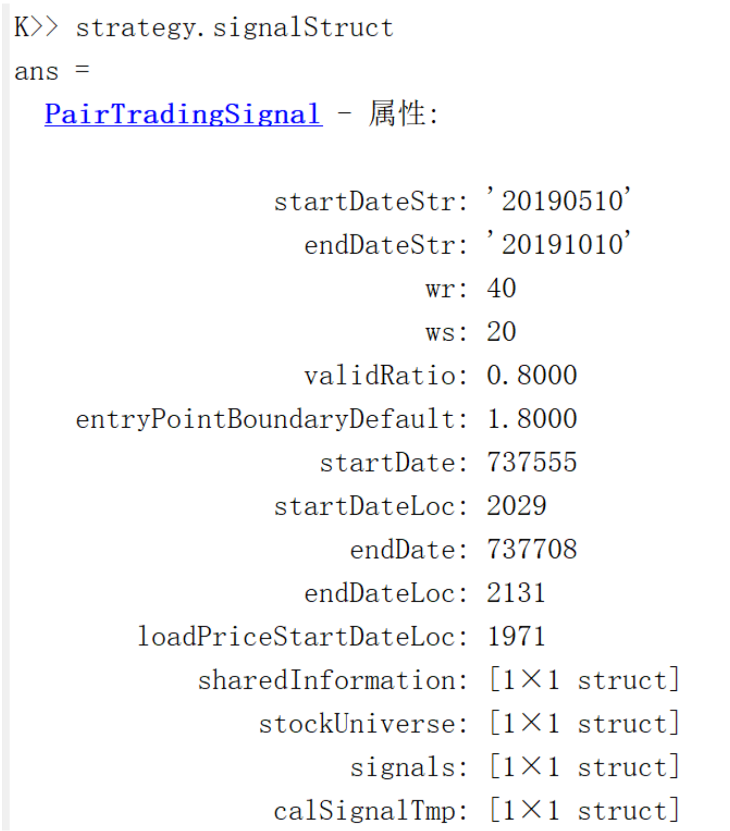
Generate order, update orderlist, and record PnL of each pair

- *function [orderList, delayList] = generateOrders(obj,currDate)*

Display and evaluate results

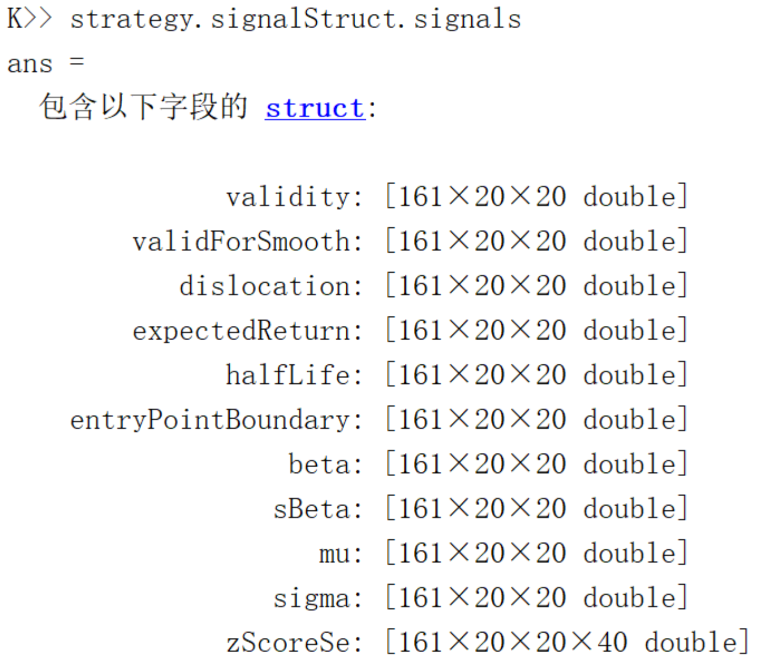
Overview: Form Test Perspective

Strategy. signalStruct



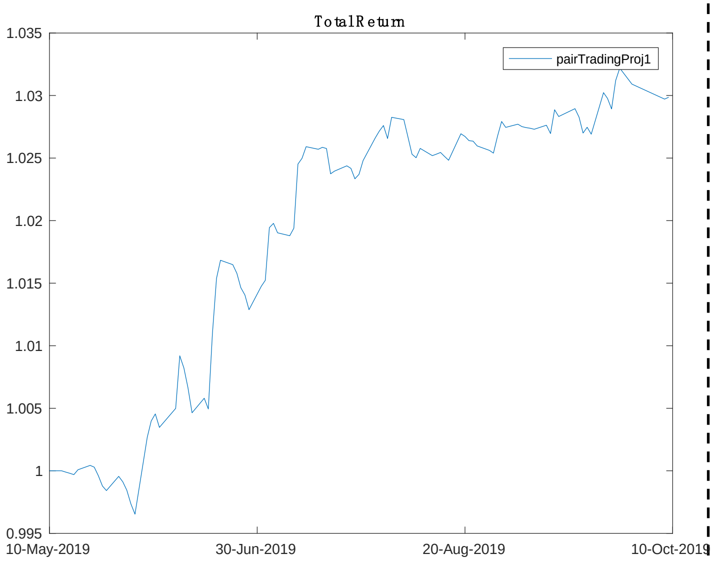
```
K>> strategy.signalStruct
ans =
  PairTradingSignal - 属性:

                 startDateStr: '20190510'
                   endDateStr: '20191010'
                           wr: 40
                           ws: 20
                   validRatio: 0.8000
    entryPointBoundaryDefault: 1.8000
                    startDate: 737555
                 startDateLoc: 2029
                      endDate: 737708
                   endDateLoc: 2131
        loadPriceStartDateLoc: 1971
            sharedInformation: [1×1 struct]
                stockUniverse: [1×1 struct]
                      signals: [1×1 struct]
                 calSignalTmp: [1×1 struct]
```

strategy. signalStruct.signals



```
K>> strategy.signalStruct.signals
ans =
  包含以下字段的 struct:

              validity: [161×20×20 double]
        validForSmooth: [161×20×20 double]
           dislocation: [161×20×20 double]
        expectedReturn: [161×20×20 double]
              halfLife: [161×20×20 double]
    entryPointBoundary: [161×20×20 double]
                  beta: [161×20×20 double]
                 sBeta: [161×20×20 double]
                    mu: [161×20×20 double]
                 sigma: [161×20×20 double]
              zScoreSe: [161×20×20×40 double]
```

Results



2. Details of Signal Class

Signal Class

PairTradingStrategy

+ **startDateStr**: string
+ **endDateStr**: string
+ **wr**: int (40)
+ **ws**: int (20)
+ **validRatio**: float (0.8)
+ **entryPointBoundaryDefault**: float (1.8)
+ **startDateLoc**: int
+ **endDateLoc**: int
+ **loadPriceStartDateLoc**: int
+ **sharedInformation**: struct
+ **stockUniverse**: struct
+ **signals**: struct

Methods

+ **PairTradingSignal** (startDateStr: string ,endDateStr: string ,sectorNum: int): void

+ **calSignals ()**: void

+ **calSignal** (currDateLoc: int,stockYLoc: int,stockXLoc: int): void

+ **testCointegration** (priceX: matrix, priceY: matrix): Boolean

+ **smoothBeta** (): struct

+ **analyzeResidual**(residualSe: matrix)

Static Methods

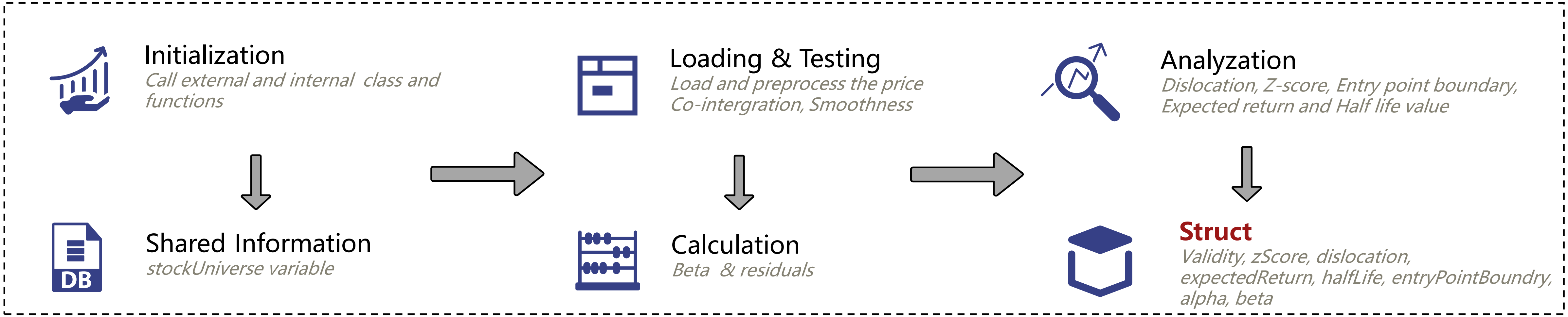
+ **processPrice**(price: matrix)

+ **calResidual** (sBeta: float,priceY: matrix, priceX: matrix)

+ **OU_Calibrate_LS** (S: matrix, delta: float)

Whole picture of *pairTradingSignal*

- **Summary:** Signal Class is used to calculate the signal identified by strategies.



Initialization

Call external and internal class and functions

Shared Information

stockUniverse variable

Loading & Testing

Load and preprocess the price

Co-intergration, Smoothness

Calculation

Beta & residuals

Analyzation

Dislocation, Z-score, Entry point boundary, Expected return and Half life value

Struct

Validity, zScore, dislocation, expectedReturn, halfLife, entryPointBoundry, alpha, beta

- The information along with the pairs stock price are stored in a ***struct*** as follows: (250 is the number of dates and we pick 50 pairs of stocks as an example)

Name	Size	Name	Size
validity	250 × 50 × 50 double	halfLife	250 × 50 × 50 double
zScore	250 × 50 × 50 double	entryPointBoundry	250 × 50 × 50 double
dislocation	250 × 50 × 50 double	alpha	250 × 50 × 50 double
expectedReturn	250 × 50 × 50 double	beta	250 × 50 × 50 double

Whole picture of *pairTradingSignal*

- The output of signal is **a structure**, which contains the information of each pair of stocks and trading day.
- Each element is a three-dimensional matrix, corresponding to ***[trading date, serial number of stock A, serial number of stock B]***. The specific elements are as follows:

validity: whether the pair of shares can be traded on the day

zscore: calculate the residual generated after the regression of stock B to stock A in the past time window and the results after de mean standardization

relocation: calculate the latest residual generated after the regression of stock B to stock A in the past time window, which is used to judge whether the signal is triggered

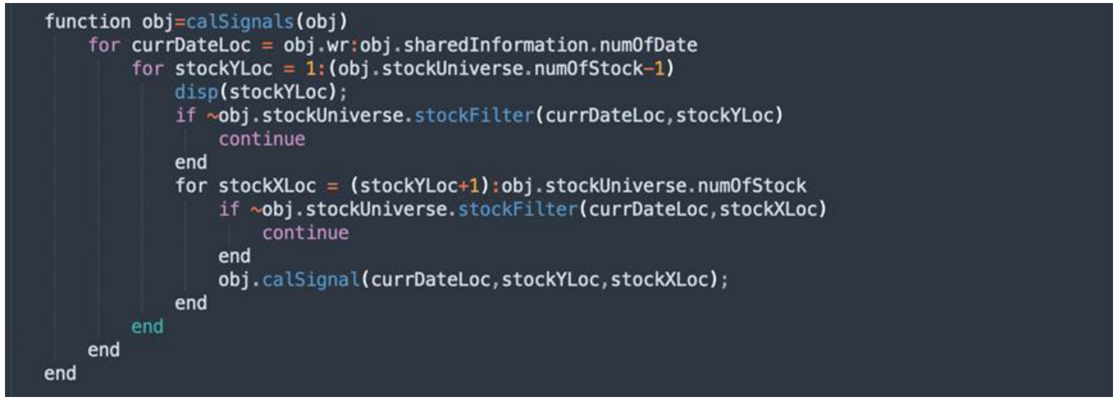
expectedreturn: calculate the expected return of each pair of stocks on each trading day

halflife: half life

entrypointboundary: the threshold value of each pair of stock transactions on each trading day. By judging the relationship between zscore and entry point boundary, select long portfolio or short portfolio, generally 2*sigma

alpha: the intercept item after the regression of stock B to stock A on each trading day

beta: the coefficient value after the regression of stock B to stock A on each trading day



```
function obj=calSignals(obj)
    for currDateLoc = obj.wr:obj.sharedInformation.numOfDate
        for stockYLoc = 1:(obj.stockUniverse.numOfStock-1)
            disp(stockYLoc);
            if ~obj.stockUniverse.stockFilter(currDateLoc,stockYLoc)
                continue
            end
            for stockXLoc = (stockYLoc+1):obj.stockUniverse.numOfStock
                if ~obj.stockUniverse.stockFilter(currDateLoc,stockXLoc)
                    continue
                end
                obj.calSignal(currDateLoc,stockYLoc,stockXLoc);
            end
        end
    end
end
```

3. Details of Strategy Class

Strategy Class

PairTradingStrategy

+ **holdingStruct**: struct
+ **signalStruct**: struct
+ **maxNumofPairs**: int
+ **holdingPairStruct** : struct
+ **closedPairStruct** : struct
+ **recentPairID**: : struct
+ **currDate**: int
+ **currDateLoc**: int
+ **cutWin**: float
+ **cutLoss**: float
+ **cutPeriod**: int
+ **startDateStr**: string
+ **endDateStr**: string
+ **sectorNum**: int
+ **maxNumPlot**: int
+ **capitalAvail**: int
+ **capitalInit**: int
+ **adjERratio**: float

+ **prepareFields**(): void
+ **generateOrders**(tradeDate): void
+ **setOrder**(numPairAvail): void
+ **updateHoldingPairPrice**(): void
+ **checkHoldingPairToClosed**(): void
+ **adjustOrder** (currAvailableCapital): void
+ **adjustPairCodePosition**(): void

```
Function PairTradingStrategy
Begin
  signalstruct = PairTradingSignal
  initialize all struct
  while currentdate <= enddate
      if stockpair meets the close position condition:
          put into closedpairstruct
      elseif stockpair meets the adjusted position condition:
          generateorder;
      elseif stockpair meets the Generate position condition:
          setorder;
      Update HoldingStruct
      currentdate ++
End
```

Properties

- **Struct:**
  - **Holdingstruct:** TradeDate*Stock, NetPosition, which stores the net position sum of each y to all x after regression
  - **Signalstruct:** Calculate the signal
  - **Holdingpairsstruct**: Record holding stock pairs’ order
  - **Closepairstruct**: Record closed stock pairs’ order
  - **recentpairID**: every stock pair has an unique id

- **Int:**
  - **maxNumOfPairs**: max numbers of pair available
  - **currDate**: current trading date
  - **Sectornum**: sector number
  - **Currdateloc**: current date location
  - **Cutwin & cutloss**: check if current pairpnl is larger than cutwin(%) or less than cutloss(%)
  - **Cutperiod**: max holding days
  - **CapitalInit& capital available**: initialize capital and available capital

Prepare Fields

- The fields to initialize all the parameters and calculate the signals.
- **Initialize Object** (capitalAvail, holdingPairStruct, holding struct, closedPairStruct)
  - *for i = 1:length(obj.holdingPairStruct.description)-1*
  - *obj.holdingPairStruct.(obj.holdingPairStruct.description{i})=zeros(obj.maxNumOfPairs,1);*
  - *end*
- **Calculate signals from PairTradingSignal Class**
  - *obj.signalStruct = PairTradingSignal(obj.startDateStr,obj.endDateStr,obj.sectorNum);*
  - *obj.signalStruct.calSignals();*

Generate Orders

- The fields to update daily orders' adjustment.
- **UpdateHoldingPairPrice:** update holding paris' price
- **AdjustPairCodePosition:** calculate the net position according to Long or Short and update to orderlist
- **HoldingPairToClosedPair:** put holding pairs to closed pairs if close position' s condition meets
- Before adjusting positions, it is necessary to check close position condition!

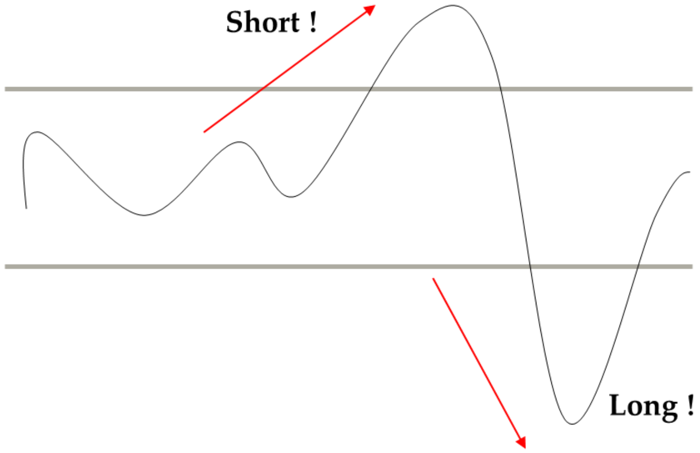
- Close Position' s condition: (if it meets the condition, close pairs and update capital available, pair → Closepairstruct)
  - (1) Judge whether the current pair is effective. If it is invalid, it will be forced to close the position
  - (2) If it works, see if it reaches cutperiod
  - (3) If you reach the stop profit or stop loss, close the position. If the profit reaches 3%, or the stop loss reaches 3%, close the position
  - (4) Update available funds after closing positions

- Adjust Position' s condition:
  - (1) If numpairavail equals to zero and there is no available capital, the position will be adjusted
  - (2) Find the pair with the smallest ER, find the pair with 2ER in the buffer pool, and adjust the position

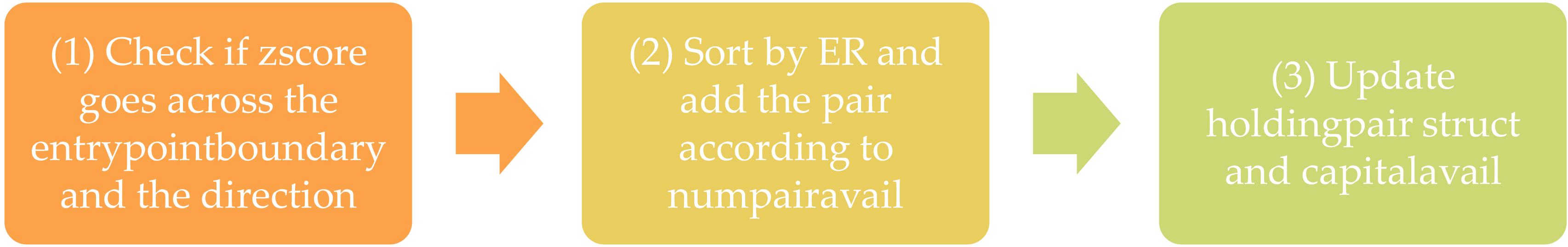
Set Orders

- The fields to set daily orders.
- **In generate orders, we set the condition:**

```
if (numPairAvail ==0) | (isempty(obj.capitalAvail))
            obj.adjustOrder();
        else
            obj.setOrder(numPairAvail);
    end
```



- **Set Order Process:**



(1) Check if zscore goes across the entrypointboundary and the direction

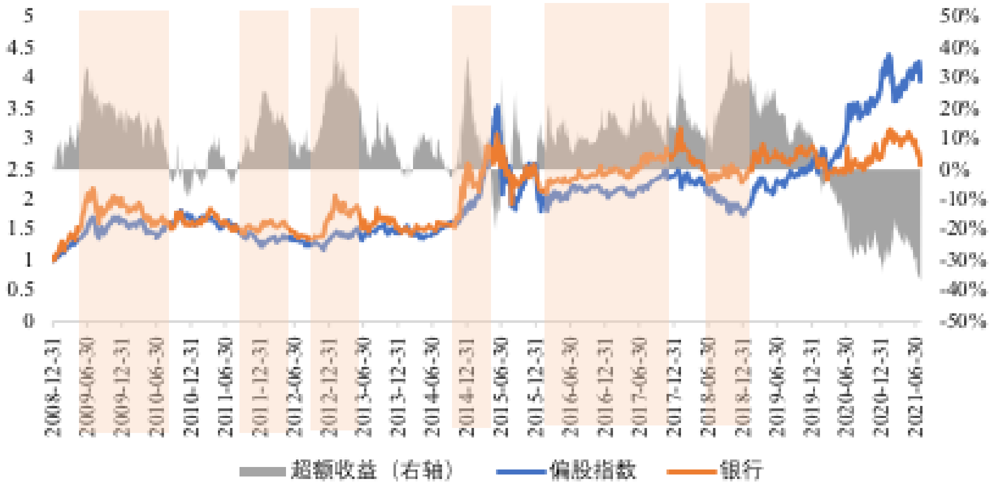
(2) Sort by ER and add the pair according to numpairavail

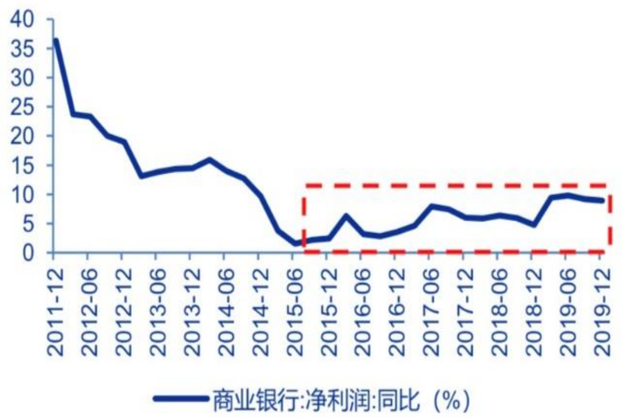
(3) Update holdingpair struct and capitalavail

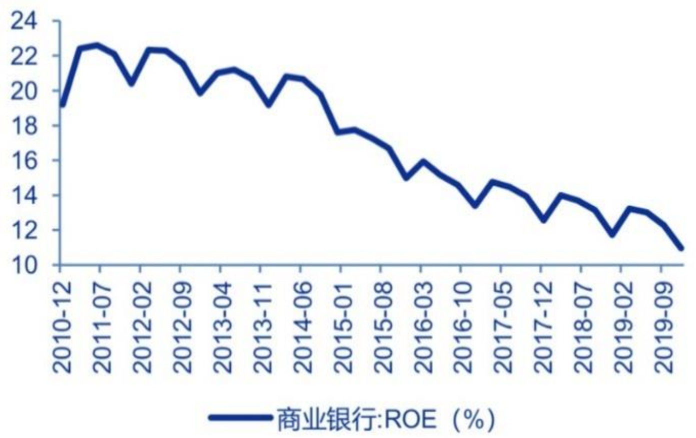
Why Choose Bank Sector?

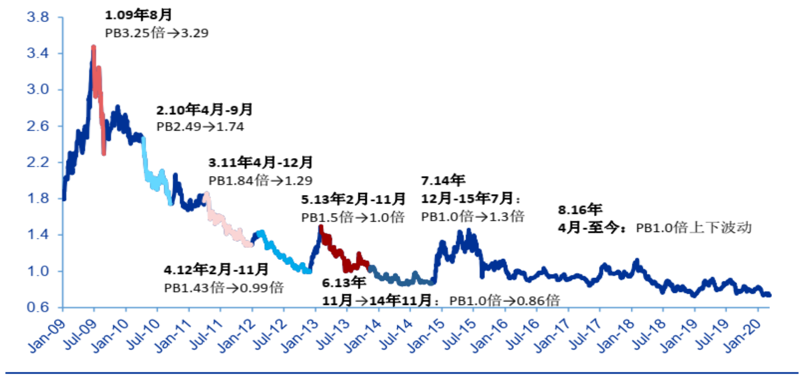
- Historically, the bank index has had six excess returns. Banks have three times to surpass the market interpretation of their peers, two sideways shocks, and an excess return reflected by a sharp decline in the market.

- Some fundamental data of banking industry:
    - The growth rate of net profit of commercial banks increased, ROE generally showed a downward trend, and the PB was low



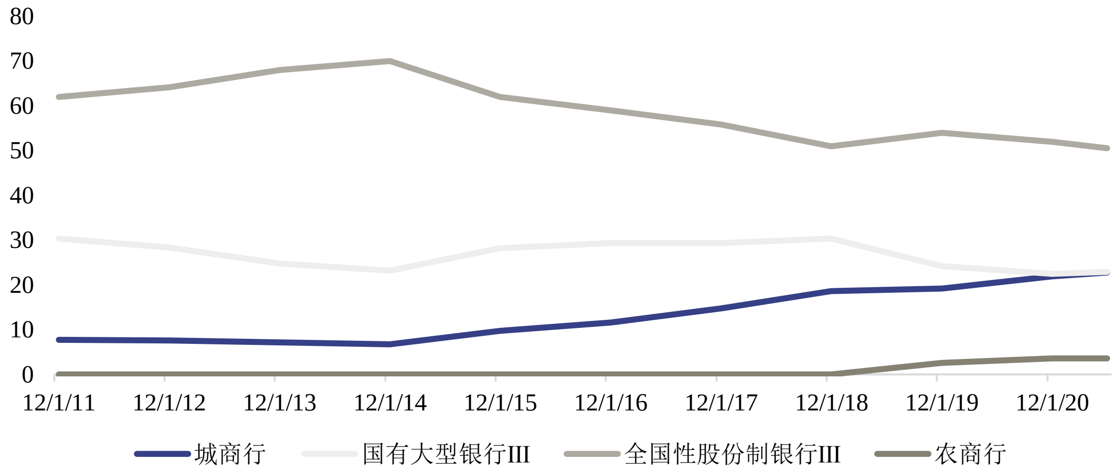






Why Choose Bank Sector?

- The bank distribution of the composite index is mainly joint-stock banks, rather than state-owned banks and urban commercial banks.
- From the top ten heavyweight stocks of the sw index, China Merchants Bank accounts for a relatively high proportion of the heavyweight and industrial bank is generally in an upward trend.
- Roe of the heavyweight stocks is basically between 10% - 15%, and dividend yield in the second quarter of 2021 vary not so greatly among different banks.

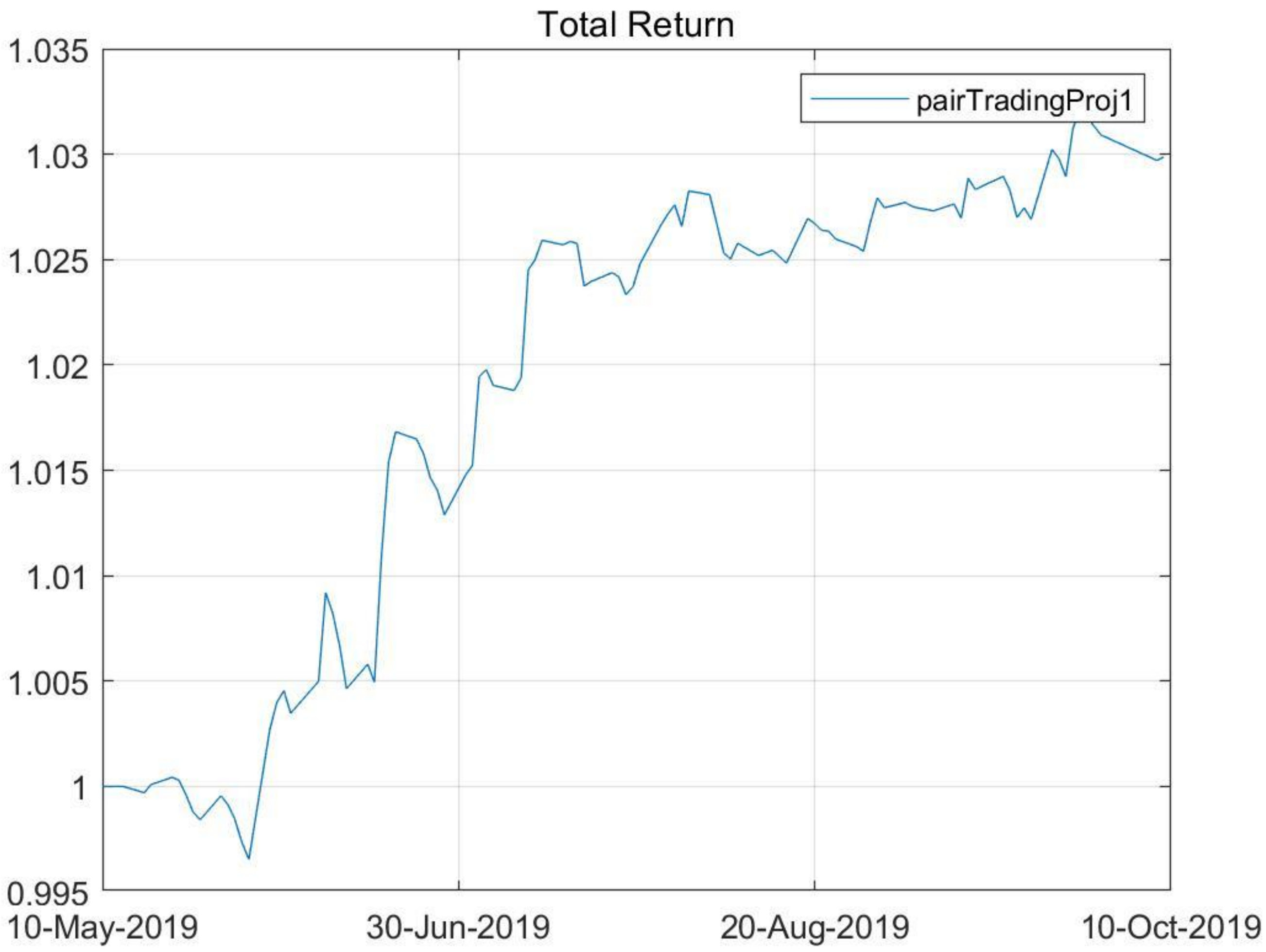


2011A		2014A		2018A		2021H	
民生银行	14.86	招商银行	15.70	招商银行	13.99	招商银行	14.98
招商银行	13.92	民生银行	15.56	兴业银行	10.67	兴业银行	12.11
浦发银行	9.77	兴业银行	11.51	交通银行	9.11	平安银行	8.89
交通银行	8.87	浦发银行	10.72	民生银行	8.15	工商银行	7.34
兴业银行	8.79	交通银行	6.52	农业银行	7.90	宁波银行	5.69
工商银行	7.89	光大银行	5.93	浦发银行	6.59	交通银行	5.45
农业银行	6.17	农业银行	5.88	工商银行	6.54	浦发银行	4.76
光大银行	5.15	平安银行	5.52	北京银行	4.76	民生银行	3.80
建设银行	5.04	工商银行	5.13	平安银行	4.61	农业银行	3.53
平安银行	4.91	北京银行	4.23	中国银行	4.36	江苏银行	3.40

2021H	PB	ROE* (%)	最新一期股息率 (%)
招商银行	2.04	17.32	2.21
兴业银行	0.75	15.19	3.90
平安银行	1.47	11.00	0.88
工商银行	0.67	11.68	4.74
宁波银行	2.16	15.67	1.28
交通银行	0.48	10.00	6.43
浦发银行	0.54	11.56	6.00
民生银行	0.38	10.98	4.83
农业银行	0.56	11.79	6.11
江苏银行	0.78	11.59	4.45

4. Performance & Sensitivity Analysis

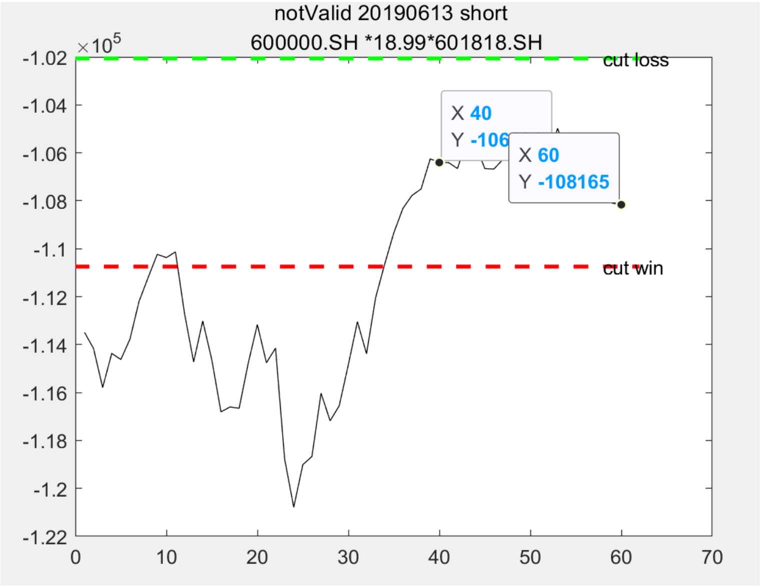
Performance Display



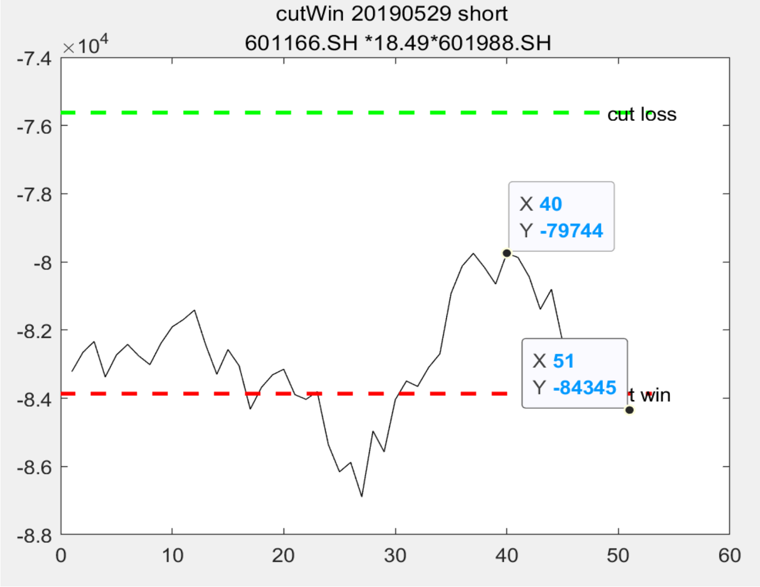
■ Result

- Sector: bank
- Stock pool: 20
- Annualized rate: 7.29%
- Sharpe ratio: 1.81

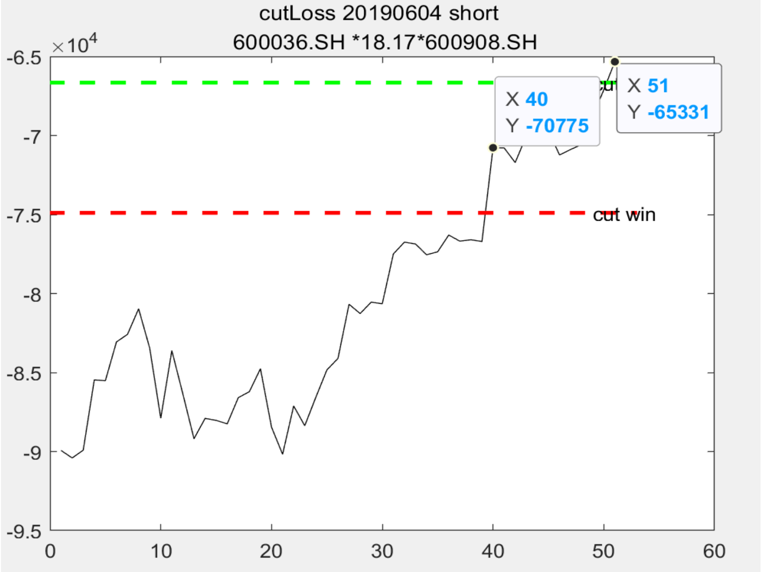
Examples of Closed a Position



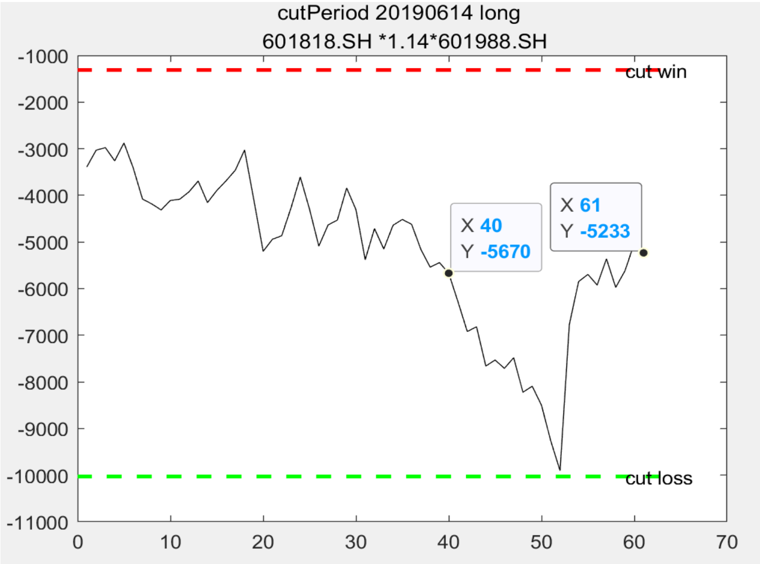
- **Invalid Pairs**



- **Cut Win**

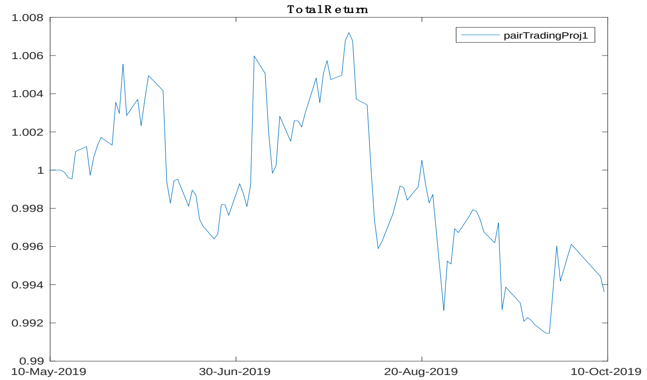


- **Cut Loss**

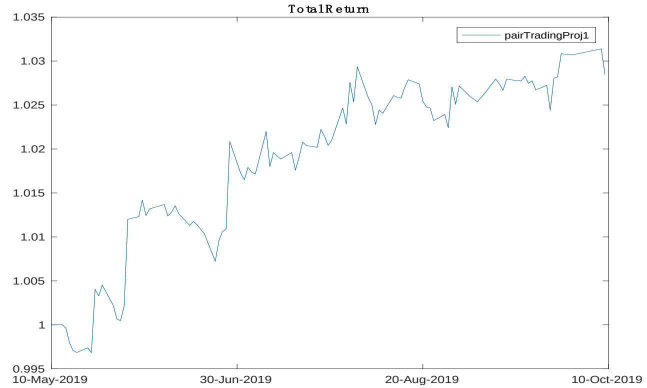


- **Cut Period**

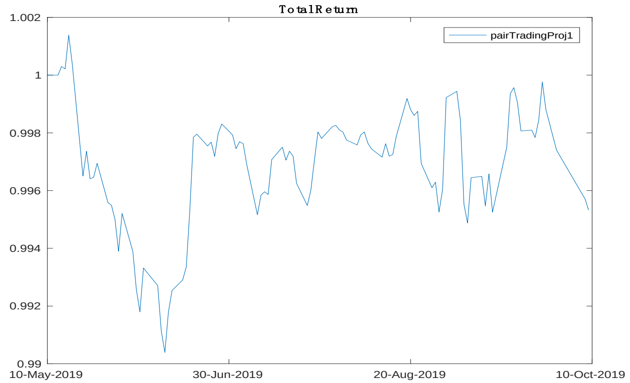
Sensitivity Analysis—Different Sectors



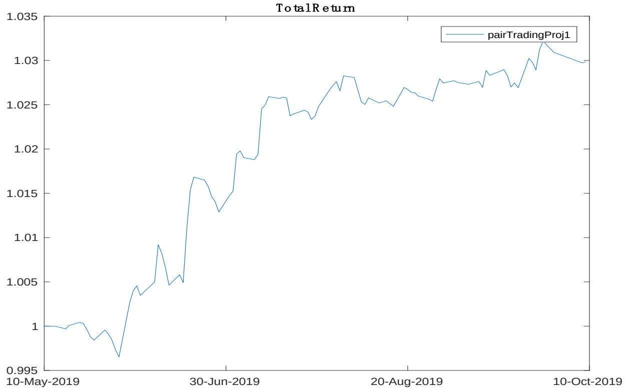
- **Mining Sector**
  - annualized return: -1.52%
  - sharpe ratio: -1.71



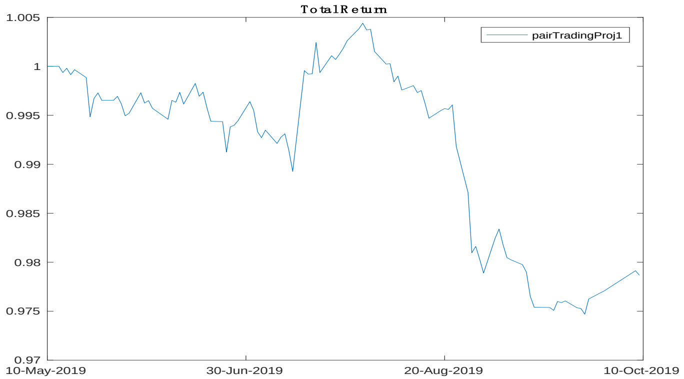
- **Nonferrous Metals Sector**
  - annualized return: 6.94%
  - sharpe ratio: 1.11



- **Chemistry Industry Sector**
  - annualized return: -1.12%
  - sharpe ratio: -2.48



- **Bank Sector**
  - annualized return: 7.29%
  - sharpe ratio: 1.81

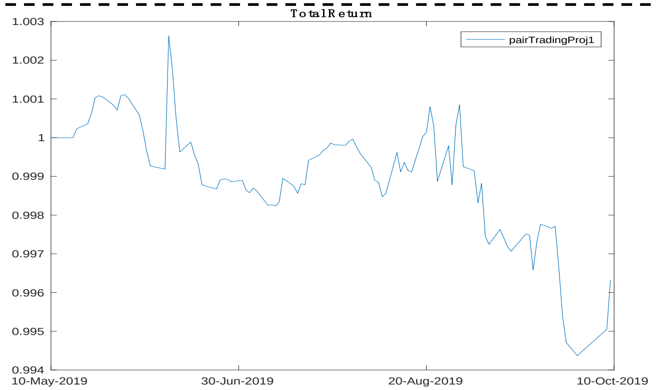


- **Steel Sector**
  - annualized return: -5.03%
  - sharpe ratio: -2.73

Note:

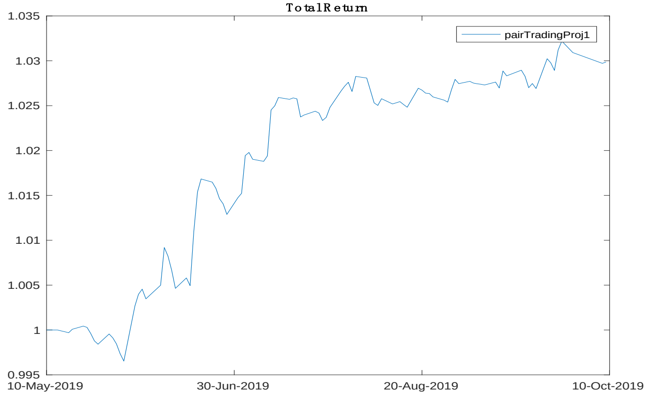
- ***Time period: 20190510-20191010***
- ***Sector number: 20***

Sensitivity Analysis—Different Stock Pools



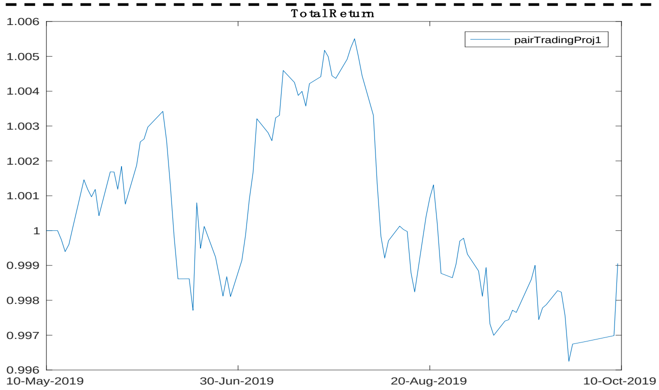
5 Stocks

- annualized return: -0.88%
- sharpe ratio: -3.89



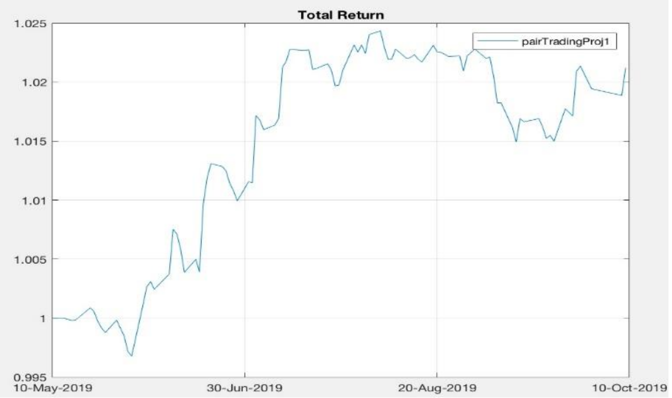
20 Stocks

- annualized return: 7.29%
- sharpe ratio: 1.81



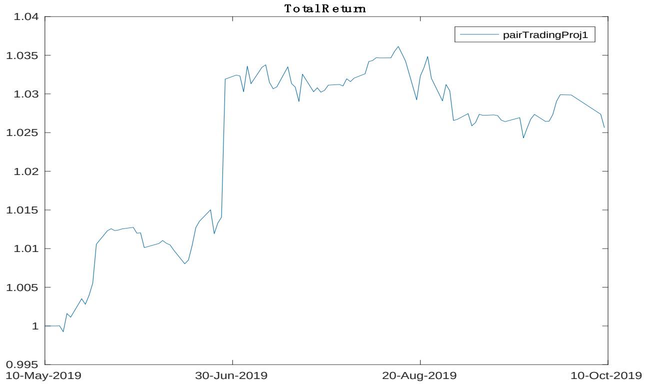
10 Stocks

- annualized return: -0.23%
- sharpe ratio: -2.40



25 Stocks

- annualized return: 5.15%
- sharpe ratio: 0.93



15 Stocks

- annualized return: 6.24%
- sharpe ratio: 0.89

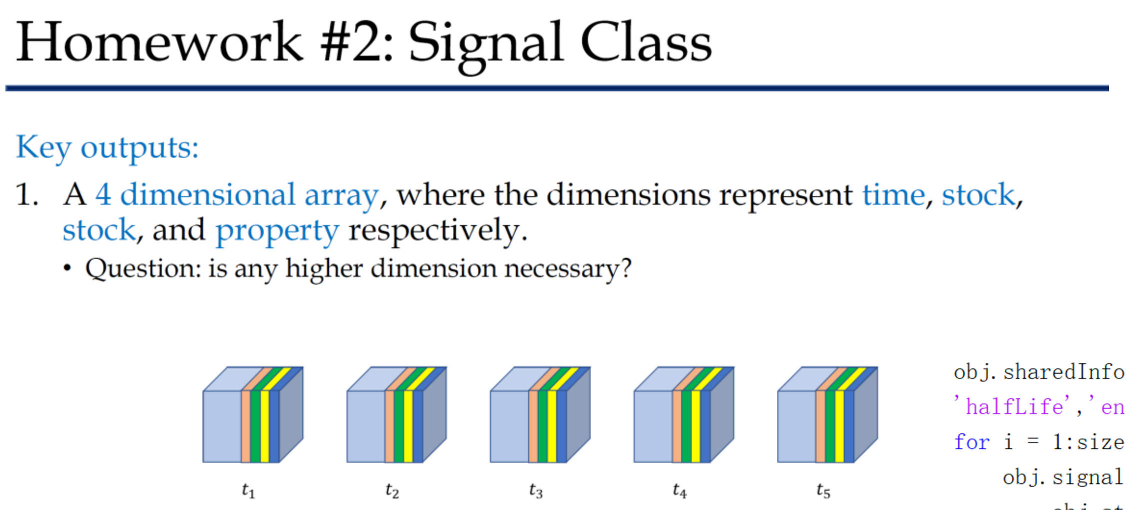
Note:

- ***Time period: 20190510-20191010***
- ***Bank Sector***

5. Highlights & Bug Reporting

dataStruct Design--PairTradingSignal

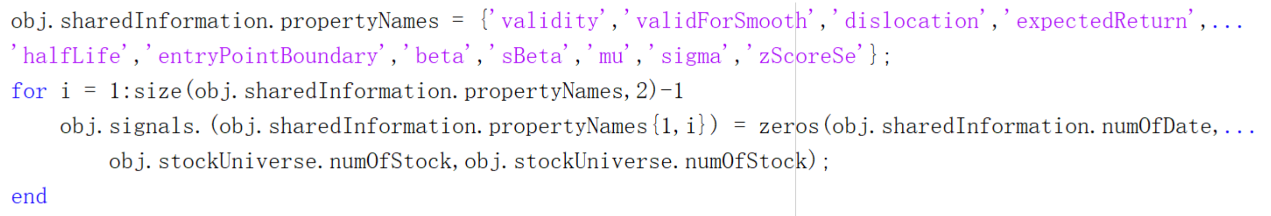
- Signals
  - From the slide, it should be a ***cell***, which is not sufficient!!
  - My design: use ***struct***
    - ***signals.propertyName = zeros(numOfDate,numOfStock,numOfStock)***



Homework #2: Signal Class

Key outputs:

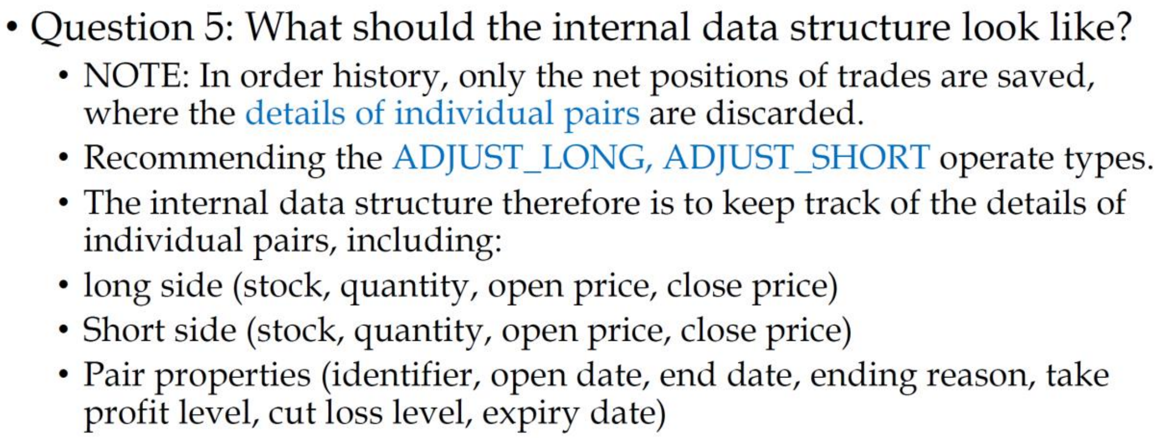
1. A 4 dimensional array, where the dimensions represent time, stock, stock, and property respectively.
   - Question: is any higher dimension necessary?



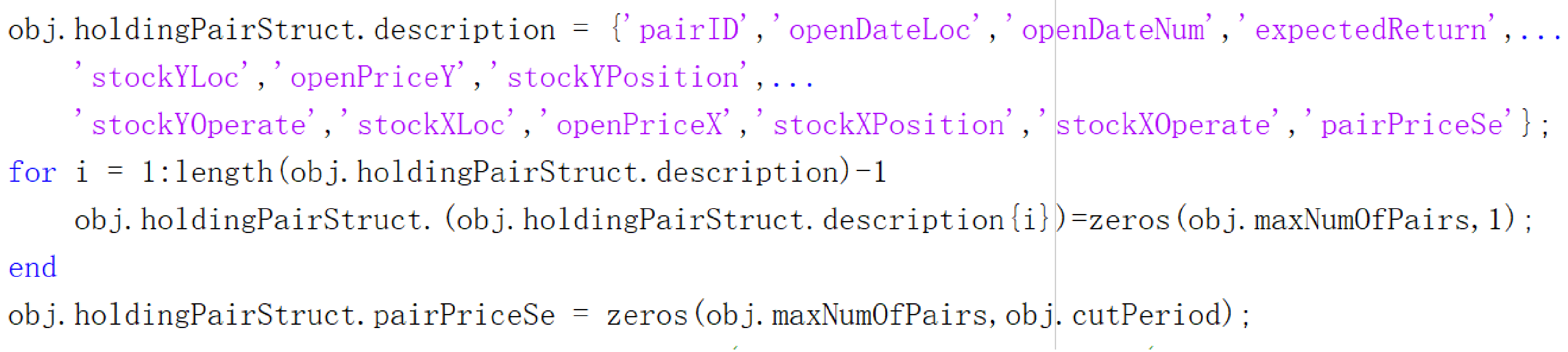
```
obj.sharedInformation.propertyNames = {'validity','validForSmooth','dislocation','expectedReturn',...
'halfLife','entryPointBoundary','beta','sBeta','mu','sigma','zScoreSe'};
for i = 1:size(obj.sharedInformation.propertyNames,2)-1
    obj.signals.(obj.sharedInformation.propertyNames{1,i}) = zeros(obj.sharedInformation.numOfDate,...
        obj.stockUniverse.numOfStock,obj.stockUniverse.numOfStock);
end
```

dataStruct Design--PairTradingStrategy

- ***holdingPairStruct***
  - Record the information of our holding pairs
  - openDate, price, position & operate → for calculate pairPriceSe → plot the performance of each pair



- Question 5: What should the internal data structure look like?
  - NOTE: In order history, only the net positions of trades are saved, where the details of individual pairs are discarded.
  - Recommending the ADJUST_LONG, ADJUST_SHORT operate types.
  - The internal data structure therefore is to keep track of the details of individual pairs, including:
  - long side (stock, quantity, open price, close price)
  - Short side (stock, quantity, open price, close price)
  - Pair properties (identifier, open date, end date, ending reason, take profit level, cut loss level, expiry date)

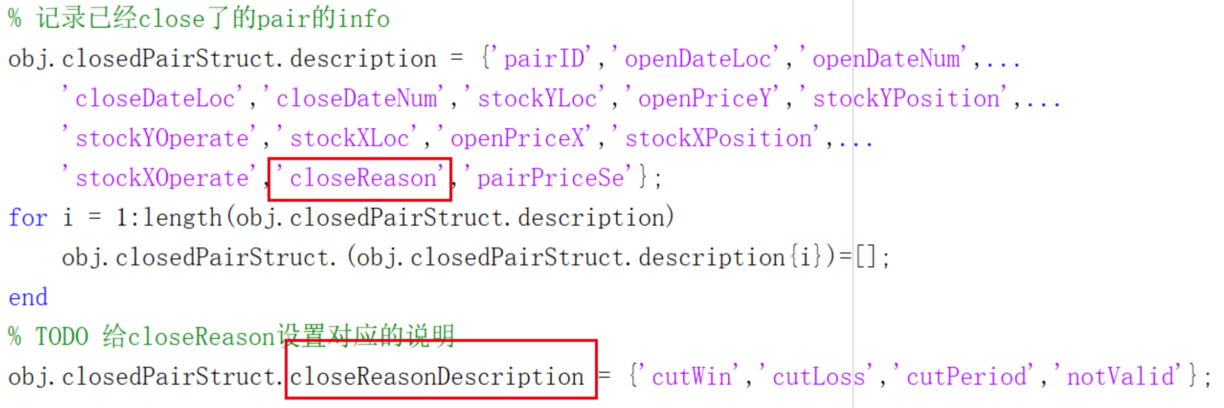


```
obj.holdingPairStruct.description = {'pairID','openDateLoc','openDateNum','expectedReturn',...
    'stockYLoc','openPriceY','stockYPosition',...
    'stockYOperate','stockXLoc','openPriceX','stockXPosition','stockXOperate','pairPriceSe'};
for i = 1:length(obj.holdingPairStruct.description)-1
    obj.holdingPairStruct.(obj.holdingPairStruct.description{i})=zeros(obj.maxNumOfPairs,1);
end
obj.holdingPairStruct.pairPriceSe = zeros(obj.maxNumOfPairs,obj.cutPeriod);
```

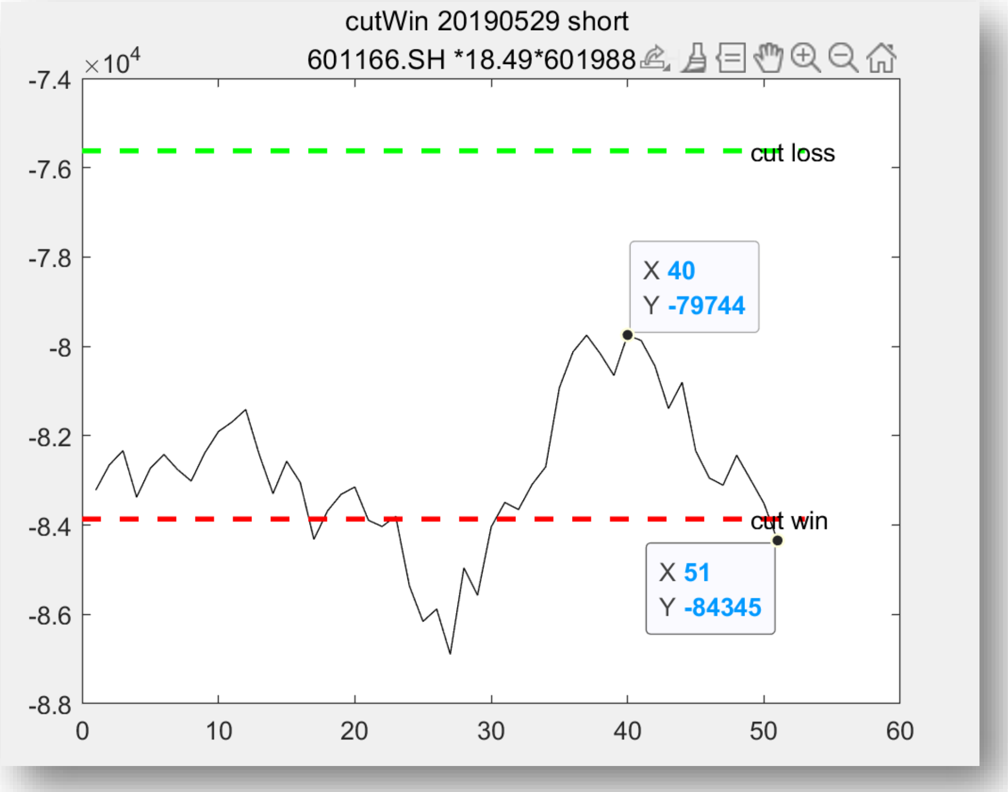
dataStruct Design--PairTradingStrategy

closedPairStruct

- Everyday check the holdingPairStuct whether to close order
- If order closed, remove this pair from **holding**PairStruct to **closed**PairStruct, then plot the performance of this pair

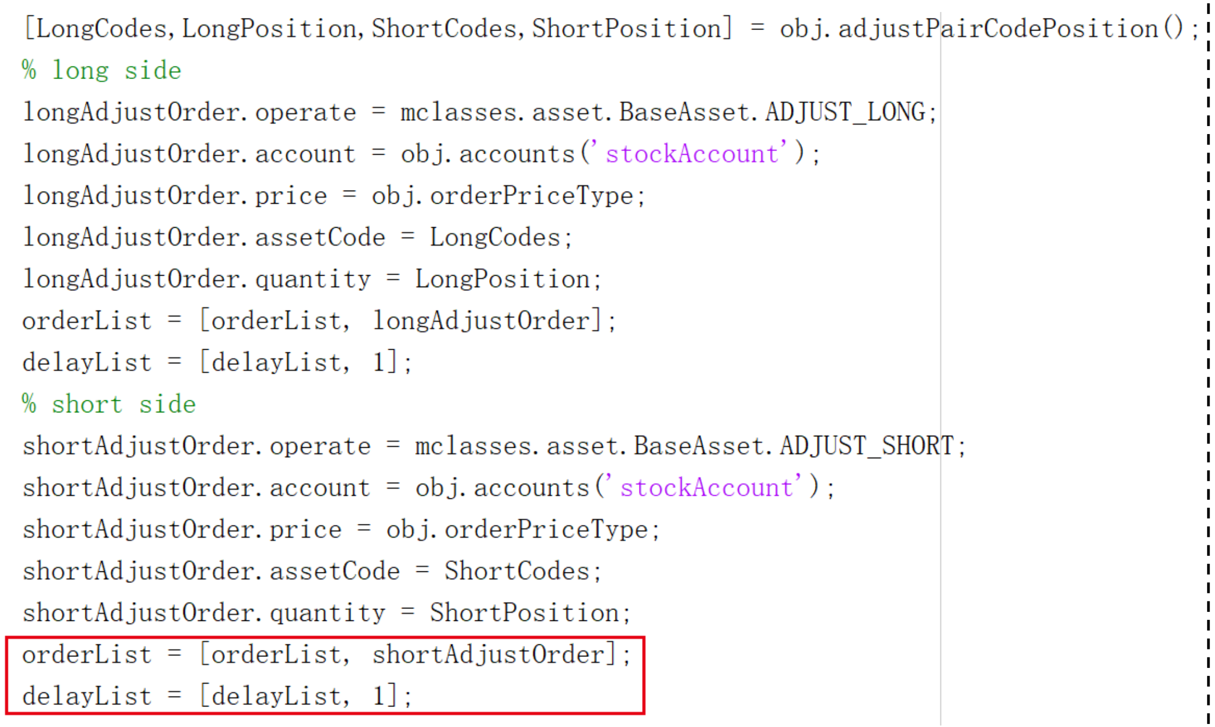


```
% 记录已经close了的pair的info
obj.closedPairStruct.description = {'pairID','openDateLoc','openDateNum',...
    'closeDateLoc','closeDateNum','stockYLoc','openPriceY','stockYPosition',...
    'stockYOperate','stockXLoc','openPriceX','stockXPosition',...
    'stockXOperate','closeReason','pairPriceSe'};
for i = 1:length(obj.closedPairStruct.description)
    obj.closedPairStruct.(obj.closedPairStruct.description{i})=[];
end
% TODO 给closeReason设置对应的说明
obj.closedPairStruct.closeReasonDescription = {'cutWin','cutLoss','cutPeriod','notValid'};
```



dataStruct Design--PairTradingStrategy

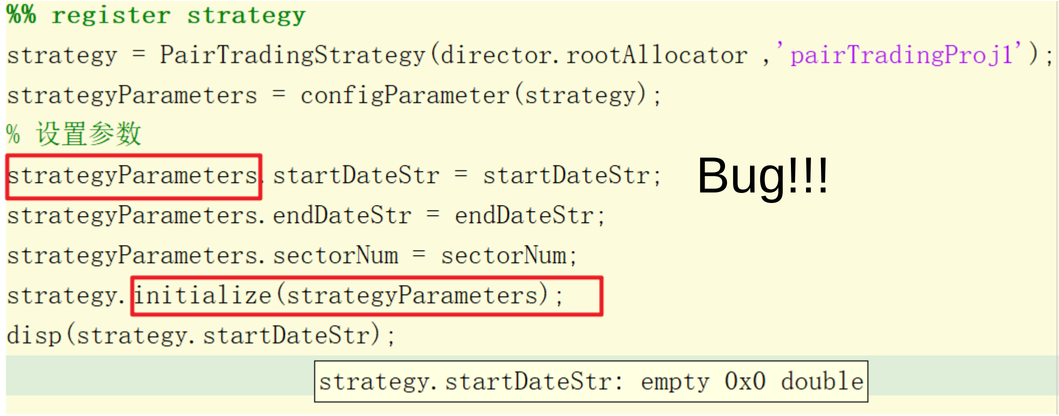
- ***holdingStruct***
  - Record the net position and price of each stock
  - Strategy offers ***orderList & delayList***
  - ***holdingPairStruct***
    - Pair1: **stock1**-stock2(**long1**)
    - Pair2: **stock1**-stock3(**long2**)
    - Pair3: stock4+**stock1**(**short2**)
  - ***holdingStruct***
    - position=(1,-1,-2,-2)
    - In fact, short operation happens in margin account.
  - →**adjOrderList**



```
[LongCodes, LongPosition, ShortCodes, ShortPosition] = obj.adjustPairCodePosition();
% long side
longAdjustOrder.operate = mclasses.asset.BaseAsset.ADJUST_LONG;
longAdjustOrder.account = obj.accounts('stockAccount');
longAdjustOrder.price = obj.orderPriceType;
longAdjustOrder.assetCode = LongCodes;
longAdjustOrder.quantity = LongPosition;
orderList = [orderList, longAdjustOrder];
delayList = [delayList, 1];
% short side
shortAdjustOrder.operate = mclasses.asset.BaseAsset.ADJUST_SHORT;
shortAdjustOrder.account = obj.accounts('stockAccount');
shortAdjustOrder.price = obj.orderPriceType;
shortAdjustOrder.assetCode = ShortCodes;
shortAdjustOrder.quantity = ShortPosition;
orderList = [orderList, shortAdjustOrder];
delayList = [delayList, 1];
```

Bugs!

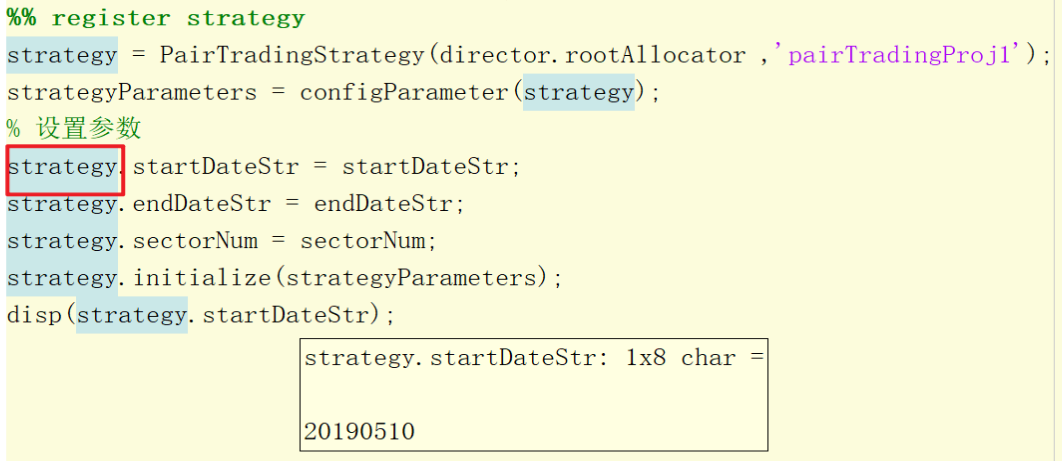
- Don’ t touch the super class TAT (in this proj…), especially when you don’ t know much about it.



```
%% register strategy
strategy = PairTradingStrategy(director.rootAllocator ,'pairTradingProj1');
strategyParameters = configParameter(strategy);
% 设置参数
strategyParameters.startDateStr = startDateStr;
strategyParameters.endDateStr = endDateStr;
strategyParameters.sectorNum = sectorNum;
strategy.initialize(strategyParameters);
disp(strategy.startDateStr);
```

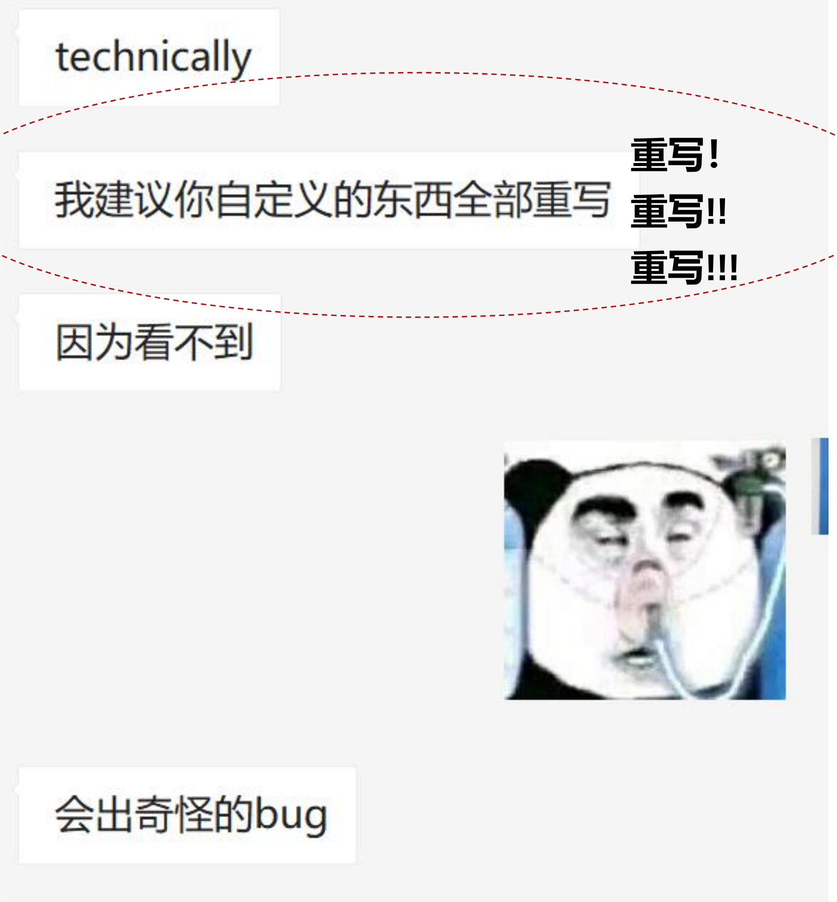
```
strategy.startDateStr: empty 0x0 double
```

Bug!!!



```
%% register strategy
strategy = PairTradingStrategy(director.rootAllocator ,'pairTradingProj1');
strategyParameters = configParameter(strategy);
% 设置参数
strategy.startDateStr = startDateStr;
strategy.endDateStr = endDateStr;
strategy.sectorNum = sectorNum;
strategy.initialize(strategyParameters);
disp(strategy.startDateStr);
```

```
strategy.startDateStr: 1x8 char =

20190510
```



technically

我建议你自定义的东西全部重写

因为看不到

会出奇怪的bug

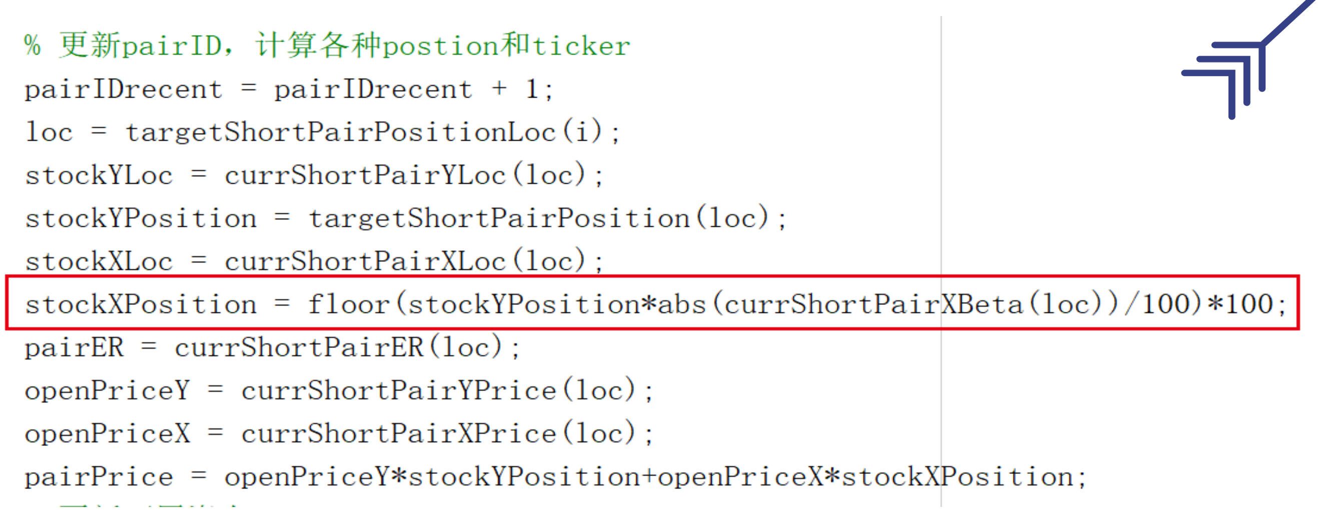
重写!

重写!!

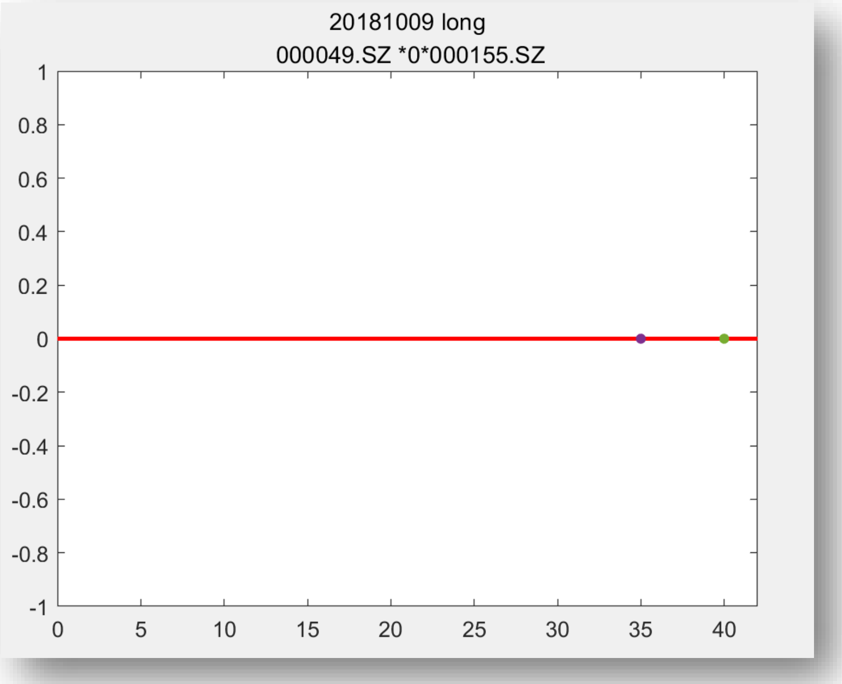
重写!!!

Bugs!

- setOrder: use signals, zScoreSe & entryPointBoundary
- closeOrder:
  - Signals
    - Use **beta** on **openDate, not** the **smoothed beta** on **currDate**
    - Not idealPosition due to the **floor** operation
  - pairPriceSe
    - obj.cutWin=0.03
    - obj.cutLoss=-0.03
    - Use pairPnL

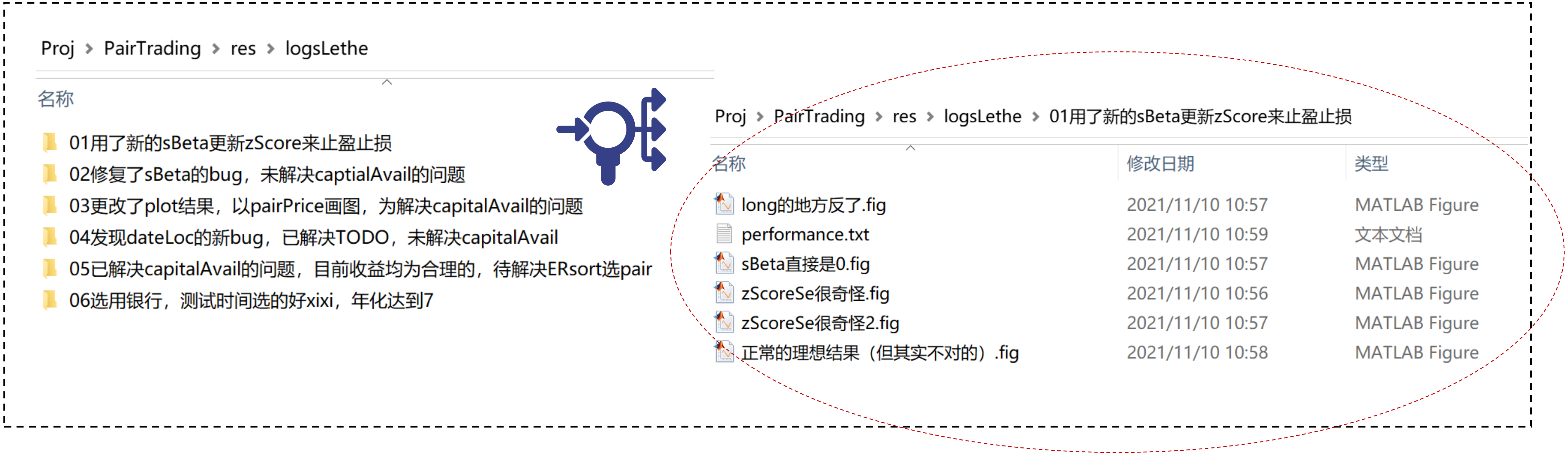


```
% 更新pairID，计算各种postion和ticker
pairIDrecent = pairIDrecent + 1;
loc = targetShortPairPositionLoc(i);
stockYLoc = currShortPairYLoc(loc);
stockYPosition = targetShortPairPosition(loc);
stockXLoc = currShortPairXLoc(loc);
stockXPosition = floor(stockYPosition*abs(currShortPairXBeta(loc))/100)*100;
pairER = currShortPairER(loc);
openPriceY = currShortPairYPrice(loc);
openPriceX = currShortPairXPrice(loc);
pairPrice = openPriceY*stockYPosition+openPriceX*stockXPosition;
```



Logs!

- Better to have log in your code
- Record important parameters and results
- Artificial Log…



Proj › PairTrading › res › logsLethe

名称

- 01用了新的sBeta更新zScore来止盈止损
- 02修复了sBeta的bug，未解决captialAvail的问题
- 03更改了plot结果，以pairPrice画图，为解决capitalAvail的问题
- 04发现dateLoc的新bug，已解决TODO，未解决capitalAvail
- 05已解决capitalAvail的问题，目前收益均为合理的，待解决ERsort选pair
- 06选用银行，测试时间选的好xixi，年化达到7

Proj › PairTrading › res › logsLethe › 01用了新的sBeta更新zScore来止盈止损

名称	修改日期	类型
long的地方反了.fig	2021/11/10 10:57	MATLAB Figure
performance.txt	2021/11/10 10:59	文本文档
sBeta直接是0.fig	2021/11/10 10:57	MATLAB Figure
zScoreSe很奇怪.fig	2021/11/10 10:56	MATLAB Figure
zScoreSe很奇怪2.fig	2021/11/10 10:57	MATLAB Figure
正常的理想结果（但其实不对的）.fig	2021/11/10 10:58	MATLAB Figure

Thank you for listening!

Thanks for the help of Jinze He and Jizhong Cao!