**Matlab获取K线和因子数据方法**

**注：函数API注意输入参数的数据结构，输出参数的数据结构。**

**1、策略内部获取行情和因子数据**

策略内获取数据，意思为在一个完整的策略中获取数据，获取的API函数一般是先注册数据，再获取数据。

**K线行情数据**

K线数据获取API：

1. KIdx = traderRegKData(KFrequency, KFreNum);
2. data = traderGetRegKData(KIdx, length, fillup);



注册数据traderRegKData在if bInit语句中，获取数据traderGetRegKData在else语句中。

**行情数据结构（最需要注意和理解的地方）：**

行情数据输出参数是矩阵，每8行为一个标的的数据

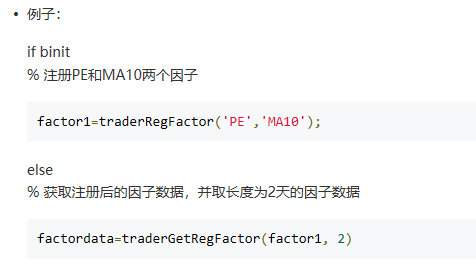


**因子数据**

因子数据API：

1) Idx = traderRegFactor(FactorName1，FactorName2，……)；

2) factor = traderGetRegFactor(Idx, length);



因子数据结构：

参数输出数据结构为矩阵，与行情数据不同的是，注意第一行数据为时间。



**举个例子：**

主函数M文件：

function factortest(bInit,bDayBegin,cellPar)

%% 函数说明

% 上面的三个参数是一种固定结构。

% bInit,在策略逻辑运行前为1，类似优矿等平台的initialize函数。当bInit为0时，交易启动。

% bDayBegin判断是否为一天的开始。

%% 全局变量声明 %%

global g\_idxK; % 注册数据的index.

global TLen; % 标的数量

global factor; % 注册因子

%% 传递参数 %%

len = cellPar{1};

%% 程序开始 %%

if bInit

%% 初始化回测帐户，注册数据 %%

traderSetParalMode(false);

g\_idxK = traderRegKData('day',1); % 注册K线日数据

factor = traderRegFactor('PB','PE','RSI','DA'); % 注册因子数据（市盈率、总市值、总资产）

TLen = length(g\_idxK(:,1)); % 标的个数

warning ('off');

else

%% K线数据 %%

% 获取len+1根bar数据 [1.time 2.open 3.high 4.low 5.close 6.vol 7.turn 8.oi]

% 数据结构为 （8\*N）行，len+1列的数据，N为标的数量

datas = traderGetRegKData(g\_idxK,len+1,true);

lendatas = length(datas(:,1));

time = datas(1:8:lendatas,:);

open = datas(2:8:lendatas,:); % 获取所有标的开盘价的数据

high = datas(3:8:lendatas,:);

low = datas(4:8:lendatas,:);

close = datas(5:8:lendatas,:); % 获取所有标的收盘价的数据

if isnan(close(1)) % 收盘价要全部有数据

return;

end

%% BP因子数据 %%

% 因子数据结构为（M\*N+1）行，len+1 列的数据，第一行为时间，M为标的数量，N为因子数量

factordata = traderGetRegFactor(factor,len+1);

lenfactor = length(factordata(:,1)); % 因子矩阵的总行数

PB = factordata(2:4:lenfactor,:);

PE = factordata(3:4:lenfactor,:); % 从整体的因子数据中获取单个因子PB的数据

RSI = factordata(4:4:lenfactor,:);

ROE = factordata(5:4:lenfactor,:);

[a,b] = size(datas);

[c,d] = size(factordata);

disp(['今天回测日期为',datestr(time(end),29),'；标的个数为',num2str(TLen),'；因子矩阵为',num2str(c),'\*',num2str(d),'；K线数据矩阵为',num2str(a),'\*',num2str(b)]) %调试专用

%% 策略逻辑计算 %%

% XX

% XX

%% 策略下单 %%

% XX

% XX

end

运行脚本函数：

clear;

clc;

%% 标的信息 %%

% 标的为所有A股

A = traderGetCodeList('sse\_a');

for m = 1:length(A)

targetList(m).Market = A(m).Market;

targetList(m).Code = A(m).Code;

end

B = traderGetCodeList('szse\_a');

for n = 1:length(B)

targetList(n+length(A)).Market = B(n).Market;

targetList(n+length(A)).Code = B(n).Code;

end

% %可将标的换成指定具体的标的，回测速度大大加快

% targetList(1).Market = 'sse';

% targetList(1).Code = '600000'; % 浦发银行

% targetList(2).Market = 'sse';

% targetList(2).Code = '601699'; % 潞安环能

%% 参数设置 %%

len = 30; %滑窗长度

Freq = 1; %刷新频率

begintime = 20160401; %开始回测时间

endtime = 20160901; %结束回测时间

%% 账户资金、手续费、滑点设置 %%

initial = 10000000;

traderSetBacktest(initial,0.000026,0.02,0,1,0,0);

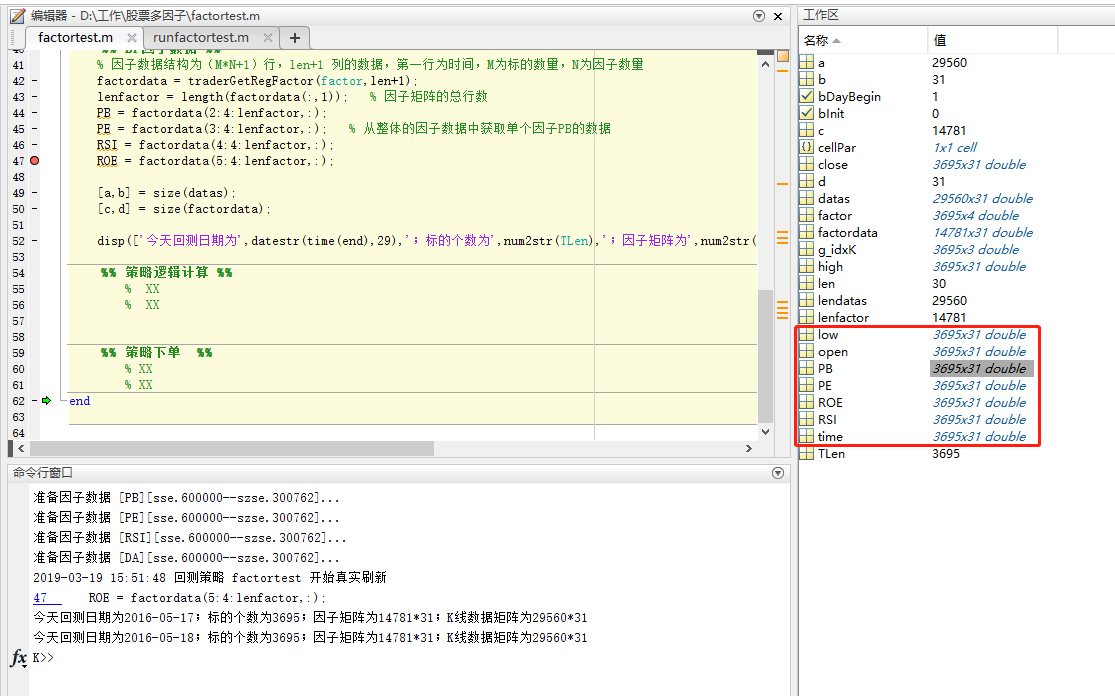
AccountList(1) = {'FutureBackReplay'};

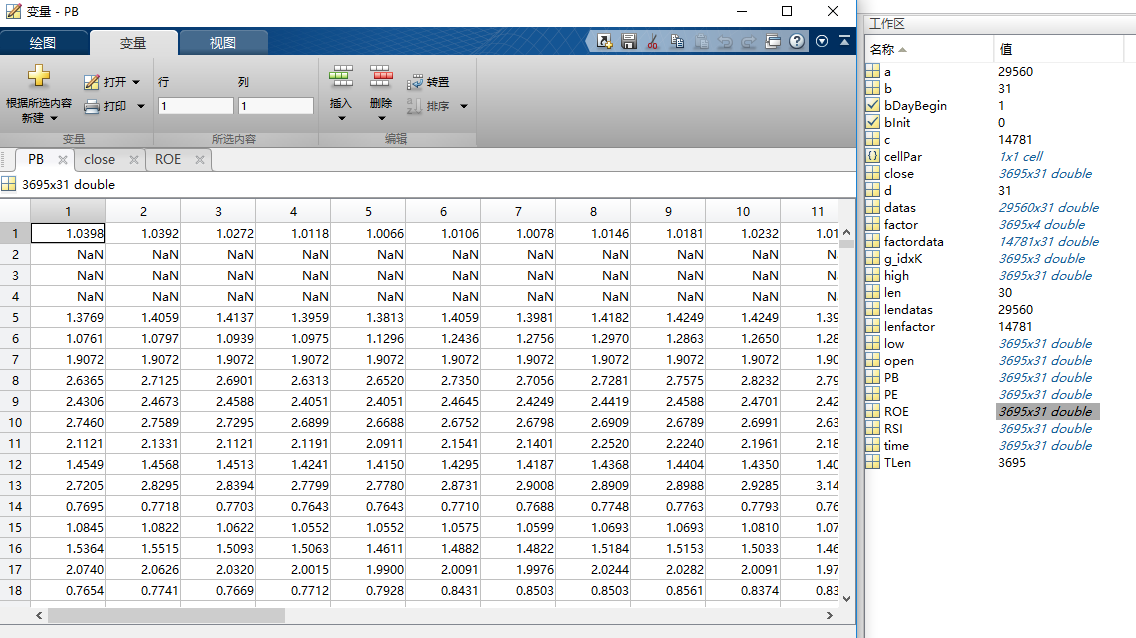
%% 回测函数设置 %%

traderRunBacktestV2('factortest',@factortest,{len},AccountList,targetList,'day',Freq,begintime,endtime,'FWard');

结果：

策略在运行的时候是看不出数据的类型和结构的，可以在主函数中添加断点调试的方式，逐步观察数据的变化。下面结果为数据断点调试的结果：





**2、策略外获取因子数据**

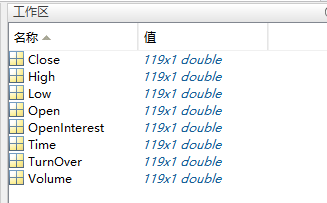
策略外意思是不是一个完整的策略当中，是随时可获取的数据，可视化效果好。

行情数据API：

traderGetKData(Market, Code, KFrequency, KFreNum, BeginDate, EndDate, Filledup, FQ);

[Time, Open, High, Low, Close, Volume, TurnOver, OpenInterest] = traderGetKData('SZSE', '000004', 'day', 1, 20160101, 20161231, false, 'FWard');

输出：



因子数据API：

traderGetFactor(FactorName, TargetList, BeginDate, EndDate)；



例子：

TargetList(1).Market = 'sse';

TargetList(1).Code = '600000'; % 浦发银行

TargetList(2).Market = 'sse';

TargetList(2).Code = '601699'; % 潞安环能

[Date, targetList, Data] = traderGetFactor('PE', TargetList, 20160301,20161001)

