

法律声明

本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。



加班主任入群

《进阶！量化交易实战：迭代式的量化策略研发》第5期

第7课：期货趋势型策略开发 主 讲：宋战江

内容介绍



趋势型量化策略必备的期货知识

期货量化工具介绍

趋势型策略原理

用TB开发商品期货趋势型策略

多空双向 T+0 杠杆交易

趋势型量化策略必备的期货知识

典型的交易标的

- ☐ 股票
- ☐ 期货
- ☐ 债券
- ☐ 外汇
- ☐ 期权
- ☐ 数字货币
- ☐



期货交易的基本特征

- 合约标准化
- 交易集中化
- 双向交易和对冲机制
- 杠杆机制
- 每日无负债结算制度
- T+0交易

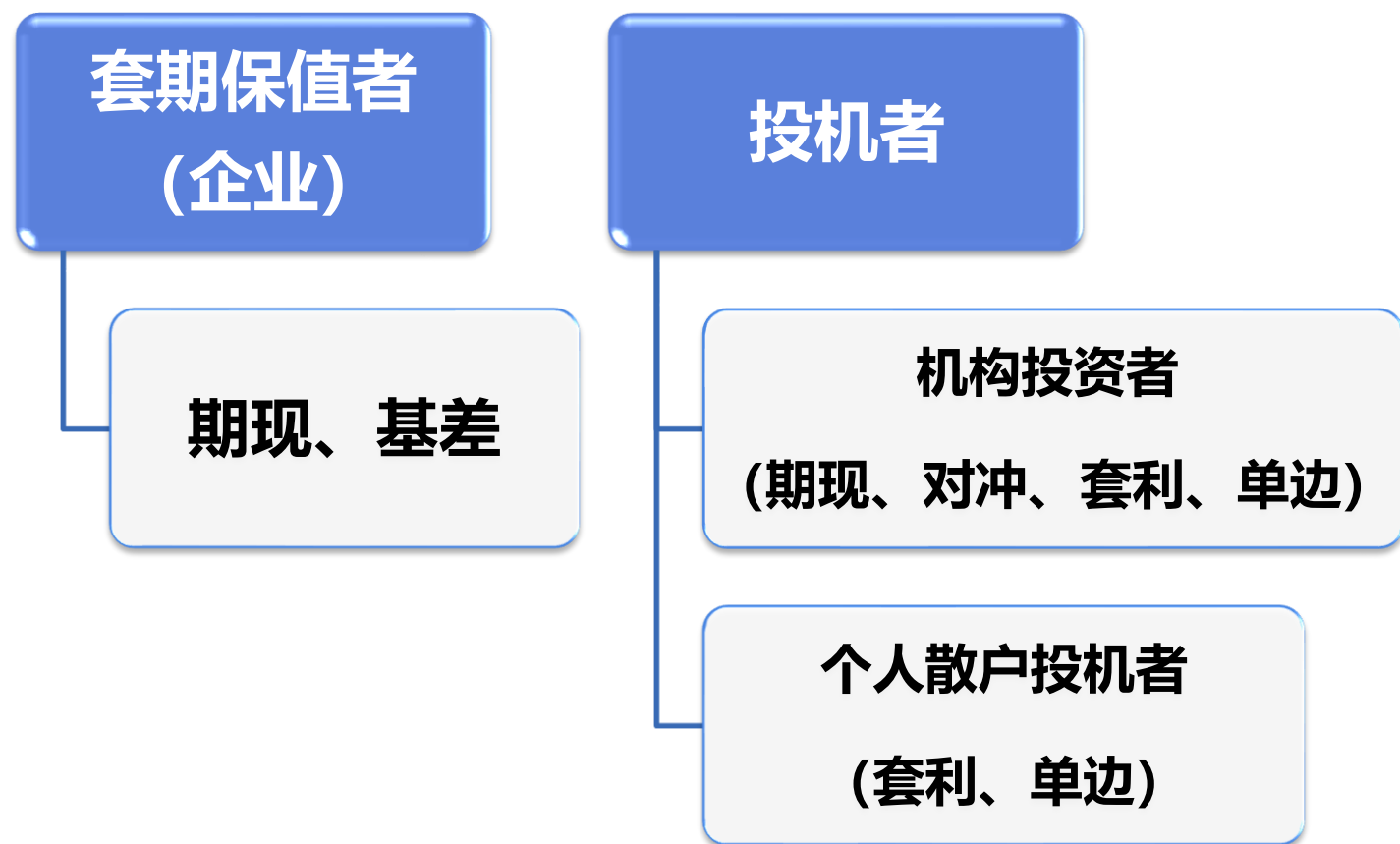
交易所名称	交易品种
上海期货交易所	黄金、白银、铜、铝、锌、橡胶、 燃油、螺纹钢、线材、石油、沥青、铅
郑州商品交易所	小麦、棉花、白糖、PTA、绿豆、红小 豆、花生、菜油籽、玻璃
大连商品交易所	大豆、豆粕、玉米、豆油、啤酒、焦 炭、棕榈油、聚氯乙烯
中国金融期货交易所	沪深300指数、5年期国债期货

螺纹钢 期货合约

交易品种	螺纹钢
交易单位	10吨/手
报价单位	元（人民币）/吨
最小变动价位	1元/吨
涨跌停板幅度	不超过上一交易日结算价 $\pm 7\%$
合约交割月份	1、2、3、4、5、6、7、8、9、10、11、12月
交易时间	每周一至周五 上午：9:00—10:15 10:30—11:30 下午：13:30—15:00 连续交易时间：21:00至23:00
最后交易日	合约交割月份的15日（遇法定假日顺延）
最后交割日	最后交易日后第五个工作日
交割品级	标准品：符合国标GB1499.2-2007《钢筋混凝土用钢 第2部分：热轧带肋钢筋》HRB400或HRBF400牌号的 $\phi 16\text{mm}$ 、 $\phi 18\text{mm}$ 、 $\phi 20\text{mm}$ 、 $\phi 22\text{mm}$ 、 $\phi 25\text{mm}$ 螺纹钢。 替代品：符合国标GB1499.2-2007《钢筋混凝土用钢 第2部分：热轧带肋钢筋》HRB335或HRBF335牌号的 $\phi 16\text{mm}$ 、 $\phi 18\text{mm}$ 、 $\phi 20\text{mm}$ 、 $\phi 22\text{mm}$ 、 $\phi 25\text{mm}$ 螺纹钢。

交割地点	交易所指定交割仓库
最低交易保证金	合约价值的9%
交割方式	实物交割
交易代码	RB
上市交易所	上海期货交易所
最小交割单位	300吨
上市日期	2009年03月27日

期货市场的构成



期货投资分析方法综述

技术分析

- ★ 市场行为涵盖一切信息
- ★ 价格沿趋势移动
- ★ 历史会重演

基本分析

- ★ 因果关系明确，结论确定
- ★ 完备的即时资料
- ★ 极高的行业经验专业素质

程序化交易

- ★ 技术分析程序化
- ★ 数量统计分析、交易
- ★ 高频交易

价·
量·
时·
空·
仓

期货投资者的通病

- ❑ 满仓操作
- ❑ 逆势开仓
- ❑ 持仓综合症
- ❑ 判顶断底
- ❑ 死不认输
- ❑ 逆势抢反弹
- ❑ 频繁全天候操作
- ❑ 把握不好进出场时机
- ❑ 盲目跟风
- ❑ 主力盯单心态

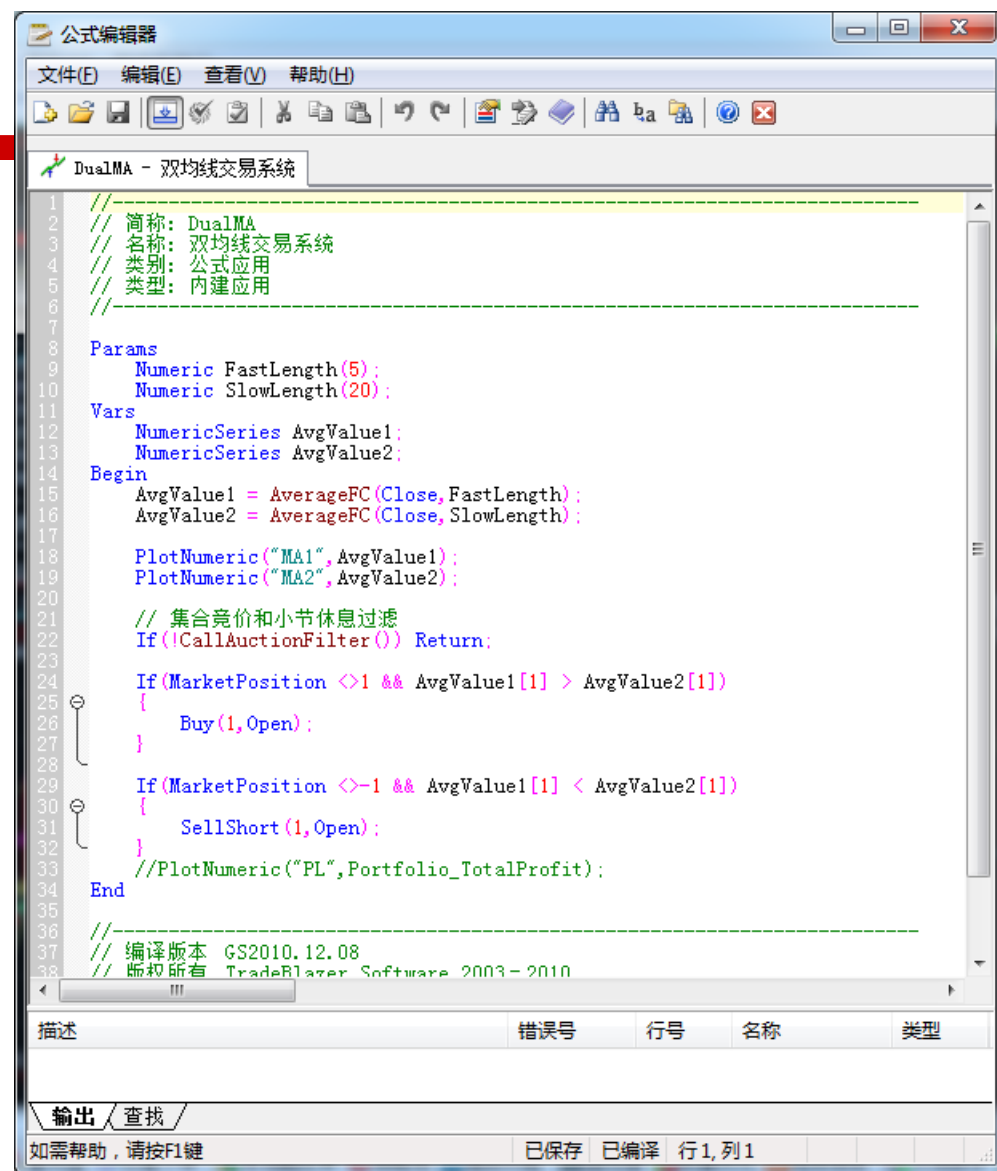
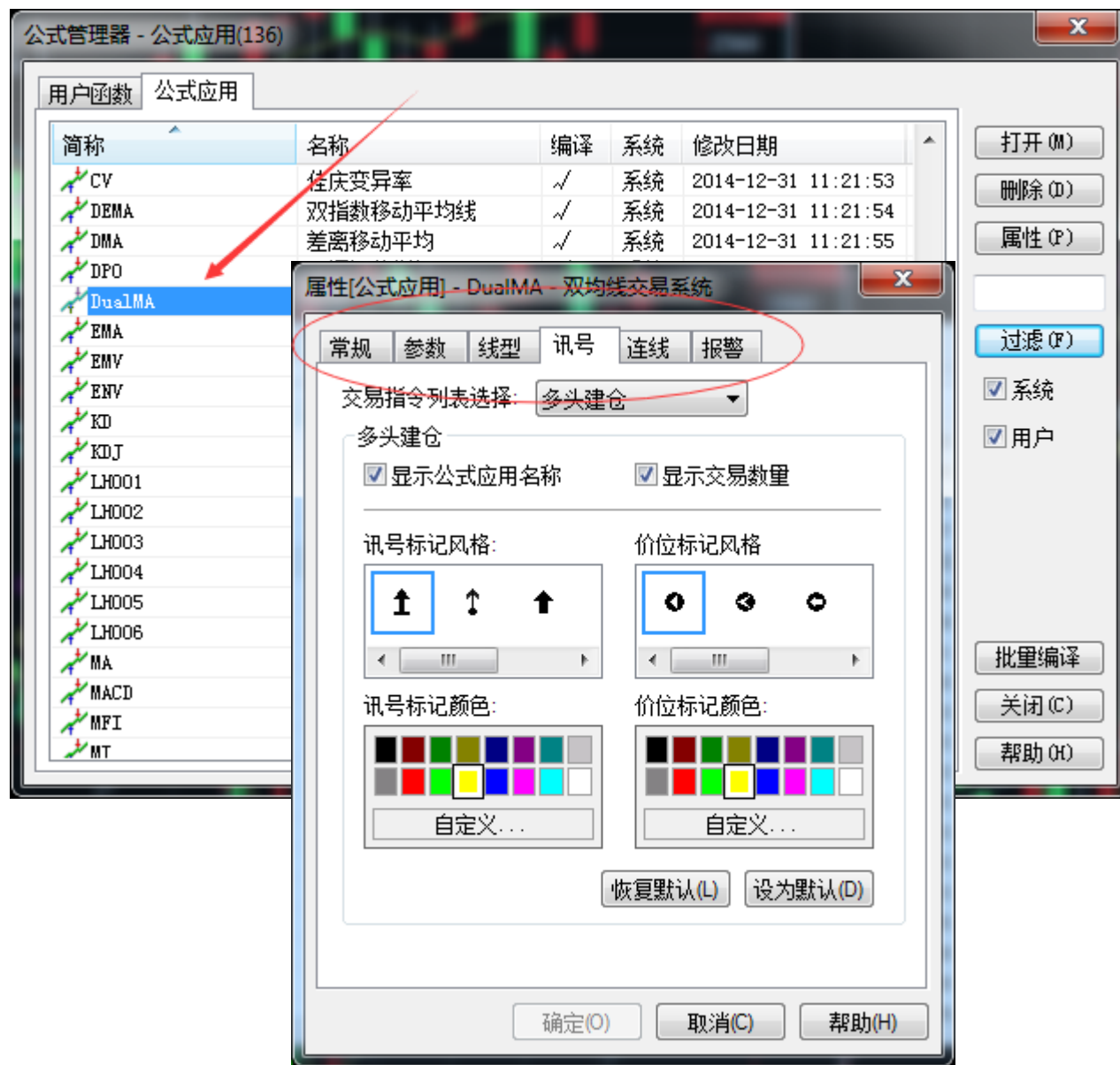
工欲善其事 必先利其器

期货量化工具介绍

期货量化平台选择

- ☐ TB（交易开拓者）
- ☐ MC（Multicharts）
- ☐ 金字塔
- ☐ WH8（文华赢智）
- ☐ VN.PY（开源框架）
- ☐ 自行研发







VN.PY

- ❑ 丰富的Python交易和数据API接口
- ❑ 事件驱动引擎 (vn.event)
- ❑ 开发示例 (vn.demo)
- ❑ 交易平台 (vn.trader)
- ❑ RPC模块 (vn.rpc)
- ❑ 官方网站 (www.vnpy.com) /知乎专栏
- ❑ 官方交流QQ群

```
eventEngine.py: vt.wpr: Wing IDE
File Edit Source Refactor Project Debug Testing Git Tools Window Help
eventEngine.py vtEngine.py
MainEngine ▾ __init__ ▾
170
171 #-----
172 def exit(self):
173     """退出程序前调用, 保证正常退出"""
174     # 安全关闭所有接口
175     for gateway in self.gatewayDict.values():
176         gateway.close()
177
178     # 停止事件引擎
179     self.eventEngine.stop()
180
181     # 保存数据引擎里的合约数据到硬盘
182     self.dataEngine.saveContracts()
183
184 #-----
185 def writeLog(self, content):
186     """快速发出日志事件"""
187     log = VtLogData()
188     log.logContent = content
189     event = Event(type_=EVENT_LOG)
190     event.dict['data'] = log
191     self.eventEngine.put(event)
192
193 #-----
194 def dbConnect(self):
195     """连接MongoDB数据库"""
196     if not self.dbClient:
197         try:
198             self.dbClient = MongoClient()
199             self.writeLog(u'MongoDB连接成功')
200         except ConnectionFailure:
201             self.writeLog(u'MongoDB连接失败')
202
203 #-----
eventEngine.py vtEngine.py
test ▾ simpletest ▾
12 from eventType import *
13
14
15 #####
16 class EventEngine(object):
17     """
18     事件驱动引擎
19
20     事件驱动引擎中所有的变量都设置为了私有, 这是为了防止不小心
21     从外部修改了这些变量的值或状态, 导致bug。
22
23     变量说明
24     __queue: 私有变量, 事件队列
25     __active: 私有变量, 事件引擎开关
26     __thread: 私有变量, 事件处理线程
27     __timer: 私有变量, 计时器
28     __handlers: 私有变量, 事件处理函数字典
29
30     方法说明
31     __run: 私有方法, 事件处理线程连续运行用
32     __process: 私有方法, 处理事件, 调用注册在引擎中的监听函数
33     __onTimer: 私有方法, 计时器固定事件间隔触发后, 向事件队列中存入计时器事件
34     start: 公共方法, 启动引擎
35     stop: 公共方法, 停止引擎
36     register: 公共方法, 向引擎中注册监听函数
37     unregister: 公共方法, 向引擎中注销监听函数
38     put: 公共方法, 向事件队列中存入新的事件
39
40     事件监听函数必须定义为输入参数仅为一个event对象, 即:
41
42     函数
43     def func(event)
44         ...
45 """
```

CTA策略

加载策略 全部初始化 全部启动 全部停止

double ema

初始化 启动 停止

name	className	author	vtSymbol	fastK
double ema	DoubleEm...	用Python...	IF1602	0.9

inited	trading	pos	fastMa0	fastMa1
False	False	0	0.0	0.0

21:43:35 CTA引擎启动成功
21:43:36 double ema的交易合约IF1602无法找到
21:43:36 策略加载成功

VnTrader

系统 功能 算法 帮助

代码 p1609 卖五
名称 棕榈油p1609 卖四
方向类型 多 卖三
开平 开仓 卖二
价格 0.0000 卖一 5518.0 1355
数量 0 最新 5516.0 -0.04%
价格类型 限价 买一 5516.0 263
交易所 买二
货币 买三
产品类型 买四
交易接口 买五

发单
全撤

合约代码	名称	最新价	成交量	持仓量	开盘价	最高价	最低价	买一价	买一量	卖一价	卖一量	时
p1609	棕榈油p1609	5516.0	200268	788240	5504.0	5520.0	5482.0	5516.0	263	5518.0	1355	21:44
p1608	棕榈油p1608	5476.0		6				5464.0	1	5570.0	1	99:99
p1607	棕榈油p1607	5510.0		2				5400.0	1	5498.0	1	99:99
p1606	棕榈油p1606	5436.0		14				5338.0	1	5434.0	1	99:99
p1605	棕榈油p1605	5426.0	9212	145462	5430.0	5432.0	5406.0	5426.0	311	5428.0	84	21:44
m1609	豆粕m1609	2341.0	177590	2653822	2340.0	2346.0	2337.0	2341.0	1349	2342.0	1187	21:44
m1608	豆粕m1608	2351.0		146				2348.0	6	2354.0	1	21:43
m1607	豆粕m1607	2368.0		324				2345.0	3	2362.0	6	21:44
m1605	豆粕m1605	2306.0	4742	421072	2305.0	2309.0	2302.0	2305.0	94	2306.0	12	21:44
IF1609	沪深300IF1609	3015.0	458	4051	3015.4	3023.2	2994.6	3013.2	1	3017.4	1	99:99
IF1606	沪深300IF1606	3139.0	1100	7759	3139.6	3144.8	3124.0	3139.0	1	3142.0	2	99:99
IF1605	沪深300IF1605	3193.8	1040	1602	3191.0	3199.0	3179.6	3191.2	1	3194.6	1	99:99
IF1604	沪深300IF1604	3239.4	14521	35953	3227.8	3244.4	3222.2	3239.0	2	3239.2	1	99:99

日志 错误 账户

账户	昨结	净值	可用	手续费	保证金	平仓盈亏	持仓盈亏	接口
000200002874	9619242413.52	9613794263.52	9528094488.52		85699771.0		-5448150.0	XSPEED

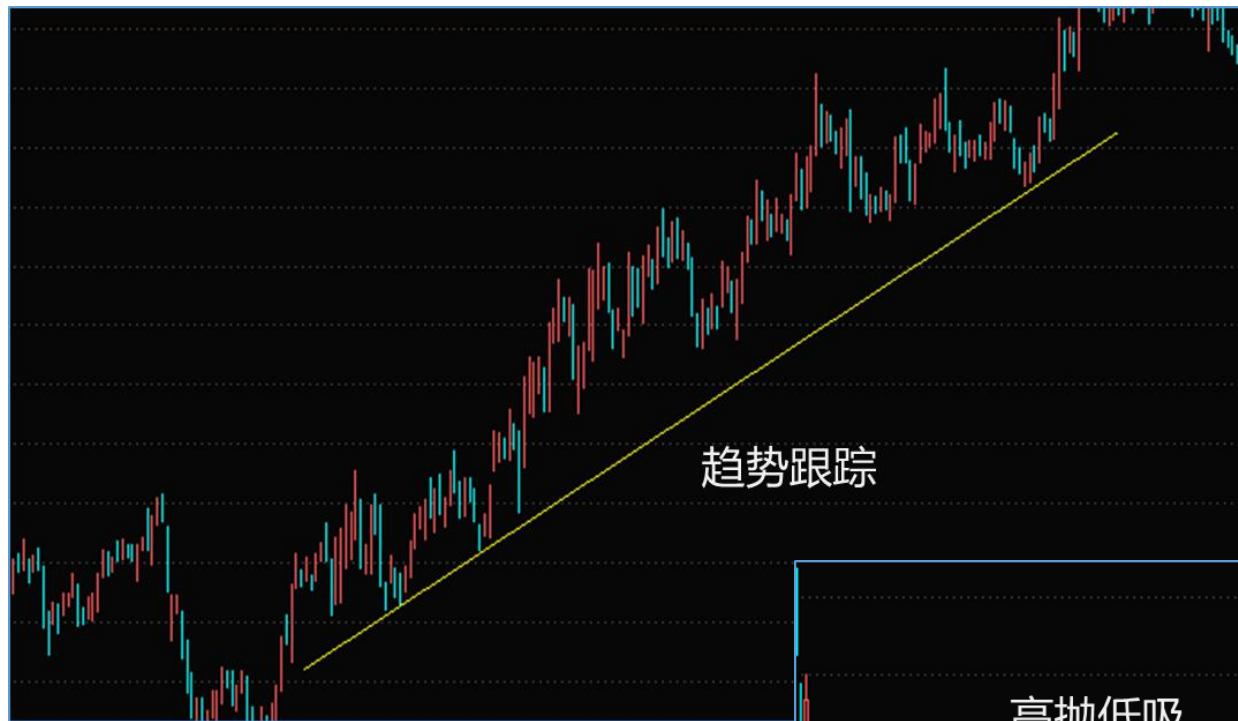
成交 委托 持仓

合约代码	名称	方向	持仓量
p1605	棕榈油p1605	空	1700
i1606	铁矿石i1606	空	30
i1606	铁矿石i1606	多	1
i1605	铁矿石i1605	空	37010
i1605	铁矿石i1605	多	9

以“海龟交易法则”为例

趋势型策略原理

常见的交易策略类型



交易系统的核心要素

- 市场：买卖什么？
- 头寸规模：买卖多少？
- 入市：什么时候买进？
- 止损：什么时候放弃一个亏损的头寸？
- 退出：什么时候退出一个盈利的头寸？
- 战术：怎么买卖？

海龟交易系统就是一个
包含了上述所有要素的
“完整的”交易系统

头寸规模计算

- 波动性N: $N = (19 * PDN + TR) / 20$
 - PDN = 前一日的N值
 - TR = 当日的真实波动幅度
- 真实波幅 = $\text{Max} (H - L, |H - PDC|, |PDC - L|)$
 - H = 当日最高价
 - L = 当日最低价
 - PDC = 前一日收盘价
- 头寸规模单位 (手数)
= 账户的 1% / ($N * \text{每一最小交易单位}$)

(举例)

- 假设股票中国平安的N值 ($\approx$ ATR) 为5元
 - 账户的本金为100万元, 那么账户的1%就是10000元
 - 最小交易单位是1手股票, 也就是100股
- 那么对于中国平安这只股票, 可以分配的头寸规模单位是:
 - $10000 / (5 * 100) = 20$ (手股票)
- 即: 给中国平安分配的头寸规模单位是20手

□ 系统1：以20日突破为基础的短期系统

- 如上次突破是赢利性突破，那么当前的入市信号将被忽略
- 55日突破点保障信号

□ 系统2：以55日突破为基础的长期系统

□ 在突破点建立1个单位的头寸

- 按 $N/2$ 的价格间隔逐步扩大头寸
- 以上一份订单的实际成交价为基准

□ 直到总头寸达到规模上限

止损

退出

- 任何一笔交易风险程度不超过账户的2%

- 价格变动上限是 $2N$

- 对于加仓情况，止损点上浮

- 系统1:

- 10日反向突破退出

- 系统2:

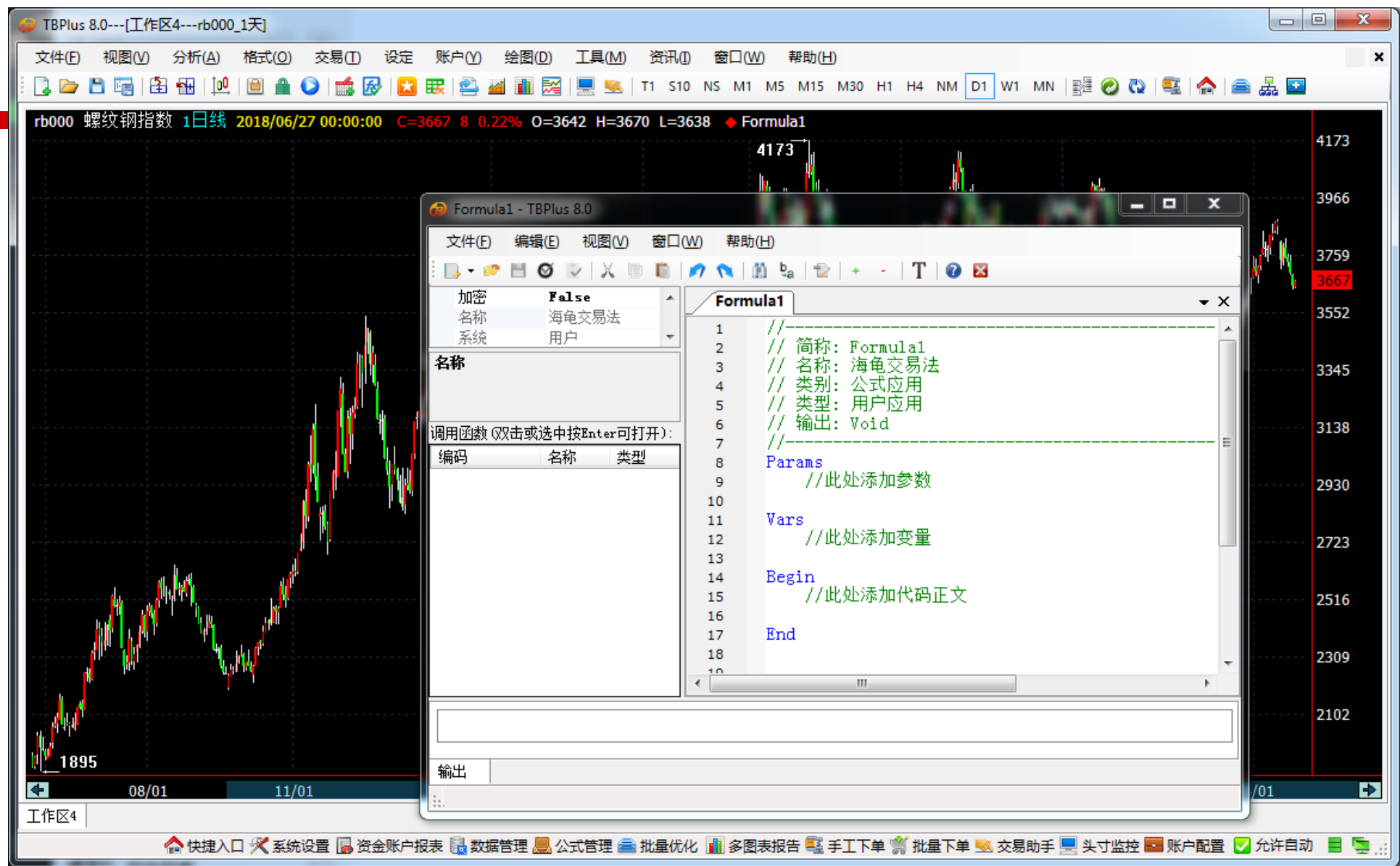
- 20日反向突破退出

动手实践，从策略思想到编程实现，逐步求精

用TB开发商品期货趋势型策略

环境准备

- ❑ 运行TB，登录，新建工作区
- ❑ 加载交易品种：螺纹钢（指数），日K级别
- ❑ 设置商品，增加样本（K线）数到3000
- ❑ 设置公式应用，允许连续建仓，最多10次，最大持仓限制增加到2000合约
- ❑ 新建“用户应用”Formula1；文件菜单，公式管理，找到Formula1，选中并执行编译
- ❑ 插入公式应用，加载Formula1
- ❑ 如下页图，开始代码编辑.....



休息一下
5分钟后回来

初步实现

□ 建仓信号

- 突破（20日）前高：开多信号
- 跌破（20日）前低：开空信号

□ 仓位计算（无持仓情况下）

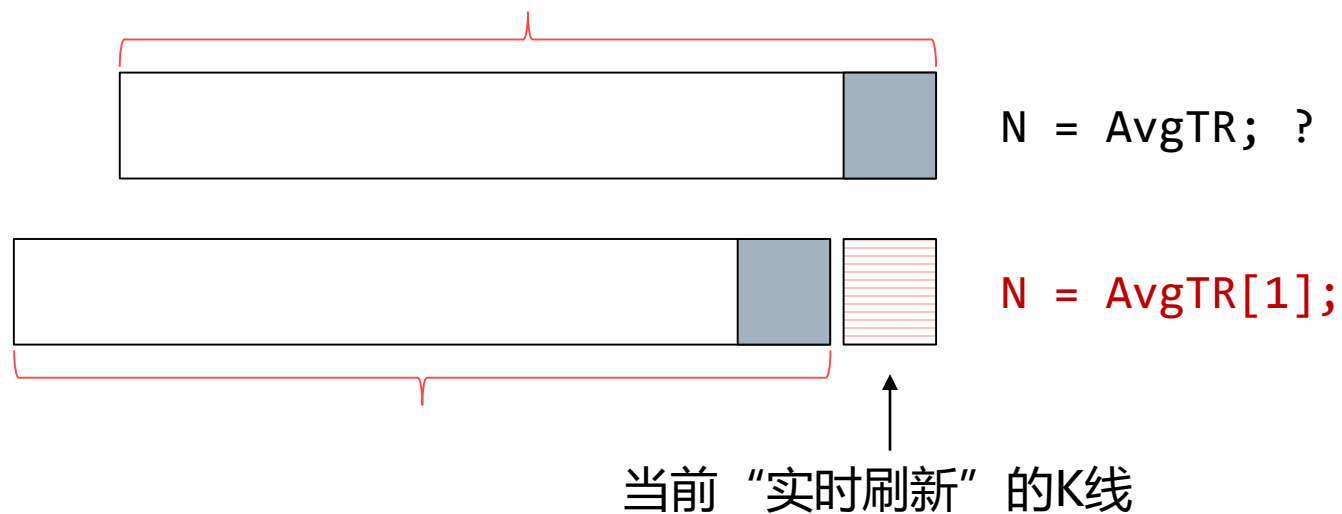
- 突破时建立一个初始的头寸单位

□ 退出（有持仓情况下）

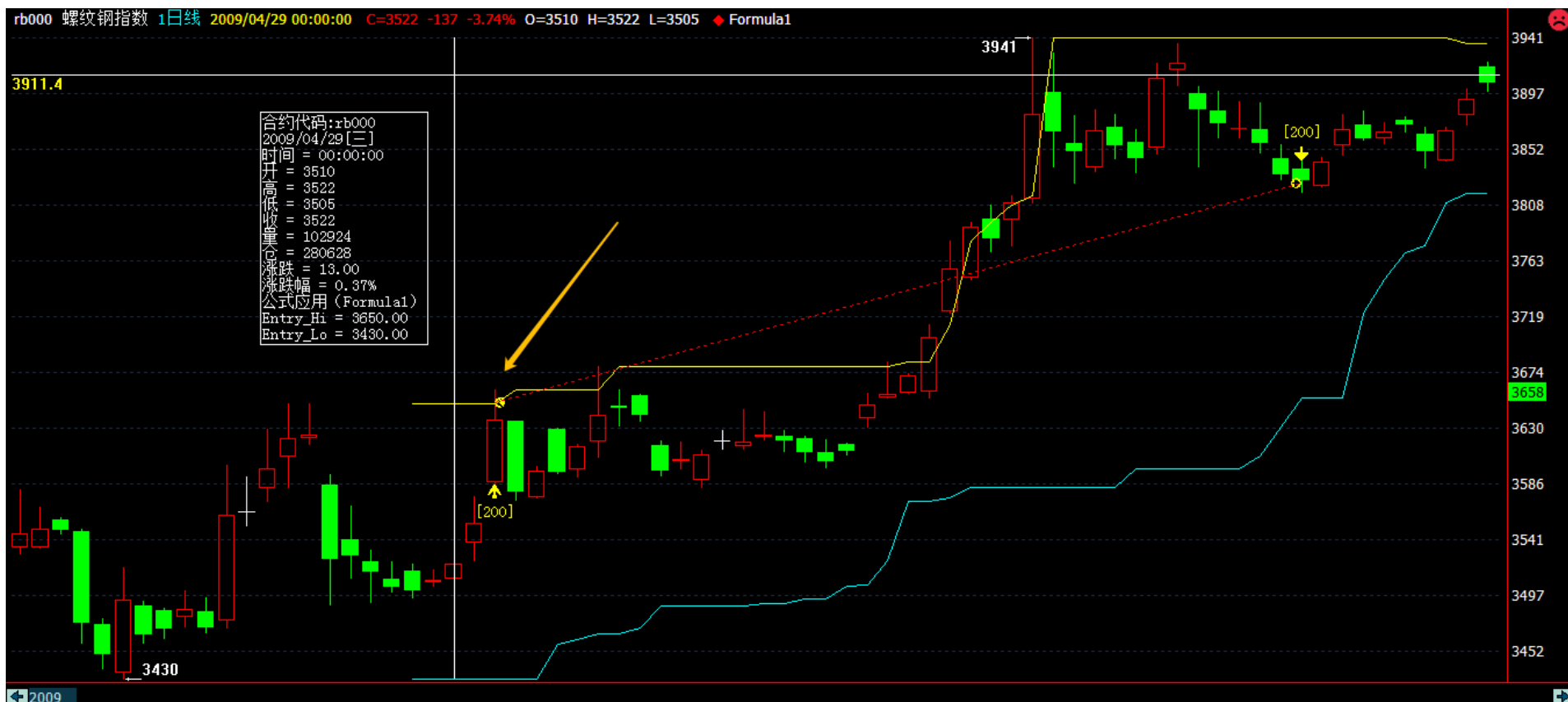
- 跌破（10日）前低时平多
- 突破（10日）前高时平空

序列变量平移取值

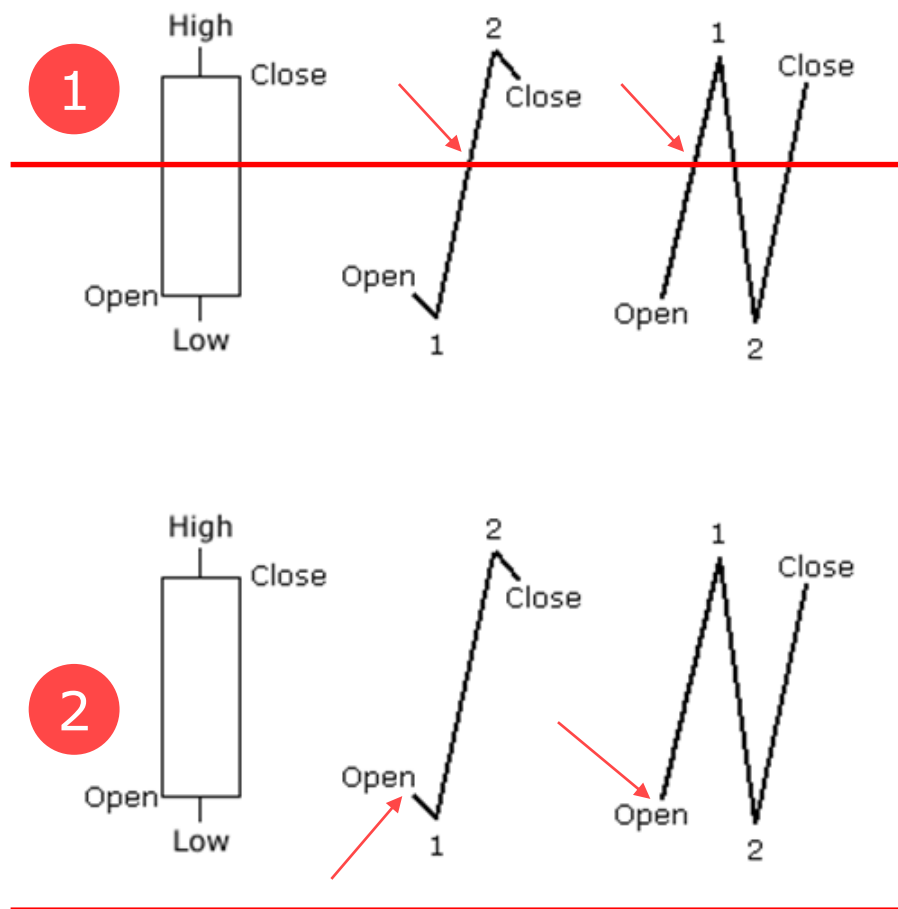
```
AvgTR = XAverage(TrueRange, ATRLength);  
N = AvgTR[1];
```



突破入场示意图



入场点代码解析



DonchianHi

```
// 多头开仓  
if (high > DonchianHi && TurtleUnits >= 1)  
{
```

1

```
myEntryPrice = min(high,  
DonchianHi+MinPoint);
```

2

```
myEntryPrice = IIF(myEntryPrice < Open,  
Open,  
myEntryPrice); // 应对跳空
```

```
Buy(TurtleUnits, myEntryPrice);
```

```
}
```

DonchianHi

Params

```
Numeric RiskRatio(1); // 1% risk ratio  
Numeric ATRLength(20); // ATR window size  
Numeric boLength(20); // breakout  
Numeric teLength(10); // trailing exit
```

Vars

```
Numeric N;  
NumericSeries AvgTR;  
Numeric TotalEquity;  
Numeric TurtleUnits;  
NumericSeries DonchianHi;  
NumericSeries DonchianLo;  
Numeric MinPoint;  
Numeric ExitLowestPrice;  
Numeric ExitHighestPrice;  
Numeric myEntryPrice;  
Numeric myExitPrice;
```

Begin

```
if (!CallAuctionFilter()) return;  
  
MinPoint = MinMove * PriceScale;  
AvgTR = XAverage(TrueRange, ATRLength);  
N = AvgTR[1];  
TotalEquity = Portfolio_CurrentCapital() + Portfolio_UsedMargin();  
TurtleUnits = (TotalEquity*RiskRatio/100) / (N*ContractUnit()*BigPointValue());  
TurtleUnits = IntPart(TurtleUnits);
```

【初步实现的（简化版）代码】

```
DonchianHi = HighestFC(High[1], boLength);
DonchianLo = LowestFC(Low[1], boLength);
// PlotNumeric("Entry_Hi", DonchianHi);
// PlotNumeric("Entry_Lo", DonchianLo);

ExitLowestPrice = LowestFC(Low[1], teLength);
ExitHighestPrice = HighestFC(High[1], teLength);

if (MarketPosition == 0)
{
    // 多头开仓
    if (high > DonchianHi && TurtleUnits >= 1)
    {
        myEntryPrice = min(high, DonchianHi+MinPoint);
        myEntryPrice = IIF(myEntryPrice < Open, Open, myEntryPrice); // 应对跳空情况
        Buy(TurtleUnits, myEntryPrice);
    }
    // 空头开仓
    if (Low < DonchianLo && TurtleUnits >= 1)
    {
        myEntryPrice = max(low, DonchianLo-MinPoint);
        myEntryPrice = IIF(myEntryPrice > Open, Open, myEntryPrice); // 应对跳空情况
        SellShort(TurtleUnits, myEntryPrice);
    }
}
```

```
if (MarketPosition == 1)
{
    if (Low < ExitLowestPrice)
    {
        // 多头退出
        myExitPrice = max(Low, ExitLowestPrice-MinPoint);
        myExitPrice = IIF(myExitPrice > Open, Open, myExitPrice); // 应对跳空情况
        Sell(0, myExitPrice);
    }
    else {
        // 多头加仓逻辑... 多头止损逻辑...
    }
}
else if (MarketPosition == -1)
{
    if (High > ExitHighestPrice)
    {
        // 空头退出
        myExitPrice = min(High, ExitHighestPrice+MinPoint);
        myExitPrice = IIF(myExitPrice < Open, Open, myExitPrice); // 应对跳空情况
        BuyToCover(0, myExitPrice);
    }
    else {
        // 空头加仓逻辑... 空头止损逻辑...
    }
}
End
```

【简化版回测结果】



【螺纹钢】总收益: 263.10%, 最大回撤: 14.98%, 胜率: 51.76%, 盈亏比: 2.49

完整逻辑

□ 增仓动作（有持仓情况下）

- 多头：相对于上次入场价格，每上涨 $N/2$ 就新增一个单位的多头头寸，每根K线内可多次加仓
- 空头：相对于上次入场价格，每下跌 $N/2$ 就新增一个单位的空头头寸，每根K线内可多次加仓

□ 退出（有持仓情况下）

- 新增规则：相对于上次入场价格，如超过最大允许的损失幅度（ $2*N$ ），则清仓
- 加仓所在的那根K线内，不进行止损

其它细节

- 长周期突破点保障信号
 - 入市过滤条件
- 前次突破失败与否的判别
 - 盈利性退出
 - 亏损性退出

```
if (MarketPosition == 0 && \  
    (!LastProfitableTradeFilter or PreBreakoutFailure))  
{  
    // ...  
}
```

Params

```
Numeric RiskRatio(1);           // % Risk Per N ( 0 - 100)
Numeric ATRLength(20);          // 平均波动周期 ATR Length
Numeric boLength(20);           // 短周期 BreakOut Length
Numeric fsLength(55);           // 长周期 FailSafe Length
Numeric teLength(10);           // 离市周期 Trailing Exit Length
Bool LastProfitableTradeFilter(True); // 使用入市过滤条件
```

Vars

```
Numeric MinPoint;               // 最小变动单位
NumericSeries AvgTR;            // ATR
Numeric N;                      // N 值
Numeric TotalEquity;            // 按最新收盘价计算出的总资产
Numeric TurtleUnits;           // 交易单位
NumericSeries DonchianHi;       // 唐奇安通道上轨, 延后1个Bar
NumericSeries DonchianLo;       // 唐奇安通道下轨, 延后1个Bar
NumericSeries fsDonchianHi;     // 唐奇安通道上轨, 延后1个Bar, 长周期
NumericSeries fsDonchianLo;     // 唐奇安通道下轨, 延后1个Bar, 长周期
Numeric ExitHighestPrice;       // 离市时判断需要的N周期最高价
Numeric ExitLowestPrice;        // 离市时判断需要的N周期最低价
Numeric myEntryPrice;           // 开仓价格
Numeric myExitPrice;            // 平仓价格
Bool SendOrderThisBar(False);   // 当前Bar有过交易
NumericSeries preEntryPrice(0); // 前一次开仓的价格
BoolSeries PreBreakoutFailure(false); // 前一次突破是否失败
```

Begin

```
// 集合竞价过滤
if (!CallAuctionFilter()) return;
```

【TB自带示例（完整版）代码】

```
If(BarStatus == 0)
{
    preEntryPrice = InvalidNumeric;
    PreBreakoutFailure = false;
}

MinPoint = MinMove*PriceScale;
AvgTR = XAverage(TrueRange,ATRLength);
N = AvgTR[1];
TotalEquity = Portfolio_CurrentCapital() + Portfolio_UsedMargin();
TurtleUnits = (TotalEquity*RiskRatio/100) /(N * ContractUnit()*BigPointValue());
TurtleUnits = IntPart(TurtleUnits); // 对小数取整

DonchianHi = HighestFC(High[1],boLength);
DonchianLo = LowestFC(Low[1],boLength);

fsDonchianHi = HighestFC(High[1],fsLength);
fsDonchianLo = LowestFC(Low[1],fsLength);

ExitLowestPrice = LowestFC(Low[1],teLength);
ExitHighestPrice = HighestFC(High[1],teLength);

Commentary("N="+Text(N));
Commentary("preEntryPrice="+Text(preEntryPrice));
Commentary("PreBreakoutFailure="+IIFString(PreBreakoutFailure,"True","False"));
```

```
// 当不使用过滤条件, 或者使用过滤条件并且条件为PreBreakoutFailure为True进行后续操作
If(MarketPosition == 0 && ((!LastProfitableTradeFilter) Or (PreBreakoutFailure)))
{
    // 突破开仓
    If(High > DonchianHi && TurtleUnits >= 1)
    {
        // 开仓价格取突破上轨+一个价位和最高价之间的较小值, 更接近真实情况, 并能尽量保证成交
        myEntryPrice = min(high,DonchianHi + MinPoint);
        myEntryPrice = IIF(myEntryPrice < Open, Open,myEntryPrice); // 跳空用开盘价代替
        preEntryPrice = myEntryPrice;
        Buy(TurtleUnits,myEntryPrice);
        SendOrderThisBar = True;
        PreBreakoutFailure = False;
    }

    If(Low < DonchianLo && TurtleUnits >= 1)
    {
        // 开仓价格取突破下轨-一个价位和最低价之间的较大值, 更接近真实情况, 并能尽量保证成交
        myEntryPrice = max(low,DonchianLo - MinPoint);
        myEntryPrice = IIF(myEntryPrice > Open, Open,myEntryPrice); // 跳空用开盘价
        preEntryPrice = myEntryPrice;
        SendOrderThisBar = True;
        SellShort(TurtleUnits,myEntryPrice);
        SendOrderThisBar = True;
        PreBreakoutFailure = False;
    }
}
```

```
// 长周期突破开仓 Failsafe Breakout point
If(MarketPosition == 0)
{
    Commentary("fsDonchianHi="+Text(fsDonchianHi));
    If(High > fsDonchianHi && TurtleUnits >= 1)
    {
        // 开仓价格取突破上轨+一个价位和最高价之间的较小值, 更接近真实情况, 并能尽量保证成交
        myEntryPrice = min(high,fsDonchianHi + MinPoint);
        myEntryPrice = IIF(myEntryPrice < Open, Open,myEntryPrice); // 跳空用开盘价代替
        preEntryPrice = myEntryPrice;
        Buy(TurtleUnits,myEntryPrice);
        SendOrderThisBar = True;
        PreBreakoutFailure = False;
    }

    Commentary("fsDonchianLo="+Text(fsDonchianLo));
    If(Low < fsDonchianLo && TurtleUnits >= 1)
    {
        // 开仓价格取突破下轨-一个价位和最低价之间的较大值, 更接近真实情况, 并能尽量保证成交
        myEntryPrice = max(low,fsDonchianLo - MinPoint);
        myEntryPrice = IIF(myEntryPrice > Open, Open,myEntryPrice); // 跳空用开盘价代替
        preEntryPrice = myEntryPrice;
        SellShort(TurtleUnits,myEntryPrice);
        SendOrderThisBar = True;
        PreBreakoutFailure = False;
    }
}
```

```

If(MarketPosition == 1) // 有多仓的情况
{
    Commentary("ExitLowestPrice="+Text(ExitLowestPrice));
    If(Low < ExitLowestPrice)
    {
        myExitPrice = max(Low,ExitLowestPrice - MinPoint);
        myExitPrice = IIF(myExitPrice > Open, Open,myExitPrice); // 跳空用开盘价代替
        Sell(0,myExitPrice);    // 数量用0的情况下将全部平仓
    }Else
    {
        If(preEntryPrice!=InvalidNumeric && TurtleUnits >= 1)
        {
            If(Open >= preEntryPrice + 0.5*N) // 如开盘超过设定的1/2N,则直接用开盘价增仓
            {
                myEntryPrice = Open;
                preEntryPrice = myEntryPrice;
                Buy(TurtleUnits,myEntryPrice);
                SendOrderThisBar = True;
            }

            while(High >= preEntryPrice + 0.5*N) // 以最高价为标准, 判断能进行几次增仓
            {
                myEntryPrice = preEntryPrice + 0.5 * N;
                preEntryPrice = myEntryPrice;
                Buy(TurtleUnits,myEntryPrice);
                SendOrderThisBar = True;
            }
        }
    }
}

```

```

// 止损指令
If(Low <= preEntryPrice - 2 * N && SendOrderThisBar == false) // 加仓Bar不止损
{
    myExitPrice = preEntryPrice - 2 * N;
    myExitPrice = IIF(myExitPrice > Open, Open, myExitPrice); // 跳空用开盘价代替
    Sell(0, myExitPrice); // 数量用0的情况下将全部平仓
    PreBreakoutFailure = True;
}
}
}Else If(MarketPosition == -1) // 有空仓的情况
{
    // 求出持空仓时离市的条件比较值
    Commentary("ExitHighestPrice="+Text(ExitHighestPrice));
    If(High > ExitHighestPrice)
    {
        myExitPrice = Min(High, ExitHighestPrice + MinPoint);
        myExitPrice = IIF(myExitPrice < Open, Open, myExitPrice); // 跳空用开盘价代替
        BuyToCover(0, myExitPrice); // 数量用0的情况下将全部平仓
    }Else
    {
        If(preEntryPrice != InvalidNumeric && TurtleUnits >= 1)
        {
            If(Open <= preEntryPrice - 0.5*N) // 如开盘超过设定的1/2N,则直接用开盘价增仓
            {

```

```

        myEntryPrice = Open;
        preEntryPrice = myEntryPrice;
        SellShort(TurtleUnits,myEntryPrice);
        SendOrderThisBar = True;
    }

    while(Low <= preEntryPrice - 0.5*N) // 以最低价为标准, 判断能进行几次增仓
    {
        myEntryPrice = preEntryPrice - 0.5 * N;
        preEntryPrice = myEntryPrice;
        SellShort(TurtleUnits,myEntryPrice);
        SendOrderThisBar = True;
    }
}

// 止损指令
If(High >= preEntryPrice + 2 * N &&SendOrderThisBar==false) // 加仓Bar不止损
{
    myExitPrice = preEntryPrice + 2 * N;
    myExitPrice = IIF(myExitPrice < Open, Open,myExitPrice); // 跳空用开盘价代替
    BuyToCover(0,myExitPrice); // 数量用0的情况下将全部平仓
    PreBreakoutFailure = True;
}

}

}

End

```



【完整版回测结果】



【螺纹钢】总收益：700.22%，最大回撤：21.79%，胜率：50.09%，盈亏比：2.01

对比 - 【简化版回测结果】



【螺纹钢】总收益：263.10%，最大回撤：14.98%，胜率：51.76%，盈亏比：2.49

思考

- 完整实现的效果一定比简化版好吗？
- 思考问题的本质
 - 突破：趋势型策略，顺势加仓，在什么情况下回调风险大？
 - 对品种、周期的选择有什么要求？
- 尝试改写完整版的代码？
 - 同时进行跨市场、多品种的交易
- 测试不同级别：60分钟、30分钟、周线
- 测试多个品种：RB -> J, RU, PTA

期货交易策略的注意事项

- 移仓换月（主力合约切换）
- 杠杆效应（保证金交易）
- 波动幅度（隔夜跳空风险）
- 套利机会（跨期、跨品种）

课后练习

- 用TB测试海龟交易法在不同周期、不同品种上的表现，撰写测试报告
 - 测试不同级别：60分钟、30分钟、周线
 - 测试多个品种：RB -> J, RU, PTA
 - 对比完整版和简化版的表现，分析原因

下节课预告

☐ 题目：用Python开发均值回复型股票策略

- 编程语言：Python

- 运行平台：一创·聚宽 - <https://ycjq.95358.com/>

- 准备工作：

- ☐ 提前在平台上注册账号

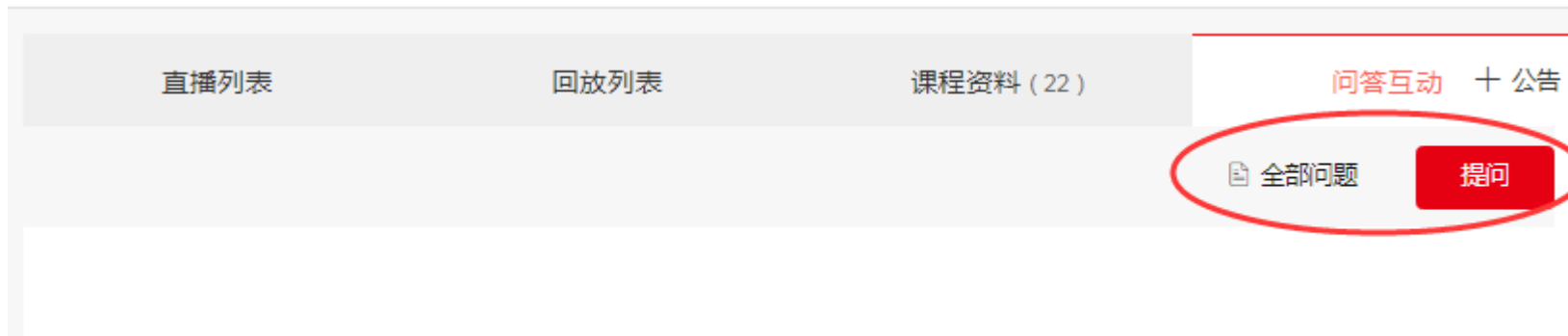
- ☐ 阅读并熟悉平台的API文档

- ☐ 自学Python和pandas

问答互动

在所报课的课程页面，

- 1、点击“全部问题”显示本课程所有学员提问的问题。
- 2、点击“提问”即可向该课程的老师 and 助教提问问题。



联系我们

小象学院：互联网新技术在线教育领航者

— 微信公众号：小象学院



THANKS