

# TB 共学计划第二周 TB 共学计划第三周

胡琛胡琛

2017 年 2 月 18 日

## 目录

1 软件介绍	4
2 建立图表	6
3 模拟账户注册	7
4 手工下单	7
5 公式管理	9
6 建立公式	11
6.1 举例：四周规则 . . . . .	11
7 加载公式和设置交易属性	13
8 性能测试	14
9 优化与批量优化	14
10 自动选参数	18
11 多品种组合优化	18
12 开启自动交易	18

目录	2
<b>13 头寸监控和交易助手</b>	<b>19</b>
<b>14 资金曲线</b>	<b>19</b>
<b>15 辅助功能</b>	<b>19</b>
<b>16 TB 公式</b>	<b>20</b>
16.1 初步认识 TB 公式 . . . . .	20
16.2 通过实例了解 TB 公式运行机制 . . . . .	21
<b>17 TB 语法基础</b>	<b>21</b>
17.1 TBL 语言基本元素介绍 . . . . .	21
17.2 命名规则 . . . . .	22
17.3 标点符号含义 . . . . .	23
17.4 操作运算符 . . . . .	23
<b>18 数据类型</b>	<b>24</b>
18.1 数据类型 . . . . .	24
18.1.1 基本类型 . . . . .	24
18.1.2 扩展类型 . . . . .	24
18.2 类型转换 . . . . .	25
<b>19 Bar 数据</b>	<b>26</b>
19.1 Bar 数据基本概念 . . . . .	26
19.2 序列数据与非序列数据区别 . . . . .	26
19.3 每根 Bar 包含哪些数据 . . . . .	27
<b>20 使用 Bar 数据</b>	<b>27</b>
<b>21 参数</b>	<b>28</b>
21.1 概念 . . . . .	28
21.2 声明 . . . . .	28
21.3 使用举例 . . . . .	29

目录	3
<b>22 变量</b>	<b>29</b>
22.1 变量声明与赋值 . . . . .	29
22.2 Commentary 函数的使用 . . . . .	30
22.3 变量使用举例 . . . . .	31
<b>23 期货基础</b>	<b>31</b>
<b>24 000 和 888</b>	<b>31</b>
<b>25 TBL 的三种数据</b>	<b>31</b>
<b>26 If Else 分支语句与循环语句</b>	<b>33</b>
<b>27 用户函数编写</b>	<b>33</b>
<b>28 全局变量</b>	<b>35</b>
<b>29 MA (移动平均线)</b>	<b>36</b>
<b>30 EMA (指数移动平均线)</b>	<b>37</b>
<b>31 DEMA (双重指数移动平均线)</b>	<b>38</b>
<b>32 MACD (指数平滑移动平均线)</b>	<b>40</b>
<b>33 KDJ (随机指标)</b>	<b>41</b>
<b>34 唐奇安通道</b>	<b>45</b>
<b>35 布林通道 (Bollinger Band) 交易系统</b>	<b>45</b>
<b>36 测试与评估</b>	<b>49</b>
<b>37 优化</b>	<b>52</b>
<b>38 模拟交易</b>	<b>53</b>

1 软件介绍

4

39 发单价格偏移

53

40 信号闪烁

56

41 使用序列函数的注意事项

56

42 多品种交易

58

43 控制开平仓、止盈止损

61

44 跟踪止损 (移动止损)

64

45 加减仓

67

46 收盘平仓

67

# 1 软件介绍

1. 官网 [www.tradeblazer.net](http://www.tradeblazer.net)
2. 软件下载安装

当前位置： 首页 > 软件下载 > 交易开拓者·极速版

软件下载

交易开拓者·极速版

交易开拓者·智能交易版

交易开拓者·旗舰版

交易开拓者·旗舰版V4

交易开拓者·CTP终端版

交易开拓者·期权Beta版

交易开拓者·极速版

最近版本升级日志详情

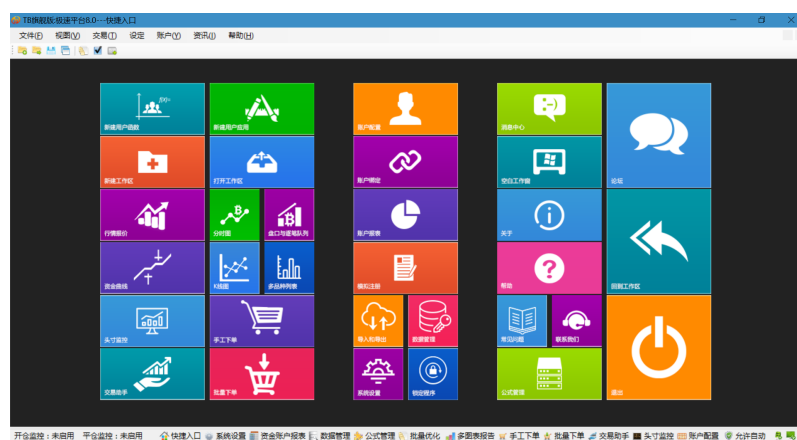
TB公式指南下载

帮助文档

培训视频

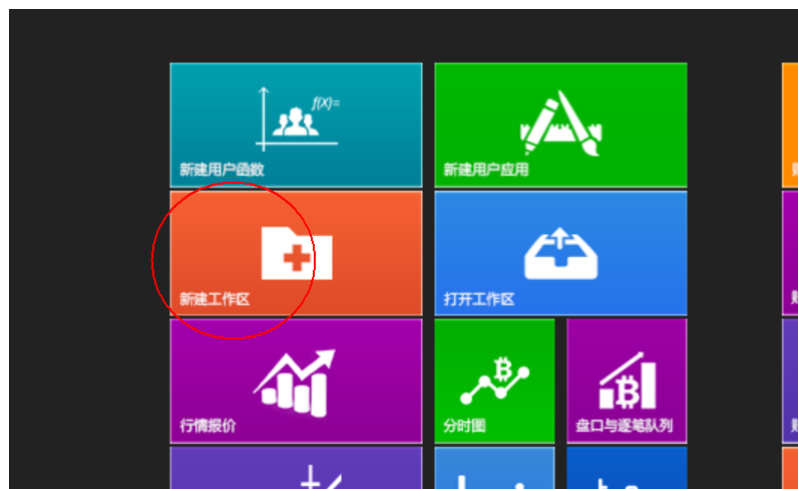
软件版本	v8.1.2.8(32位)	v8.1.2.8(64位)
发布日期:	2017-01-08	2017-01-08
软件大小:	105M	100M
旧版下载:	v8.1.2.6 (详情)	v8.1.2.6 (详情)
软件语言:	简体中文	
运行环境:	Windows 7 及以上	
软件介绍:	<div>交易开拓者极速版TBplus简介（8大亮点新功能）：</div> <div>1、多品种组合参数优化——同一个策略可以在多个品种不同周期或同周期上进行参数优化。</div> <div>2、实时资金曲线——图表上按*可以显示资金曲线，可以新建资金曲线图表，把N个策略或图表组合成新的资金曲线，可以任意组合。</div> <div>3、编程选参数——参数选择是一个复杂的过程，但可以有一定规则，把选择的规则写出编程语句，可以自动选择。</div> <div>4、自动选参数——参数优化自动选择你规则定义的参数值，运用到具体公式图表上交易。</div> <div>5、CTP飞马行情直连——满足先登、直连，托管机房可以直接行情报盘机，比转发快3MS。</div>	

3. 登陆及登陆后界面

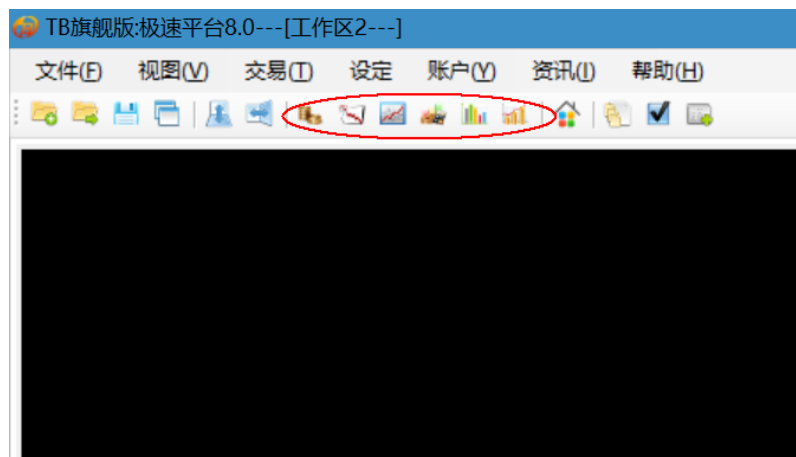


## 2 建立图表

### 1. 新建工作区



### 2. 插入分时图、行情报价、K 线图



### 3. 水平/垂直拆分



### 3 模拟账户注册



### 4 手工下单

#### 1. 手工下单

4 手工下单



2. 批量下单

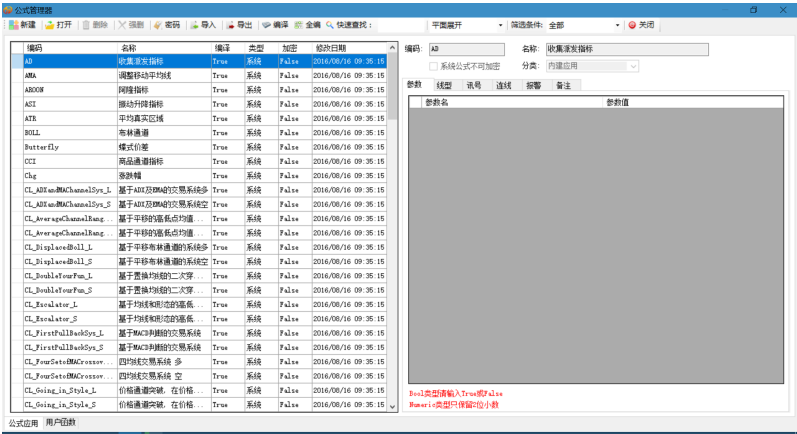




3. 一键撤单/一键全平

5 公式管理

1. 公式管理器

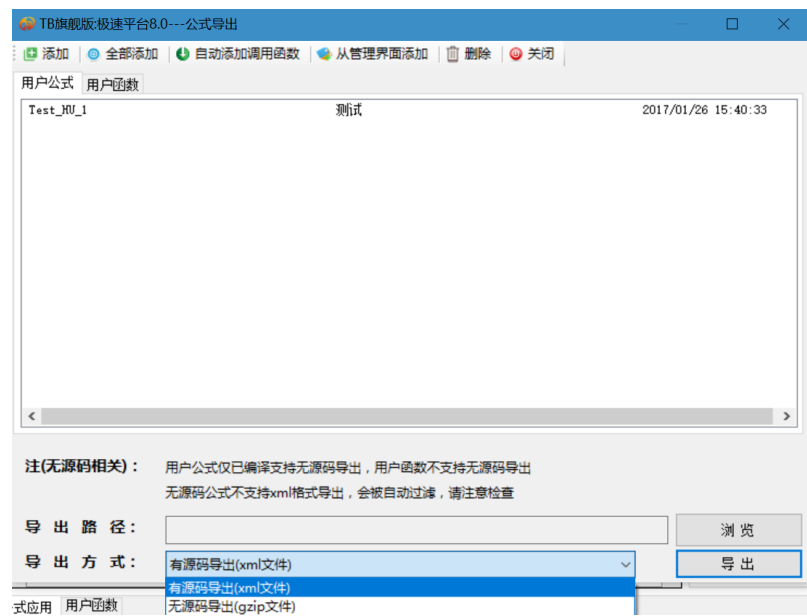


2. 用户函数与公式应用

- 用户函数在编写策略中，有些功能可能反复用到，为提高编程效率，封装为用户函数
- 公式应用一般指交易策略或技术指标

3. 公式应用导入导出

- 有源码
- 无源码



## 6 建立公式

### 6.1 举例：四周规则

1. 内容
- 价格突破四周高点的时候，做多；突破四周低点的时候，做空
  - 如果时间周期是日线的话，四周对应 20 天
2. 集合竞价过滤

```
If (CallAuctionFilter() == False) Return;
```

需要注意的是，TB 内置的 CallAuctionFilter 仅处理 1、3、5、15、30 分钟，1 小时，日线，周线，月线等常用周期，如果需要使用特殊周期，需要注意使用自己的集合竞价过滤函数，新版 TB 有处理任意周期的函数 ‘IsCallAuctionTime()’

3. 代码

## Params

```
Numeric Offset(2);  
// Lots 默认值设为 0 的时候，表示这次买卖的手数由公式应用设置中某个选项决定  
Numeric Lots(0);
```

## Vars

```
Numeric HiBand;  
Numeric LoBand;
```

## Begin

```
// 计算上轨和下轨通道  
// 注意此处不用 HiBand = Highest(High, 20);  
// 因为当前 K 线也包含在内，突破前 20 根 K 线高点永远无法满足  
// 变通的做法，改为 HiBand = Highest(High[1], 20);  
// 代表取前一根 K 线开始，往前 20 根 K 线的高点  
HiBand = Highest(High[1], 20);  
LoBand = Lowest(Low[1], 20);  
  
// 集合竞价过滤  
If(CallAuctionFilter() == False) Return;  
  
// 简单交易指令：开多 Buy，平多 Sell，开空 SellShort，平空 BuyToCover  
// 复杂交易指令：A_SendOrder  
  
// 第一种写法：最高价突破上轨 If(High > HiBand)  
// 为了避免连续加仓的情况，可以加入头寸控制，当没有开多的时候，开多  
If(High > HiBand And MarketPosition <> 1)  
{  
    // 注意价格合理性，有些价格不可能成交  
    // 最高价突破 HiBand 的时候，High 至少要比 HiBand 高一跳
```

```

// 这时不能说成交价就是 HiBand
// 如果价格突破上轨后，一路突破，没有机会以 HiBand 成交
// 至少要加上一跳，此时可以设置一个 Offset，考虑滑点因素
// 如果价格突破上轨时，有可能是从下面慢慢突破，也可能开盘跳空
// 开盘跳空的时候，就可以开盘价和考虑滑点后价格的最高价作为委托价
Buy(Lots, Max(Open, HiBand+Offset*MinMove*PriceScale));
}
// 另外一种写法： 最高价与上轨持平
//If(High >= HiBand)
//{
// Buy(1, Max(Open, HiBand));
//}

If(Low < LoBand And MarketPosition <> -1)
{
    SellShort(Lots, Min(Low, LoBand-Offset*MinMove*PriceScale));
}
End

```

## 7 加载公式和设置交易属性

在图表中，右击鼠标，选择对应的商品设置和应用公式设置注意：

- 默认值设置：应用公式设置 -> 图表交易设定

8 性能测试

交易策略性能测试报告 工作区1[r000--1天线]+HU_Four_Square				
保存 关闭				
交易汇总 交易分析 交易记录 平仓分析 阶段总结 资产变化 图表分析 系统设置				
性能概要				
统计指标	全部交易	多头	空头	
净利润	8600.49	11131.84	-2531.35	
总盈利	14277.14	13300.23	976.91	
总亏损	-5676.65	-2168.39	-3508.26	
总盈利/总亏损	2.52	6.13	0.28	
交易手数	7	4	3	
盈利比率	71.43%	75.00%	66.67%	
盈利手数	5	3	2	
亏损比率	28.57%	25.00%	33.33%	
亏损手数	2	1	1	
持平手数	0	0	0	
平均利润	1228.64	2782.96	-843.78	
平均盈利	2855.43	4433.41	488.45	
平均亏损	-2838.32	-2168.39	-3508.26	
平均盈利/平均亏损	1.01	2.04	0.14	
最大盈利	5812.67	5812.67	843.66	
最大亏损	-3508.26	-2168.39	-3508.26	
最大盈利/总盈利	0.41	0.44	0.86	
最大亏损/总亏损	0.62	1.00	1.00	
净利润/最大亏损	2.45	5.13	0.72	
最大连续盈利手数	5	3	2	
最大连续亏损手数	2	1	1	
平均持仓周期	38	52	20	
平均盈利周期	49	67	23	
平均亏损周期	12	10	14	
平均持平周期	0	0	0	
最大使用资金	3495.00	3495.00	3200.00	
佣金合计	49.51	28.16	21.35	
收益率	0.09%			
年度收益率	203.67%			

商品测试，测试周期，滑点，手续费，委托偏移 (启动偏移后，按照真正的主力合约价格加上一定跳数成交)

9 优化与批量优化

1. 参数优化



2. 在公式中设置参数优化范围，步长

3. 批量处理

(a) 批量设置



(b) 将当前图标公式应用到其他工作区



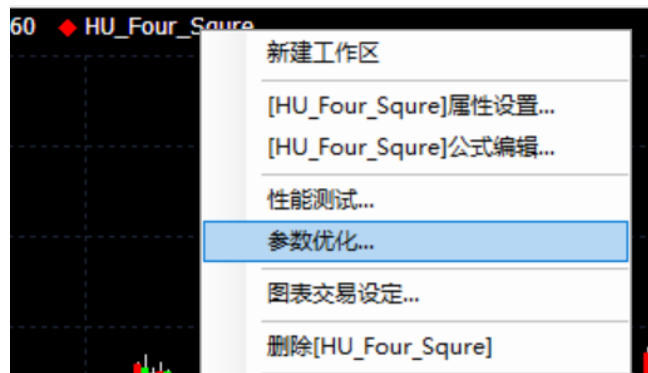
- 将当前公式另存为模板

- 批量设置，应用模板

(c) 批量优化

- 方式一：优化窗口 -> 加入队列 -> 批量优化

i. 参数优化



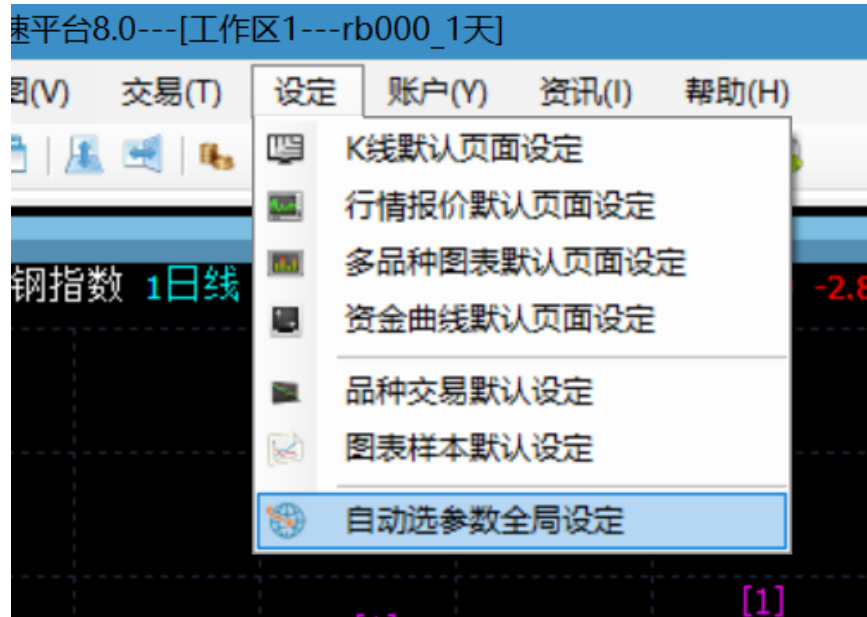
ii. 加入队列

iii. 批量优化

- 方式二：

可以将当前所有工作区的内容加入到队列中，然后选择想要批量优化的内容

## 10 自动选参数



缺点：仅对当前品种效果表现好，可能过拟合

## 11 多品种组合优化

希望找到一个参数，在所有品种上表现都好

1. 打开多品种图标
2. 添加公式
3. 参数优化

## 12 开启自动交易

## 13 头寸监控和交易助手

计算机问题或网络问题，可能导致交易失败，为了保证交易安全稳定，需要打开头寸监控和交易助手

头寸监控处理的是，发单不成功的情况

1. 头寸监控查询 -> 检查点 (账户仓-系统仓，理论上策略应该有的仓位 = 手工仓) -> 一键同步
2. 自动同步设置同步设置，可以将手工仓设为自动同步

如果发单了，但是没有成交，此时需要交易助手

1. 撤单设置发单后，如果成交价离我的价格太远，可以设置撤单

## 14 资金曲线

资金曲线的相关设置，可以在页面设置中看到；区域报告；  
设置不同品种，不同权重下的资金曲线图

1. 新建工作区
2. 图表类型设置资金曲线
3. 设置数据源
4. 添加品种
5. 不同品种的权重

## 15 辅助功能

1. 消息中心
2. 行情测速与切换 -> 图标右下角
3. 行情故障自动切换

4. 行情自动刷新
5. 风控系统设定 -> 交易



## 16 TB 公式

### 16.1 初步认识 TB 公式

1. 什么是 TB 公式
2. TB 公式的类型
  - 公式应用
  - 用户函数
3. 如何查看 TB 公式
4. TB 公式的一般结构

Params

...

...

```
Vars
```

```
...
```

```
...
```

```
Begin
```

```
...
```

```
...
```

```
End
```

## 16.2 通过实例了解 TB 公式运行机制

1. 创建、编辑和编译 TB 公式

2. 举例: Welcome To TB

3. TB 公式的运行机制

- 基于图表运行

- ‘FileAppend’ 的使用

```
Begin
```

```
FileAppend("C:\\Users\\Curio\\demo_welcome.txt", "Welcome to TB!");
```

```
End
```

- 从左到右

- 公式在开市和闭市时运行的不同

- 开市状态下，在最新的一根 Bar 上公式会随着数据不断更新  
反复调用公式

## 17 TB 语法基础

### 17.1 TBL 语言基本元素介绍

- 保留字: TB 中有独特含义或用途的字符, 主要指:

- 功能关键字、系统函数名、数据类型、数据源
- 操作符、标点符号
- 函数
  - 实现一定功能的代码，分系统函数和用户函数
- 表达式
  - 由常量、变量、操作符、函数和圆括号组成，通过运算能得到结果的式子，运算结果的类型由数据和操作符共同决定
- 语句：符合 TBL 语言语法的一行命令，比如：赋值语句、控制语句、循环语句等

## 17.2 命名规则

1. 详细描述，在 TB 官网下载 TB 公式电子指南，可以看到详细的有关命名规则描述
2. 命名规则的一些要点：
  - 不区分大小写
  - 不能和系统保留字、C++ 关键字重名
  - 名称只能使用字母、数字和下划线
  - 长度不超过 32 个英文字符
3. 公式正文规则：
  - 不区分大小写
  - 公式语句以';' 结尾
  - 公式中正文字符因含义不同，公式编辑器会以不同颜色加以区分

17.3 标点符号含义

符号	名称	说明
;	分号	表示当前语句的结束
,	逗号	当函数带有多个参数时，用于分隔参数
()	小括号	设置初始值，限定函数参数，改变运算优先级
" "	双引号	表示字符串常量
[ ]	中括号	指定回溯读取的偏移量，指定数组元素下标
{ }	大括号	指定语句块的范围，即语句块的开始与结束
.	点	表示数据源的数据调用
//	双斜杠	表示本命令行的之后部分为注释部分
* *	块注释	符号之间的部分是注释部分

17.4 操作运算符

类别	运算符	说明
算术运算符	+	加
	-	减
	*	乘
	/	除
	%	求模 (求余数)
关系运算符	>	大于
	>=	大于等于
	<	小于
	<=	小于等于
	==	等于
	!= 或 <>	不等于
逻辑运算符	And(&&)	与
	OR(两个竖线)	或
	Not	非

## 18 数据类型

### 18.1 数据类型

#### 18.1.1 基本类型

- 三种基本数据类型
  - 数值型 (Numeric)
    - \* e.g. 10000, 20160808, 0.1415
  - 字符串 (String)
    - \* e.g. "EntryPrice=" , "Buy At 2085"
  - 布尔型 (Bool)
    - \* 布尔型只有两个取值: True 和 False
    - \* 譬如:  $1000 > 900$  , Close > Open

#### 18.1.2 扩展类型

- 序列类型 (Series)
  - 和公式运行所在图表 K 线有密切关系，一一对应关系
  - 可以回溯读取
  - 变量的值系统会自动向下传递，确保序列变量在每根 K 线上都有一个有效值

数值序列类型	字符串序列类型	布尔型序列类型
NumericSeries	StringSeries	BoolSeries

- 引用类型

将来编写用户函数，可以用引用类型向用户函数传递参数，并且将用户函数中运算之后的结果，通过引用类型返回给调用用户函数的函数

数值引用类型	字符串引用类型	布尔引用类型
NumericRef	StringRef	BoolRef



- 数组类型 (Array)

一些相同类型的变量或者数据，按照一定顺序排列在一起的集合，集合中的元素，可以通过编号访问，方便用户对大批量同类型数据处理

数值型数组	字符串型数组	布尔型数组
NumericArray	StringArray	BoolArray

## 18.2 类型转换

- 从一种基本数据类型转型为另一种基本数据类型

- 类型转换必须在有意义的情况下进行

- e.g. "2000" 这个字符串转为数值：2000；"Buy Entry" 这个字符串转为数值，没有意义

- TBPlus 类型转换常用函数

- 字符串转数值型：‘Value’

```
Numeric Value(String str)
```

- 数值型转字符串：‘Text’

```
String Text(Numeric value)
```

- 布尔型转字符串：‘IIFString’

```
String IIFString(Bool Condition, String TrueValue, String FalseValue)
```

```
// Condition 为条件表达式
```

```
// TrueValue 是条件为真时的返回值
```

```
// FalseValue 是条件为假时的返回值
```

- 布尔型转数值型：‘IIF’

```
Numeric IIF(Bool Condition, Numeric TrueValue, Numeric FalseValue)
```

```
// Condition 条件表达式
```

```
// TrueValue 条件为真时返回的数值
```

```
// FalseValue 条件为假时返回的数值
```

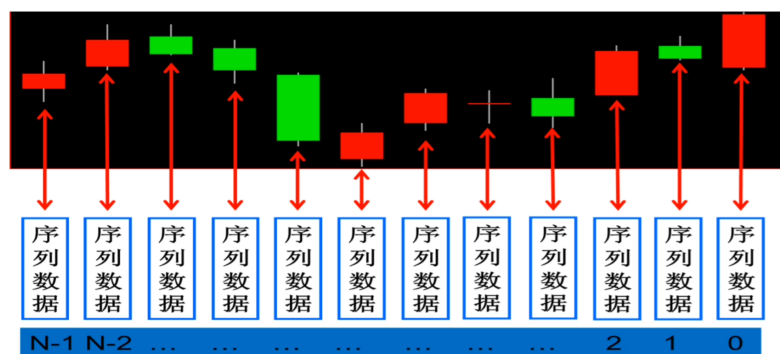
## 19 Bar 数据

### 19.1 Bar 数据基本概念

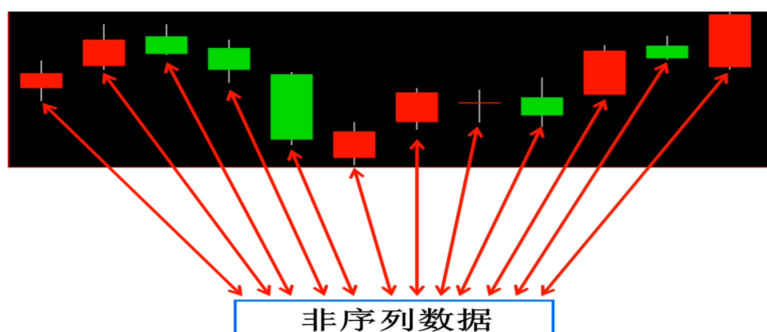
- 图表中的商品在所选择的时间周期下的所有 K 线样本数据，按照时间从先到后的顺序排列而成的序列数据
- Bar 数据是序列数据，序列数据支持回溯读取。具体使用方法：
  - 变量名 [偏移 Bar 数目] 或函数名 [偏移 Bar 数目]

### 19.2 序列数据与非序列数据区别

- 序列数据支持回溯读取



- 非序列数据无法实现回溯读取



### 19.3 每根 Bar 包含哪些数据

函数名	简写	返回值
Date	D	当前 Bar 的日期, 返回值是整型数
Time	T	当前 Bar 的时间, 返回值是小数
Open	O	当前 Bar 的开盘价
High	H	当前 Bar 的最高价 (Tick 图, 盘口的叫卖价, 即委卖价)
Low	L	当前 Bar 的最低价 (Tick 图, 盘口的叫买价, 即委买价)
Close	C	当前 Bar 的收盘价 (开市期间最后一根 Bar 最新价)
Vol	V	当前 Bar 的成交量
OpenInt	无	当前 Bar 的持仓量
CurrentBar	无	当前 Bar 的索引值 (编号), 从图标最左边开始, 从 0 开始计数
BarStatus	无	当前 Bar 的状态值, 0-图表中的第一根 Bar, 2-图表中最后一根 Bar, 也就是最

## 20 使用 Bar 数据

### 1. Version1

```

Begin
    FileAppend("C:\\Users\\Curio\\Downloads\\Demo_BarData.txt", "Welcome To TB");
    // 添加当前 Bar 索引
    FileAppend("C:\\Users\\Curio\\Downloads\\Demo_BarData.txt", CurrentBar);
End

```

### 2. Version2: Version1 编译出错, 原因在于 'FileAppend' 函数接收的函数都是 String 类型, 修改如下

```

Begin
    FileAppend("C:\\Users\\Curio\\Downloads\\Demo_BarData.txt", "Welcome To TB");
    // 添加当前 Bar 索引
    FileAppend("C:\\Users\\Curio\\Downloads\\Demo_BarData.txt", Text(CurrentBar));
End

```

3. Version3: 将 Version2 修改, 尝试 `String` 类型的拼接功能

```
Begin
  FileAppend("C:\\Users\\Curio\\Downloads\\Demo_BarData.txt", "Welcome To TB! 运
    + " 根 K 线上, BarStatus = " + Text(BarStatus));
End
```

4. Version4: 回溯读取 Bar 数据

```
Begin
  FileAppend("C:\\Users\\Curio\\Downloads\\Demo_BarData.txt", "Welcome To TB! 运
    + " 根 K 线上, BarStatus = "+Text(BarStatus)+"Close = "+Text(Close)+"Close[2] :
End
```

## 21 参数

### 21.1 概念

- 预先声明的地址, 用于存放输入的值, 声明后可以在公式中使用该参数名称引用其值
- 定义参数的好处:
  - 修改参数而不许更改公式代码就可以使公式以不同状态运行
  - 参数可以优化

### 21.2 声明

- 声明方法
  - 参数数据类型参数名 (默认值);
  - 参数数据类型参数名 (默认值, 最小值, 最大值, 步长);
  - 参数可声明的数据类型和公式类型有关

- \* 公式应用的参数只支持三种基本类型，而用户函数支持全部类型
- 参数值在公式内部不能被修改 (引用参数除外)，因此不能对参数进行赋值操作

### 21.3 使用举例

Params

Numeric Offset(3);

Begin

```
FileAppend("C:\\Users\\Curio\\Downloads\\Demo_Param.txt", "Close = " + Text(Close) +
+ Text(Close[Offset]));
```

End

## 22 变量

### 22.1 变量声明与赋值

1. 存储值的地址，用于存放公式在运算过程中会发生改变的值，变量声明后就可以在公式中使用该变量进行操作
2. 定义变量的好处：
  - 变量主要用于存储计算或比较的结果，之后的公式代码可以通过直接引用变量得到结果，无需重现计算过程，提高执行效率
  - 可高效地实现各种算法，提高代码可读性
  - e.g. 计算最近 20 根 K 线的收盘价平均价

```
sum = 0;
```

```
For i=0 To 20
```

```
{
```

```

        sum = sum+C[i]
    }

```

```

ma = sum/20;

```

### 3. 声明方法

- ‘变量类型变量名 (默认值);’
- 变量可声明的类型：除引用类型之外的其他数据类型

### 4. 变量赋值

- ‘变量名 = 表达式;’
- 表达式的数据类型必须与变量数据类型一致

## 22.2 Commentary 函数的使用

1. 在当前 Bar 的输出信息中添加一行注释信息

```

Commentary(String strTip)

```

2. e.g. 判断是否三连阳

```

Vars

```

```

    Bool bPattern(False);

```

```

Begin

```

```

    bPattern = Close > Open And Close[1] > Open[1] And Close[2] > Open[2];

```

```

    Commentary(IIFString(bPattern, "这是三连阳","这不是三连阳形态"));

```

```

End

```

### 22.3 变量使用举例

## 23 期货基础

1. 交易所
2. 交易品种
3. 与股票区别

## 24 000 和 888

1. 000 – 商品指数
  - 权重 = 持仓量/总合约持仓量
  - 000 对应的价格 =  $\text{Sum}(\text{各合约价格} \times \text{权重})$
  - 一般用于交易分析
  - 并不完全代表主力合约走势，可能出现问题
2. 888 – 主力连续
  - 一直映射当前主力合约，如果主力合约发生发生换月，888 会按照一定规则，自动切换为主力合约
  - 换月时可能发生跳空，程序化的时候需要考虑如何处理这种情况

## 25 TBL 的三种数据

1. Numeric – 作图

`Numeric PlotNumeric(String Name, Numeric Number, Numeric Locator=0, Integer Color=-`

- `Name` 输出值的名称，不区分大小写；
- `Number` 输出的数值；

- **Locator** 输出值的定位点，默认时输出单点，否则输出连接两个值线段，用法请看例 3；
- **Color** 输出值的显示颜色，默认表示使用属性设置框中的颜色；
- **BarsBack** 从当前 Bar 向前回溯的 Bar 数，默认值为当前 Bar。

```
PlotNumeric("收盘价与 3370 连线", Close, 3370);
```

设置了公式连线为柱状线

## 2. String – 作图

```
String PlotString(String Name,String str,Numeric Locator=0,Integer Color=-1,Integer BarsBack=1);
```

- **Name** 输出值的名称，不区分大小写；
- **str** 输出的字符串；
- **Locator** 输出值的定位点；
- **Color** 输出值的显示颜色，默认表示使用属性设置框中的颜色；
- **BarsBack** 从当前 Bar 向前回溯的 Bar 数，默认值为当前 Bar。

## 3. Bool – 作图

```
Bool PlotBool(String Name,Bool bPlot,Numeric Locator=0,Integer Color=-1,Integer BarsBack=1);
```

- **Name** 输出值的名称，不区分大小写；
- **bPlot** 输出的布尔值；
- **Locator** 输出值的定位点；
- **Color** 输出值的显示颜色，默认表示使用属性设置框中的颜色
- **BarsBack** 从当前 Bar 向前回溯的 Bar 数，默认值为当前 Bar。

```
PlotBool ("con",con,high); //在 bar 的最高价位置输出条件 con 的布尔值。
```

```
If (PlotBool ("con1",con1)) Alert("con1 is true"); //输出 con1 的值，并且当 con1 为真时
```



## 26 If Else 分支语句与循环语句

## 27 用户函数编写

### 1. 举例：计算 10 周期收盘价的平均值

- 新建一个新的函数，函数名可以设为 `func_test`

Params

```
NumericSeries Price(0);  
Numeric length(10);
```

Vars

```
Numeric Sum(0);  
Numeric i(0);
```

Begin

```
For i = 0 To length-1  
{  
    sum = sum + Price[i];  
}
```

```
Return sum/length;
```

End

编译成功后，在公式应用中直接调用函数即可

```
Commentary("Avg Close:"+Text(func_test(Close, 10)));
```

### 2. 关于'NumericRef' 的使用

- 如果在函数中希望能返回多个参数，可以使用'NumricRef'

- 譬如, 上例中, 不仅希望返回平均值, 还希望返回收盘价之和, 可以对函数进行修改

Params

```
NumericSeries Price(0);
Numeric length(10);
NumericRef Sumat; // 注意, 引用型参数, 不能赋初值
```

Vars

```
Numeric Sum(0);
Numeric i(0);
```

Begin

```
For i = 0 To length-1
{
    sum = sum + Price[i];
}
Summat = sum;
```

```
Return sum/length;
```

End

- 相应的, 公式调用函数方法

Vars

```
Numeric Sum(0);
```

Begin

```
Commentary("Avg Close: "+Text(func_test(Close, 10, Sum));
Commentary("Sum: " + Text(Sum));
```

End

## 28 全局变量

- 简单变量

仅在当根 Bar 上有效

- 序列变量

值可以传递，如果当根 Bar 没有做运算，值会从上根 Bar 上传下来，每根 Bar 上都有值，这些 Bar 的值集合作为一个变量称为序列变量

- 数组变量

在每根 Bar 上都有很多值，用下标取值

- 全局变量

独立于 K 线之外的变量，仅有一个值

- 注意数组变量于序列变量的区别

- e.g.

Params

```
Numeric a;  
NumericSeries b;  
NumericArray arr;
```

Begin

```
If(BarStatus == 0)  
{  
    a = 1;  
    b = 1;  
    arr[1] = 2;  
    arr[2] = 3;  
    SetGlobalVar(0, a); // 全局变量一共可以设置 255 个，这里设置第一个全局变量值  
}
```

```

Commentary("a = " + Text(a));
Commentary("b = " + Text(b));
Commentary("arr[1] = " + Text(arr[1]));
Commentary("arr[2] = " + Text(arr[2]));
Commentary("globalVar 0 = " + Text(GetGlobalVar(0))); // 获取第一个全局变量值
End

```

## 29 MA (移动平均线)

### 1. 说明

将某一段的收盘价之和除以该周期，譬如，日线 MA5 指的是 5 日内的收盘价除以 5。

### 2. e.g. 画出 5 日/10 日均线，颜色分别为黄色和蓝色

- 新建计算均值的用户函数 CalMA，返回值为'Numeric' 类型

#### Params

```

NumericSeries Price(0); // 接受收盘价
Numeric Length(0); // 周期

```

#### Vars

```

Numeric Sum(0);
Numeric i(0);

```

#### Begin

```

For i = 0 To Length - 1
{
    Sum = Sum + Price[i];
}

```

```

Return Sum/Length;

```

End

- 新建用户公式

Param

Numeric Length1(5);

Numeric Length2(10);

Begin

PlotNumeric("短周期均线", CalMA(Close, Length1), 0, Yellow);

PlotNumeric("短周期均线", CalMA(Close, Length2), 0, Blue);

End

## 30 EMA (指数移动平均线)

### 1. 说明

EMA (Exponential Moving Average), 指数平均数指标, 也叫 EXPMA 指标, 它也是一种趋向性指标, 指标平均数指标是以指数式递减加权的移动平均。

### 2. 公式

```
\begin{equation}
EMA_{\text{today}} = \alpha \times Price_{\text{today}} + (1-\alpha) \times EMA_{\text{ye}}
\end{equation}
```

其中,  $\alpha$  为平滑指数, 一般取做  $2/(N+1)$ .

3. 解读 EMA 指标中, 每天价格的权重系数以指数等比形式缩小, 时间越靠近当今时刻, 其权重越大, 说明 EMA 函数对近期价格加强了权重比, 更能及时反映近期价格波动情况。所以 EMA 比 MA 更具有参考价值, 而 EMA 也不容易出现死叉和金叉, 所以一旦出现需要立即做出反映。对周线处理, EMA 就更加稳定了。

## 4. 源码

```

Params
    Numeric Length(10);

Vars
    NumericSeries Ema(0);
    Numeric Alpha(0);

Begin

    If(CurrentBar == 0) Ema = Close;
    Else{
        Alpha = 2/(Length+1);
        Ema = Alpha*Close + (1-Alpha)*Ema[1];
    }
End

```

## 31 DEMA (双重指数移动平均线)

## 1. 说明

双重指数移动平均线是为了减少普通移动平均线中出现滞后时间而开发的更快速更平滑的移动平均线。

## 2. 公式

```

\begin{equation}
(N \text{ 日的 EMA 的 DEMA}) = 2 \times N \text{ 日的 EMA} - N \text{ 日的 EMA 的 EMA}
\end{equation}

```

变形有

```

\begin{equation}

```

$$((N \text{ 日的 EMA 的 DEMA}) + N \text{ 日的 EMA 的 EMA})/2 = N \text{ 日的 EMA}$$

`\end{equation}`

### 3. 解读

DEMA 不是简单对 EMA 再进行指数平均，也不是移动平均线的移动平均线，是单一 EMA 和双重 EMA 的结合，比两种都更少滞后。

### 4. 源码

```
Params
    Numeric Length(10);

Vars
    NumericSeries Ema(0);
    NumericSeries Eema(0);
    NumericSeries my_Dema(0);
    Numeric Alpha(0);

Begin
    If(CurrentBar == 0)
    {
        Ema = Close;
        Eema = Close;
    }Else{
        Alpha = 2/(1+Length);
        Ema = Alpha * Close + (1-Alpha)*Ema[1];
        Eema = Alpha * Ema + (1-Alpha)*Eema[1];
        my_Dema = 2* Ema - Eema;
    }
    PlotNumeric("DEMA", my_DEMA, 0, Blue);
End
```

## 32 MACD (指数平滑移动平均线)

### 1. 说明

从双指数移动平均线发展而来, 由快的指数移动平均线 (EMA12) 减去慢的指数移动平均线 (EMA26) 得到快线 DIF, 再用 2x(快线 DIF-DIF 的 9 日加权移动平均线 DEA) 得到 MACD 柱, 即, 由快、慢均线的离散、聚合表征当前的多空状态和股价可能的发展变化趋势, 但阅读起来更加方便。当 MACD 由负数转向正数, 是买的信号, 当 MACD 由正数转为负数, 是卖的信号。当 MACD 以大角度变化, 表示快的移动平均线和慢的移动平均线的差距非常迅速地拉开, 代表了一个市场大趋势的转变。

### 2. 源码

#### Params

```
Numeric FastLength(12);  
Numeric SlowLength(26);  
Numeric MACDLength(9);
```

#### Vars

```
Numeric Dif;  
Numeric MACD;  
Numeric EmaDif;
```

#### Begin

```
Dif = XAverage(Close, FastLength) - XAverage(Close, SlowLength);  
EmaDif = XAverage(Dif, MACDLength);  
MACD = Dif - EmaDif;  
PlotNumeric("Dif", Dif);  
PlotNumeric("EmaDif", EmaDif);  
If(MACD > 0){  
    PlotNumeric("MACD", MACD, 0, Red);
```



```

}Else{
    PlotNumeric("MACD", MACD, 0, Green);
}
PlotNumeric("0 轴", 0, 0);
End

```

### 33 KDJ (随机指标)

#### 1. 说明

KDJ 又称随机指标，最早以 KD 指标形式出现，而 KD 指标又是在威廉指标的基础上发展而来，目前是金融市场上常用的技术分析工具之一。

K, D 指标实在 WMS 的基础上发展起来的，所以 K, D 指标是 WMS 的一些特性。在反映股市价格变化时，J 最快，K 指标其次，D 指标最慢。K 指标反应敏捷，但容易出错；D 指标反映稍慢，但稳重可靠。

#### 2. 计算方法

- (a) 首先计算周期内的 RSV 值，即未成熟随机指标值，然后再计算 K 值，D 值，J 值。

```

\begin{equation}
n \text{ 日 } RSV = (C_n - L_n) / (H_n - L_n) \times 100
\end{equation}

```

即，N 日内的 RSV 值等于这段周期内，最新的一根 Bar 的收盘价-周期内最低价，然后除以周期内最高价-周期内最低价的值。

可以看出，RSV 越小，越接近最低价，RSV 值越大，则越接近最高价。

- (b) 计算 K、D 值

```

\begin{eqnarray}
\text{当日 K 值} &= & \text{Avg}(RSV, 3) \\
\end{eqnarray}

```

```

    当日 D 值 &= & Avg(K, 3)
\end{eqnarray}

```

如果前一日 K、D 值，则可分别用 50 来代替。

(c) 计算 J 值

```

\begin{equation}
    J \text{ 值} = 3 \times \text{当日 K 值} - 2 \times \text{当日 D 值}
\end{equation}

```

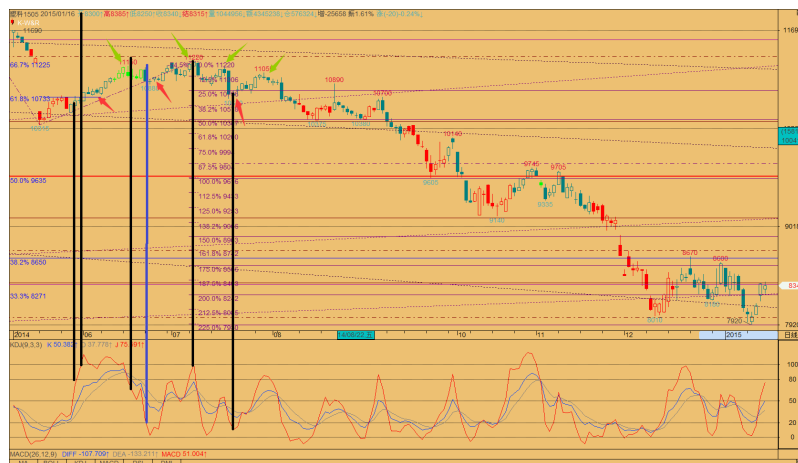
### 3. 解读

由于  $K_0 = R_2 + R_1 + R_0 \frac{1}{3}, D_0 = K_2 + K_1 + K_0 \frac{1}{3}, J_0 = 3K_0 - 2D_0$ ，可以推出  $J_0 = (7R_0 + 5R_1 + 3R_2 - 4R_3 - 2R_4)/9$ ，可见，对于 J 的计算，近期的 RSV 值对其影响越大。另外，由于 RSV 的取值在 0 到 +100 之间，由此可以得到 J 的取值范围在 -66.67 到 +166.67 之间，在该取值范围内，如果取黄金分割点，值分别在 20 和 80 左右，再加上考虑到 20/80 法则 (RSV 取极值 100 和 0 的时候，J 取值为 0 到 100)，又是在三分位上，因此，可以取 20 和 80 作为指标分割线。

为什么 J 要用  $3 \times K - 2 \times D$ ? K 和 D 为什么要用 3 天平均? RSV 为什么要选取 9 日内? RSV 权重，依次为 7、5、3、4、2，被减数是奇数权重，减数是偶数权重，为什么? 简单一句话——自然数列，任何指标参数，包括 KDJ 在内，不能随便改，或者说，如果要改，得保证 RSV 都取极值时，J 值等于 100 或 0。这些常用指标参数都是经过数十年市场检验，不断优化，最后确定下来的，已经是最优选择了。

作者：何子垠链接：<https://www.zhihu.com/question/27652388/answer/37491605> 来源：知乎著作权归作者所有。  
商业转载请联系作者获得授权，非商业转载请注明出处。

### 4. 举例



作者：何子垠链接：<https://www.zhihu.com/question/27652388/answer/37491605> 来源：知乎著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

看图中的例子。L1505 合约开始后，伴随 1501 继续缓慢上涨，一根大阳突破 10700 平台上沿，J 值超过 100，行情是继续涨呢，还是要回调呢？在短期（KDJ 截取周期）内，价格总体是不断走高的，J 值也随价格的升高，而不断走高，没有出现背离。当 J 值超过 80 以上，进入超买区。啥叫超买？买的人多。为什么买的人多？因为看涨。此时最有可能的行情是什么？多头趋势延续。K 线突破平台，多头信号，J 值进入超买区，就是绝佳多单进场点。

当 K 线创出阶段新高后（比如图中涨到 11150，接近重要关口 11225 附近），出现滞涨，K 线有小幅回落，此时观察 KDJ。J 值迅速下落直到接近 0 线，而 K 线价格仍在前期平台 10700 以上并稳住，那么，KDJ 与 K 线的背离就出来了，多头强势，J 值回落表明短线获利盘跑了，这是洗筹的过程。此时，利用 J 值进入超卖区进场做多，以跌破前平台上沿为止损，也是绝佳的机会。

随后，K 线再创高点 11220，非常接近 11225。重要关口位，一定要谨慎对待，可以突破，也可以被压制，观察此时的

KDJ。J 值在高点又达到了 100。在 11150-11220 这段，高点没有明显差别，且数个低点也没有明显差别，震荡幅度在不断加大，此时 J 值进入超买区，就真的是超买了，因为短期（KDJ 截取周期）内，趋势是震荡。且对比 11220 和 11500 时的 J 值，价格创新高，但 J 值没有创新高，背离出来了。对于中长线多头，当高点接近重要关口时，要做相应准备。能不能突破关口不知道，但一定要防关口的压制，所以在高位，要么减仓，要么锁上，没突破关口就不能加多单。

再向后看，K 线受关口压制下跌，之后反弹，高点没有创新高，与 11150 持平，且比 11150-11220 间的高点 11200 低，J 值也比 11200 对应的值低，短期内仍是震荡且有由多转空的节奏，那就空呗。后市下跌后反弹至 11055，高点依次降低，对应的 J 值也依次降低，且在超买区，继续空。当然这中间的反弹也可以用 J 值与 K 线的背离来抓，比如从下跌至 10670 时，K 线创短期新低，但 J 值大于前低，背离出现，可抓反弹，因为短期是震荡市。

总结一下：所谓趋势延续，是当下你要进场的位置前，总体走势是上涨或下跌的，当 J 值进入超买超卖区时，表明当前市场集中了大量的顺势动能，并且即将释放，选一个 K 线上比较安全的位置——突破，或回撤后顺原趋势运行时进场，就是绝佳的搭顺风车的机会。如果进场前，趋势是震荡，那么 J 值在超买超卖区，就是绝佳的反手机会，而不是顺势，因为短期根本就没有趋势可言。KDJ 在短期趋势运行的过程中，J 值超买超卖区，表示趋势继续延续；在 20-80 间利用位置关系，背离关系，顺势加仓或抓回撤（在 20-80 间的数值要一视同仁，50 这个数没什么太大意义）。在短期震荡市中，J 值超买超卖，表明可抓反手；在 20-80 间，同样利用位置关系，背离关系做加仓，抓回撤动作。所以一定要理解指标的含义，看当前短期内的 K 线走势，而不是见到超买超卖就反手，不能教条。

## 34 唐奇安通道

### 1. 说明

由 Richard Donchian 发明，由三条不同颜色的线组成，该指标用周期内的最高价和最低价来显示市场价格的波动性，当其通道窄时表示市场波动较小，反之，通道宽表示市场波动比较大。

具体而言，当价格冲破上轨时是可能的买信号，突破下轨则是对应的卖信号。当时由于曲线是由最高价和最低价计算出来的，价格很少突破上下轨道线，多数实在轨道之间运动，因此建议配合其他指标一起使用。

### 2. 计算公式

```
\begin{eqnarray}
  上线 &=& \text{Max}(\text{最高价}, n) \\
  下线 &=& \text{Min}(\text{最低价}, n) \\
  中线 &=& (\text{最高价} + \text{最低价}) / 2 \\
\end{eqnarray}
```

## 35 布林通道 (Bollinger Band) 交易系统

### 1. 说明

布林线是根据统计学中的标准差原理设计出来的一种非常实用的技术指标。它由三条轨道线组成，其中上下两条线分别可以看成是价格的压力线和支撑线，在两条线之间是一条价格平均线，一般情况价格线是在由上下轨道组成的带状区间内游走，而且随价格变动而自动调整轨道的位置。当波带变窄时，激烈的价格波动有可能随机产生，若高低点穿越边线时，立刻又回到波带内，则会有回档产生。

布林线在中间的通常为 20 天平均线，而在上下的两条线则分别为 Up 线和 Down 线，算法首先是通过计算出过去 20 日收盘价的均价和标

准差, Up 线是 20 均线加上 2 倍标准差, Down 线是 20 日均线减去 2 倍标准差。

## 2. 计算公式

```
\begin{eqnarray}
  \text{中间线} \&=& 20 \text{ 日均线} \\
  \text{Up 线} \&=& 20 \text{ 日均线} + 2\sigma \\
  \text{Down 线} \&=& 20 \text{ 日均线} - 2\sigma \\
\end{eqnarray}
```

## 3. 进出场规则

- 进场
  - 上一根 Bar 的收盘价突破上一根 Bar 的上轨, 做多, 默认手数, 当前 Bar 开盘价发单
  - 上一根 Bar 的收盘价突破上一根 Bar 的下轨, 做空, 默认手数, 当前 Bar 开盘价发单
- 出场
  - 无论多单空单, 价格回落至中轨, 平仓, 当前 Bar 开盘价平仓。

## 4. 源码

```
Params
  Numeric Length(20);
  Numeric Offset(2);

Vars
  NumericSeries MidLine(0);
  NumericSeries UpLine(0);
  NumericSeries DownLine(0);
```

```

Begin
  MidLine = Average(Close, Length);
  UpLine = MidLine+Offset*StandardDev(MidLine,Length,2);
  DownLine = MidLine-Offset*StandardDev(MidLine,Length,2);
  PlotNumeric("UpLine", UpLine);
  PlotNumeric("DownLine", DownLine);
  PlotNumeric("MidLine", MidLine);

  // 入场条件
  If(Close[1] > UpLine[1]) Buy(0, Open);
  If(Close[1] < DownLine[1]) SellShort(0, Open);

  // 出场条件
  If(MarketPosition == 1 And Low[1] < MidLine[1]) Sell(0, Open);
  If(MarketPosition == -1 And High[1] > MidLine[1]) BuyToCover(0, Open);
End

```



但是，如上图所示，如果允许连续建仓，可能出现问题。因此，可以改进如下，加入对当前仓位的判断，只有当前没有持仓才会开仓。

```

Begin

```





- (b) 加入止盈
- (c) 信号闪烁问题如果我们不用前一根 Bar 的信息来进行开平仓判断，而是使用盘中 Bar 信息进行开平仓判断，譬如盘中价格突破上轨就做多，但是有可能收盘的时候这个 Bar 的收盘价在轨道区间内，这个就是信号闪烁。

## 36 测试与评估

### 1. 回测

- 回测速度快
- 与实际运行很大区别

### 2. 模拟盘

- 冲击成本无法模拟 – 什么是冲击成本？譬如卖盘挂了 10 单，买盘挂了 20 单，那么吃掉 10 单后，卖盘价格肯定会再涨。
- 如果是日线级别的策略，模拟盘运行需要时间长，很可能需要几个月时间
- 很可能模拟盘运行的时候，行情处于单一行情

### 3. 测试设置

#### (a) 商品设置

- 样本 -> 范围
  - 注意：夜盘按道理是第二天的行情，但是如果设置样本范围是 1 天的时候，是没有夜盘数据的
- 属性
  - 最小变动：一跳
  - 交易单位：每一手交易多少单位，K 线图上价格是 1 个单位的价格

- 每点价值：股指期货比较特殊，一般是 1 点 300 元；商品期货一般就是 1 点 1 元

- 交易

- 保证金率：可以稍微调大些，保证有一定回撤余地
- 委托偏移：与模拟盘有关
- 手续费：Cu -> 不收平今仓手续费；大部分是双向收手续费；Au -> 费率方式一般是固定 1 手 10 这样的方式收费
- 滑点：一般用‘跳/手’，一般最大不会超过 10 跳，5 跳就是比较大的波动

(b) 公式应用设置

- 图表交易设定

- 初始资金：如果设置的值比较小，可能会出现爆仓，因此一定要设置够
- 允许连续建仓：如果策略需要连续建仓，这里要打开
- 计算前 N 个较大资产回撤的平均值：因此测试报告中已经有了最大资产回撤，因此这边可以设置 5 左右，看看整体回撤情况

(c) 测试报告

可以在工具 -> 测试报告里看到策略的性能测试报告，可以保存，保存格式为网页格式

(d) 滑点

价格一般不大可能以当前价位成交，可能出现价格损耗，价格损耗的影响可以用滑点来表示，滑点损耗在测试报告中展示在手续费中。

交易记录													
#	公式应用	类型	商品	建仓时间	建仓价格	平仓时间	平仓价格	数量	佣金	净利	累计净利	收益率%	累计收益率%
1	DualMA	空头	au888	2016/11/11 21:50:00	280.90	2016/11/15 11:50:00	275.15	1	20.00	5730.00	5730.00	20.40	20.40

上图是不设滑点，应用双均线策略，测试报告中成交记录的某一单，买一手，卖一手，双向收费，手续费是每手 10 元，因此对应的佣金是 20 元。

#	公式应用	类型	商品	建仓时间	建仓价格	平仓时间	平仓价格	数量	佣金	净利	累计净利	收益率	累计收益率
1	DonM	空头	au999	2016/11/11 21:00:00	280.90	2016/11/15 11:00:00	275.15	1	220.00	5530.00	5530.00	19.69	19.69

上图则是加上了滑点后佣金变化，滑点设为 2 跳，黄金 1 手 1000 克，1 跳对应 0.05 元，因此，每手对应的滑点是 100 元，买一手，卖一手，正好是 200 元，加上 20 元手续费，佣金变为 220 元。

#### (e) 测试报告解读

- 净利润：总盈利 - 总亏损（包含了手续费和滑点）
- 交易手数：表示做了多少次交易，不是指买了或者卖了多少手合约
- 盈利比率：盈利手数/交易手数
- 平均利润：净利润/交易总手数
- 平均盈利：总盈利/盈利手数
- 平均亏损：总亏损/亏损手数
- 平均盈利/平均亏损：体现了盈利的效率值 -> 凯利公式
- 最大盈利：所有盈利的交易里，盈利最多的一笔交易盈利额
- 最大亏损：所有亏损交易里，亏损最多的一笔交易亏损额
- 最大盈利/总盈利：可以看到盈利的分布
- 最大使用资金：Max（交易手数 x 保证金率 x 市值）
- 收益率：净利润/初始资金
- 有效收益率：净利润/最大使用资金
- 收益曲线斜率：收益曲线终值与收益曲线初值的连线斜率
- 收益曲线 R 平方值：
- 夏普比率：
- 收益风险比：年化收益/最大资产回撤 -> 回测报告中没有 -> 策略评估最重要的指标之一
- 调整的收益风险比：年化收益/平均资产回撤 -> 回测报告中没有 -> 策略评估重要指标之一
- 平均浮动盈亏：所有交易最大浮动盈利/最大浮动亏损
- 建仓效率：(最佳平仓价-开仓价)/(最佳平仓价-最佳开仓价)

- 平仓效率:  $(\text{平仓价} - \text{最佳开仓价}) / (\text{最佳平仓价} - \text{最佳开仓价})$
- 总效率:  $(\text{平仓价} - \text{开仓价}) / (\text{最佳平仓价} - \text{最佳开仓价})$ 
  - 最佳平仓价: 建仓后浮动盈利最大的价格
  - 最佳开仓价: 建仓后浮动亏损最大的价格 -> 就是最差平仓价

## 37 优化

1. 优化目标
2. 优化参数
3. 自动选参设置 -> 模板设置

`Max(净利润, 0.3)`

表示取净利润最大的前 30% 的参数。

`Ptg(净利润, 0.3, 0.5)`

表示净利润在 30% 到 50% 之间的一个范围。

4. 加入队列 -> 批量优化
5. 多品种优化
6. 默认设置
  - 设定 -> 品种交易默认设定
  - 文件 -> 设置所有图表参数

## 38 模拟交易

1. 可以同时运行多个公式
2. 多品种同时交易 -> 每个品种单独设置
3. 委托偏移
  - 如果用指数 (000 或 888) 进行交易  
公式应用设置 -> 图表交易设定 -> 交易账户 (D0-> 主力合约)
  - 偏移 x 跳 -> 如果以心理价位进行交易可能不能成交，可以设置跳数，这样，每次发单都会偏移 x 跳进行发单，不一定会以委托的价格成交，按实际撮合交易价格为准，譬如，发单价格比委卖价高，那么交易所会以委卖价进行撮合成交。

## 39 发单价格偏移

1. 说明
  - 发单不准确是程序化交易者在刚开始学习编写模型时常犯的一个错误。
  - 发单价格不准确发生原因往往是因为发单价格和触发交易条件的价格不一样。
  - 偷价属于发单价格不准确的一部分，但是偷价和实盘交易没有关系，仅仅体现在测试报告中，在 If 语句中，如果逻辑判断条件和发单价格逻辑没有关联或者逻辑错误，可能造成发单价格不准确。
  - 在测试报告中看不出来，只能实盘模拟，比较交易记录才能看出问题发生在哪里。
  - 短线中高频的交易策略如果发生偷价，往往非常致命。
2. 示例

```

MidLine = Average(Close, Length);
Band = StandardDev(Close, Length, 2);
UpBand = MidLine + Offset*Band;
DownBand = Midline - Offset*Band;

PlotNumeric("MidLine", MidLine);
PlotNumeric("UpBand", UpBand);
PlotNumeric("DownBand", DownBand);

```

#### (a) 震荡反手开仓

如果是在震荡势中，我们认为，当价格突破下轨后，很大概率后市价格会回到中轨，反之，当价格突破上轨后，价格会回落到中轨。

下面的代码会出现偷价问题：

```

// 反手写法，突破上轨，开空单，突破下轨，开多单
If(High > UpBand[1] And MarketPosition <> -1){
    SellShort(0, Close); // 最新价发单，close 既是收盘价，在当前 Bar 柱又是最新价
}
If(Low < DownBand[1] And MarketPosition <> 1){
    Buy(0, Close);
}

```



在图中绿色箭头位置是模拟盘中实际应该发单价格，但是回测报告中，用的是收盘价，即红色箭头位置，这就造成了偷价格。

## (b) 趋势突破平仓条件

```
// 持有多仓，且最低价突破中轨，以最新价平仓
If(MarketPosition == 1 And Low < MidLine){
    Sell(0, Close);
}
```

```
// 持有空仓，且最高价突破中轨，以最新价平仓
If(MarketPosition == -1 And High > MidLine)
{
    BuyToCover(0, Close);
}
```

上述代码同样会造成偷价问题。

如果将上述代码改为与前一根 Bar 的价格比较

```
// 持有多仓，且最低价突破中轨，以最新价平仓
If(MarketPosition == 1 And Low < MidLine[1]){
    Sell(0, MidLine[1]);
}
```

```
// 持有空仓，且最高价突破中轨，以最新价平仓
If(MarketPosition == -1 And High > MidLine[1])
{
    BuyToCover(0, MidLine[1]);
}
```

同样会有问题，假如前一根 Bar 的 MidLine 价格是 10，我当前的 Low 是 9，那么满足 'Low < MidLine[1]' 了，此时，我以前一根 Bar 的 MidLine 价格试图卖出，回测中是可以卖的，但是，实盘中，Low 都已经小于 10 了，还以 10 的加个挂卖单，很明显会卖不出去。

### 3. 其他可能的偷价情况

#### (a) 跳空

如果以当前 Bar 柱的价格作为条件判断分支语句，出现跳空的时候，很可能会出现偷价格问题。

## 40 信号闪烁

### 1. 说明

盘中模拟或者实盘交易的时候，有时候公式会遇到开平仓信号在最新 Bar 上时有时无的情况。这种情况我们称为信号闪烁，造成这种问题原因：

当我们用 IF 分支语句作为交易动作的一个条件判断时，如果 IF 后面的 Bool 值和 High, Low, Close, Vol, Openint 等这些在盘中变化的值有关，那么就会很容易造成信号闪烁。

最直接的解决办法：条件判断都用历史数据，不用盘中的实时数据。

### 2. 查看信号闪烁方法

跑模拟盘，看委托记录，注意查看价格

## 41 使用序列函数的注意事项

### 1. 说明

- TBL 的计算机制：对于用 AND 连接的 BOOL 值，依次连接连接项的值，如果判断到 FALSE，如果判断到 FALSE，那么直接返回 FALSE，不再计算后面连接项的值。
- 什么是序列函数  
类似与序列变量，该函数是可以回溯的，需要在历史 BAR 上保存相应的值。



- e.g. BarCount 返回当前公式应用在指定图层上的 Bar 总数;  
CurrentBar 返回当前公式应用在指定图层上的当前 Bar 的索引值。

## 2. 一个容易出错的地方举例

### Vars

```
BoolSeries Condition1;  
BoolSeries Condition2;  
  
NumericSeries MacdDiff(0);  
NumericSeries MacdValue(0);  
NumericSeries AvgMacd(0);
```

```
BoolSeries Cond1;
```

### Begin

```
MacdValue = XAverage(Close, FastLength) - XAverage(Close, SlowLength);  
AvgMacd = XAverage(MacdValue, MacdLength);  
MacdDiff = MacdValue - AvgMacd;  
  
Condition1 = MacdDiff[1] > 0 And CrossOver(MacdValue[1], AvgMacd[1]);  
Condition2 = CrossOver(MacdValue[1], AvgMacd[1]) And MacdDiff[1] > 0;  
  
PlotBool("Condition1", Condition1, Low);  
PlotBool("Condition2", Condition2, High);
```

逻辑上而言, 'Condition1' 应该与 'Condition2' 的值一样, 但是会有差别, 原因在于 CrossOver 这个函数是返回值是 BoolSeries, 又因为 TB 默认两个 Bool 类型连接时, 第一项是 False, 第二项不会再计算, 直

接返回 False, 这会碰到一种情况, 某连续两根 Bar, 第一根 Bar 满足  $\text{MacdDiff}[1] > 0$ , 但是  $\text{CrossOver}(\text{MacdValue}[1], \text{AvgMacd}[1])$  是 False, 第二根 Bar 时  $\text{MacdDiff}[1] > 0$  是 False, 则直接返回 False, 此时, 语句  $\text{CrossOver}(\text{MacdValue}[1], \text{AvgMacd}[1])$  不会运行, 则尽管此时实际情况是  $\text{CrossOver}(\text{MacdValue}[1], \text{AvgMacd}[1])$  为 True, 但因为这个语句没有运行, 则默认其结果是 False, 这样, 到了第三根 Bar, 如果两个条件都是 True 的话, 那么 Condition2 返回是 True, 而 Condition1 的结果却是 False.

### 3. 另一个示例

假如当前 K 线图由 300 根 K 线, 我想要在每根 K 线上打印出当前 Bar 柱的索引, 只需要用 'CurrentBar' 即可, 但是, 假如我采用下面语句

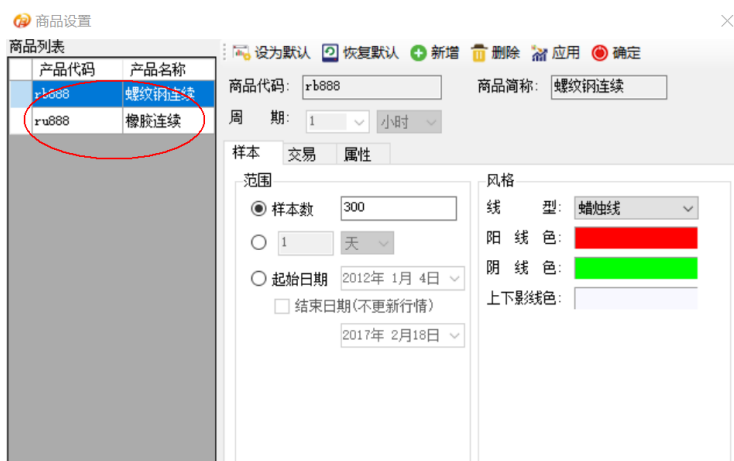
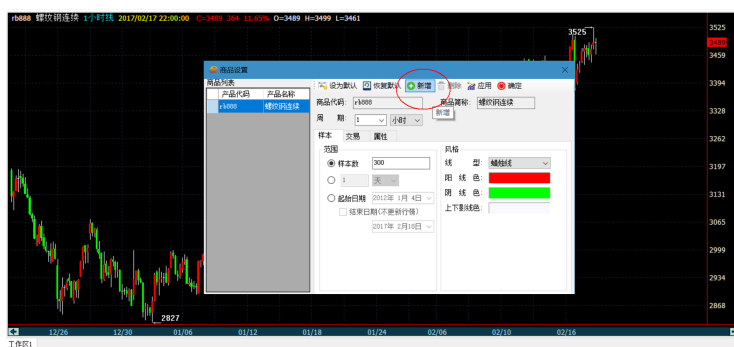
```
If(BarStatus == 2)
{
    Commentary("BarCount[1] = ", Text(BarCount[1]));
    Commentary("BarCount = ", Text(BarCount));
    Commentary("CurrentBar[1] = ", Text(CurrentBar[1]));
    Commentary("CurrentBar = ", Text(CurrentBar));
}
```

显示的结果分别是 301, 301, 300, 300, 原因在于, BarCount 是数值类型变量, 就是一个内存地址保存的值, 因此, 用不用索引, 其值都是一样的, 但是, 对于 CurrentBar, 它是序列型变量, 由于加入了条件判断语句, 在之前的 K 线上, 该函数并不会运行, 因此, 我们要索引前一根 K 线的索引值, 只能用当前 K 线的索引值来作为前一根 K 线的索引值。

## 42 多品种交易

### 1. 说明

- 多品种交易指的是图表中叠加多个商品，在一个公式中对多个商品进行交易。
- 默认情况下，如果公式中没有指定交易的品种，仅针对主图商品进行交易。
- 一个 K 线图最多可以叠加商品 4 个。
- 多个商品周期是固定的
- 查看方式，K 线图 -> 商品设置



- 默认的开、高、低、收、量等值在多品种交易中，指的就是主图的值

## 2. 示例 (双均线系统)

Params

```
Numeric ShortLength1(5);  
Numeric LongLength1(20);  
Numeric ShortLength2(5);  
Numeric LongLength2(20);
```

Vars

```
NumericSeries AvgShort_Com1(0);  
NumericSeries AvgLong_Com1(0);  
NumericSeries AvgShort_Com2(0);  
NumericSeries AvgLong_Com2(0);
```

Begin

```
// 主图设置
```

```
AvgShort_Com1 = AverageFC(data0.Close, ShortLength1);  
AvgLong_Com1 = AverageFC(data0.Close, LongLength1);  
AvgShort_Com2 = AverageFC(data1.Close, ShortLength2);  
AvgLong_Com2 = AverageFC(data1.Close, LongLength2);
```

```
If(data0.MarketPosition <> 1 And AvgShort_Com1 > AvgLong_Com1){  
    Buy(0, data0.Close);
```

```
}
```

```
If(data1.MarketPosition <> 1 And AvgShort_Com2 > AvgLong_Com2){  
    Buy(0, data1.Close);
```

```
}
```

```
If(data0.MarketPosition <> -1 And AvgShort_Com1 < AvgLong_Com1){  
    SellShort(0, data0.Close);
```

```
}
```

```
If(data1.MarketPosition <> -1 And AvgShort_Com2 < AvgLong_Com2){
```

```
SellShort(0, data1.Close);  
}  
  
End
```

## 43 控制开平仓、止盈止损

### 1. 两个函数

- **MarketPosition**

- 返回数值型变量的函数，分别是 -1 (空仓)、0 (没有持仓)、1 (多仓)
- 返回的仓位并不是我们账户的仓位，其判断的仓位逻辑如下：当前图标是其样本空间，策略从第一根 K 线开始运行，这时，系统仓是 0，然后策略按照条件不断开平仓，一直持续到当前最新的一根 Bar，最后的系统仓结余的仓位就是 **MarketPosition** 返回的值。
- 通过头寸监控，可以将系统仓与账户仓进行同步

- **BarSinceEntry**

- 主要用于控制平仓，在止盈止损编写模式中，可能会遇到下面情况，当前 Bar 既出现止盈条件，又出现止损条件，假如开仓后，价格下跌到止损条件，此时出现止损信号，这个出场是正确的，但是如果这个 Bar 柱先到了止损价格，再到开仓价格，这个 Bar 柱上同样出现了开平仓信号，先平仓，然后再开仓，这个是不正确的，K 线图无法描述哪个价格是先到，哪个价格是后来的，测试报告无法区分这个开平仓是否正确，它认为都是可以实现的，会让测试报告不准确。
- 为避免上述情况发生，在进场那根 Bar 柱不进行平仓，**BarSinceEntry** 返回当前图表中，第一根开仓的 Bar 柱到当前 Bar 柱的距离，中间是否由加仓对其没有影响，如果中间有平仓，然后再开仓，那么最新的开仓的 Bar 就是初始点。

## 2. 止盈止损

- 提供了一种出场方式，示例：

止盈 30 跳；止损 10 跳

Params

```
Numeric FastLength(5);  
Numeric SlowLength(20);  
Numeric Offset_Profit(30);  
Numeric Offset_Loss(10);
```

Vars

```
NumericSeries AvgValue1;  
NumericSeries AvgValue2;  
Numeric PriceOpen(0);  
// 尤其小心止盈止损价位，需要是序列型变量  
NumericSeries ProfitCut;  
NumericSeries LossCut;
```

Begin

```
AvgValue1 = AverageFC(Close, FastLength);  
AvgValue2 = AverageFC(Close, SlowLength);
```

```
PlotNumeric("MA1", AvgValue1, 0, Red);  
PlotNumeric("MA2", AvgValue2, 0, Green);
```

// 集合竞价和小节休息过滤

```
If(!CallAuctionFilter()) Return;
```

```
If(MarketPosition <> 1 && AvgValue1[1] > AvgValue2[1])  
{  
    Buy(1, Open);  
    PriceOpen = Open;
```

```

    ProfitCut = PriceOpen + Offset_Profit*MinMove*PriceScale;
    LossCut = PriceOpen - Offset_Loss*MinMove*PriceScale;
}

If(MarketPosition <> -1 && AvgValue1[1] < AvgValue2[1])
{
    SellShort(1,Open);
    PriceOpen = Open;
    ProfitCut = PriceOpen - Offset_Profit*MinMove*PriceScale;
    LossCut = PriceOpen + Offset_Loss*MinMove*PriceScale;
    Commentary("止盈价: "+Text(ProfitCut));
    Commentary("止损价: "+Text(LossCut));
    Commentary("最小变动: "+Text(MinMove));
    Commentary("最小变动单位: "+Text(PriceScale));
}

// 加入止盈止损
// 30 跳止盈
// 10 跳止损
If(MarketPosition == 1 And High >= ProfitCut And BarsSinceEntry >= 1)
{
    Sell(0, ProfitCut);
}
If(MarketPosition == 1 And Low <= LossCut And BarsSinceEntry >= 1)
{
    Sell(0, Min(Open, LossCut));
}
If(MarketPosition == -1 And Low <= ProfitCut And BarsSinceEntry >= 1)
{
    BuyToCover(0, ProfitCut);
}

```

```

        Commentary("最低价: "+Text(Low));
        Commentary("止盈价: "+Text(ProfitCut));
    }

    If(MarketPosition == -1 And High >= LossCut And BarsSinceEntry >= 1)
    {
        BuyToCover(0, Max(Open, LossCut));
        Commentary("最高价: "+Text(High));
        Commentary("止损价: "+Text(LossCut));
    }
End

```

## 44 跟踪止损 (移动止损)

### 1. 说明

在趋势类行情中, 如果设定固定止盈, 很可能错过一波大行情, 跟踪止损是一种自适应的出场方式。如果开仓后以后记录出现的最高价, 如果现在的价格距离最高点回撤了  $n\%$ , 那么就离场, 这属于跟踪止损。

### 2. 示例 (双均线策略, 移动止损加入固定止损)

```

Params
    Numeric FastLength(5);
    Numeric SlowLength(20);
    Numeric Offset_Profit(30);
    Numeric Offset_Loss(10);
Vars
    NumericSeries AvgValue1;
    NumericSeries AvgValue2;
    NumericSeries PriceOpen(0);
    NumericSeries ProfitCut;

```



```

NumericSeries LossCut;
NumericSeries TakeProfitPrice(0);

Begin
    AvgValue1 = AverageFC(Close, FastLength);
    AvgValue2 = AverageFC(Close, SlowLength);

    PlotNumeric("MA1", AvgValue1, 0, Red);
    PlotNumeric("MA2", AvgValue2, 0, Green);

    // 集合竞价和小节休息过滤
    If(!CallAuctionFilter()) Return;

    If(MarketPosition <> 1 && AvgValue1[1] > AvgValue2[1])
    {
        Buy(1, Open);
        PriceOpen = Open;
        /*ProfitCut = PriceOpen + Offset_Profit*MinMove*PriceScale;
        LossCut = PriceOpen - Offset_Loss*MinMove*PriceScale;*/
        // 保护性止损
        // 如果错估形式，可能开仓后行情是反向的
        // 需要加入保护性止损
        LossCut = PriceOpen - Offset_Loss/1000*PriceOpen;
        Commentary("保护性止损价格: "+Text(LossCut));
    }

    If(MarketPosition <> -1 && AvgValue1[1] < AvgValue2[1])
    {
        SellShort(1, Open);
        PriceOpen = Open;
    }

```

```

    /*ProfitCut = PriceOpen - Offset_Profit*MinMove*PriceScale;
    LossCut = PriceOpen + Offset_Loss*MinMove*PriceScale;*/
    // 保护性止损
    LossCut = PriceOpen - Offset_Loss/1000*PriceOpen;
}

// 加入跟踪止盈止损
If(MarketPosition == 1){
    If(High > PriceOpen) PriceOpen = High;
    TakeProfitPrice = PriceOpen - Offset_Profit/1000*PriceOpen;
    Commentary("移动止盈价格: "+Text(TakeProfitPrice));
}
If(MarketPosition == 1 And Low < TakeProfitPrice[1] And BarsSinceEntry >= 1){
    If(LossCut > TakeProfitPrice[1]){
        Sell(0, Min(Open, LossCut));
    }
    If(LossCut < TakeProfitPrice[1]){
        Sell(0, Min(Open, TakeProfitPrice[1]));
    }
}
If(MarketPosition == -1){
    If(Low < PriceOpen) PriceOpen = Low;
    TakeProfitPrice = PriceOpen + Offset_Profit/1000*PriceOpen;
    Commentary("移动止盈价格: "+Text(TakeProfitPrice));
}
If(MarketPosition == -1 And Low < TakeProfitPrice[1] And BarsSinceEntry >= 1){
    If(LossCut > TakeProfitPrice[1]){
        BuyToCover(0, Min(Open, TakeProfitPrice[1]));
    }
    If(LossCut < TakeProfitPrice[1]){

```

```
        BuyToCover(0, Min(Open, LossCut));  
    }  
}  
End
```

## 45 加减仓

### 1. 说明

- 首次开仓 2 手后每盈利 30 跳加仓一次，每次 1 手，最多加仓 3 次 (意味着 `CurrentEntries` 最大值是 4)
- 每亏损 30 跳减仓 1 手
- `CurrentEntries` 返回指定图层上当前持仓的建仓次数

### 2. 示例

## 46 收盘平仓