参数输入模块

数据库

主循环 Main loop

结果表单

Out\_dt

Traiding\_dt

数据分析，制图

数据清洗

Dominant\_Bars（list））

Day,product\_id…

资金管理

* 去空值
* 计算要增加参数（$/point, n日均线）
* 从list数据合并到dt

System1

System2

Data\_dt;cdt（ data.frame））

* 生成交易参数
* Account: 记录现有资产总值
* ShadowAccount: 用以生成Unit （开始等于Account,当Accoutn = 90% Shadow Account时进行调整（80%），当Account等于1.25 shadow时上调，直至等于初始Account）。当Account > Acc\_origin 就不用调整。
* 二次判定投资约束[4,6,10,12]，不符合则报错退出

Account；ShadowAccount; Slippage; vm; fee …

ShadowAccount； cdt; system.select(1|2)， Units

进场开仓

Trading\_In(empty), Standing\_Record(empty)， Cash，开仓表…

* 判定系统（1、2），（Cash小于所有Unit跳过）
* 判定通道突破
  + 记录开仓信号 （date, sig\_price, high, good/bad）
  + 信号强弱排序 strength(5,3,1,4,2…)

（looping）j in 1:length(strength), index strength to get prices

* + Standing\_Contracty(enter\_price,enter\_date,stop\_price,fee)
  + Position (0,0,0,0,0) -> （1,0,0,0,0），测试投资约束[4,6,10,12]
* 判断日内加仓并测试投资约束[4,6,10,12]，然后计入trade\_in，否则不改position
* Trade\_In : append by standing buy, 以及其他参数 /另外需要更新各种profit和position
* Cash <0 也不改position

退场

Position； Trading\_Record; cdt

* 根据position(1,0,0,0,0)确定哪些资产需要判定（=0 skip）
* 判定日内平仓（如果low小于止损点，close接近low则日内平仓）
* 判定平仓，最后修改trade\_out （根据Standing\_Contract增加quit\_price,profit,direction），更新profit、fee等等
* 写一个留给Whipsaw的接口，以后可以更新这个止损功能

Position； Trading\_Record; cdt

获利了结

* 未突破止损价，但突破了n天最高/最低（对应空仓/多仓）判定
* 与退场模块要做的事情完全相同，只是利润可能为正。

日结

* 更新资产表

临时数据存储向量

Trade\_Out

临时数据存储list

Standing\_Contract

（临时存储现有合约）

资产表（更新每日的日期，position（资产总额），cash，closed\_profit, position\_profit）

Data(list)

存储了所有产品数据

cdt(data.frame)

存储了所有产品数据

((5+i)\*k列)

Update From Standing Contract

Enter. Date

Exit. Date

Enter/Exit

Contract (long/short?)

Profit/Lost