

# 股票池择时策略 实验手册

详细介绍

# 策略逻辑

初始化

- 设定回测基准标的 : `set_benchmark('000300.XSHG')`
- 设定交易成本 : `set_order_cost ()`

交易

- 计算20分钟K线 : `g.pl_bar_number % PL_TRADE_BAR_DURATION==0`
- 卖出持仓股票 : `pl_sell(context)`
  - 跌停判断 : `pl_is_low_limit ()`
- 仓位计算 : `pl_calc_position(context,pl_code)`
- 买入 : `pl_buy(context)`
  - 涨停判断 : `pl_is_high_limit ()`

盘后统计

- 更新股票池 : `pl_stock_pool(context)`
  - 剔除**ST**股票
- 获取指标数据 : `pl_load_fundamentals_data(context)`
- 市值过滤 : `pl_filtered_market_cap`
- PE过滤 : `pl_filtered_pe`
- 涨幅过滤 : `pl_load_change_pct_data ()`

# 全局参数

- # 股票池计算涨跌幅的窗口大小  
PL\_CHANGE\_PCT\_DAY\_NUMBER = 25
- # 更新股票池的间隔天数  
PL\_CHANGE\_STOCK\_POOL\_DAY\_NUMBER = 25
- # 两次处理交易逻辑的窗口大小  
PL\_TRADE\_BAR\_DURATION = 20
- # 计算数据时的bar的单位  
PL\_UNIT = str(PL\_TRADE\_BAR\_DURATION) + 'm'
- # 买入信号中的短时均线长度  
PL\_BUY\_SHORT\_MA = 3
- # 买入信号中的长时均线长度  
PL\_BUY\_LONG\_MA = 200
- # 卖出信号中的短时均线长度  
PL\_SELL\_SHORT\_MA = 3
- # 卖出信号中的长时均线长度  
PL\_SELL\_LONG\_MA = 200

# 关键函数及变量说明

- `initialize(context)` : 策略初始化函数，策略启动时被调用
  - `set_order_cost(OrderCost(close_tax=0.001, open_commission=0.0003, close_commission=0.0003, min_commission=5), type='stock')` : 设定交易成本, `close_tax=0.001`千分之一的印花税, `open_commission=0.0003`万分之三开仓费用, `close_commission=0.0003`万分之三平仓费用, `min_commission=5`每次交易最低5元
  - `pl_init_global(context)` : 初始化全局变量, 全局变量为 g.xxx 格式
  - `run_daily()` : 最重要的函数, 指定开盘前、盘中、收盘后的交易逻辑

# 关键函数及变量说明

- `run_daily(pl_before_market_open, time='before_open', reference_security='000300.XSHG')`
- `run_daily(pl_trade, time='every_bar',reference_security='000300.XSHG')`
- `run_daily(pl_after_market_close, time='after_close', reference_security='000300.XSHG')`
  - `run_daily`定义了在指定的时间，执行指定的交易函数
  - 参数1：被自动调用的交易函数
  - 参数2：指定的时间
    - `before_open`：每个交易日开盘前
    - `every_bar`：每根K线的开始，K线的时间级别在聚宽的界面上设定（每天、分钟、tick）
    - `after_close`：每个交易日收盘后

# 关键函数及变量说明

- `pl_trade(context)` : 盘中交易逻辑函数
  - `g.pl_bar_number` : 20分钟K线计数器
  - `pl_sell(context)` : 卖出持仓股函数
    - `pl_is_low_limit(pl_code)` : 跌停判断
    - `pl_cross(pl_short_ma,pl_long_ma)` : 死叉计算
  - `pl_buy(context)` : 买入函数
    - `pl_is_high_limit(pl_code)` : 涨停判断
    - `pl_cross(pl_short_ma,pl_long_ma)` : 金叉计算
    - `pl_calc_position(context,pl_code)` : 计算仓位

# 关键函数及变量说明

- `pl_after_market_close(context)` : 盘后更新股票池
  - `g.pl_stock_pool_update_day` : 再平衡周期计数器
  - `pl_stock_pool(context)` : 更新股票池
- `pl_stock_pool(context)` : 更新股票池
  - `pl_current_data.is_st` : 剔除ST股票
  - `pl_load_fundamentals_data(context)` : 获取财务指标数据
    - `get_fundamentals(query(valuation,indicator), context.current_dt.strftime("%Y-%m-%d"))`
    - `Valuation` : 市值数据
    - `Indicator` : 财务指标数据
  - `pl_filtered_market_cap` : 根据市值过滤股票
  - `pl_filtered_pe` : 根据PE过滤股票
  - `pl_load_change_pct_data()` : 根据涨幅过滤股票

# 想一想，练一练

- 请修改策略的参数并回测，看看不同的参数会对策略性能有什么影响？
  - 把策略择时信号的K线时间级别改为30分钟、60分钟，看看回测结果有什么变化？
- 请修改代码，并投稿