30 行代码讲透多因子模型

回测区间：2016-01-12 至 2017-06-30
回测频率：日级
回测资金：100 万

二八轮动

回测区间：2013-06-28 至 2017-07-19
回测频率：日级
回测资金：10 万

二八轮动源代码：

```
#初始化账户
def initialize(account):
    #调仓频率
    account.trade_date = range(1,13,1)
    ## 按月调用程序
    run_weekly(trade,date_rule=1)
#-----------------------------------------函 数 区
-----------------------------------
#-----------------------------------判 断 牛 熊 值
-----------------------------------
def zhong2(account,data):
    price = data.attribute_history('000300.SH', ['close'], 25, '1d')
    p20=price['close'].iloc[-20]
    p1=price['close'].iloc[-1]
    n=(p1-p20)/p20
    log.info(n)
    return n
def zhong8(account,data):
    price = data.attribute_history('399006.SZ', ['close'], 25, '1d')
    p20=price['close'].iloc[-20]
    p1=price['close'].iloc[-1]
    m=(p1-p20)/p20
    log.info(m)
    return m
#---------------------------------- 牛熊值止损函数----------------------
def trade(account,data):
    #获取牛熊值
    n=zhong2(account,data)
    m=zhong8(account,data)
```

```
    date = get_datetime()
    months = get_datetime().month
    if months in account.trade_date:
        if n <= 0 and m <= 0:
            if len(account.positions) > 0:
                for stock in list(account.positions):
                    order_target(stock, 0)
        elif n > m:
            if '510500.OF' in list(account.positions):
                order_target('510500.OF', 0)
            if len(account.positions) < 1:
                order_target_value('510300.OF', account.cash)
        elif n < m:
            if '510300.OF' in list(account.positions):
                order_target('510300.OF', 0)
            if len(account.positions) < 1:
                order_target_value('510500.OF', account.cash)
```

30 行代码讲透多因子模型，源代码

```
import pandas as pd
import numpy as np
import datetime

#----------------------------------基本参数设置
------------------------------------------------

# 账户初始化函数
def initialize(account):
    # 使用智能选股函数 get_iwencai 生成股票池
    get_iwencai('沪深 300')
    # 设置最大持股数
    account.max_stocks = 5
    # 设置调仓周期，每月第 1 个交易日运行
    run_monthly(func = reallocation, date_rule = 1)

def reallocation(account,data):
    #       ----------------------------------------获取数据
----------------------------------------
    # 获得调仓日前一个交易日日期（系统默认当日开盘执行，所以应获取前一个交易日）
    yst_date = get_datetime().strftime('%Y%m%d')
```

```
    # 获得股票池列表（get_iwencai 函数的选股结果自动存储于 account.iwencai_securities 中）
    sample = account.iwencai_securities

    # 筛选需要获取的因子数据
    q = query(profit.symbol, # 股票代码
                profit.date, # 日期
                profit.roic, # 投资回报率
                valuation.pb, # 市净率
                valuation.pe, # 市盈率
                ).filter(
                    profit.symbol.in_(sample)
                    )
    # 获取前一个交易日的因子数据，并将缺失值填充为 0
    df = get_fundamentals(q, date = yst_date).fillna(0)

    #----------------------------------------数据处理----------------------------------------
    # 因子极值处理（中位数去极值法）
    # 因子名称为后面 3 个字段
    for i in list(df.columns)[-3:]:
        m = np.mean(df[i]) # 求均值
        s = np.std(df[i]) # 求标准差
        df[i] = df[i].where(df[i] > m-3*s).fillna(m-3*s) # 将小于 m-3*s 的值替换成 m-3*s
        df[i] = df[i].where(df[i] < m+3*s).fillna(m+3*s)# 将大于 m+3*s 的值替换成 m+3*s

    # 因子无量纲处理
    for j in list(df.columns)[-3:]:
        m = np.mean(df[j]) # 求均值
        s = np.std(df[j]) # 求标准差
        df[j] = (df[j]-m)/s # 标准化

    #-------------------------------------因子选股-------------------------------------------------------------------------------
    # 计算综合因子（因子方向为正向，则+；为反向，则-）
    df['score'] = df['profit_roic'] - df['valuation_pb'] - df['valuation_pe']
    # 按综合因子值由大到小排序
    df = df.sort_values(by = 'score', ascending = False)
    # 选取前 N 只股票，N 为最大持股数
    buy_list = df['profit_symbol'].values[:account.max_stocks]
```

```
    # 遍历当前持仓股票
    for stk in list(account.positions):
        # 若持仓股不在待买股票池中，说明该股应该调出
        if stk not in buy_list:
            #清仓该股
            order_target_percent(stk,0)

    #-----------------------------------调                                仓
-------------------------------------------------------------------------------
    # 遍历待买股票池
    for stk in buy_list:
        # 若待买股不在持仓股票池中，说明该股应该调入
        if stk not in account.positions:
            # 买入该股，买入资金量为 1/N，N 为最大持股数
            order_target_percent(stk,1/account.max_stocks)
```