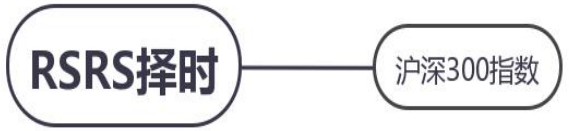
巴菲特的alpha增强版

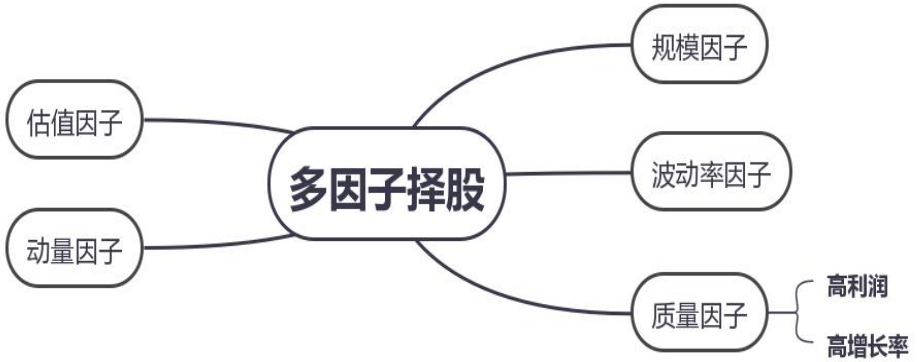
汇报人：陈鸿达

策略

择时



择股



✓ 回测结果

回测日期：2015-01-01 至 2021-05-31 | 资金：10000000 | 频率：每日 | 总运行时长：2130秒（完成）

导出结果

收益概览 | 交易明细 | 历史持仓 | 输出日志 | 收益 | Alpha | Beta | Sharpe | Sortino | Information Ratio | Volatility | 最大回撤

策略收益	基准收益	策略年化收益率	基准年化收益	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	最大回撤	Tracking Error	Downside Risk	胜率
381.78%	50.88%	28.68%	6.82%	0.24	0.39	1.45	1.9	0.82	0.18	20.26%	0.21	0.14	39%

缩放：1周 1月 3月 6月 1年 全部 — 基准收益 — 策略收益 — 超额收益

普通轴 对数轴 从 2014-12-31 至 2021-05-31

目录

Contents

RSRS择时

数据剔除

因子数据处理

交易函数

个股止损

1

RSRS择时

✓ RSRS择时

第一种方法是直接将斜率作为指标值。当日RSRS斜率指标择时策略如下：

1、取前N日最高价与最低价序列。（N = 18）

2、将两个序列进行OLS线性回归。

3、将拟合后的β值作为当日RSRS斜率指标值。

4、当RSRS斜率大于S_{buy}时，全仓买入，小于S_{sell}时，卖出平仓。（$S_{buy} = 1, S_{sell} = 0.8$）

由于市场处于不同时期时，斜率的均值有比较大的波动。因此，直接采用斜率均值作为择时指标并不太合适。我们尝试下面的方法。

第二种方法是在斜率基础上进行标准化，取标准分作为指标值。RSRS斜率标准分指标择时策略如下：

1、取前M日的RSRS斜率时间序列。（M = 600）

2、计算当日RSRS斜率的标准分$RSRS_{std}$：

$$RSRS_{std} = \frac{RSRS - \mu_M}{\sigma_M}$$

其中μ_M为前M日的斜率均值，σ_M为前M日的标准差。

3、若$RSRS_{std}$大于S_{buy}，则全仓买入；若$RSRS_{std}$小于S_{sell}，则卖出平仓。（$S_{buy} = 0.7, S_{sell} = -0.7$）

2

剃除数据

✓ 去除ST，停牌，去新股，以及去除开盘涨停股

```
#去除开盘涨停股票
def fun_highlimit(bar_dict,stock_list):
    return [stock for stock in stock_list if bar_dict[stock].open!
        =bar_dict[stock].high_limit]
#去除st股票
def fun_st(bar_dict,stock_list):
    return [stock for stock in stock_list if not bar_dict[stock].is_st]
#去除停牌
def fun_unpaused(bar_dict, stock_list):
    return [s for s in stock_list if not bar_dict[s].is_paused]
#去新股
def fun_remove_new(_stock_list, days):
    deltaDate = get_datetime() - dt.timedelta(days)
    stock_list = []
    for stock in _stock_list:
        if get_security_info(stock).listed_date < deltaDate:
            stock_list.append(stock)
    return stock_list
```

3

因子处理

✓ 获取多因子数据

```
#规模因子
cap_df = market_cap(stock_list, 'valuation_market_cap',last_date)
cap_df = cap_df * -1
```

```
#估值因子
PB_df = PB(stock_list, 'valuation_pb',last_date)
PB_df = PB_df * -1
```


```
#波动率因子
ATR20_df = ATR20(stock_list, 'ATR20')
ATR20_df = ATR20_df * -1
```

```
def ATR20(stocklist, new_factor):
    # 取数据
    for stock in stocklist:
        Data_ATR = history(stock,['close','high','low'],20,'1d')
        close_ATR = np.array(Data_ATR['close'])
        high_ATR = np.array(Data_ATR['high'])
        low_ATR = np.array(Data_ATR['low'])

        ATR = ta.ATR(high_ATR, low_ATR, close_ATR, timeperiod=1)

        indices = ~np.isnan(ATR)
        result = np.average(ATR[indices])
        s = pd.Series(result.astype(float), index=[stock])
        if 'ATR_df' in locals():
            ATR_df = ATR_df.append(s)
        else:
            ATR_df = s
    df = ATR_df.to_frame()
    df.index.name = 'valuation_symbol'
```

```
#动量因子
MTM20_df = MTM20(stock_list, 'MTM20')
MTM20_df=MTM20_df* -1
```

```
def MTM20(stocklist, factor):
    # 取数据
    for stock in stocklist:
        df1=history(stock,['close'],20,'1d')
        s = pd.DataFrame([(df1['close'][-1]-df1['close'][0]
            )/df1['close'][0]], index=[stock])
        if 'df' in locals():
            df = df.append(s)
        else:
            df = s
    df.columns = ['MTM20']
    df.index.name = 'valuation_symbol'
```

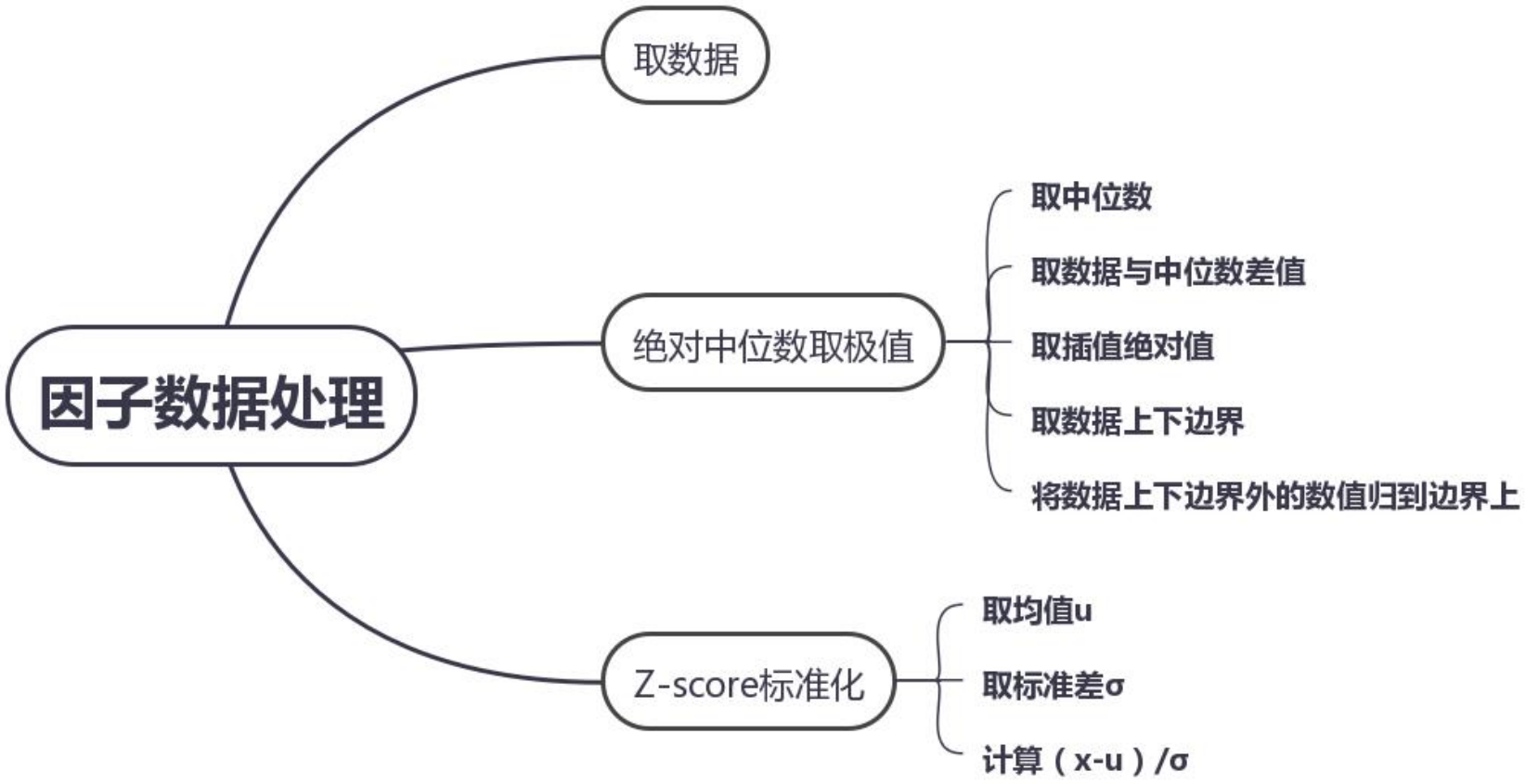
```
#2.净利润同比增长率（高成长）
net_profit_growth_ratio_df=net_profit_growth_ratio(stock_list
    ,'growth_net_profit_growth_ratio',last_date)
```

```
def net_profit_growth_ratio(stocklist, factor,last_date):
    # 取数据
    df = get_fundamentals(query(valuation.symbol, growth
        .net_profit_growth_ratio).filter(valuation.symbol.in_(stocklist
        )),date=last_date)
    df = df.set_index('valuation_symbol')
```

```
#1.ROE（高利润）
roe_df = roe(stock_list, 'profit_roe_ths',last_date)
```

```
def roe(stocklist, factor,last_date):
    # 取数据
    df = get_fundamentals(query(valuation.symbol, profit.roe_ths).filter
        (valuation.symbol.in_(stocklist)),date=last_date)
```

✓ 因子数据处理


因子数据处理
取数据
绝对中位数取极值
取中位数
取数据与中位数差值
取插值绝对值
取数据上下边界
将数据上下边界外的数值归到边界上
Z-score标准化
取均值u
取标准差σ
计算（x-u）/σ

✓ 多因子合并，得到选出的股票列表

- 按股票代码合并，然后加总因子得分，去除空值的股票代码

```
#合并多因子
concat_obj = [cap_df, PB_df,MTM20_df,roe_df
    ,net_profit_growth_ratio_df,ATR20_df]
df = pd.concat(concat_obj, axis=1)
df = df.dropna()
sum = df.sum(axis=1)
log.info(sum)
```

- 从大到小排序，取前5%的股票

```
sum.sort_values(ascending = False,inplace=True)
stock_list1 = sum[0:int(len(stock_list)*g.long_pct)].index
```

4

交易函数

✓ 不累计买入

```
buy_list = []
for stock in stock_list1:
    macd = get_ma(stock)
    if (macd[-1] > macd[-3]):
        buy_list.append(stock)
#sell
for stock in list(context.portfolio.positions):
    macd = get_ma(stock)
    if (macd[-1] < macd[-3]):
        order_target(stock, 0)
#去除现有持仓
buy_list2 = list(set(buy_list)-set(list(context.portfolio.positions
    )))
cash = context.portfolio.portfolio_value
position=cash*g.total_positionprevious
num=int(len(buy_list2))
## 买入
for stock in buy_list2:
    order_target_value(stock,position/num)
```

✓ 不累计买入回测

回测日期：2015-01-01 至 2021-05-31 | 资金：10000000 | 频率：每日 | 总运行时长：1825秒 (完成)

导出结果

收益概览 | 交易明细 | 历史持仓 | 输出日志 | 收益 | Alpha | Beta | Sharpe | Sortino | Information Ratio | Volatility | 最大回撤

策略收益	基准收益	策略年化收益率	基准年化收益	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	最大回撤	Tracking Error	Downside Risk	胜率
371.78%	50.88%	28.25%	6.82%	0.23	0.5	1.35	2.09	0.92	0.19	21.01%	0.19	0.12	53.43%

缩放: 1周 1月 3月 6月 1年 全部 — 基准收益 — 策略收益 — 超额收益 普通轴 对数轴 从 2014-12-31 至 2021-05-31

✓ 累积买入

```
buy_list = []

for stock in stock_list1:
    buy_list.append(stock)

#买卖操作
for stock in list(context.portfolio.positions):
    if stock not in buy_list:
        order_target(stock, 0)
cash = context.portfolio.portfolio_value
position=cash*g.total_positionprevious
num=int(len(buy_list))
## 买入
for stock in buy_list:
    order_target_value(stock,position/num)
```

5

个股止损

✓ 个股止损

每天运行一次，当下跌超过3%则平仓止损

```
#个股止损
def stop_loss(context,bar_dict):
    for stock in list(context.portfolio.positions):
        cumulative_return=bar_dict[stock].close/context.portfolio
            .positions[stock].cost_basis
        if cumulative_return<0.97:
            order_target_value(stock,0)
```

6

改进

✓ 改进1：价格优化交易策略

一个直接的想法是从近期历史价格趋势进行判断，在回测中，使用前1日的20日均线值和3日前的20日均线值的相对大小牌判断近期市场状态。策略如下（取值来自研报）：

1、计算RSRS右偏标准分指标$RSRS_{rightdev}$。（N = 18, M = 600）

2、若$RSRS_{rightdev}$大于S_{buy}，同时满足前1日的MA20的值大于前3日的MA20的值，则全仓买入。（$S_{buy} = 0.7$）

3、若$RSRS_{rightdev}$小于S_{sell}，同时满足前1日的MA20的值小于前3日的MA20的值，则卖出平仓。（$S_{sell} = -0.7$）

```
def get_ma(stock):
    price = history(stock, ['close'], 500, '1d', True, 'pre', is_panel=1
        )['close']
    MA = ta.SMA(price.values,
            timeperiod = 20)
    return MA
```

```
#buy
for stock in stock_list1:
    macd = get_ma(stock)
    if (macd[-1] > macd[-3]) and total_position==1:
        buy_list.append(stock)

#sell
for stock in list(context.portfolio.positions):
    macd = get_ma(stock)
    if (macd[-1] < macd[-3]) and total_position==0:
        order_target(stock, 0)
```

回测结果

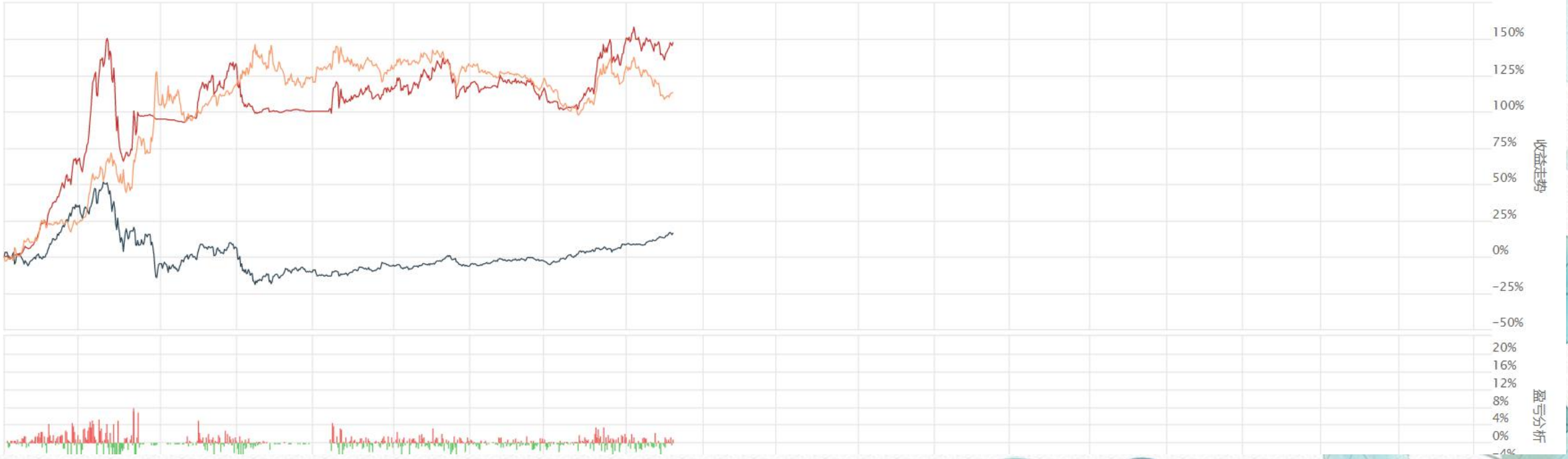
回测日期：2015-01-01 至 2021-05-31 | 资金：10000000 | 频率：每日 | 总运行时长：1568秒（完成）

导出结果

收益概览 | 交易明细 | 历史持仓 | 输出日志 | 收益 | Alpha | Beta | Sharpe | Sortino | Information Ratio | Volatility | 最大回撤

策略收益	基准收益	策略年化收益率	基准年化收益	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	最大回撤	Tracking Error	Downside Risk	胜率
220.84%	50.88%	20.56%	6.82%	0.16	0.52	0.82	1.28	0.54	0.22	33.8%	0.21	0.14	52.31%

缩放: 1周 1月 3月 6月 1年 全部 — 基准收益 — 策略收益 — 超额收益 ◉普通轴 ○对数轴



✓ 改进2：利用持股数据 择时

```
#rsrs计算历史数据窗口
def rsrs(context):
    initlast_date=context.now-timedelta(days=1)
    prices = get_price(g.stock, '2011-06-05', initlast_date, '1d', ['high', 'low'])
    #获取最高价和最低价
    highs = prices.high
    lows = prices.low
    for i in range(len(highs))[g.N:]:
        data_high = highs.iloc[i-g.N+1:i+1]
        data_low = lows.iloc[i-g.N+1:i+1]
        X = sm.add_constant(data_low)
        model = sm.OLS(data_high,X)
        results = model.fit()
        g.ans.append(results.params[1])
#每天开盘前运行，初始化rsrs数据窗口
def before_trading(context):
    #如果持股不为空
    if g.stock_list:
        #对手中的股票建立择时历史数据窗口
        rsrs_init(context)
        try:
            g.stock = g.stock_list
            rsrs(context)
        except:
            #如果持股数据存在确实，那么仍然利用沪深300指数择时
            g.stock = '000300.SH'
            rsrs(context)
    elif context.flag:
        #先建立过去的数据窗口
        rsrs(context)
        context.flag=False
```

回测结果

回测日期：2015-01-01 至 2021-05-31 | 资金：10000000 | 频率：每日 | 总运行时长：1383秒（完成）

导出结果

收益概览 | 交易明细 | 历史持仓 | 输出日志 | 收益 | Alpha | Beta | Sharpe | Sortino | Information Ratio | Volatility | 最大回撤

策略收益	基准收益	策略年化收益率	基准年化收益	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	最大回撤	Tracking Error	Downside Risk	胜率
328.45%	50.88%	26.28%	6.82%	0.22	0.37	1.34	1.71	0.72	0.18	20.25%	0.21	0.14	38.87%

缩放：1周 1月 3月 6月 1年 全部 — 基准收益 — 策略收益 — 超额收益

普通轴 对数轴 从 2014-12-31 至 2021-05-31

放松买入条件，加紧卖出条件（个股3%止损，无均线）

回测日期：2015-01-01 至 2021-05-31 | 资金：10000000 | 频率：每日 | 总运行时长：1577秒（完成）

导出结果

收益概览 | 交易明细 | 历史持仓 | 输出日志 | 收益 | Alpha | Beta | Sharpe | Sortino | Information Ratio | Volatility | 最大回撤

策略收益	基准收益	策略年化收益率	基准年化收益	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	最大回撤	Tracking Error	Downside Risk	胜率
257.84%	50.88%	22.68%	6.82%	0.19	0.32	1.42	1.51	0.59	0.14	19.77%	0.2	0.13	50.16%

加紧买入条件（个股3%止损）

```
#个股止损
def stop_loss(context,bar_dict):
    for stock in list(context.portfolio.positions):
        cumulative_return=bar_dict[stock].close/context.portfolio.positions[stock].cost_basis
        if cumulative_return<0.97:
            order_target_value(stock,0)
```

```
buy_list = []
for stock in stock_list1:
    macd = get_ma(stock)
    if (macd[-1] > macd[-3]):
        buy_list.append(stock)
#sell
for stock in list(context.portfolio.positions):
    # macd = get_ma(stock)
    # if (macd[-1] < macd[-3]):
    if stock not in buy_list:
        order_target(stock, 0)
```

收益概览 | 交易明细 | 历史持仓 | 输出日志 | 收益 | Alpha | Beta | Sharpe | Sortino | Information Ratio | Volatility | 最大回撤

策略收益	基准收益	策略年化收益率	基准年化收益	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	最大回撤	Tracking Error	Downside Risk	胜率
363.56%	50.88%	27.88%	6.82%	0.24	0.29	1.51	1.75	0.73	0.17	14.15%	0.23	0.14	37.59%

7

调参

✓ 调仓频率 10

巴菲特pro10天调一次仓

返回策略目录 | 返回策略编辑 | 返回回测列表

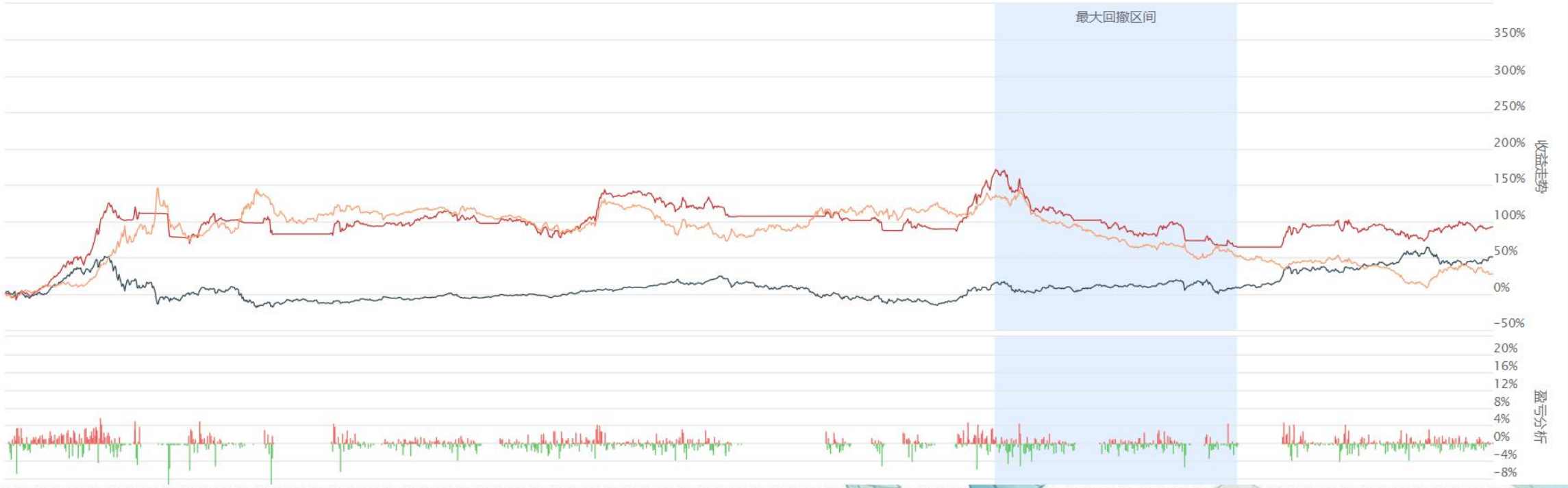
回测日期：2015-01-01 至 2021-05-31 | 资金：10000000 | 频率：每日 | 总运行时长：2209秒（完成）

导出结果

收益概览 | 交易明细 | 历史持仓 | 输出日志 | 收益 | Alpha | Beta | Sharpe | Sortino | Information Ratio | Volatility | 最大回撤

策略收益	基准收益	策略年化收益率	基准年化收益	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	最大回撤	Tracking Error	Downside Risk	胜率
92.47%	50.88%	11.07%	6.82%	0.07	0.46	0.41	0.57	0.14	0.21	39.48%	0.22	0.15	41.82%

缩放：1周 1月 3月 6月 1年 全部 — 基准收益 — 策略收益 — 超额收益 ◉普通轴 ○对数轴 从 2014-12-31 至 2021-05-31



✓ 调仓频率 30

巴菲特proMonth

返回策略目录 | 返回策略编辑 | 返回回测列表

回测日期：2015-01-01 至 2021-05-31 | 资金：10000000 | 频率：每日 | 总运行时长：253秒（完成）

导出结果

收益概览 | 交易明细 | 历史持仓 | 输出日志 | 收益 | Alpha | Beta | Sharpe | Sortino | Information Ratio | Volatility | 最大回撤

策略收益	基准收益	策略年化收益率	基准年化收益	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	最大回撤	Tracking Error	Downside Risk	胜率
64.87%	50.88%	8.35%	6.82%	0.04	0.41	0.37	0.44	-0.01	0.16	44.02%	0.19	0.13	54.59%

✓ 剔除周期行业

回测日期：2015-01-01 至 2021-05-31 | 资金：100000 | 频率：每日 | 总运行时长：625秒（完成）

导出结果

收益概览 | 交易明细 | 历史持仓 | 输出日志 | 收益 | Alpha | Beta | Sharpe | Sortino | Information Ratio | Volatility | 最大回撤

策略收益	基准收益	策略年化收益率	基准年化收益	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	最大回撤	Tracking Error	Downside Risk	胜率
270.35%	50.88%	23.36%	6.82%	0.19	0.43	1.23	1.66	0.67	0.17	23.27%	0.19	0.12	53.18%

缩放：1周 1月 3月 6月 1年 全部 — 基准收益 — 策略收益 — 超额收益

普通轴 对数轴 从 2014-12-31 至 2021-05-31

✓ 累计买入 +3%个股止损+均线买入

```
buy_list = []
for stock in stock_list1:
    macd = get_ma(stock)
    if (macd[-1] > macd[-3]):
        buy_list.append(stock)
#sell
for stock in list(context.portfolio.positions):
    # macd = get_ma(stock)
    # if (macd[-1] < macd[-3]):
    if stock not in buy_list:
        order_target(stock, 0)
#去除现有持仓
#buy_list2 = list(set(buy_list)-set(list(context.portfolio.positions)))
cash = context.portfolio.portfolio_value
position=cash*g.total_positionprevious
num=int(len(buy_list))
## 买入
for stock in buy_list:
    order_target_value(stock,position/num)
```

回测日期：2015-01-01 至 2021-05-31 | 资金：10000000 | 频率：每日 | 总运行时长：1534秒（完成）

导出结果

收益概览 | 交易明细 | 历史持仓 | 输出日志 | 收益 | Alpha | Beta | Sharpe | Sortino | Information Ratio | Volatility | 最大回撤

策略收益	基准收益	策略年化收益率	基准年化收益	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	最大回撤	Tracking Error	Downside Risk	胜率
418.04%	50.88%	30.18%	6.82%	0.27	0.22	1.86	1.9	0.77	0.15	11.63%	0.23	0.15	32.97%

缩放：1周 1月 3月 6月 1年 全部 — 基准收益 — 策略收益 — 超额收益 ◉普通轴 ○对数轴 从 2014-12-31 至 2021-05-31

✓ Compared to the origin

感谢您的观看