



严谨 认真 持续

# 用Python写一个均值回复策略

量化投资实战交易体系必修课

系列4 - 快速开发一个量化策略

# 目录

CONTENT

策略原理

PART ONE

---

编程实现


PART TWO

---

改进思路

PART THREE

---



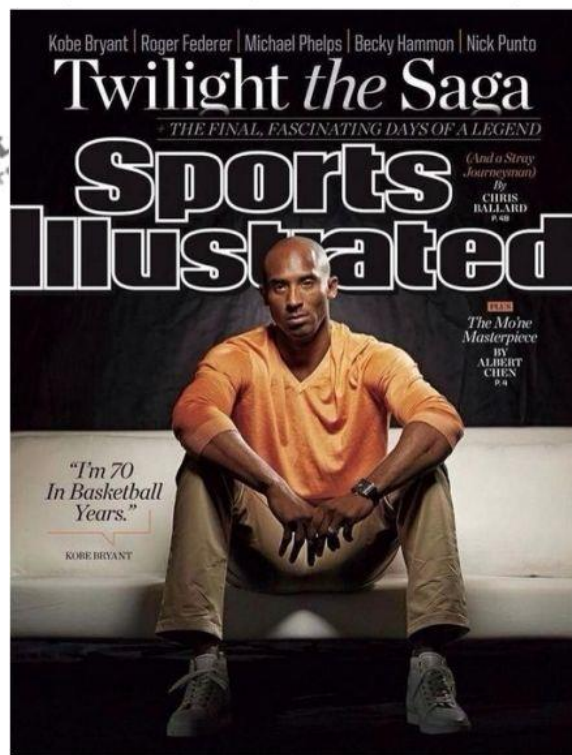
# 策略原理

## PART ONE

---

# 《体育画报》的厄运

Kobe Bryant on the new cover of Sports Illustrated: "I'm 70 in basketball years."



一个运动员的照片如果出现在杂志封面，  
那他注定要在接下来的赛季中表现不佳。

- 丹尼尔·卡尼曼 2011

一个运动员的表现可以被认为是围绕均值随机分布的

## 短期市场反转

建立一个股票池：

过去3（或1）个月表现最差的N只股票构成的组合

再平衡周期：

- 1个月

头寸管理：

- 所有入选股票均仓
- 按照市值加权

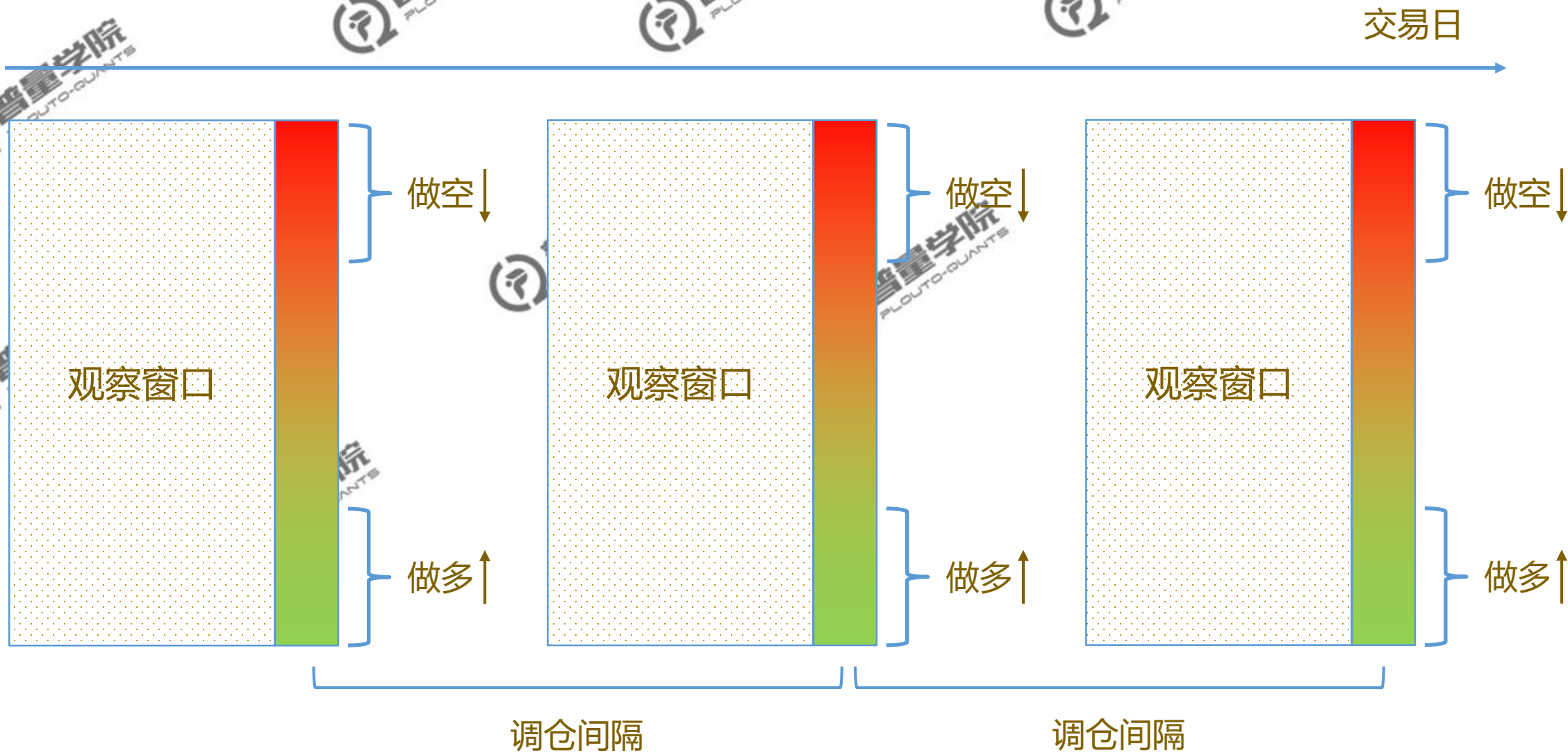
多空组合：

做多表现最差组合

同时

做空表现最好组合

# 均值回复策略示意图





# 编程实现

## PART TWO



## 策略编写的考虑

- 投资标的选择
  - 投资组合的容量
  - 多空双向交易
  - 做多/做空机制



A股

融资/融券

Python



# 编程平台工作界面1

JoinQuant 聚宽

首页 我的策略 策略广场 基金经理孵化 策略大赛 数据 帮助 量化课堂 社区 个人账号

< Mean Regression

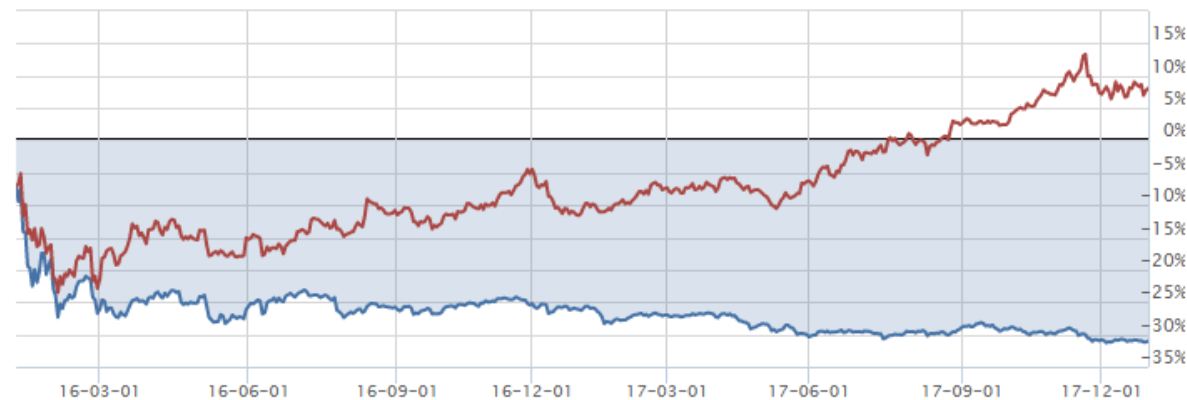
编辑策略 回测详情 编译运行列表 回测列表

已保存 编译运行 函数库

API 🔍 ⚙️ 🧩 >>

2016-01-01 至 2017-12-31 ¥ 10000000 每天 运行回测

策略收益	基准收益	Alpha	Beta	Sharpe	最大回撤
-31.03%	8.04%	-0.21	0.69	-1.44	31.29%



6月 2016 12月 2016 6月 2017 12月 2017

www.joinquant.com

日志 错误

```
2016-01-04 09:00:00 - INFO - Rebalance at 2016-01-04
2016-01-04 09:00:00 - INFO - [{'code': '002558.XSHE', 'profit_r': 4.1213270142180098}, {'code': '0005
20.XSHE', 'profit_r': 3.8102766798418974}, {'code': '600053.XSHG', 'profit_r': 3.468036529680365}, {'c
ode': '600734.XSHG', 'profit_r': 3.0805369127516782}, {'code': '000676.XSHE', 'profit_r': 3.0759803921
56862}, {'code': '300368.XSHE', 'profit_r': 2.2172433352240501}, {'code': '002667.XSHE', 'profit_r':
3.1429571429571429}, {'code': '1000538.XSHE', 'profit_r': 2.1305311543300100}, {'code': '1000468.XSHE'
```

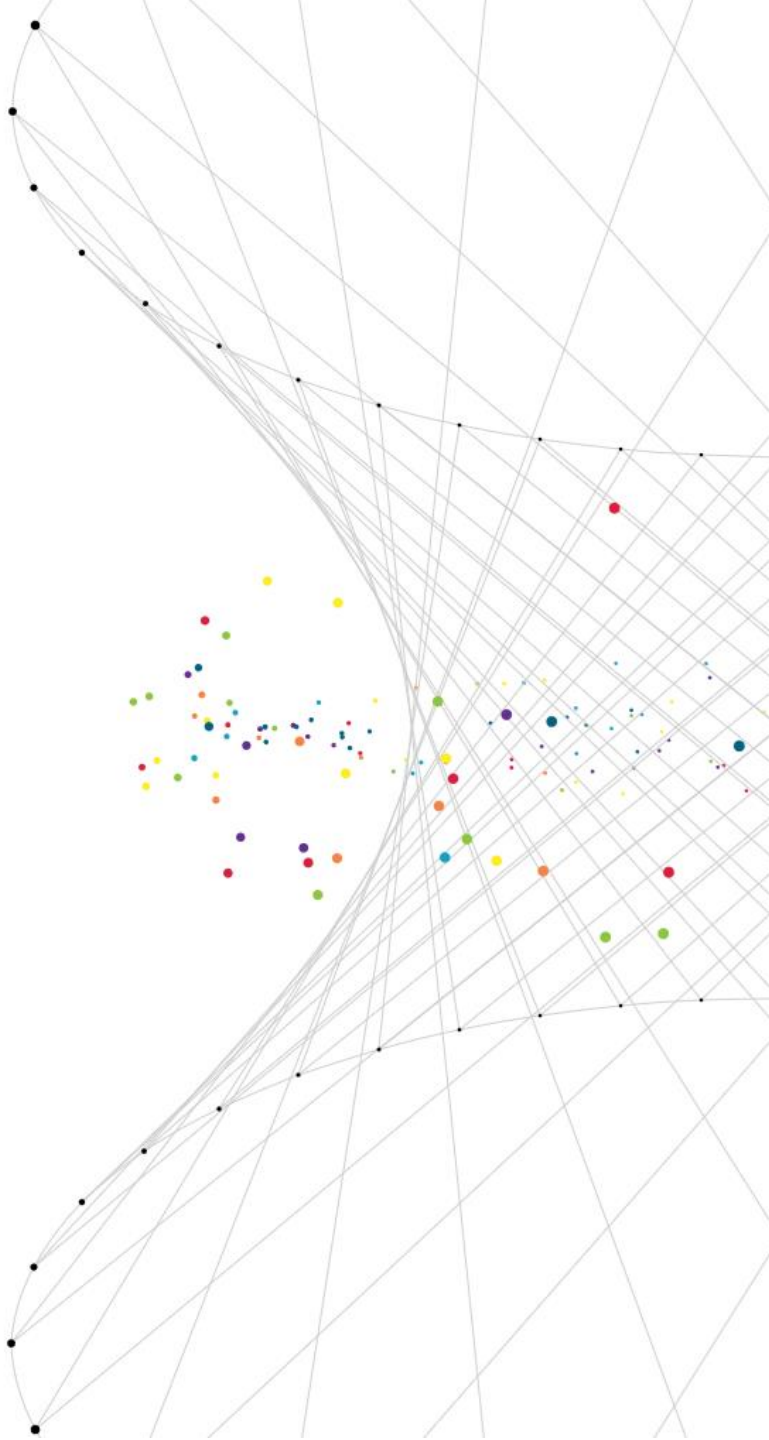
# 编程平台工作界面2





# 比较基准

纯多头方案



仅做多表现最差的

### 收益概述

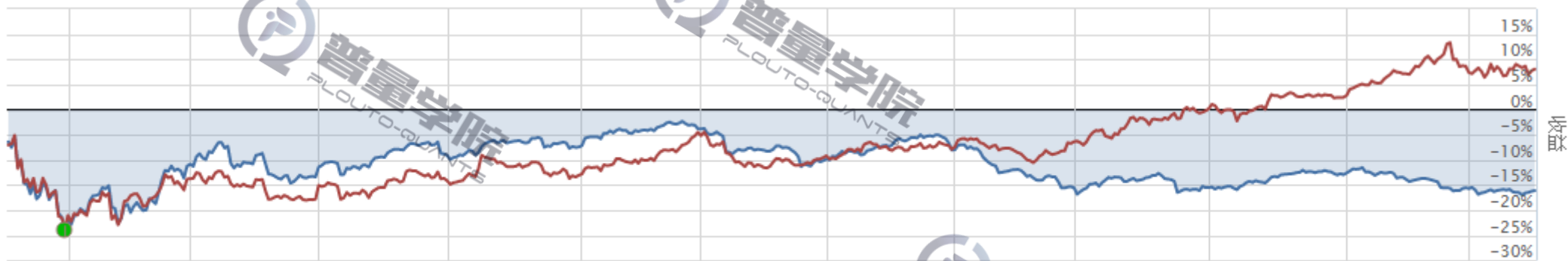
策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤
-16.10%	-8.60%	8.04%	-0.126	0.762	-0.786	0.484	0.731	23.883%

其他指标

缩放: 1个月 1年 全部

策略收益 基准收益 超额收益 普通轴 对数轴 超额收益

时间: 2016-01-04 - 2017-12-29

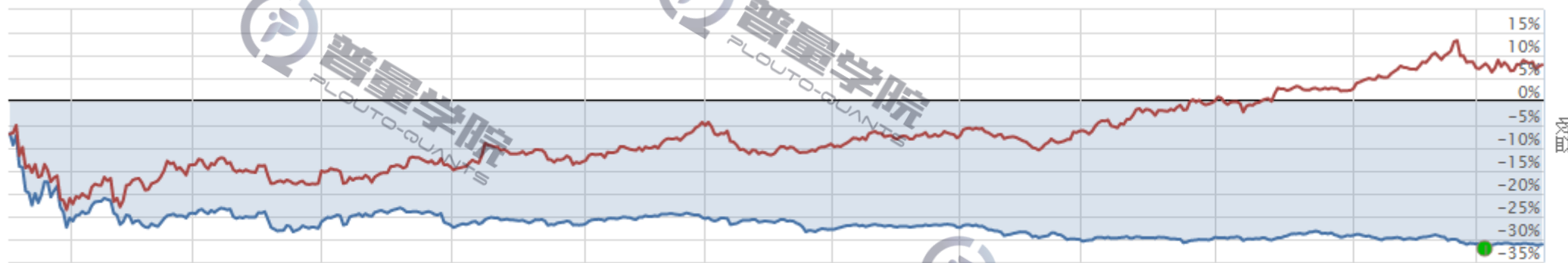


仅做多表现最好的 ●

### 收益概述

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤 <sup>?</sup>	其他指标 <sup>?</sup>
-31.03%	-17.33%	8.04%	-0.214	0.694	-1.438	0.408	0.443	31.290%	

缩放: 1个月 1年 全部  策略收益  基准收益  超额收益  普通轴  对数轴  超额收益 时间: 2016-01-04 - 2017-12-29

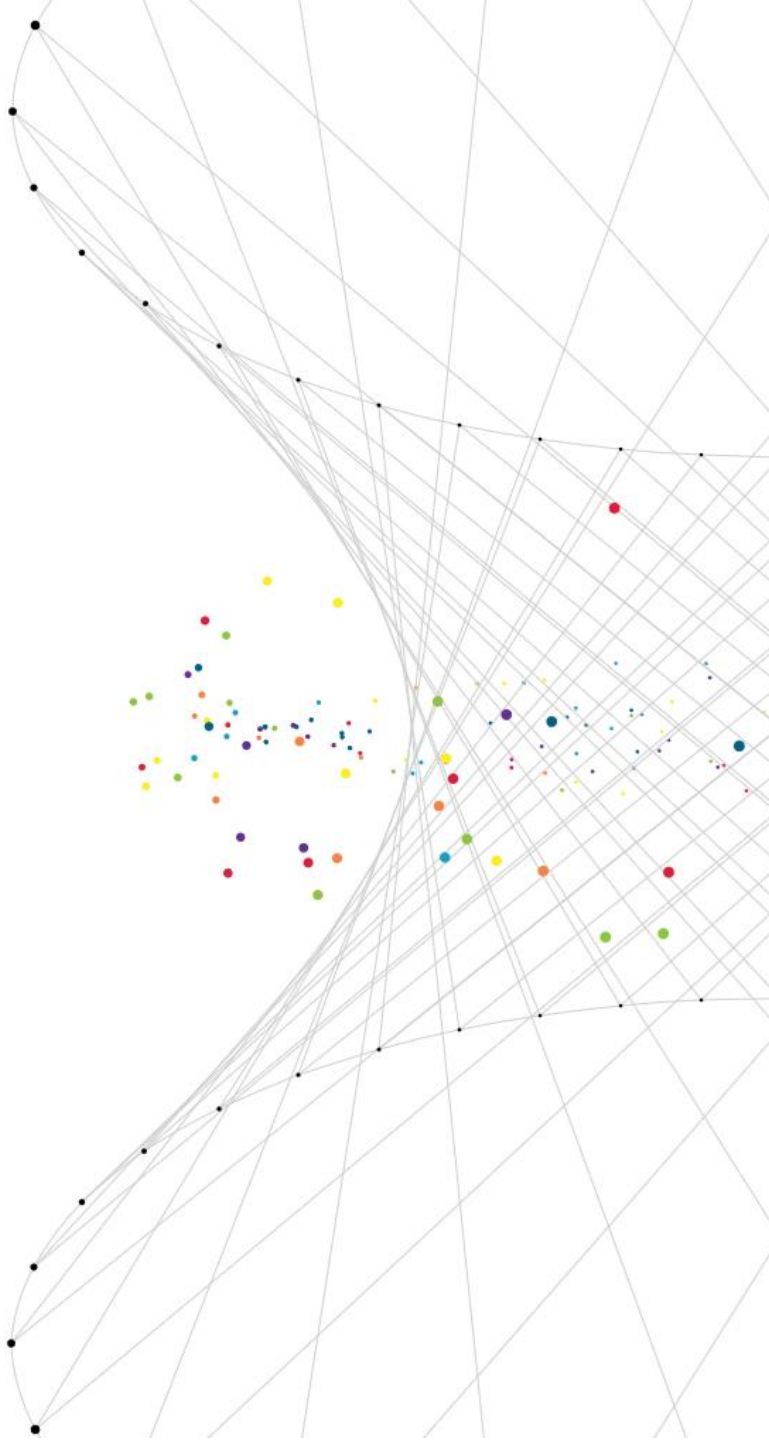






# 多空双向

构造均值回复型组合






## 初始化参数和变量

```
1 import jqdata
2
3 ADJUST_INTERVAL = 20 # 调仓周期
4 CHK_PF_INTERVAL = 60 # 观测窗口
5 PERCENT_TO_KEEP = 10 # 组合容量
6
7 # 初始化函数，回测开始时执行一次
8 def initialize(context):
9     set_benchmark('000300.XSHG') # 设定比较基准指数
10    set_option('use_real_price', True) # 默认复权模式
11
12    init_cash = context.portfolio.starting_cash # 启动资金
13
14    # 设定融资/融券账户
15    set_subportfolios([SubPortfolioConfig(cash=init_cash, type='stock_margin')])
16
17    # 设定交易成本
18    set_order_cost(OrderCost(open_tax=0, close_tax=0, open_commission=0,
19                             close_commission=0, close_today_commission=0, min_commission=0), type='stock')
19
20    # 设定融资/融券的利率和保证金比例
21    set_option('margincash_interest_rate', 0.0)
22    set_option('margincash_margin_rate', 1.0)
23    set_option('marginsec_interest_rate', 0.0)
24    set_option('marginsec_margin_rate', 1.0)
25
```

```
26 # 设置定时运行函数
27 run_daily(before_market_open, time='before_open', reference_security='000300.XSHG')

28 run_daily(market_open, time='open', reference_security='000300.XSHG')
29 run_daily(after_market_close, time='after_close', reference_security='000300.XSHG')
30
31 # 记录交易日期及其索引
32 g.all_tdays = jqdata.get_all_trade_days()
33 g.date_xlat = dict([(doc[1], doc[0]) for doc in enumerate(g.all_tdays)])
34
35 # 设定若干全局变量
36 g.elapsed_days = 0
37 g.rebalance_today = False
38 g.new_long_pos = {}
39 g.new_short_pos = {}
40
```

2018-05-07	2018-05-08	2018-05-09	2018-05-10	2018-05-11	2018-05-14	2018-05-15	2018-05-16	...
143	144	145	146	147	148	149	150	...



## 开盘前运行函数 •

```
41 # 在每个交易日开盘前运行
42 def before_market_open(context):
43     cur_dt = context.current_dt.date() # 当前日期
44     if g.elapsed_days % ADJUST_INTERVAL == 0: # 是否是调仓日
45         log.info("Rebalance at %s", cur_dt)
46         g.rebalance_today = True
47         idx = g.date_xlat[cur_dt]
48
49         # 观察窗口首末日期
50         tail_date = g.all_tdays[idx-1]
51         head_date = g.all_tdays[idx-1-CHK_PF_INTERVAL]
52
53         # 调仓日前一天的股票列表
54         candidates = list(get_all_securities(["stock"], tail_date).index)
55
56         # 观察窗口末尾日期的股票价格
57         tail_prices = get_price(candidates, tail_date, tail_date, frequency="1d",
58                                 fields=["close"])
59         g.tail_values = dict(tail_prices["close"].iloc[0])
60
61         # 观察窗口起始日期的股票价格
62         head_prices = get_price(candidates, head_date, head_date, frequency="1d",
63                                 fields=["close"])
64         g.head_values = dict(head_prices["close"].iloc[0])
```

```
64 # 在观察窗口内按收益率对股票排序（从小到大）
65 merged_list = []
66 for code, tail_value in g.tail_values.items():
67     if math.isnan(tail_value):
68         continue
69     if code in g.head_values:
70         head_value = g.head_values[code]
71         if math.isnan(head_value):
72             continue
73         profit_r = tail_value/head_value - 1.0
74         merged_list.append({
75             "code": code,
76             "profit_r": profit_r
77         })
78
79 # 从首尾分别取出一定比例的股票，构造新的多空投资组合
80 merged_list.sort(key=lambda x: x["profit_r"], reverse=False)
81 num_to_keep = len(merged_list) * PERCENT_TO_KEEP / 100 // 2
82 g.new_long_pos = {doc["code"] for doc in merged_list[0:num_to_keep]}
83 g.new_short_pos = {doc["code"] for doc in merged_list[-num_to_keep:]}
84
85 else:
86     g.rebalance_today = False
87
88 g.elapsed_days += 1
89
```

## 开盘时运行函数 •

```
90 # 在每个交易日开盘时运行
91 def market_open(context):
92     p = context.portfolio.subportfolios[0] # 融资/融券保证金账户
93     if g.rebalance_today:
94         # 再平衡步骤1: 平掉原有多空仓位
95         prev_long_pos = p.long_positions
96         prev_short_pos = p.short_positions
97
98         for code, pos in prev_long_pos.items():
99             margincash_close(code, pos.closeable_amount)
100
101         for code, pos in prev_short_pos.items():
102             marginsec_close(code, pos.closeable_amount)
103
104         # 再平衡步骤2: 开立新的多空仓位
105         each_long_cash = round(p.available_margin * 0.5 / len(g.new_long_pos), 2)
106         each_short_cash = round(p.available_margin * 0.5 / len(g.new_short_pos), 2)
107
108         for code in g.new_long_pos:
109             num_to_buy = each_long_cash / g.tail_values[code] // 100 * 100
110             margincash_open(code, num_to_buy)
111
112         for code in g.new_short_pos:
113             num_to_sellshort = each_short_cash / g.tail_values[code] // 100 * 100
114             marginsec_open(code, num_to_sellshort)
115
```



## 收盘后运行函数

```
116     else:
117         # 非调仓日, 可以增加止盈/止损等额外操作.....
118         pass
119
120 # 在每个交易日收盘后运行
121 def after_market_close(context):
122     # 查看融资融券账户相关相关信息(更多请见API-对象-SubPortfolio)
123     p = context.portfolio.subportfolios[0]
124     log.info('-----')
125     log.info('查看融资融券账户相关相关信息(更多请见API-对象-SubPortfolio): ')
126     log.info('总资产: ', p.total_value)
127     log.info('净资产: ', p.net_value)
128     log.info('总负债: ', p.total_liability)
129     log.info('融资负债: ', p.cash_liability)
130     log.info('融券负债: ', p.sec_liability)
131     log.info('利息总负债: ', p.interest)
132     log.info('可用保证金: ', p.available_margin)
133     log.info('维持担保比例: ', p.maintenance_margin_rate)
134     log.info('账户所属类型: ', p.type)
135     log.info('#####')
136
```



# 多空双向：震荡市 vs. HS300

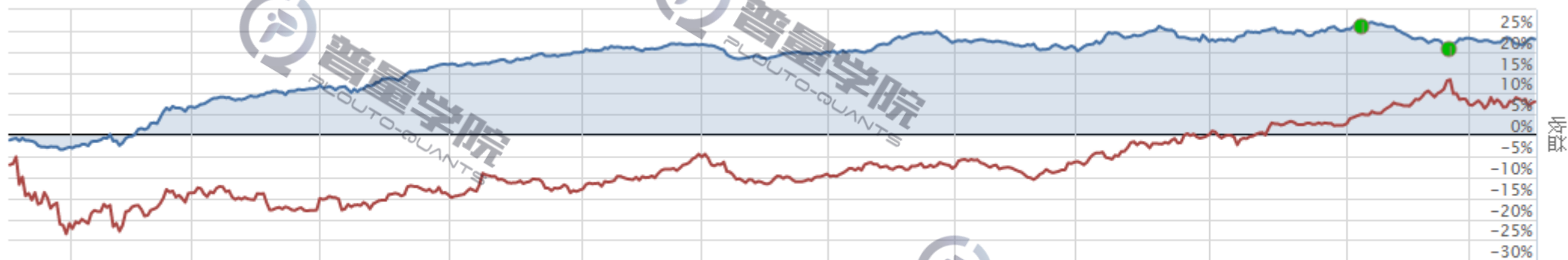
## 收益概述

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤	其他指标
<b>22.97%</b>	<b>11.17%</b>	<b>8.04%</b>	<b>0.072</b>	<b>0.109</b>	<b>1.253</b>	<b>0.547</b>	<b>1.259</b>	<b>4.558%</b>	

缩放：1个月 1年 全部

策略收益 基准收益 超额收益 普通轴 对数轴 超额收益

时间：2016-01-04 - 2017-12-29



# 多空双向：震荡市 vs. 上证指数

## 收益概述

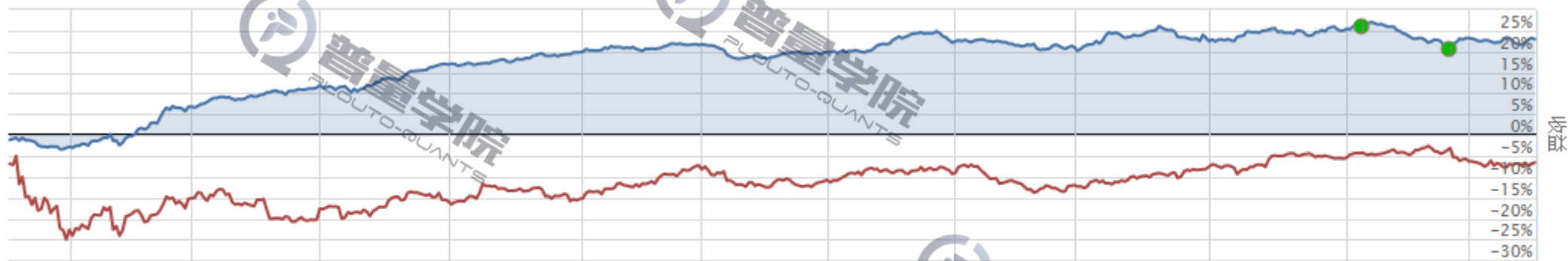
策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤	其他指标
22.97%	11.17%	-6.56%	0.081	0.123	1.253	0.547	1.259	4.558%	

缩放: 1个月 1年 全部

策略收益 基准收益 超额收益

普通轴 对数轴 超额收益

时间: 2016-01-04 - 2017-12-29



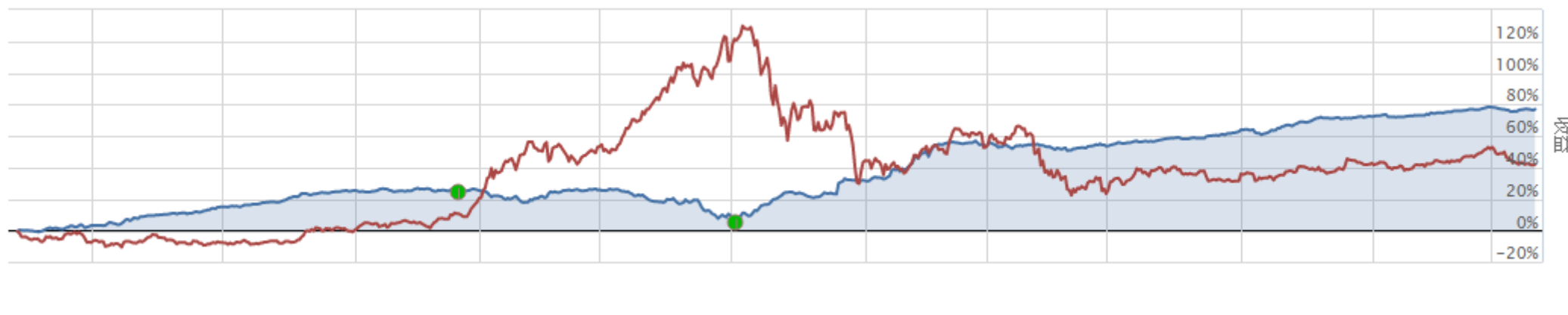
# 多空双向：牛熊市 vs. HS300

## 收益概述

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤	其他指标
76.98%	21.50%	42.06%	0.176	-0.014	1.970	0.576	1.407	15.405%	

缩放： 1个月 1年 全部  策略收益  基准收益  超额收益  普通轴  对数轴  超额收益

时间： 2013-12-30 - 2016-12-30



## 压力测试-考虑各种成本

- 各类交易费用
  - 交易所手续费
  - 券商佣金
- 保证金账户设置
  - 融资/融券利率
  - 保证金比率

## 保证金分配比例的计算:

	交易额	保证金
多头 (融资)	X	1 * X
空头 (融券)	X	1.5 * X

$$1 * X + 1.5 * X = 1.0 (100\%)$$

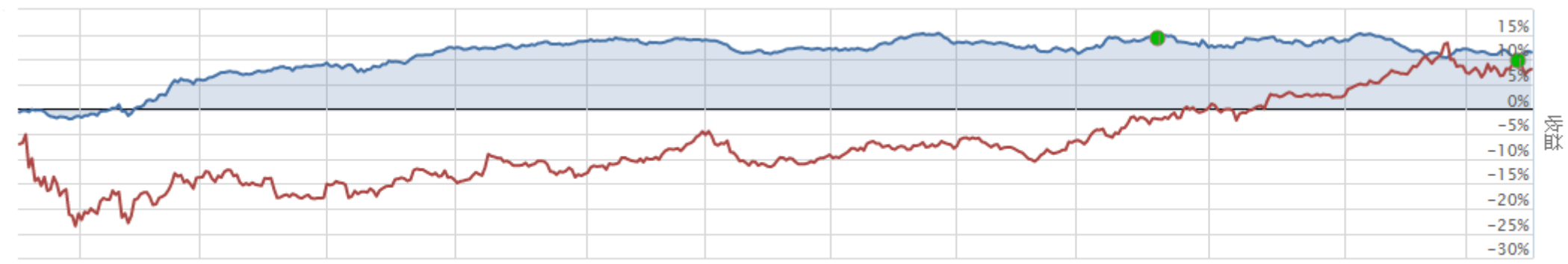
$$\text{多空资金分别占比: } X = 0.4 (40\%)$$




### 收益概述

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤 <sup>?</sup>	其他指标 <sup>?</sup>
<b>11.44%</b>	<b>5.71%</b>	<b>8.04%</b>	<b>0.017</b>	<b>0.062</b>	<b>0.351</b>	<b>0.549</b>	<b>1.275</b>	<b>4.352%</b>	

缩放: 1个月 1年 全部 ■ 策略收益 ■ 基准收益 ■ 超额收益 ● 普通轴 ● 对数轴 ■ 超额收益 时间: 2016-01-04 - 2017-12-29







# 改进思路

## PART THREE

---

## 头寸管理

- 分配资金方式：均仓（现有方式） vs. 按波动率
  - 波动率增大意味着风险增加，分配的仓位就小，反之亦然
- 平均真实波幅  $ATR = MA(TR, N)$ 
  - $TR = \text{当日的真实波动幅度} = \text{Max} (H-L, |H-PDC|, |PDC-L|)$
  - 其中：H=当日最高价，L=当日最低价，PDC=前一日收盘价
- 头寸规模单位 = 账户的%风险 / (ATR \* 每一最小交易单位)

## 计算公式

- 假设： $R\%$ 是每只股票承担的%风险， $C$ 是可用资金， $P_i$ 是每只股票的最新价格， $V_i$ 是波动率，假如全仓买入所有候选股票，则：

$$\sum_{i=1}^N \frac{C \times R\%}{V_i} \times P_i = C$$

$$R\% = \frac{1}{\sum_{i=1}^N \frac{P_i}{V_i}}$$

个股承担的%风险

$$K_i = \text{Int}\left(\frac{C \times R\%}{V_i \times 100}\right) \times 100$$

个股的头寸

## 等波动率分配仓位

```
78     if code in g.head_values:
79         head_value = g.head_values[code]
80         if math.isnan(head_value):
81             continue
82
83         # 计算相对收益率
84         profit_r = tail_value/head_value - 1.0
85
86         # 计算ATR指标
87         df = get_price(code, end_date=tail_date, count=ATR_WINDOW_SIZE+1, frequency
88                     = "1d", skip_paused=True, fields=["high", "low", "close"])
89         df["pdc"] = df["close"].shift(1)
90         tr = df.apply(lambda x: max(x["high"]-x["low"], abs(x["high"]-x["pdc"]),
91                                 abs(x["pdc"]-x["low"]))), axis=1)
92         atr = tr[-ATR_WINDOW_SIZE:].mean()
93         if math.isnan(atr):
94             continue
95
96         # 保存中间结果
97         merged_list.append({
98             "code": code,
99             "profit_r": profit_r,
100            "price": tail_value,
101            "atr": atr
102        })
```

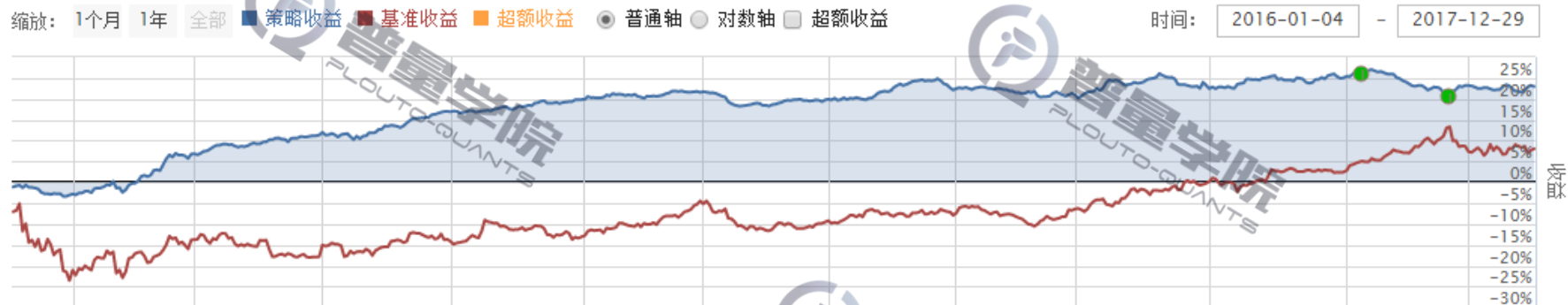
```
3 import functools
4 ATR_WINDOW_SIZE = 20 # 计算ATR的窗口宽度
```

```
102 # 从首尾分别取出一定比例的股票，构造新的多空投资组合
103 merged_list.sort(key=lambda x: x["profit_r"], reverse=False)
104 num_to_keep = len(merged_list) * PERCENT_TO_KEEP // 100 // 2
105 g.new_long_pos = {doc["code"]:doc for doc in merged_list[0:num_to_keep]}
106 g.new_short_pos = {doc["code"]:doc for doc in merged_list[-num_to_keep:]}
107
108 # 计算每个个股承担的风险百分比
109 sum = functools.reduce(lambda x, y: x + y["price"]/y["atr"], g.new_long_pos.values(
    ), 0)
110 sum += functools.reduce(lambda x, y: x + y["price"]/y["atr"], g.new_short_pos.
    values(), 0)
111 g.each_risk = 1 / sum
```

```
140 total_cash = p.available_margin
141
142 for code, doc in g.new_long_pos.items():
143     # num_to_buy = each_long_cash / g.tail_values[code] // 100 * 100
144     num_to_buy = total_cash * g.each_risk / doc["atr"] // 100 * 100
145     margincash_open(code, num_to_buy)
146
147 for code, doc in g.new_short_pos.items():
148     # num_to_sellshort = each_short_cash / g.tail_values[code] // 100 * 100
149     num_to_sellshort = total_cash * g.each_risk / doc["atr"] // 100 * 100
150     marginsec_open(code, num_to_sellshort)
```

### 收益概述

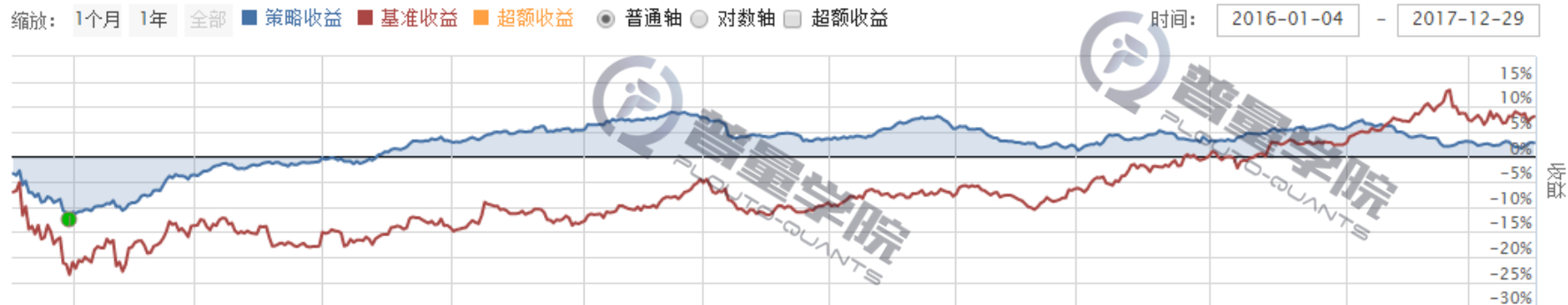
策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤 <sup>?</sup>	其他指标
<b>22.97%</b>	<b>11.17%</b>	<b>8.04%</b>	<b>0.072</b>	<b>0.109</b>	<b>1.253</b>	<b>0.547</b>	<b>1.259</b>	<b>4.558%</b>	



均仓 / 2016...

### 收益概述

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤 <sup>?</sup>	其他指标
<b>2.79%</b>	<b>1.42%</b>	<b>8.04%</b>	<b>-0.026</b>	<b>0.278</b>	<b>-0.358</b>	<b>0.545</b>	<b>1.040</b>	<b>11.997%</b>	



波动率 / 2016...



### 收益概述

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤 <sup>?</sup>	其他指标
<b>76.98%</b>	<b>21.50%</b>	<b>42.06%</b>	<b>0.176</b>	<b>-0.014</b>	<b>1.970</b>	<b>0.576</b>	<b>1.407</b>	<b>15.405%</b>	

缩放: 1个月 1年 全部 ■ 策略收益 ■ 基准收益 ■ 超额收益 ● 普通轴 ● 对数轴 ■ 超额收益

时间: 2013-12-30 - 2016-12-30



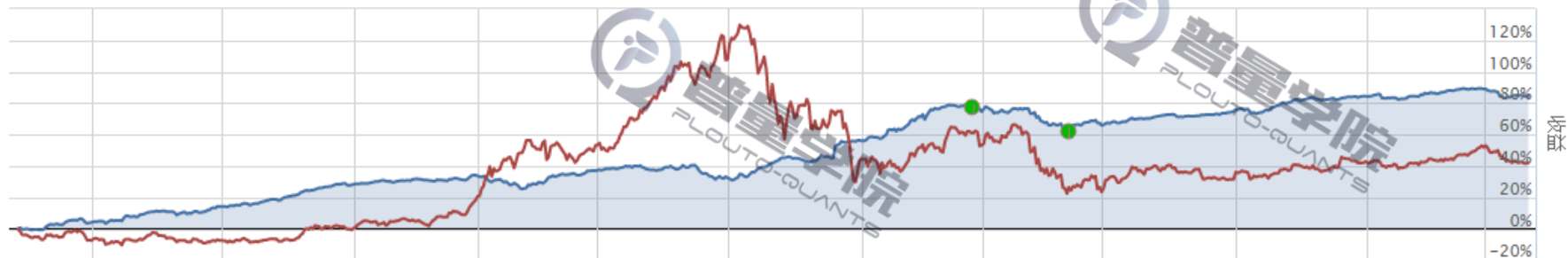
均仓 / 2014...

### 收益概述

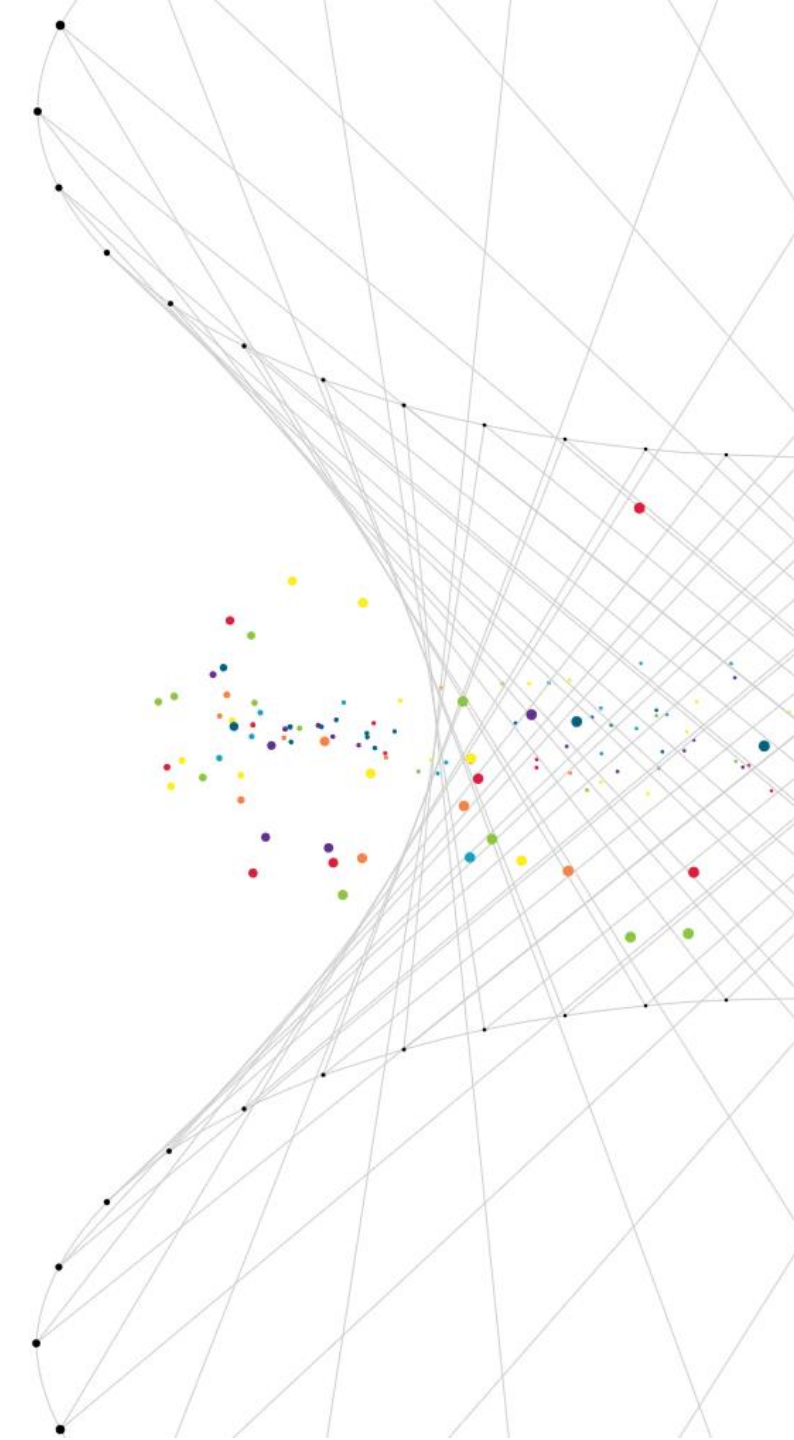
策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤 <sup>?</sup>	其他指标
<b>84.50%</b>	<b>23.23%</b>	<b>42.06%</b>	<b>0.187</b>	<b>0.065</b>	<b>2.336</b>	<b>0.576</b>	<b>1.468</b>	<b>8.614%</b>	

缩放: 1个月 1年 全部 ■ 策略收益 ■ 基准收益 ■ 超额收益 ● 普通轴 ● 对数轴 ■ 超额收益

时间: 2013-12-30 - 2016-12-30



波动率 / 2014...



# 纯多头的改进实验

基准, 不加成本:

均仓 / 2016...

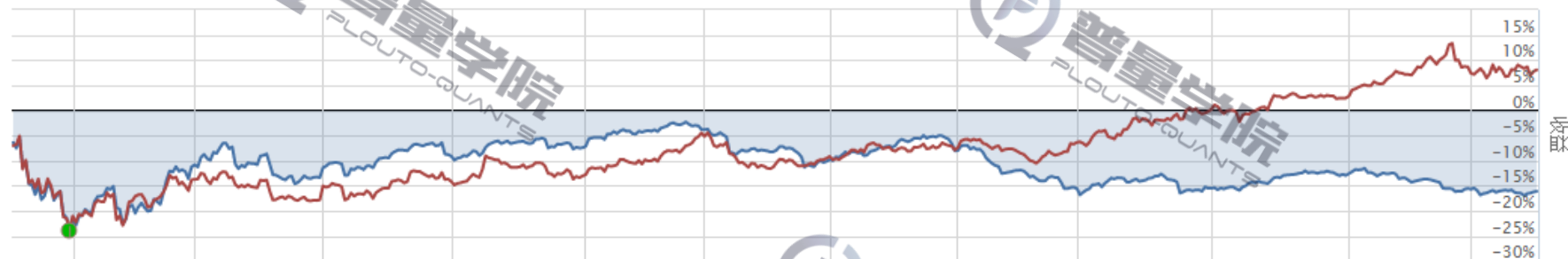
均仓 / 2014...

### 收益概述

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤 <sup>?</sup>	其他指标 <sup>&gt;</sup>
-16.10%	-8.60%	8.04%	-0.126	0.762	-0.786	0.484	0.731	23.883%	

缩放: 1个月 1年 全部  策略收益  基准收益  超额收益  普通轴  对数轴  超额收益

时间: 2016-01-04 - 2017-12-29



### 收益概述

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤 <sup>?</sup>	其他指标 <sup>&gt;</sup>
106.41%	28.04%	42.06%	0.196	0.510	1.305	0.603	1.729	27.186%	

缩放: 1个月 1年 全部  策略收益  基准收益  超额收益  普通轴  对数轴  超额收益

时间: 2013-12-30 - 2016-12-30



对比, 不加成本:

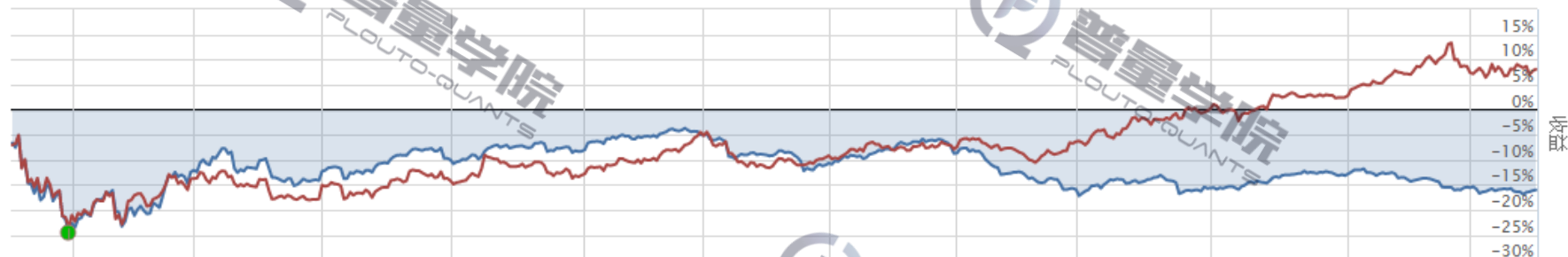
波动率 / 2016...

波动率 / 2014...

### 收益概述

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤
-16.00%	-8.54%	8.04%	-0.126	0.773	-0.781	0.487	0.724	24.568%

缩放: 1个月 1年 全部



### 收益概述

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤
109.95%	28.78%	42.06%	0.203	0.512	1.346	0.604	1.765	25.850%

缩放: 1个月 1年 全部



对比，加上成本：

波动率 / 2016...

波动率 / 2014...

### 收益概述

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤
-17.57%	-9.42%	8.04%	-0.135	0.772	-0.837	0.487	0.723	24.583%

其他指标 >

缩放: 1个月 1年 全部

策略收益

基准收益

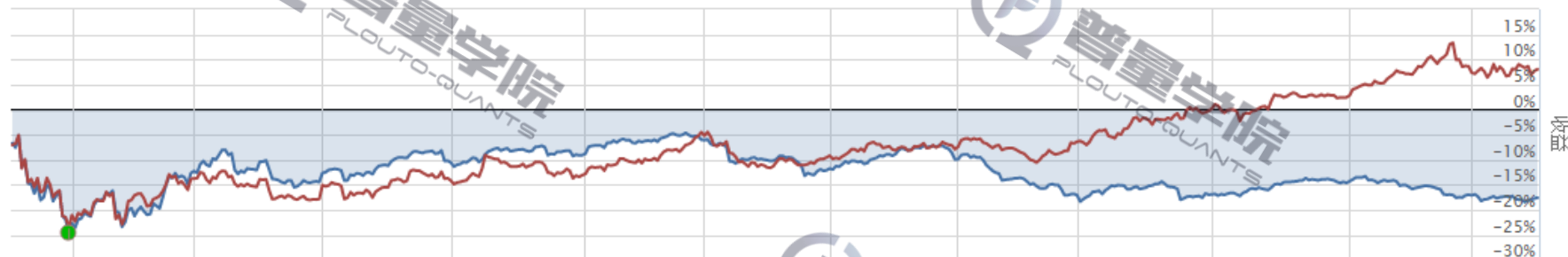
超额收益

普通轴

对数轴

超额收益

时间: 2016-01-04 - 2017-12-29



### 收益概述

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤
103.84%	27.49%	42.06%	0.190	0.511	1.277	0.604	1.774	25.963%

其他指标 >

缩放: 1个月 1年 全部

策略收益

基准收益

超额收益

普通轴

对数轴

超额收益

时间: 2013-12-30 - 2016-12-30





# — 答疑热线 —

牛小秘

微信：niuxiaomi1

