

# 追涨杀跌财富自由短线王基金量化策略研究报告

## 一、前言

鉴于本小组成员大多数是在 A 股中已被收割或正被收割的韭菜，遂想利用本次机会，团结组员之力，构建追涨杀跌财富自由之短线王策略来证明在 A 股市场中采用追涨杀跌方式仍然是能够赚钱的，本策略便由此诞生。

## 二、赛道研究方法及赛道选择

### （一）自回归积分滑动平均模型（ARIMA）

ARIMA 模型全称为自回归积分滑动平均模型，是指将非平稳时间序列转化为平稳时间序列，然后将因变量仅对它的滞后值以及随机误差项的现值和滞后值进行回归所建立的模型。可以被使用于时间序列预测，如海关税收预测、GDP 走势预测等等。在此我们使用 1-7 月的行业数据代入 ARIMA 模型，用以预测未来 3 个月的收益率曲线。

对从 wind 上抓取的行业指数数据进行 ARIMA 建模以及预测。首先观察行业数据的自回归项的阶数（d 值）、使序列平稳的最小差分阶数（p 值）、和滑动平均项的阶数（q 值），得到结果如下（图 1），代码见附录二、三。

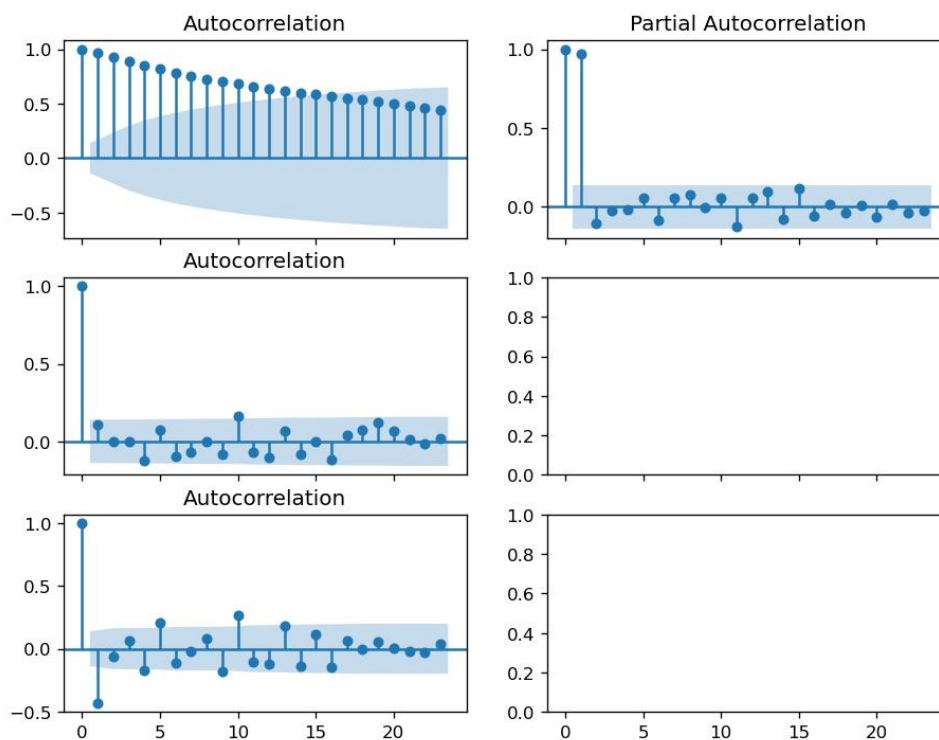


图-1 ARIMA 模型结果

将得到参数再代入 ARIMA 模型内，算出该行业回测期间预测收益率，得到行业排名。行业预期收益增长如下。

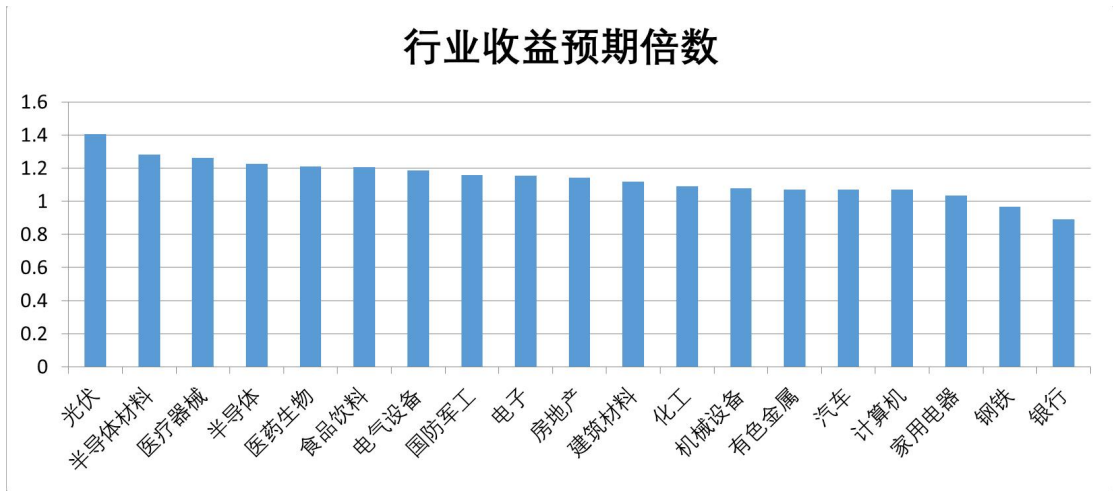


图-2 行业预期收益

### （二）相对旋转图（RRG, Relative Rotation Graphs）

相对旋转图（Relative Rotation Graphs），通常称为 RRG，是一种独特的可视化相对强度分析工具。在 RRG 中，我们可以同时分析多种证券相对于同一基准的相对强弱趋势。此工具的真正强大之处在于旋转。我们都听说过行业和资产类别的轮换，但很难在线性图表上看到这种“轮换”的过程，而 RRG 可以清晰的展示这一过程。

在建立相对旋转图之前，我们要先建立两个指标，JdK RS-Ratio（相对比率）和 JdK RS-Momentum（相对动量），分别代表相对强度和基于相对强度的动量。

#### 1. 相对比率 RS-Ratio

相对比率（JdK RS-Ratio）是衡量相对强度趋势的一个指标。RS-Ratio 使用比率来比较一种证券与另一种证券（通常是基准）的走势强弱。这样就可以确定股票是处于领先还是落后。善用相对强度可以提前发现股票与大盘的走势差异，在大盘下跌期间找到良好上升趋势的股票（用于做多）或在大盘上涨期间找到疲软的股票。（用于做空）

$$\text{相对比率} = \frac{\text{基本证券价格}}{\text{基准证券价格（证券组合）}}$$

#### 2. 相对动量 JdK RS-Momentum

RS-Momentum 是衡量 RS-Ratio 变化率的指标。作为一种动量指标，它引导着 RS-Ratio，可以用来预测 RS-Ratio 的转向。通常，当 RS-Ratio 形成低谷并

开始上升时，RS-Momentum 超过 100。相反，当 RS-Ratio 形成峰值并开始下降时，RS-Momentum 就会跌破 100。

### 3. 绘制相对旋转图

相对旋转图类似于散点图，有一条 X 轴和一条 Y 轴。X 轴表示 RS-Ratio，Y 轴表示 RS-Momentum。注意，这里的原点不是 (0,0)，而是 (100,100)。X 轴和 Y 轴将整个图形分成了四个象限。

在某一时刻，每个股票的 RS-Ratio 和 RS-Momentum 在相对旋转图上形成了一个点，如果我们要分析 9 个标普行业指数，那么我们把标普 500 设置为基准，此时我们就可以在相对旋转图上看到了 9 个点，每个点都代表了该行业的 RS-Ratio 和 RS-Momentum。

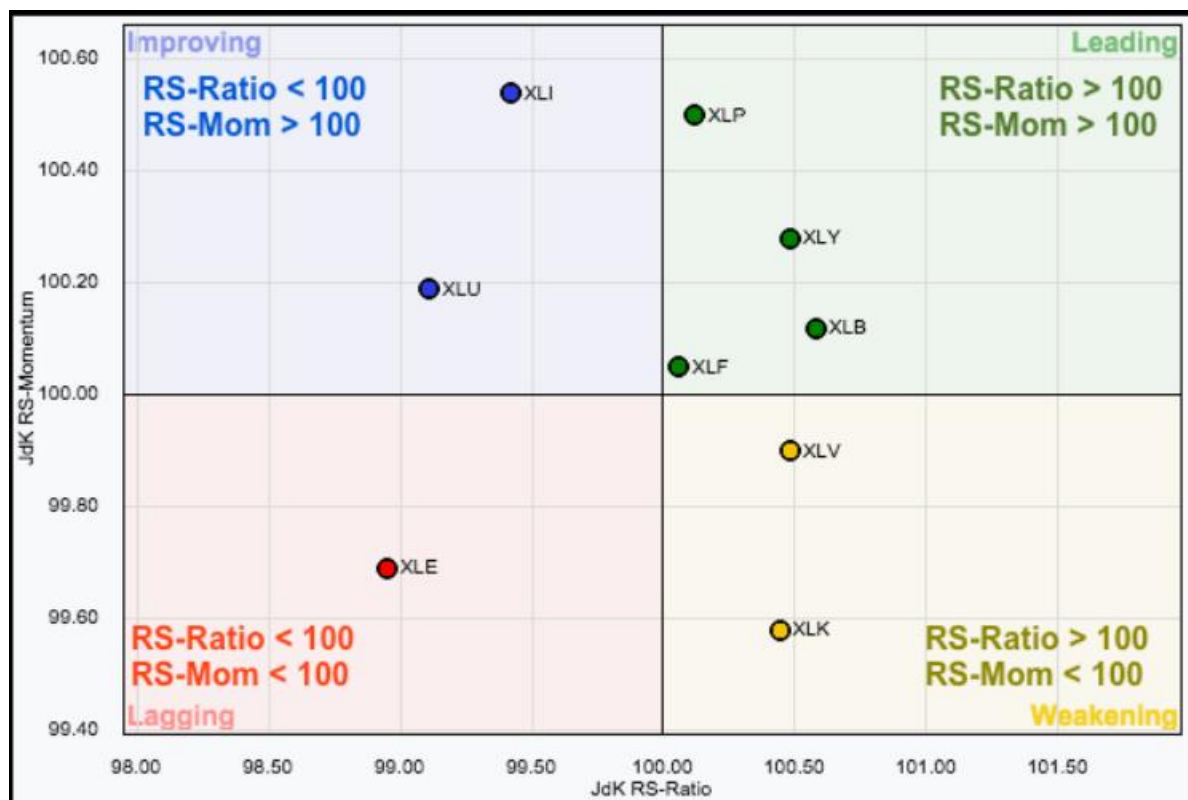


图-3 RRG 四象限 (图源)

RS-Ratio 和 RS-Momentum 的不同数值有四种组合。分别对应相对旋转图的四个象限。

①当 RS-Ratio 和 RS-Momentum 大于 100 时 (+/+), 该行业处于领先象限 (绿色)。大于 100 的 RS-Ratio 表明了相对强度处于上升趋势 (个股相对基准越来越强), 大于 100 的动量意味着这一趋势仍在向更高的方向推进。

②当 RS-Ratio 高于 100, RS-Momentum 小于 100 时 (+/-), 该行业处于停滞

象限(黄色)。大于 100 的 RS-Ratio 表明了相对性能的处于上升趋势，但小于 100 的动量意味着这种上升趋势正在停滞或失去动力。

③当 RS-Ratio 和 RS-Momentum 均小于 100 时 (-/-)，该行业处于落后象限(红色)。小于 100 的 RS-Ratio 表明了相对强度处于下降趋势（个股相对基准越来越若），而小于 100 的动量意味着这种下降趋势仍然在向下加强。

④当 RS-Ratio 低于 100，RS-Momentum 大于 100 时 (-/+），该行业处于改善象限（蓝色）。小于 100 的 RS-Ratio 表明了相对强度处于下降趋势，大于 100 的动量意味着这种下降趋势正在有改善或逆转的可能。

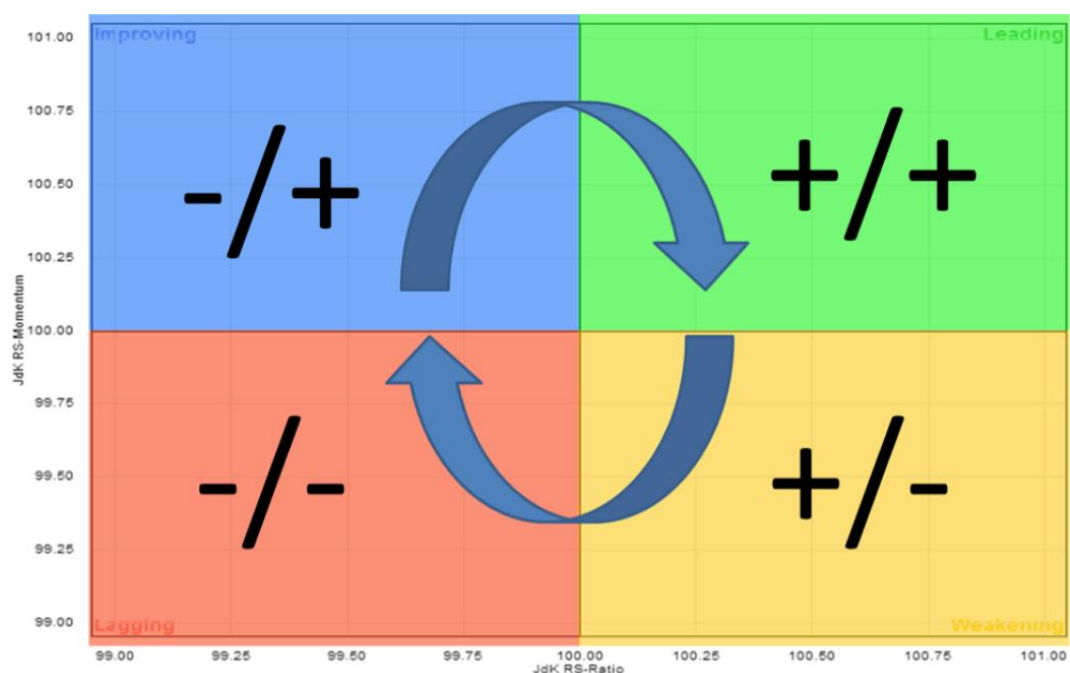


图-4 RRG 旋转顺序

当把点依据时间连起来时，历史数据不再是某一时刻孤单的点，能够呈现改股票的历史轨迹。如图-4 显示了九个行业 ETF 的 12 周历史轨迹：每条线不断的从一个象限转到另一个象限，带代表其处于不同的行情状态。例如，从最上面开始看，XLF 从改善象限进入了领先象限。在底部，XLU 由加速变弱开始缓慢变弱。蓝线表示 XLP 从停止到落后，然后从落后到改善。改善象限意味着 RS-Momentum 增加到 100 以上，但是 RS-Ratio 保持在 100 以下。当 RS-Ratio 和 RS-Momentum 都在 100 以上时，XLY 进入领先象限。

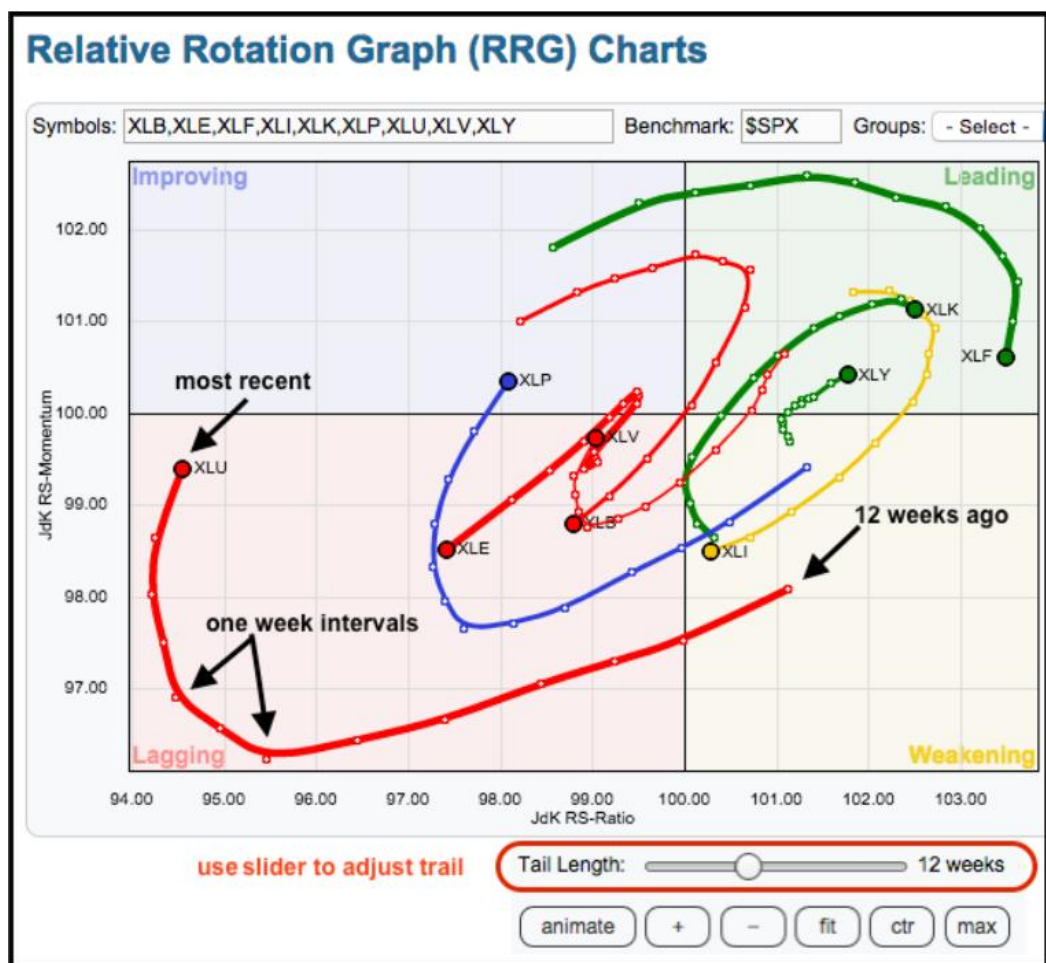


图-5 旋转轨迹

我们还可以观察同一条线中的两点间隔。上图中，所有的线都拥有 12 个点，代表他们过去 12 周的走势，很明显，有的线条很长，红色的 XLU 是轨迹最长的，他有着最大的移动范围，意味着他巨大的波动。右上角的绿色 XLY 线是最短的，这意味着它的移动最小，并且最稳定。

#### 4. RRG 的实现

详见附件 *rrg-月.xlsx*。

#### 5. RRG 实证结果

基于 ARIMA 模型挑选出来的结果，对预测收益率前六名的行业版块进行 RRG 分析，基准选取为沪深 300 组合，行业指数数据来源于 Wind。结果如下图（图 6-图 11）所示。

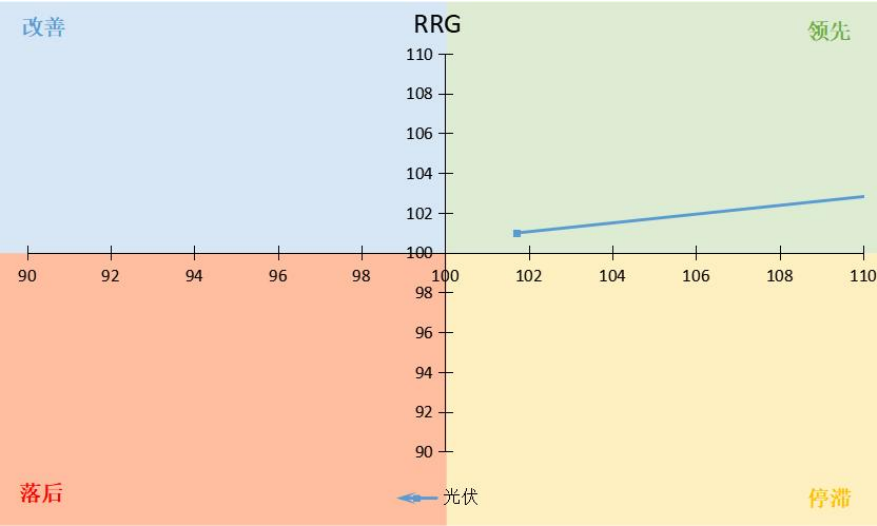


图-6 光伏行业 RRG 图

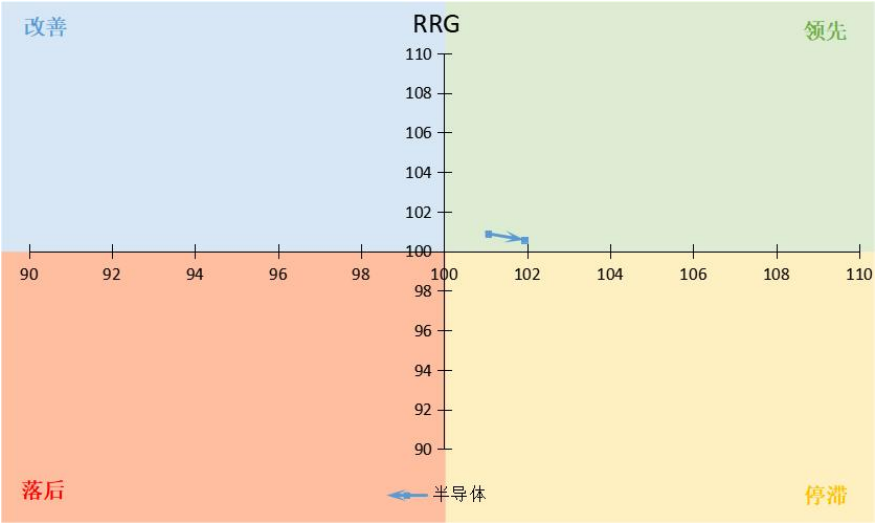


图-7 半导体行业 RRG 图

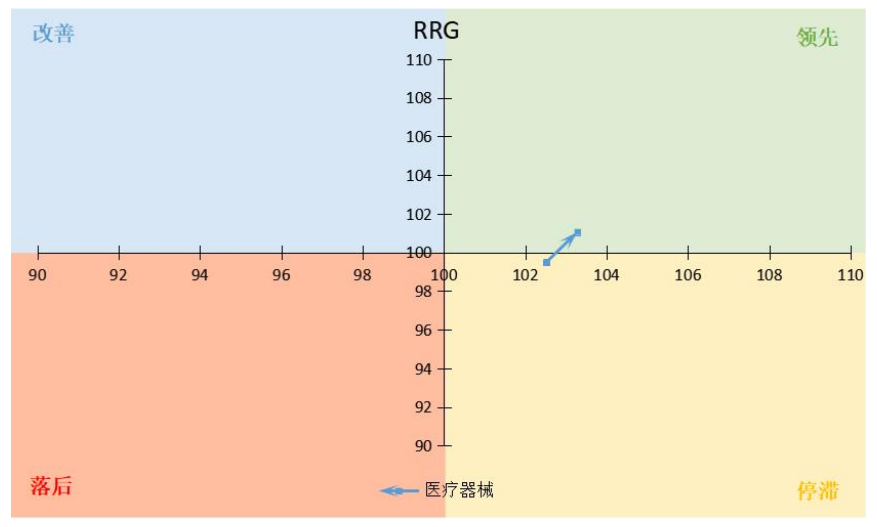


图-8 医疗器械行业 RRG 图

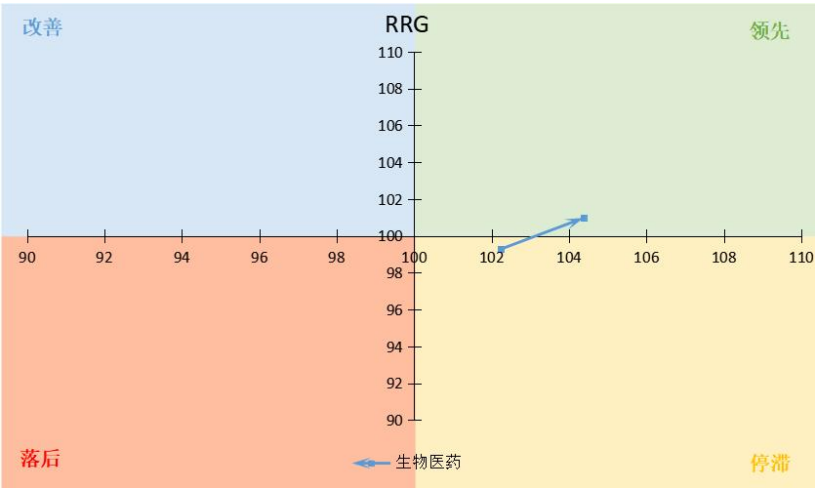


图-9 生物医药行业 RRG 图

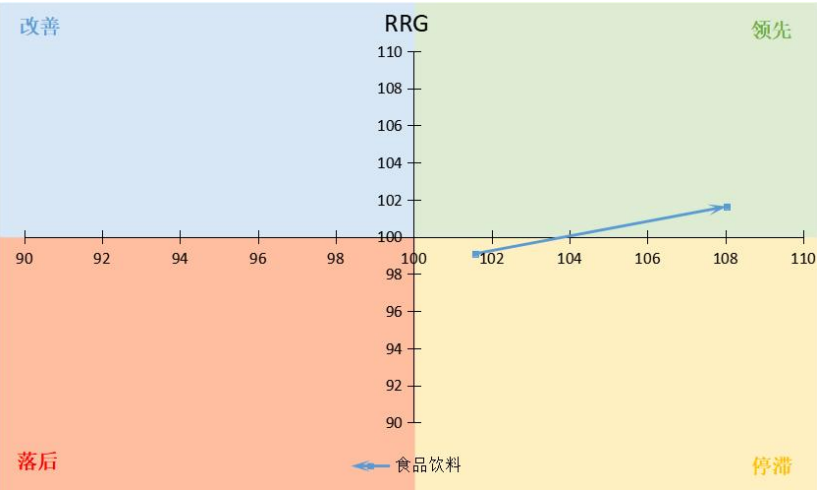


图-10 食品饮料行业 RRG 图

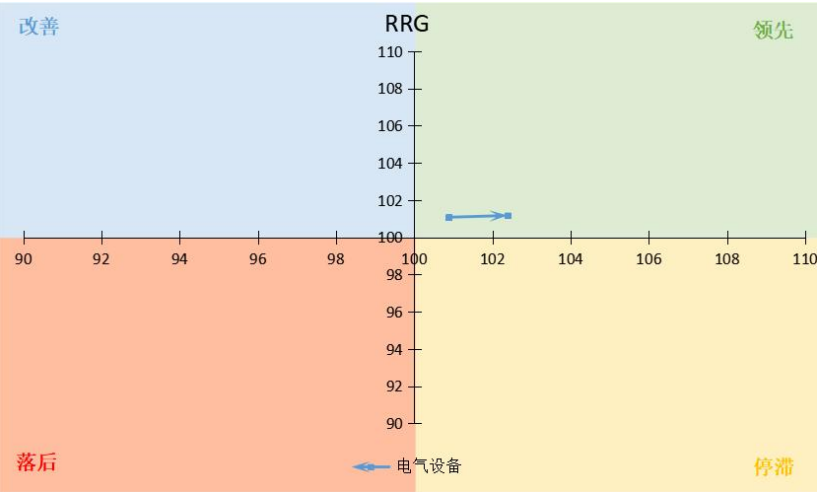


图-11 电气设备行业 RRG 图



由上图看出，所有行业最近一个数据均位于领先象限，除半导体以外的行业均处于持续领先状态，而半导体则有从领先向停滞状态转变的趋势。单个行业来看，光伏行业走势明显强于沪深 300 走势，是所有行业中领先程度最大的行业；医疗器械、生物医药、食品饮料刚刚从停滞象限转向领先象限，其中食品饮料行业线段长度最长，波动最大；而电气设备停留于领先象限，暂时没有向停滞象限转向的趋势。

### （三）结论

结合 ARIMA 模型与 RRG 模型，确定本策略的赛道选择为光伏（857333.SI）、食品饮料（801120.SI）、电气设备（801730.SI）。

## 三、策略优化

首先，我们先获取了所有的申万行业代码，然后仅改变行业（图），选择有利赛道，而不改变策略其他代码，查看未改进前策略收益率，如下图所示（图）。在初始多因子交易策略下，策略最主要表现为  $\beta$  值低，波动小。原始策略开平仓条件较为粗糙，导致策略收益不高，但是选取了走势较为强势的赛道，即使在原策略下，策略收益率也能够大于 0。故策略优化核心思路为提高策略选股能力，提高收益率。

```
# 获取申万行业，光伏设备
df_tscodes = pd.DataFrame(columns = ['in_date', 'out_date', 'index_code', 'con_code'])
df_tscodes1 = yq.concept.sw_detail(index_code= '801120.SI')#食品饮料
df_tscodes2 = yq.concept.sw_detail(index_code= '857333.SI')#光伏
df_tscodes3 = yq.concept.sw_detail(index_code= '801730.SI')#电气
df_tscodes = pd.concat([df_tscodes1,df_tscodes2,df_tscodes3])
ts_codes = df_tscodes['con_code']
```

图-12 赛道选择



图-13 原始多因子交易策略运行结果



### （一）整体思路

整体思路：选取有大幅上涨潜力的股票，捕捉短期资产增值机会。虽然多因子策略本质上为长期策略，但是三个月的回测周期不利于长期策略效果的展示，故团队决定将多因子策略改造为短期策略：利用因子选取短期内可能出现大幅上涨的股票，并结合择时策略，实现短期收益。

团队的策略基于趋势突破的思路开发，价格突破前一日高点是必要的，其次我们发现高成交量的K线对后市的走势有锚定作用——即支撑位，如果价格能保持在当前高量柱的最低价上方，是安全的，出现信号可以进行买入开仓，而价格跌破了当前高量柱的最低点，是危险的信号，如果有持仓则需要将股票清仓离场，而这个有锚定作用的高量柱随着市场更新——如果后面的K线最高价高于该K线的最高价或前者的成交量高于锚定K线并且价格在锚定K线的范围内，则更新锚定K线为前述的K线，所有操作都需要通过当前价是否在锚定高量柱最低价之上判断当前股票是安全还是危险。



图-14 趋势突破图解①



图-15 趋势突破图解②

## （二）确定回测频率

团队曾经尝试分钟策略，但未能实现。由于趋势策略为短期策略，选用分钟较于选用日频率更能够实现其效果。故在最开始，团队决定采取分钟策略，然而在编写完代码后，团队发现数据库并不能完全支持分钟策略（图-16），故放弃，转回日策略。此次尝试给团队带来了不少编程上、团队合作方面的经验，团队也并不是一无所获。

```
ValueError: 未在 2020-09-01 和 2020-11-01 区间内查询到数据，请检查并升级您的 data bundle 或者选择其他回测区间。
[2021-12-23 21:37:38.457387] ERROR: user_system_log: 未在 2020-09-01 和 2020-11-01 区间内查询到数据，请检查并升级您的 data bundle 或者选择其他回测区间。
```

图-16 数据库不支持分钟策略

## （三）数据提取及异常值处理

异常值处理。为了使 IC 检验的结果更有说服力，即为了使因子与收益率之间线性关系更为明显，在取数据时，对因子值进行了处理。基于中心极限定理的思想，认为因子值大于  $\mu \pm 3\sigma$  的值对模型解释力度下降。故剔除因子值不在  $(\mu - 3\sigma, \mu + 3\sigma)$  区间的因子值。异常值部分代码如图-17 所示。在保证其他条件不变情况下，表-1 展示了光伏、电气、饮料三个行业股票池针对 1-7 月换手率（turnover\_rate）、总市值（total\_mv）、净利润增长率（netprofit\_margin）、经营现金流增长率（op\_yoy）、净利润（net\_profit）在异常值处理前后 IC 值结果，结果表明异常值处理能够显著提升线性关系，有助于因子的选择。

```
#异常值处理
for i in factors_df.columns:
    temp_var = np.var(factors_df[i])**0.5
    temp_mean = factors_df[i].mean()
    factors_df = factors_df[factors_df[i] < temp_mean + 3 * temp_var]
    factors_df = factors_df[factors_df[i] > temp_mean - 3 * temp_var]
```

图-17 异常值处理

表-1 异常值处理前后 IC 值

|                              | 保留异常值       | 删去异常值       | 删后对比 |
|------------------------------|-------------|-------------|------|
| 换手率（turnover_rate）           | 0.108931678 | 0.215101963 | 增加   |
| 总市值（total_mv）                | 0.228039692 | 0.26018947  | 增加   |
| 净利润增长率<br>（netprofit_margin） | 0.041160317 | 0.071399133 | 增加   |
| 经营现金流增长率（op_yoy）             | 0.056599567 | 0.027780319 | 增加   |
| 净利润（net_profit）              | 0.192973016 | 0.041902189 | 增加   |

## （四）基于 IR 的因子打分法以及与因子排名的比较

首先计算单因子与当期收益率的相关系数，即 IC 值，再剔除 p 值较大的因

子，一定程度上可以等同为在不同股票上 IC 值正负号不一致。因此在代码操作时剔除该类因子。

```
for key,value in a.items():  
    if len([i for i in value if i > 0])==0 or len([i for i in value if i < 0])==0:  
        b[key]= a[key]
```

图-18 剔除正负号不同的股票

再将 IC 均值除以其标准差，得到 IR 值，作为各因子的权重，以各股票的因子值加权作为股票分数，从高到低排名，取前几只股票到我们的股票池中。

相较于原来的因子排序法，因子打分考虑了各因子的权重，给予不同因子在选择中不同的占比因素。权重越高，就越会考虑该因子，比因子排序法更加精细。

而且当所取因子越多时，因子排名的并集就会越小，可能出现取得股票过少的窘境。而因子打分法完全可以精确控制所取股票的数量，在这方面更加便捷。

## 四、因子研究

### （一）因子选取

根据相关文献及各券商因子选股报告，我们列举出了 40 余个对有助于选股的因子，并分别在所选择的行业中进行 IC 值检验，所输出的数据有：2020 年 1 月至 7 月各因子在各行业中的 IC 值与 IC 值平均值、IC 值波动率、t 统计量值与 P 值。

我们将原始策略中 find\_order\_stock 函数重命名为 sub\_find\_order\_stock 并进行一系列修改，设为子函数。母函数为 find\_order\_stock 针对子函数进行循环调用。由于不同行业有不同特征，因此其在行业之间，因子对行业收益率的解释力度也不尽相同。分行业研究因子，能使多因子研究结果更为有解释力度。修改的代码如下图所示（图-19）

```
def find_order_stocks(context,bar_dict):
    start_time = (context.now - datetime.timedelta(days=context.observation)).strftime("%Y%m%d")
    start_time_long = (context.now - datetime.timedelta(days=context.observation + 330)).strftime("%Y%m%d") # 观测期为1年，保证数据完整
    end_time_long = (context.now - datetime.timedelta(days=context.observation + 300)).strftime("%Y%m%d")
    end_time = context.now.strftime("%Y%m%d")
    order_stocks=[]
    context.yq_order_stocks = []

    #选定行业
    industry=['857333.SI','801120.SI','801730.SI']

    #分别对行业修改因子
    sector1 = {'factors1':['total_mv','turnover_rate','pb'],
              'factors2':['profit_to_gr','op_yoy','roe_yoy','netprofit_margin'],'factors3':[],'factors4':[],'factors5':[]}
    sector2 = {'factors1':['total_mv','pe','dv_ratio'],
              'factors2':['op_yoy','roe_yoy','assets_turn'],'factors3':[],'factors4':[],'factors5':[]}
    sector3 = {'factors1':['total_mv','turnover_rate','pb'],
              'factors2':['profit_to_gr','roa','op_yoy','netprofit_margin','ocf_to_debt'],'factors3':[],'factors4':[],'factors5':[]}
    sector_dict={sector1,sector2,sector3}

    #各行业所取股票数
    weight=[6,2,2]

    #for循环单个行业取股票池
    for i in range(3):
        order_stocks=[]
        order_stocks+=sub_find_order_stocks(sector_dict[i],weight[i],industry[i],start_time,start_time_long,end_time,end_time_long)
        context.yq_order_stocks = context.yq_order_stocks + [stock.replace('SH','XSHG').replace('SZ','XSHE') for stock in order_stocks]

def sub_find_order_stocks(factor_d,weight,industry,start_time,start_time_long,end_time,end_time_long):
    global IC
    factors_dict = {}
```

图-19 因子 IC 值检验代码

所选择的 40 余种因子有：

表-2 因子库

| 因子名称          | 因子代码             |
|---------------|------------------|
| 换手率           | turnover_rate    |
| 总市值           | total_mv         |
| 发行股           | float_share      |
| 自由流通股         | free_share       |
| 市净率           | pb               |
| 分红率           | dv_ratio         |
| 市盈率           | pe               |
| 每股收益          | eps              |
| 权益增长率         | equity_yoy       |
| 每股净资产         | roe              |
| 产权比率          | debt_to_eqt      |
| 权益乘数          | assets_to_eqt    |
| 加权平均资产收益率     | roe_waa          |
| 扣除非经常性损益后的净利润 | profit_dedt      |
| 净利润/营业总收入     | profit_to_gr     |
| 总资产报酬率        | roa              |
| 营业利润同比增长率(%)  | op_yoy           |
| 权益乘数(杜邦分析)    | dp_assets_to_eqt |

|                     |                   |
|---------------------|-------------------|
| 资产负债率               | debt_to_assets    |
| 流动比率                | current_ratio     |
| 销售费用/营业总收入          | saleexp_to_gr     |
| 非营业利润               | cogs_of_sales     |
| 净资产收益率(摊薄)同比增长率(%)  | roe_yoy           |
| 每股净资产               | bps               |
| 息税前利润               | ebit              |
| 每股净资产相对年初增长率(%)     | bps_yoy           |
| 销售净利率               | netprofit_margin  |
| 经营活动产生的现金流量净额/流动负债  | ocf_to_shortdebt  |
| 每股营业总收入             | total_revenue_ps  |
| 留存收益                | retained_earnings |
| 速动比率                | quick_ratio       |
| 每股经营活动产生的现金流量净额     | ocfps             |
| 经营活动产生的现金流量净额/负债合计  | ocf_to_debt       |
| 股权自由现金流量            | fcfe              |
| 每股经营活动产生的现金流量净额同比增长 | cfps_yoy          |
| 流动资产/总资产            | ca_to_assets      |
| 每股企业自由现金流量          | fcff_ps           |
| 总资产周转率              | assets_turn       |
| 每股资本公积              | capital_rese_ps   |
| 应收账款周转率             | ar_turn           |
| 净利润                 | net_profit        |

将上述因子代入各行业进行 IC 值检验，首先是食品饮料行业：

表-3 食品饮料因子检验

| 因子名称  | IC 值平均值 | IC 值标准差 | P 值     |
|-------|---------|---------|---------|
| 换手率   | 0.0206  | 0.03381 | 0.7926  |
| 总市值   | 0.1562  | 0.02851 | 0.06398 |
| 发行股   | -0.0282 | 0.01927 | 0.63563 |
| 自由流通股 | -0.0298 | 0.03429 | 0.7068  |
| 市净率   | 0.3347  | 0.05745 | 0.01413 |

|                        |          |          |         |
|------------------------|----------|----------|---------|
| 分红率                    | -0.223   | 0.04118  | 0.03579 |
| 市盈率                    | 0.4257   | 0.0425   | 0.00231 |
| 每股收益                   | 0.0681   | 0.0189   | 0.27139 |
| 权益增长率                  | 0.1098   | 0.06270  | 0.3237  |
| 每股净资产                  | 0.0573   | 0.01415  | 0.2824  |
| 产权比率                   | -0.0739  | 0.0117   | 0.14510 |
| 权益乘数                   | -0.0739  | 0.0117   | 0.14510 |
| 加权平均资产收益率              | 0.0610   | 0.01397  | 0.2527  |
| 扣除非经常性损益后的净利润          | -0.0744  | 0.0128   | 0.15871 |
| 净利润/营业总收入              | 0.0374   | 0.03270  | 0.62970 |
| 总资产报酬率                 | 0.0565   | 0.0115   | 0.2458  |
| 营业利润同比增长率(%)           | 0.1723   | 0.0231   | 0.0322  |
| 权益乘数(杜邦分析)             | -0.068   | 0.0074   | 0.10018 |
| 资产负债率                  | -0.076   | 0.0148   | 0.1767  |
| 流动比率                   | 0.01559  | 0.01129  | 0.7317  |
| 销售费用/营业总收入             | -0.00855 | 0.018355 | 0.8821  |
| 非营业利润                  | 0.1392   | 0.07732  | 0.2658  |
| 净资产收益率(摊薄)同比增长率<br>(%) | 0.1992   | 0.00758  | 0.00137 |
| 每股净资产                  | 0.00704  | 0.0108   | 0.8739  |
| 息税前利润                  | -0.0640  | 0.01008  | 0.1692  |
| 每股净资产相对年初增长率(%)        | 0.1388   | 0.01835  | 0.0458  |
| 销售净利率                  | 0.0374   | 0.03270  | 0.629   |
| 经营活动产生的现金流量净额/流<br>动负债 | 0.1669   | 0.0276   | 0.0491  |
| 每股营业总收入                | 0.1568   | 0.015    | 0.0203  |
| 留存收益                   | -0.1122  | 0.0071   | 0.0176  |
| 速动比率                   | 0.0661   | 0.0098   | 0.1529  |
| 每股经营活动产生的现金流量净额        | 0.1897   | 0.0156   | 0.0098  |



|                     |         |        |        |
|---------------------|---------|--------|--------|
| 经营活动产生的现金流量净额/负债合计  | 0.167   | 0.023  | 0.0366 |
| 股权自由现金流量            | -0.0734 | 0.0081 | 0.0925 |
| 每股经营活动产生的现金流量净额同比增长 | -0.0497 | 0.0279 | 0.4931 |
| 流动资产/总资产            | -0.0612 | 0.0158 | 0.2780 |
| 每股企业自由现金流量          | -0.0255 | 0.0150 | 0.6281 |
| 总资产周转率              | 0.1968  | 0.0250 | 0.0225 |
| 每股资本公积              | 0.1062  | 0.0107 | 0.0461 |
| 应收账款周转率             | -0.0489 | 0.0167 | 0.3897 |
| 净利润                 | -0.0492 | 0.0082 | 0.2316 |

光伏行业：

表-4 光伏行业因子检验

| 因子名称          | IC 值平均值 | IC 值标准差  | P 值      |
|---------------|---------|----------|----------|
| 换手率           | 0.2677  | 0.0410   | 0.0177   |
| 总市值           | 0.33892 | 0.103    | 0.0414   |
| 发行股           | 0.1426  | 0.07628  | 0.2528   |
| 自由流通股         | 0.0771  | 0.079693 | 0.52816  |
| 市净率           | 0.5571  | 0.03076  | 0.00023  |
| 分红率           | -0.0569 | 0.0509   | 0.5591   |
| 市盈率           | 0.24942 | 0.0630   | 0.050933 |
| 每股收益          | 0.2528  | 0.2013   | 0.2167   |
| 权益增长率         | 0.11137 | 0.0940   | 0.4080   |
| 每股净资产         | 0.3989  | 0.09497  | 0.01929  |
| 产权比率          | -0.2323 | 0.06922  | 0.07376  |
| 权益乘数          | -0.2323 | 0.0692   | 0.0737   |
| 加权平均资产收益率     | 0.3250  | 0.1269   | 0.0668   |
| 扣除非经常性损益后的净利润 | 0.1376  | 0.1072   | 0.3430   |
| 净利润/营业总收入     | 0.43545 | 0.0169   | 0.0001   |
| 总资产报酬率        | 0.4905  | 0.0644   | 0.0032   |
| 营业利润同比增长率(%)  | 0.4590  | 0.04371  | 0.00169  |

|                         |          |          |          |
|-------------------------|----------|----------|----------|
| 权益乘数(杜邦分析)              | -0.1302  | 0.0508   | 0.20657  |
| 资产负债率                   | -0.26550 | 0.0539   | 0.03113  |
| 流动比率                    | 0.2164   | 0.0541   | 0.062975 |
| 销售费用/营业总收入              | -0.1988  | 0.114    | 0.2008   |
| 非营业利润                   | -0.06343 | 0.06774  | 0.57231  |
| 净资产收益率(摊薄)同比增长率<br>(%)  | 0.42664  | 0.03767  | 0.00168  |
| 每股净资产                   | 0.0884   | 0.2569   | 0.6838   |
| 息税前利润                   | 0.1662   | 0.093    | 0.2313   |
| 每股净资产相对年初增长率(%)         | 0.0773   | 0.1895   | 0.6786   |
| 销售净利率                   | 0.4354   | 0.01698  | 0.0001   |
| 经营活动产生的现金流量净额/流<br>动负债  | 0.39174  | 0.0323   | 0.0017   |
| 每股营业总收入                 | 0.04384  | 0.23807  | 0.8330   |
| 留存收益                    | 0.04543  | 0.02128  | 0.47441  |
| 速动比率                    | 0.22185  | 0.048717 | 0.04897  |
| 每股经营活动产生的现金流量净额         | 0.1228   | 0.02322  | 0.0958   |
| 经营活动产生的现金流量净额/负<br>债合计  | 0.4132   | 0.0154   | 0.0001   |
| 股权自由现金流量                | -0.0647  | 0.24165  | 0.75789  |
| 每股经营活动产生的现金流量净额<br>同比增长 | -0.0187  | 0.0125   | 0.6957   |
| 流动资产/总资产                | 0.0303   | 0.1155   | 0.8338   |
| 每股企业自由现金流量              | 0.0040   | 0.1931   | 0.98268  |
| 总资产周转率                  | 0.1200   | 0.1719   | 0.50497  |
| 每股资本公积                  | 0.0414   | 0.224    | 0.8374   |
| 应收账款周转率                 | 0.1774   | 0.0505   | 0.101506 |
| 净利润                     | 0.14883  | 0.11653  | 0.32662  |

电气设备行业：

表-5 电气设备因子检验

| 因子名称 | IC 值平均值 | IC 值标准差 | P 值 |
|------|---------|---------|-----|
|------|---------|---------|-----|

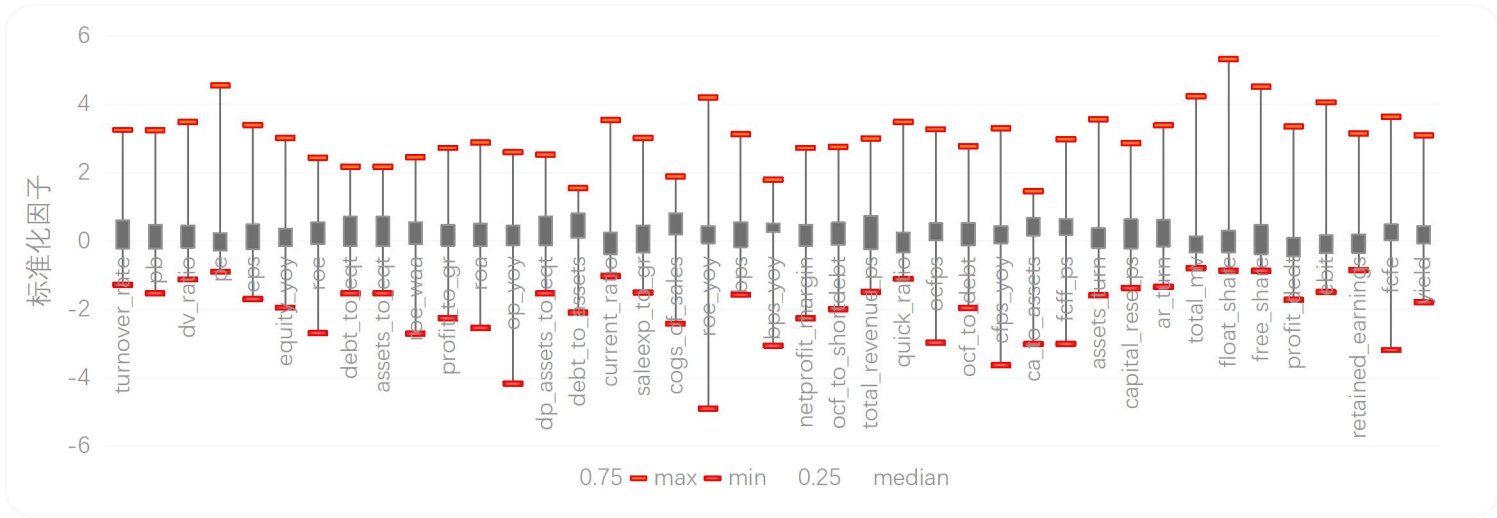
|                    |         |         |          |
|--------------------|---------|---------|----------|
| 换手率                | 0.383   | 0.0048  | 1.00980  |
| 总市值                | 0.2959  | 0.0202  | 0.0022   |
| 发行股                | 0.1276  | 0.010   | 0.02358  |
| 自由流通股              | 0.0898  | 0.00457 | 0.0174   |
| 市净率                | 0.1971  | 0.014   | 0.0073   |
| 分红率                | 0.0336  | 0.0061  | 0.33251  |
| 市盈率                | 0.0275  | 0.00899 | 0.503    |
| 每股收益               | 0.1608  | 0.0321  | 0.070    |
| 权益增长率              | 0.0696  | 0.017   | 0.2447   |
| 每股净资产              | 0.1726  | 0.0186  | 0.0212   |
| 产权比率               | -0.0856 | 0.0100  | 0.081698 |
| 权益乘数               | -0.0856 | 0.0100  | 0.0816   |
| 加权平均资产收益率          | 0.1755  | 0.0178  | 0.0180   |
| 扣除非经常性损益后的净利润      | 0.1939  | 0.0109  | 0.0039   |
| 净利润/营业总收入          | 0.232   | 0.0225  | 0.0090   |
| 总资产报酬率             | 0.1783  | 0.0176  | 0.01673  |
| 营业利润同比增长率(%)       | 0.2135  | 0.0279  | 0.0204   |
| 权益乘数(杜邦分析)         | -0.1074 | 0.0084  | 0.0289   |
| 资产负债率              | -0.0864 | 0.0084  | 0.0606   |
| 流动比率               | 0.08089 | 0.0071  | 0.05758  |
| 销售费用/营业总收入         | 0.0299  | 0.0187  | 0.6113   |
| 非营业利润              | -0.098  | 0.01791 | 0.1211   |
| 净资产收益率(摊薄)同比增长率(%) | 0.1974  | 0.02623 | 0.0244   |
| 每股净资产              | 0.0601  | 0.03139 | 0.4375   |
| 息税前利润              | 0.16505 | 0.0098  | 0.0065   |
| 每股净资产相对年初增长率(%)    | 0.1293  | 0.0198  | 0.0657   |
| 销售净利率              | 0.2322  | 0.022   | 0.0090   |
| 经营活动产生的现金流量净额/流动负债 | 0.0483  | 0.01627 | 0.38885  |
| 每股营业总收入            | 0.0327  | 0.009   | 0.4368   |

|                     |          |         |          |
|---------------------|----------|---------|----------|
| 留存收益                | 0.169    | 0.0113  | 0.0081   |
| 速动比率                | 0.0748   | 0.0064  | 0.0630   |
| 每股经营活动产生的现金流量净额     | 0.06446  | 0.0211  | 0.3187   |
| 经营活动产生的现金流量净额/负债合计  | 0.1554   | 0.017   | 0.3401   |
| 股权自由现金流量            | -0.0284  | 0.00548 | 0.3831   |
| 每股经营活动产生的现金流量净额同比增长 | 0.01565  | 0.0009  | 0.2546   |
| 流动资产/总资产            | 0.078053 | 0.0045  | 0.0294   |
| 每股企业自由现金流量          | -0.0086  | 0.0129  | 0.85869  |
| 总资产周转率              | 0.0338   | 0.0103  | 0.44573  |
| 每股资本公积              | 0.0127   | 0.0204  | 0.8343   |
| 应收账款周转率             | -0.012   | 0.0102  | 0.764181 |
| 净利润                 | 0.17603  | 0.0136  | 0.0101   |

根据以上结果，在不同行业中，我们选取 IC 值绝对值更大，IC 值波动率更小，P 值更小的因子，并考虑不同因子所反映股票情况的不同方面，选入最终的多因子选股策略，同时，因为我们的限制条件中包含总市值这一因子，所以我们将总市值因子也纳入到最后的多因子选股策略中。最终选入策略的因子已经在上表中标红。

(二) 特征分析

1.特征分析——因子分布箱型图



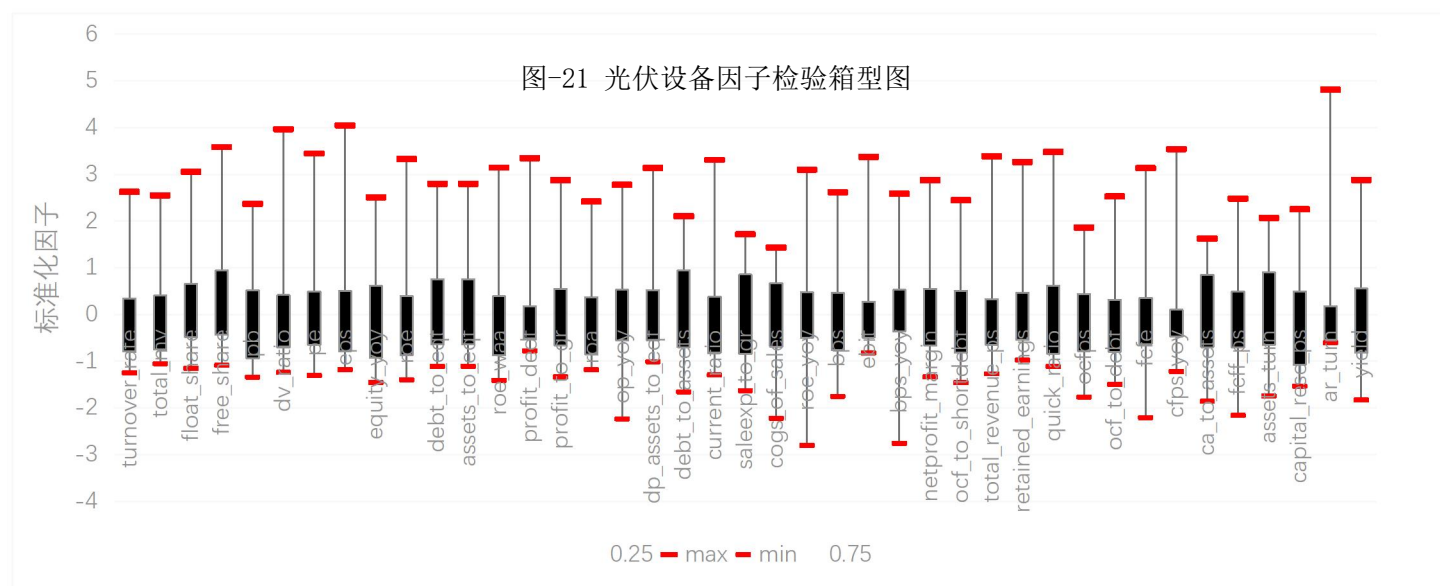
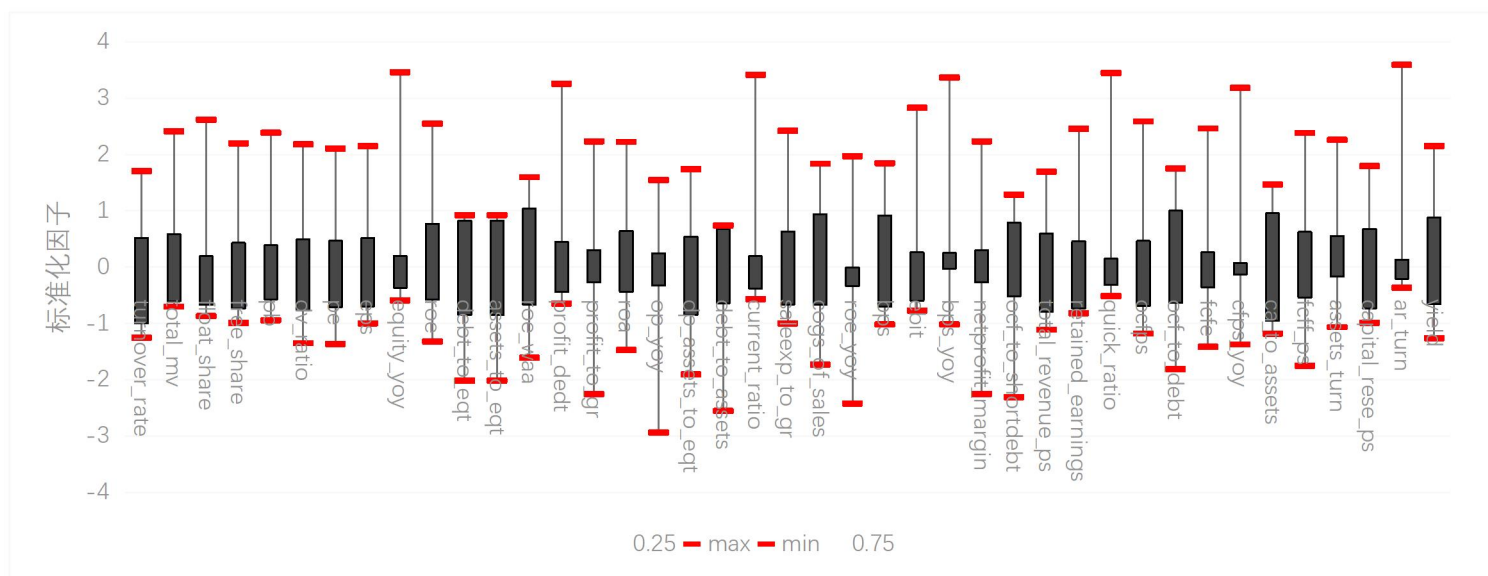


图-22 食品饮料因子检验箱型图

根据上面的三幅行业因子箱型图，我们发现，有的因子具有很长的分布范围，有的分布则更为集中，而有相同分布范围的，我们则更倾向于中间箱体更小的因子。上述因子的 IC 值可以结合我们这里的数据集中程度进行综合判断最终因子的选取。

## 2.特征分析——因子自相关分布

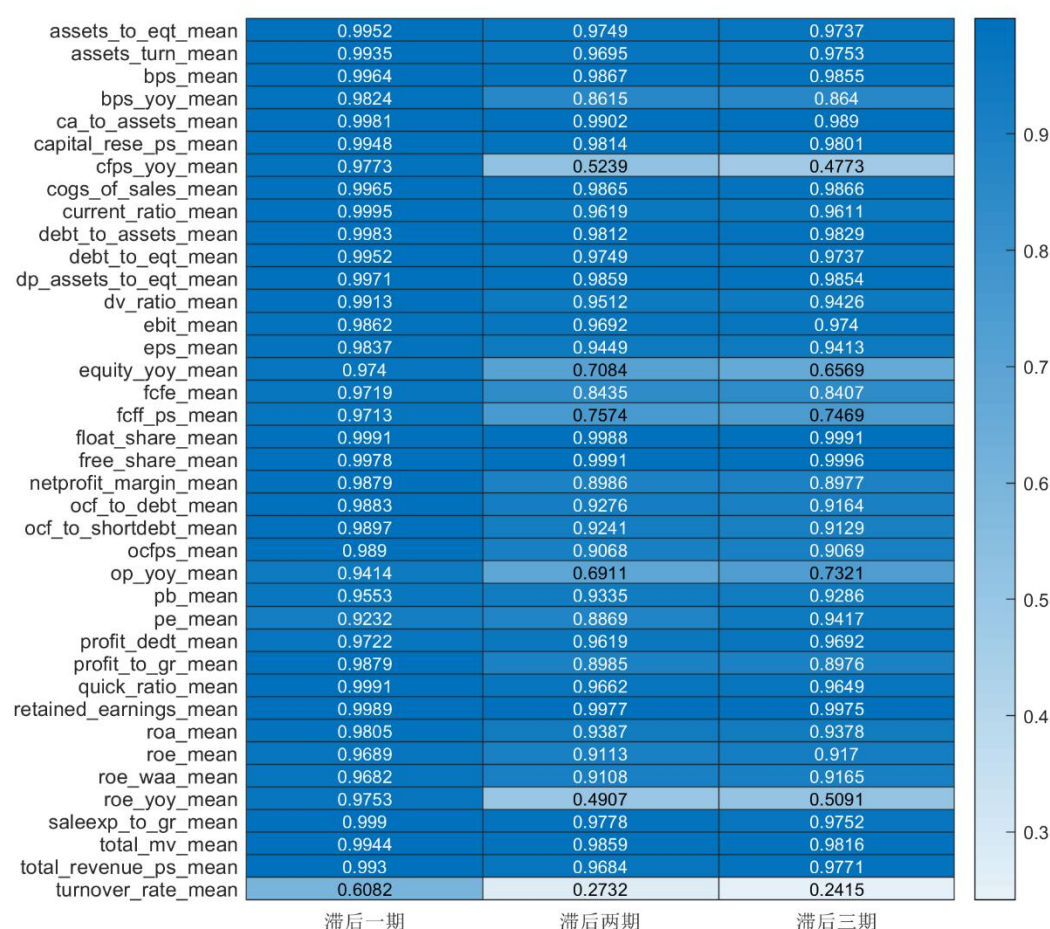


图-23 因子自相关分布热力图

在图-23 中，颜色较深的因子的变动较为规律，在实际市场中，由于它们几乎都是年报上的财务数据，在一季度之内变化不大或基本不变，所以较为稳定。而对于颜色较浅的因子，诸如换手率、加权净资产增长率等，我们难以预判到它们的变化。显而易见的是，因子数据相隔期数越长，自相关性越差，

## 五、结果分析

下图分别是纯多因子选股策略的回测结果和加入择时策略后的多因子选股策略的回测结果，观察回测结果可以发现：在加入多因子策略之后，回测结果显著高于同期沪深 300 的表现，特别是在后一个月中，策略的优势明显体现。

加入择时策略后，我们可以看到该策略  $\beta$  值 0.682 小于单纯多因子策略的 1.136，表明加入择时策略后更能有效抵御市场风险，而且在低  $\beta$  值的情况下，我们加入择时后的策略的阿尔法  $\alpha$  为 1.967，夏普率为 3.505，信息比率为 3.578，



相较于单纯因子策略分别提升了 207.3%，103%，72.6%表明该策略具有很强的盈利能力。

比较两组回测的结果，易得加入择时策略后，更容易识别股票下行信号，从而及时止损。且最大回撤持续期以及最长回撤持续期显著低于单纯因子策略。

加入择时策略后的波动率、最大回撤、跟踪误差和下行风险都高于单纯因子策略，说明加入择时策略会使风险显著增加，这也正是追涨杀跌策略的劣势所在。但也应看到在承受风险的同时，收益也在相应增加，故波动率的提升是可以接受的。



图-24 纯多因子选股策略



图-25 加入择时策略后的多因子选股策略

附录

小组成员及分工  
略

## 附录二

```
import pandas as pd
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import matplotlib.pyplot as plt

#得到数据并 dataframe 化
df=pd.read_excel('C:/Users/leo/Desktop/行业指数.xlsx').values[:,39]
df=pd.DataFrame(df)
plt.rcParams.update({'figure.figsize':(9,7), 'figure.dpi':120})

#设定图像数目
fig1, axes1 = plt.subplots(3, 1, sharex=True)

#原序列、一阶差分、二阶差分
axes1[0].plot(df)
axes1[0].set_title('Original Series')
axes1[1].plot(df.diff())
axes1[1].set_title('1st Order Differencing')
axes1[2].plot(df.diff().diff())
axes1[2].set_title('2nd Order Differencing')
plt.show()

#自相关图，用以观察 d 值和 q 值
fig2, axes2 = plt.subplots(3, 2, sharex=True)
plot_acf(df, ax=axes2[0, 0])
plot_acf(df.diff().dropna(), ax=axes2[1, 0])
plot_acf(df.diff().diff().dropna(), ax=axes2[2, 0])

#偏自相关图，用以观察 p 值
plot_pacf(df, ax=axes2[0, 1])

plt.show()
```

### 附录三

```
from statsmodels.tsa.arima_model import ARIMA
import pandas as pd
import time
yoy=[]
start=time.time()
def arima_desu(d,p,q,i):
    df=pd.read_excel('C:/Users/leo/Desktop/行业指数.xlsx').values[:,i]

    model= ARIMA(df, order=(d,p,q))
    model_fit = model.fit(dispatch=0, method='css')

    # predict_ts=model_fit.predict()
    # print(model_fit.summary())
    a=model_fit.forecast(90)

    c=list(list(a)[0])[-1]/list(list(a)[0])[0]
    return c
for i in range(1,39,2):
    df=pd.read_excel('C:/Users/leo/Desktop/行业数.xlsx',sheet_name='Sheet4').values[:,i]
    yoy.append(arima_desu(df[0],df[1],df[2],i))

tic=time.time()-start
print('tic',tic)
```

### 附录四

#### 因子自相关分布代码

```

import pandas as pd
factor={}
f={}
sub_factor={}
a=pd.read_csv(str(1)+''.csv')
del a['Unnamed: 0']
c=pd.DataFrame(columns=a.columns)
d=pd.DataFrame(columns=a.columns)
for i in range(5):
    e={}
    a=pd.read_csv(str(i+1)+''.csv')
    b=pd.read_csv(str(i+2)+''.csv')
    a.index=a[a.columns[0]]
    b.index=b[b.columns[0]]
    for t in a.index:
        for m in b.index:
            if t==m:
                c.loc[t]=a.loc[t]
                d.loc[t]=b.loc[t]
c=c.dropna(how='any')
d=d.dropna(how='any')
for j in c.columns:
    e[j]=c[j].corr(d[j])
    f[i]=e
a=pd.DataFrame(f).T
for i in a.columns:
    sub_factor[i+'_mean']=a[i].mean()
factor['一月自回归']=sub_factor
f={}
sub_factor={}
a=pd.read_csv(str(1)+''.csv')
del a['Unnamed: 0']
c=pd.DataFrame(columns=a.columns)
d=pd.DataFrame(columns=a.columns)
for i in range(4):
    e={}
    a=pd.read_csv(str(i+1)+''.csv')
    b=pd.read_csv(str(i+3)+''.csv')
    a.index=a[a.columns[0]]
    b.index=b[b.columns[0]]
    for t in a.index:
        for m in b.index:
            if t==m:

```

```

                c.loc[t]=a.loc[t]
                d.loc[t]=b.loc[t]
c=c.dropna(how='any')
d=d.dropna(how='any')
for j in c.columns:
    e[j]=c[j].corr(d[j])
    f[i]=e
a=pd.DataFrame(f).T
for i in a.columns:
    sub_factor[i+'_mean'] = a[i].mean()
factor['二月自回归']=sub_factor
f={}
sub_factor={}
a=pd.read_csv(str(1)+''.csv')
del a['Unnamed: 0']
c=pd.DataFrame(columns=a.columns)
d=pd.DataFrame(columns=a.columns)
for i in range(3):
    e={}
    a=pd.read_csv(str(i+1)+''.csv')
    b=pd.read_csv(str(i+3)+''.csv')
    a.index=a[a.columns[0]]
    b.index=b[b.columns[0]]
    for t in a.index:
        for m in b.index:
            if t==m:
                c.loc[t]=a.loc[t]
                d.loc[t]=b.loc[t]
c=c.dropna(how='any')
d=d.dropna(how='any')
for j in c.columns:
    e[j]=c[j].corr(d[j])

    f[i]=e
a=pd.DataFrame(f).T
for i in a.columns:
    sub_factor[i+'_mean'] = a[i].mean()

factor['三月自回归']=sub_factor

data=pd.DataFrame(factor)

```