

单因子测试（下）——回归测试法

原创 量化小白H 量化小白上分记 2018-12-23

收录于话题

#多因子模型

12个 >

之前两篇分别总结了因子数据的预处理和单因子测试的分层测试法，本篇总结回归测试法，相较于分层测试法，回归测试法更简洁。

因子预处理

与分层测试法不同，回归法测试时，因子可以不进行中性化处理，只进行异常值处理和标准化（zscore）处理，将中性化的过程包含在测试过程中。

方法说明

每一期，对全样本做一次回归，回归时将本期到下一期的股票收益率作为因变量，当期的因子暴露值作为自变量，同时考虑到市值中性和行业中性，加入行业虚拟变量和市值变量作为自变量，市值变量可以取对数消除量纲，之后进行回归，即

$$\begin{bmatrix} r_{t1} \\ \dots \\ r_{tn} \end{bmatrix} = \begin{bmatrix} \beta_{t1} & I_{t1,1} & \dots I_{t1,k} & m_{t1} \\ \dots & \dots & \dots \dots \dots \\ \beta_{tn} & I_{tn,1} & \dots I_{tn,k} & m_{tn} \end{bmatrix} \begin{bmatrix} f_{t1} \\ \dots \\ f_{tn} \end{bmatrix} + \begin{bmatrix} u_{t1} \\ \dots \\ u_{tn} \end{bmatrix}$$

其中, r 为股票收益率, β 为因子暴露,
 I 为行业虚拟变量, m 为市值
 f 为因子收益

得到的回归结果中, 因子暴露的系数即为因子收益率, 通过多期回归后, 就可以得到因子收益率序列及回归的t值序列, 通过这两个序列可以构造指标分析因子的表现。

一些细节

1. 包含行业虚拟变量的回归中, 如果再加有截距项, 就会导致**完全共线性**, 因此回归前必须进行处理, 大概有这么几种方法: **不加截距项** (如上方程); **加截距项, 但将其中一个行业变量的暴露值全部设为0或者删除**, 消除共线性; **施加线性约束, 转化为带约束的线性回归**, 约束的选择和理由如下, 这个方法我没试过, 所以就直接贴出来吧。



2. 这里的回归一般用最常见的**OLS**，但考虑到数据的异方差性，有的研报里也会用加权最小二乘（**WHS**）或者稳健回归（**RLM**）。

评价指标

1. 因子收益率序列检验t值的均值
2. 因子收益率序列大于0的概率
3. t值序列绝对值的均值
4. t值序列绝对值大于等于2的概率
5. IC , ICIR
6. IC均值
7. IC绝对值大于0.02的概率

基本都是一些非常简单的指标，至于为什么取t值绝对值大于2，IC值大于0.02，也没有太好的原因，但也是符合常理的，t值绝对值越大，回归方程系数的显著性越高，IC表示相关系数，绝对值越大，表明因子暴露跟股票收益率的相关性更高。

因子测试实例

测试因子：**pb_lf**，需倒数

测试区间：2010年1月-2018年5月



PB因子收益序列显著大于0，因子收益状况良好



```
1 def MultiPeriodFactorTest(factor, DateStart, DateEnd, if_reciprocal, if_neutral_industry=True,
2
3
4     # 获取交易日序列
5     BargainDate = w.tdays(DateStart, DateEnd, "Period=M")
6     BargainDate = pd.DataFrame(BargainDate.Data[0], columns = ['date'])
7     # 循环调用单期因子测试函数，得到收益率序列，IC序列，t值序列
8     result = pd.DataFrame(columns=["DateStart", "DateEnd", "factor_return", "t_values", "IC"])
9     for i in range(1, BargainDate.shape[0]):
10         datebuy = BargainDate.date[i-1]
11         datesell = BargainDate.date[i]
12         result1 = SingelePeriodFactorTest(factor, datebuy, datesell, if_neutral_industry, if_neut
13         result = result.append(result1)
14
15     result.factor_return = result.factor_return
16
17     # 计算均值
18     t_mean = result.t_values.mean()
19     return_mean = result.factor_return.mean()
20     t_abs_mean = result.t_values.abs().mean()
```

```
21
22
23     # t>0比例
24     if_t_0 = pd.DataFrame.mean(result.t_values>0)
25
26
27     # IC统计量
28     IC_mean = result.IC.mean()
29     IC_std = result.IC.std()
30     if_IC_0 = pd.DataFrame.mean(result.IC>0)
31     if_abs_IC_002 = pd.DataFrame.mean(result.IC.abs()>0.02)
32
33
34     # 计算ICIR
35     ICIR = IC_mean/IC_std
36
37     final = {"因子收益序列t均值":t_mean,
38             "因子收益序列均值":return_mean,
39             "t>0比例":if_t_0,
40             "abs(t)均值":t_abs_mean,
41             "IC均值":IC_mean,
42             "IC标准差":IC_std,
43             "IC>0比例":if_IC_0,
44             "abs(IC)>0.02比例":if_abs_IC_002,
45             "ICIR":ICIR
46         }
47     # '''
48     # 作图
49     # '''
50     # X = np.arange(result.shape[0])
51     #
52     # # 因子收益
53     # plt.figure()
54     # plt.subplot(221)
55     # plt.bar(X,result.factor_return)
56     # plt.title('factor_return')
57     # plt.subplot(222)
58     # plt.hist(result.factor_return)
59     # plt.title('factor_return')
60     #
61     # # t值
62     # plt.subplot(223)
63     # plt.bar(X,abs(result.t_values))
64     # plt.title('abs(t_value)')
65     #
66     #
67     # # IC
68     # plt.subplot(224)
69     # plt.bar(X,result.IC)
70     # plt.title('IC')
```

```

71 # plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=1.0)
72 #
73
74     return(final,result)
75
76 '''
77     单期因子测试函数
78 '''
79 def SinglePeriodFactorTest(factor,DateStart,DateEnd,if_neutral_industry, if_neutral_mktcap):
80
81     factor_name = factor['value'][0][0]
82
83     print(factor_name + '单因子测试: ' + str(DateStart) + ' -- ' + str(DateEnd))
84     data = getData(factor,DateStart)
85     # 是否取倒数
86     if if_reciprocal:
87         data.iloc[:,0] = 1/data.iloc[:,0]
88     # 通过本地数据库计算股票收益率
89     stock_list = data.index.tolist()
90     stock_list = list(map(lambda x:x.strip('.SZ'),stock_list))
91     stock_list = list(map(lambda x:x.strip('.SH'),stock_list))
92     stock_list = tuple(stock_list)
93     price_s = pd.read_sql("select stockcode,closeprice as price_s \
94                           from database\
95                           where stockcode in {} and bargaindate='{''\
96                           order by stockcode".format((stock_list),DateStart),stkbse)
97     price_s=price_s.set_index('stockcode')
98     price_e = pd.read_sql("select stockcode,closeprice as price_e \
99                           from database\
100                          where stockcode in {} and bargaindate='{''\
101                          order by stockcode".format((stock_list),DateEnd),stkbse)
102
103
104
105     price_e=price_e.set_index('stockcode')
106     price=pd.merge(price_s,price_e,left_index=True,right_index=True)
107
108
109     price['rt']=price['price_e']/price['price_s']-1
110     # 标准化
111     factor_all= norm(data,if_neutral_industry, if_neutral_mktcap)
112     factor_all.index.name = 'stockcode'
113     factor_all = factor_all.reset_index()
114     factor_all['stockcode'] = factor_all['stockcode'].apply(lambda x:x.strip('.SZ|.SH'))
115     factor_all = factor_all.set_index('stockcode')
116     Alldata = pd.merge(factor_all,price,left_index=True,right_index=True,how = 'left')
117
118
119
120

```



```

121     y = Alldata[['rt']]
122     X = factor_all
123     X['Intercept'] = 1
124
125     model = sm1.OLS(y,X)
126     result=model.fit()
127
128
129     # 计算IC,IR
130     IC = calc_IC(factor_all[factor_name], y, 'Spearman')
131
132     print(str(DateStart) + "——" +str(DateEnd)+", "+factor_name + '因子收益为:' + str(round(result
133
134     return_t=pd.DataFrame([[DateStart,DateEnd,result.params[0],result.tvalues[0],IC]],
135                           columns=["DateStart","DateEnd","factor_return","t_values","IC"])
136
137     return(return_t)

```

参考文献

1. 20170410-光大证券-光大证券多因子系列报告之一：因子测试框架

收录于话题 #多因子模型·12个 >

< 上一篇

单因子测试（中）——分层测试法

下一篇 >

多因子尝试（一）：因子加权方法在选股中的应用

喜欢此内容的人还喜欢

因未发奖金，一名程序员删代码泄愤，平台崩溃近三个小时

猿大侠



蒙古国30000头羊终于到了！全国人民流下了感动的口水(´ ͡ω ͡ `)

骑驴



暴涨40%！北京楼市，正处于启动前夜？

米宅北京

