多因子模型-国泰君安alpha191短周期量价多因子

写在前面：

这是上学期一个proj的一部分内容，历时大概1个半月，学习了multiFactor一个多学期了，我发现这个东西做的好确实很不容易，有非常多的细节需要认真check，不要用到未来信息还有code的高效性。

先读一下研报吧，我觉得写得挺好的，数学公式要是可以用latex打就更好了：

2017.06.15

基于短周期价量特征的多因子选股体系

——数量化专题之九十三

短周期交易型多因子阿尔法选股体系就是用open，high，low，close以及其他一些量价指标构造（按照交易逻辑、遗传算法、机器学习等方法遍历）出191个（甚至更多）的alpha，然后用这些alpha去fit model，再去predict model，产生一些信号。寻找哪些是使得股价上升的因子。

这些因子长这样（非常的怪异...）：

表 6: 因子明细:

因子顺序	因子构建方式
Alpha1	*(-1 * CORR(RANK(DELTA(LOG(VOLUME), 1)), RANK(((CLOSE - OPEN) / OPEN)), 6))*
Alpha2	*(-1 * DELTA(((((CLOSE - LOW) - (HIGH - CLOSE)) / (HIGH - LOW)), 1))*
Alpha3	*SUM((CLOSE=DELAY(CLOSE,1)?0:CLOSE-(CLOSE>DELAY(CLOSE,1)?MIN(LOW,DELAY(CLOSE,1)):MAX(HIGH,DELAY(CLOSE,1)))),6)*
Alpha4	*((((SUM(CLOSE, 8) / 8) + STD(CLOSE, 8)) < (SUM(CLOSE, 2) / 2)) ? (-1 * 1) : (((SUM(CLOSE, 2) / 2) < ((SUM(CLOSE, 8) / 8) - STD(CLOSE, 8))) ? 1 : ((((1 < (VOLUME / MEAN(VOLUME,20))) || ((VOLUME / MEAN(VOLUME,20)) == 1)) ? 1 : (-1 * 1))))*
Alpha5	*(-1 * TSMAX(CORR(TSRANK(VOLUME, 5), TSRANK(HIGH, 5), 5), 3))*
Alpha6	*(RANK(SIGN(DELTA((((OPEN * 0.85) + (HIGH * 0.15))), 4)))* -1)*
Alpha7	*((RANK(MAX((VWAP - CLOSE), 3)) + RANK(MIN((VWAP - CLOSE), 3))) * RANK(DELTA(VOLUME, 3)))*
Alpha8	*RANK(DELTA(((((HIGH + LOW) / 2) * 0.2) + (VWAP * 0.8)), 4) * -1)*

用的是横截面回归，这是一波正常操作，借鉴了Barra业绩归因模型。Barra Risk Model 里用了Industry Factor 以及 10类style Factor做risk decomposition，详见Barra-CNE5，里面清楚地写了styleFactor的构造方式，大类包括哪些以及每个大类下面的小类因子的权重。（值得一提的是，与Barra model有差异的做法是，Fama French Model，Fama-MacBech Model，但这些这里都不会讨论。）

数据准备：

我们需要的数据是，清洗完的量价数据(OHLW)来计算每个股票每一天的alpha，如果有N个因子，那么此处我们可以得到N张 time * stocks * alpha的表。现实项目的情况是191个alpha太多了，有一些数据得不到比如vwap之类的，所以没办法计算，所以最终测试了31个。

用清洗完的行业数据来生成行业哑变量矩阵，如果有M个行业，得到 M张time * stocks * industry 的表。

用清洗完的量价数据+一些财务数据，生成styleFactor，如果有P个styleFactor，得到P张time * stocks * style的表。

用ST和停牌的数据生成一张stockScreenTable：time * stocks 来过滤当天不能交易的股票。

用close生成一张ReturnTable：time * stocks。

单因子检验：

我们要对所有的单因子进行有效性检验，通过因子的IC来判断因子的可靠程度，因为是量价因子的关系，所以预测的周期不会很长，1-5天。

Step1:正交标准化，取目标因子残差截面ε_K^t，即

$$X_K^t = \beta_{industry} X_{industry} + \beta_{style} X_{style} + \varepsilon_K^t$$

Step2:针对给定预测周期d，通过回归方程计算单期因子收益率f_k；

$$R_{t+d} = f_{industry} X_{industry} + f_{style} X_{style} + f_k \varepsilon_K + \varepsilon_{t+d}$$

Step3:计算因子收益率序列的年化收益$E(f_k)$及信息比率$IR(f_k)$；

$$E(f_k) = 252 * (\overline{f_k / d}) \quad IR(f_k) = \sqrt{252} * (\overline{f_k / d}) / \sigma(f_k / d)$$

Step4:对于不同的收益预测周期d'，重复第2、3步。

$$R_{t+d'} = f_{industry} X_{industry} + f_{style} X_{style} + f_k' \varepsilon_K + \varepsilon_{t+d'}$$

$$E(f_k') = 252 * (\overline{f_k' / d'}) \quad IR(f_k') = \sqrt{252} * (\overline{f_k' / d'}) / \sigma(f_k' / d')$$

方法就是以上四步，在做step1前需要先对因子进行中性化，使得因子们都在同一些区间里可比。然后在每一天，把因子暴露的值（以下称之为 factor exposure）和IndustryFactor和StyleFactor进行正交，取得的残差作为真正的factor exposure。这是一个比较大的矩阵在做回归，不带截距项，取残差作为factor exposure。（此处和格拉姆施密特正交化的做法不同，格拉姆施密特是两两正交，gtja这种正交方法允许因子和因子之间存在相关性）

step2值得提的是，y是R_{t+d}而x是X_t,如果d=1，把y的returnTable向上shift一天，就可以做同一个下标的回归。

step3和4，我们改良了一下这2步，根据后文对modelIC的定义，这里定义一个叫FactorIC，这个FactorIC和百度里查到的IC定义会有不一样。通常来说，正常的因子 NormalIC是指因子暴露（factor exposure）和下一期的returnTable之间的pearson correlation，rankIC的定义是factor exposure和下一期returnTable之间的spearman correlation（秩相关系数）。

$$Normal_{IC} = pearsonCorr(f_k, R_{t+1})$$

$$Rank_{IC} = spearmanCorr(f_k, R_{t+1})$$

但根据后文的modelIC的算法，FactorIC的计算方式是（以预测1期为例，d=1）：

$$Normal_{IC} = pearsonCorr(f_k * \epsilon_t, R_{t+1})$$

$$Rank_{IC} = spearmanCorr(f_k * \epsilon_t, R_{t+1})$$

差别就在于用了factor Return * factor exposure的结果来和下一期的return算相关系数，我觉得这样更加的合理，因为是因子的这一部分乘起来的结果 代表了 下一期这个因子预测了部分return。预测的部分return和真正的return做corr相关性才更有说服力。

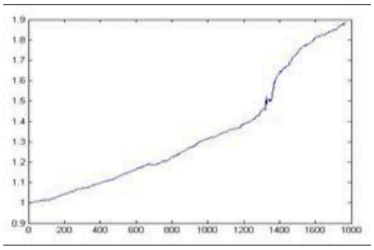
这一步会比较容易用到未来信息和遇到下标出错的问题，原因是我们在fit模型的时候，如果只有3天（记为1，2，3），假设所有拿到数据的时间都认为是14:56分，在收盘前的时刻。这个意味着，每天我们拿到新数据在14:56分，我们会在尽可能快的时间内计算完当天的multiFactorModel的结果，然后生成持仓表，在收盘之前完成当天的交易，然后第二天继续开始，其实也可以在begin的时候比如9:35分这样进行，保持换仓时间的同一性。当然这是一种理想情况hhhhh。

举个例子吧。在Day1我们不能做任何事情，因为Day1拿不到未来的return。在Day2，我们用的是Day2的return和Day1的factor exposure回归得到的系数认为是Day2的factor Return。Return这个词可以理解成一个过程量，因为Day2的stockReturn其实是Day1的结束-Day2的结束的stockReturn，因此类推一下，Day2的factor Return 是因子Day1的结束-Day2的结束的factorReturn（像绕口令一样QAQ）。factor exposure理解成一个时刻量。然后现在时间依旧是Day2收盘之前，我们会用Day2的factor Return * Day2的 factor exposure = predict Day3 stockReturn。到了Day3，会算corr(predict Day3 stockReturn, sure Day3 stockReturn) = Day2 点factorIC。注意这里算出来的是day2的factorIC！！！到了Day3才能填Day2点factorIC。所以3天才能算一个因子IC...（去头去尾，这是一个我们小组讨论了蛮久的point，真的挺久的）

在这个阶段也会算每个factor的一些统计指标，不过那些不太重要，比如factor Return序列统计量，single factor model的显著性之类的（其实就是这个模型的系数的t统计量）。最重要的就是factor IC稳不稳定。

和研报里的结果对比：差距还是有点大的...数据区间的关系还有cleanData的方式可能会不一样。

图 4 累计因子收益率

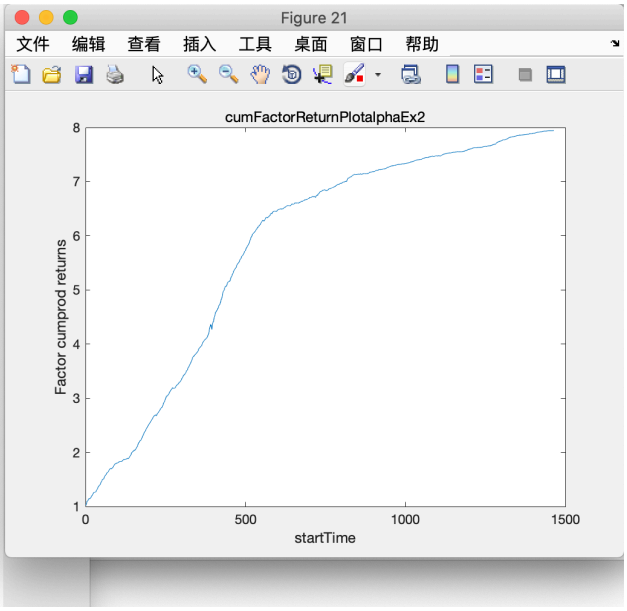


数据来源：国泰君安证券研究

当日个股开盘缺口，短周期内有较强的动量效应，累计因子收益率较为显著。

表 3：因子显著性检验

因子定义	Factor Return	Factor Return IR	IC	IC IR
开盘缺口	9.00%	4.83	0.005	2.94

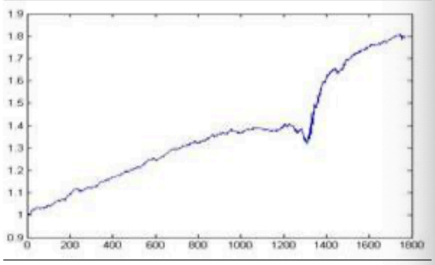


举例 3：异常成交量

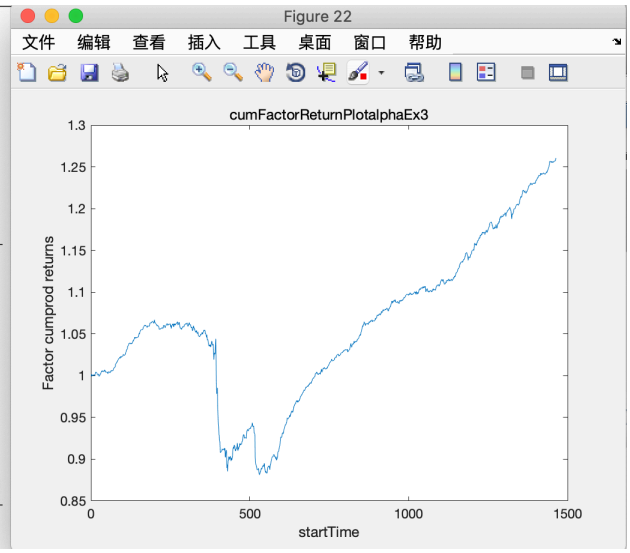
经验观察：当日成交量较短周期均值异常放大、减小。

因子定义：$Alpha_t^i = -1 * Volume_t^i / mean(Volume_{t-d:t}^i)$

图 6 累计因子收益率



数据来源：国泰君安证券研究



多因子模型：

根据研报里的说法，这191个alpha因子，他们是没有做过变量选择，直接全部一起都全都扔到Linear Model里去看模型的拟合程度和预测效果，扔进去的越多解释程度R^2越高。研报里还是一步一步做的，先扔20个，再40个，再60个，然后得到的结论是加的alpha越多，模型IC越高。

按道理这里是需要做feature selection的，比如lasso Regression或者ridge设置threhold，删去其中不太能增加模型解释程度的部分，或者是feature composition把一些相似度高的因子合成成一个因子或者扔掉其中一些因子。

Step1:对所有因子进行正交标准化处理，取目标因子残差截面ε_K^t，即

$$X_K^t = \beta_{industry} X_{industry} + \beta_{style} X_{style} + \varepsilon_K^t$$

Step2:针对给定预测周期 t=1，通过多次多元回归方程计算得到过去时间段 T 内过所有因子收益率向量簇$\{f_k^t\}_K^T$，即

$$R = f_{industry} X_{industry} + f_{style} X_{style} + f_{k1}\varepsilon_{k1} + f_{k2}\varepsilon_{k2} + \ldots + f_K \varepsilon_{K2} + \varepsilon$$

Step3:利用过去 T 时间段因子收益率均值作为下期预测值，乘以当期每个目标因子残差截面，得到阿尔法收益截面预测向量，即

$$E(\varepsilon_{t+1}) = \sum_{k=1}^{K} ((\sum_{t=1}^{T} f_k^t / T) * \varepsilon_k^t)$$

Step4:计算各期模型预测系数，进而计算模型系数 T 检验结果，即

$$IC_{AlphaModel}^t = corr(E(\varepsilon_{t+1}), \varepsilon_{t+1})$$

$$TStas_{AlphaModel} = \overline{IC}_{AlphaModel} / (\sigma_{IC_{AlphaModel}} / \sqrt{T})$$

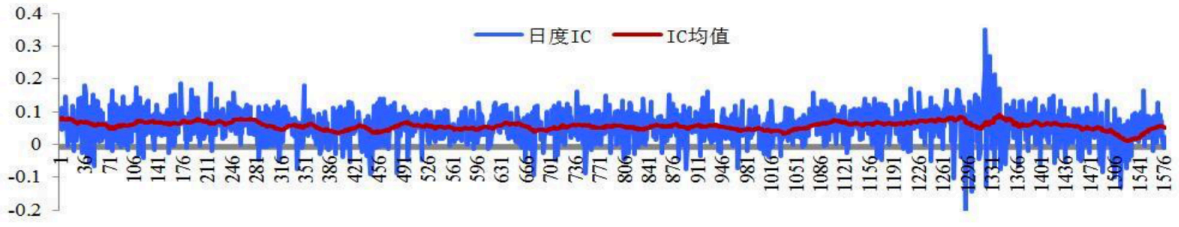
这里的Step1和singleFactor的相同。所以如果有N个alpha，可以直接生成一个正交完以后的表 time * stocks * Orth_alpha（N个）。

Step2是多元线性回归的fit，把所有的alpha（N个）都放入模型，每一个time，得到N个factor Return，得到一张 time * factor Return（N个）的表。

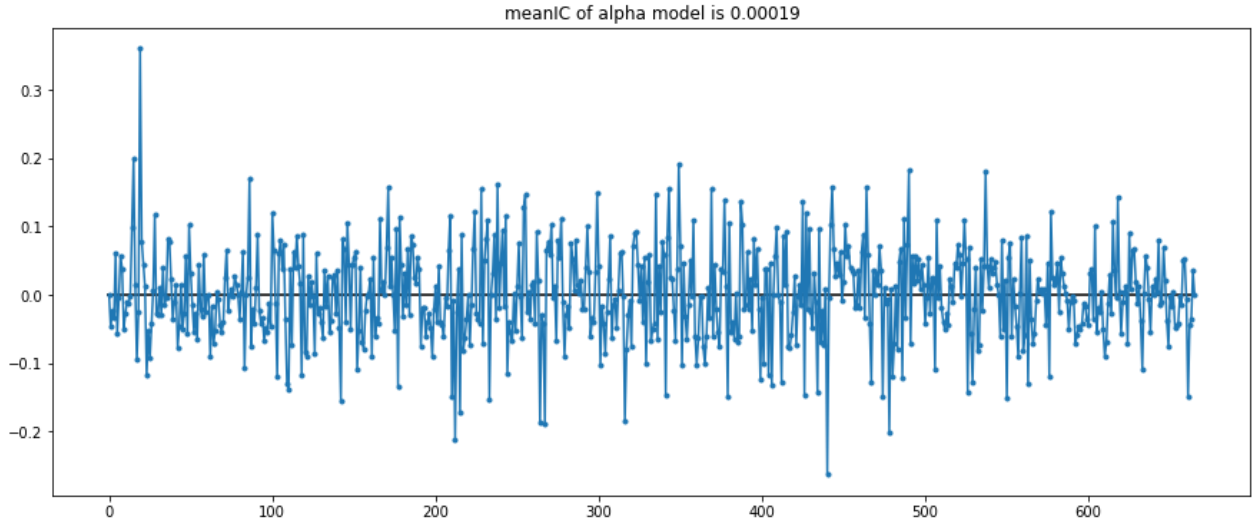
step3是预测return，方法和singleFactor的是相同的，注意下标的问题。

step4是算modelIC，通常情况下rankIC会好于normalIC。

图 21 模型预测系数



数据来源：国泰君安证券研究



然后可以画modelIC的序列，计算modelIC的mean，计算modelICIR。

多空组合：

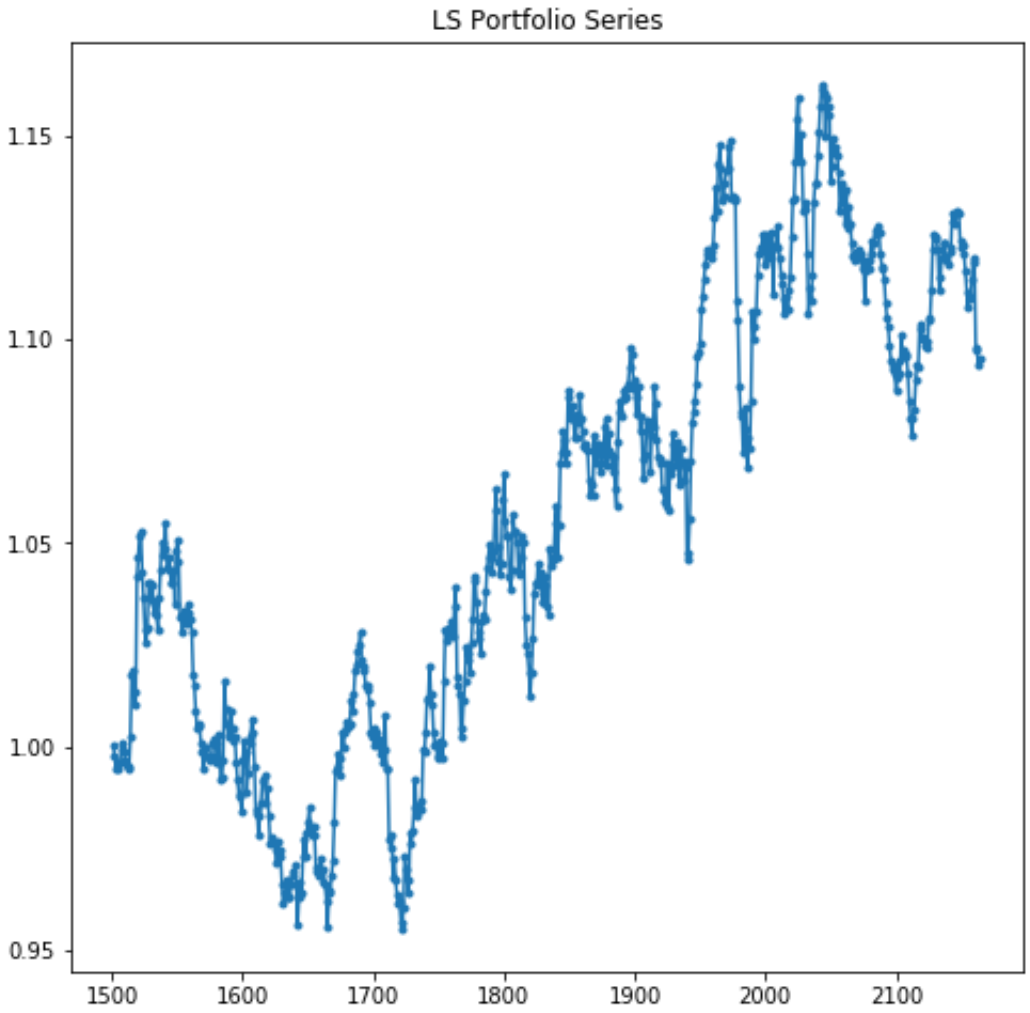
研报的部分到了以上就学得差不多了，如果是做feature engineer的话，发现有用的alpha就可以了（笑，这恰恰是最难的）。这一部分其实没有涉及到策略。我所理解的策略这部分就是，最简单像均线策略，短期均线穿过长期均线，买，生成持仓表这种，复杂的会像cta策略那种，加入很多量化突破的内容。这部分会模拟现实的交易场景，比如换手率，交易费用，涨停板之类的。当然这部分可能很复杂，要学习的地方还有很多。

但做个回测的话，可以简化emm。

这里回测了一个根据31个因子构造的多空组合，每日换仓，生成nav，没有交易费用。按照因子predict出来的stock return，每日对可交易股票且没有nan的股票进行排序，从小到大，分成10组，在每日收盘前，做多第10组做空第1组。用第二天的真实收益，生成一条group10的stockreturn-group1的stockreturn的nav，用这条nav来评价多因子模型的好坏。

然后拿这条nav可以算sharp ratio，年化收益率，max drawdown之类的评价指标。perform可以写成一个函数然后每次都是这些，不过python也有package可以直接算。

我的结果：




```
In [37]: rety,Sharp,MDD = performance(nav,LSSeries)
年化收益率为:39.43%
夏普比为: 0.02
最大回撤率为: 9.44%
```

python里哪些东西比较好用：

我的python也是初学者水平...具体来说，就是pd，np，plt + load文件的水平，sklearn里的LinearRegression, Lasso, Ridge水平。之前本科毕业论文做了个naiveBayes相关的，于是也用了用sklearn。

这里值得提一下的是，我用python的pd里的groupby和pivotTable完成多空组合这件事情，这两个东西配合起来非常好用，再加上datetime的处理可以保留xlabelTick和merge里的左连接、右连接使得多个panel data可以一起处理。

比较好用的还有lambda函数，这个函数比较适合内置函数没有然后要做的操作也非常简单。

enumerate可以循环同时生成index和值。

deque、stack（数据结构其实挺重要的，但基础的那些其实用的好就会很方便，但tree这种我想也用不到这里....）。deque可以存放固定长度的数据然后会根据deque的长度pop和push。

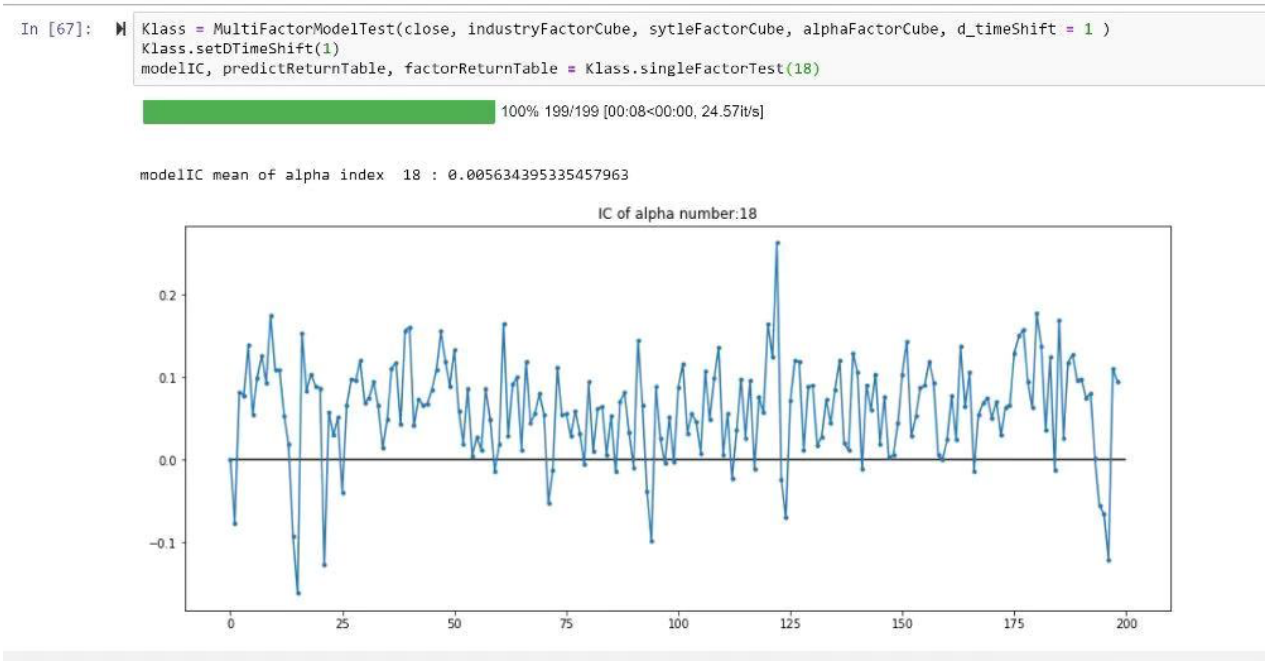
python有rolling这种神奇函数hhhhh

args用于可变参数，如果不确定要输入多少个变量，这些变量又有非常像的结构，那可以用args一下处理掉，比如我会遇到在同一个timeslice里对矩阵切片，用args可以少写一些变量以及返回值。

还有matplotlib画图的时候可以subplot，seaborn里面可以设置图片的风格比如ggplot。

还有python里的any()，all()这些和数学上比较接近的东西，逻辑与或非，在写if判断的时候可以方便很多。

tqdm这个模块可以给循环加入计时器的功能。类似这种。



面向对象oop的内容，有些重复的模块可以一层一层向上封装，但做多因子模型这个事情，更像是一个pipline，上一层的东西输出给下一层使用，我觉得并不很面向对象啦，但是有些工具封装起来直接用就会很方便。我的coding还停留在一个.py写main function，另一个.py 一个一个测这些写完的function。希望以后可以进步呢！

还有就是if continue, if break 和 try catch...

很pythonic的东西也没怎么用，比如高级的for else，装饰器，闭包之类的魔法方法。我觉得思想就是，能用python避免循环的地方，千万不要写循环，万不得已写循环，2层循环封顶，怎么简洁怎么来。

有的东西学的时候没什么感觉，但是如果能真正被用到才算是学会了吧。

想一想，我觉得自己目前还是喜欢这部分的，可能要是赚钱就是另外一回事情了，要是coding能力强一点就好了，另一条路就是去做data scientist，好像怎么样都逃不过写代码（哈哈哈笑～），因为觉得有一点像research，然后确实有数据集可以验证，而且可能可以遇到一些coding大神。如果一起工作的人和工作的事情，都是自己喜欢的话，真是会很幸运呢，才会觉得生活过得很有意义吧。hhhhh

时至今日，觉得非常庆幸，自己遇到了研究生时候非常好的一群朋友们和老师，帮助我成长了很多很多很多。

2020.0202