短期价量多因子选股

大框架：

1. 做完接口（用10个因子），把框架弄出来
2. 建立测试因子的框架！！
3. 修改barra风格因子的计算方法，并重新入库（因子的格式要做相应的改变）
4. 跑价量因子的程序，存入到数据库当中去
5. 建立一个回测平台！（ZipLine） https://github.com/liudengfeng/zipline

stock\_market\_data有点问题，退市的股票数据没有！！

1、先把原始的数据跑起来

2、建立简单的回测体系（回测不需要每天进行，单独用另外一个模块，当所有的日期完成之后，再进行回测） ——分年度的表现

3、将所有因子计算完成

更加细节的东西：写的不错！

参考这篇：https://wallstreetcn.com/articles/3053251

alpha\_return: t期的因子暴露，t+d期的股票收益率；在进行测算的时候，可以这么定义，比如求IC值的显著性等等；但真实的情况是在t期，并不能够得到这个alpha\_return，因为t+d的股票收益率并不可知。因此在做策略的时候需要注意这个问题！

因子137表现不错！！测的好的话，就存入数据库好了。他们对它进行了一定的改进！

利用算出的原始因子，利用barra的风格因子和行业因子做正交，获取残差，然后利用公司的因子测试平台测该残差，看各项指标？

1）问题：先正交，再去极值和标准化？ 还是先去极值和标准化，再正交？？

在找alpha的过程当中，除了构建简单的alpha因子（比如已经有相关的构建方法），还可以对之前的构建方法进行一定程度的改进或者啥，以便更加适合市场的情形。

在进行单因子分析的时候，还要记得进行分段分析！

本次报告将开启对国泰君安金工刘富兵团队报告《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》的单因子回测。该报告集中测试了股票市场191个短周期的alpha因子，与此前市面上WorldQuant101和广发证券“技术因子海量挖掘”等报告一样，是一篇值得去实践的研报。目前中低频的alpha因子衰退严重，变成了一场追逐风格变换的游戏，而短周期因子的alpha仍有很大机会，因为其容量不大，没有大资金介入。本系列报告将遵循以下模板，每次汇报3个因子。

IR = IC \* Breadth \*\* 0.05，要么一次性的提高IC的值，要么提高交易的次数。国泰君安报告所说，增加因子的数量也会使得整个策略的回测效果更好（这个可以自己以后建立一个因子库，即表现不错的alpha因子放在一起）

把基本的价量因子的东西实现!

价量因子的框架：

1）每日计算191个alpha因子，并将原始alpha因子保存到数据库（factor\_exposure\_original）当中去（数据放成191列）；（每天要让所有的因子值都有数据，如果某个股票的因子值没有数据，但stocklist里面有；则利用barra的data-replacement的方法）

一个日期得到N\*191一个dataframe，存入数据库 !

2）提取风格因子数据，同时利用当天计算的alpha因子（去极值、标准化）进行回归，获取单因子的残差截面alpha\_resid，存入factor\_exposure\_resid数据库，191列

3）（计算方法和第2条一样）利用因子暴露的残差、风格因子和行业因子的数据，以及股票收益率数据（periods=[1, 2, 3, 4, 5]），计算alpha因子收益率。将单因子收益率（alpha\_rtn\_single）存入到数据库（factor\_return）当中去

Date period 191列，因子名横着放

4）利用单因子收益率数据序列，滚动（比如利用前2年的时间滚动计算？）计算所有因子的年化收益均值和IR均值，画图，类似于图9到图18，并看显著性

5）计算单因子收益率的相关系数，如果相关系数大于某个阀值，剔除因子收益率IR较低的因子，保留IR较高的因子

可能方法：

如何做到多因子模型不过拟合？

1）每一期进行分析； 比如利用LASSO回归，每一期选择了过去2年中贡献最大的因子，以及因子之间不存在共线性的问题。这样做的最大问题是：收益率具有可比性吗？？（因为不同时期的因子数量是不一致的！）

2）简单粗暴点，在某一期算下VIF就好！（或者进行一次逐步回归？Lasso回归）这样会产生的问题：严重利用到了事后的信息——不符合真实的策略！？！

6）利用alpha\_resid，将所有因子作为自变量，并加上行业、风格因子暴露。类似于第（3）步的存储格式，数据库名为factor\_return\_multi

Date Period 191列

7）step3

8）（最后计算）计算各期模型预测系数，以及计算模型系数T检验结果

VIF只是用来判断，无法剔除。逐步回归的话，效果不好（RiceQuant的那篇文章所说）——自己先把报告的想法实现，之后再不断修改好了！

问题：

2）这东西能做吗？ 很难保证每次回归的因子数量是一致的？ 如果不一致，收益率还能比较吗？ 可不可以比如说，过去1年都存在因子收益率的因子，作为预测因子

像这种处理大量的因子该咋办呢？？ 有没有比较好的方法？

不剔除把，也有问题。存在共线性问题咋办？那么因子收益率的估计不就不准确了嘛？

它这种方法怎么能保证没有共线性呢？不应该是要计算VIF嘛？

先不考虑因子数量不一样的情况，即认为他们是可比的！ 但要用VIF进行处理，会更好一点

3）主成分分析得出的结果就一个吗？如果每次得出10个或者20个主成分，那么不同期的主成分还是一样的嘛？因为权重不一样啊！

1）每天进行计算（所有的步骤，所有参数为date就好）；并将191个alpha因子存入数据库中？这个多因子存入到数据库中的一个表格中去？（会不会爆？！）

2）公式中的？是什么意思？

4）价格是复权之后的嘛？要加入复权因子嘛？

5）(CLOSE-DELAY(CLOSE,6))/DELAY(CLOSE,6)\*100，这个为什么要乘以100？？？

6）因子计算完之后需要进行去极值和标准化嘛？

7）如何让计算的因子和生成表因子（191个）同时匹配起来——即不管怎样，都一样！（在enum里面设置一下？）建表的时候，也不用自己输入191个字符的那种！（看有没有比较智能的方法？）

8）alpha6去极值和标准化之后，全部变为-1， 之后看看如何进行处理？

9）部门是否有回测平台？

10）很重要的问题：！！ 1）阀值是多少？？ 2）要不用相关系数占比最大的10%的因子部分

每次根据过去的因子相关性，剔除一些相关性较大的因子，即里面的IR较低的因子；所以，factor\_return\_multi，因子值就不应该是not null，因为有些因子可能不会被纳入到 多元回归模型中，因为会存在共线性的问题（相关性太大）

那又会有个问题：某个因子，一直和其他的因子共线性，在某一个没有共线性了，那在利用过去250日收益率均值作为预测值的时候，不就会出现问题了嘛？

如果每一天，用来回归的alpha因子数量不一致，其不同期的alpha因子收益率还有可比性吗？

11）每一天对单因子进行分析，以便获得最优的因子？

12）data-replacement：有必要用吗？如果真的没有因子值，要不直接drop掉；因为在构建策略的时候，再买入这样的股票，其实也没啥意义!

说明：

先把功能实现，之后优化代码啥的，比如把代码拼接在一起！（全写成一行的代码即可）

0、去除ST（在提取数据的时候，就剔除！！）以及涨跌停的股票（在取数据的时候，就剔除！）——get\_stocklist函数

1、一开始的时候，将所有的数据提取出来放在内存当中去（最后整合的时候考虑）

比如价格的数据（）

2、如何把tdays、engine等放在一个地方，统一管理！比如Vars.tdays， Vars.engine就好，就不用设成参数了。

3、国泰君安的公式有些写的很复杂，比如本来是收益率，硬要写成这样：(CLOSE-DELAY(CLOSE,6))/DELAY(CLOSE,6)\*100

4、将get\_alpha里面的重复代码删除掉！比如loc[date]、dropna等

5、计算因子的时候，刚上市没多久的股票以及停牌的股票要剔除；但st股、涨跌停的股票要计算股票的因子暴露，以及利用这只股票的数据计算因子收益率（比如说小市值全部涨停了；其实就反应了小市值这个因子的收益；没必要剔除。但停牌为啥要剔除呢？因为停牌得到的收益率为0；但真实的收益率并不一定为0，那这样就会产生问题）。在做策略的适合，则要考虑ST股票以及涨跌停的股票，不能买入。当然如果使用特殊的策略，比如专门买ST类型的股票，当然没问题。

6、将update的函数全部合并起来！ 因为很多重复的其实！

7、alpha6 就先不放入模型当中去，因为结果很奇怪

8、它这种方法计算的因子收益率不会有问题吗？ 要不要借鉴alphalens的方法呢？

9、在计算VIF的时候，一定要非常长的数据，因为变量的个数为191，所以至少要252个交易日的数据。即min\_period=252

10、先不考虑如下的东西：

'''

下面的先不考虑

# 这里每期的因子数量可能会不一样！ 要进行共线性的分析 主成分分析？优于逐步回归

# Select alpha factors

# If VIF bigger than 10, means excessive collinearity

def get\_vif(df):

vif = pd.Series([variance\_inflation\_factor(df.values, i) for i in range(df.shape[1])],

index=df.columns)

return vif

vif = data.groupby(level='period').apply(get\_vif)

# vif = IR 为了初始的实验

#calculate multi factor return and save significant factors(Attention: not all alpha factors)

# 所以对表格进行修改，不能是not null的情况，因为并不是所有的表

'''

趋势，风格和择时，要用到趋势！！