1. **Python与金融数据分析期末作业**[**¶**](http://localhost:8888/notebooks/Desktop/%E5%AD%A6%E6%A0%A1/%E9%87%91%E8%9E%8D%E6%95%B0%E6%8D%AE%E5%88%86%E6%9E%90/Python%E4%B8%8E%E9%87%91%E8%9E%8D%E6%95%B0%E6%8D%AE%E5%88%86%E6%9E%90%E6%9C%9F%E6%9C%AB%E4%BD%9C%E4%B8%9A/homework.ipynb#Python与金融数据分析期末作业)

**数据说明**

price.csv 记录了国内A股市场自2010-2018年中每月末的股票价格数据。  
barra.csv 记录A股每个月末在barra 风格因子上的z-score。  
citics.csv 记录A股中信行业分类。

**作业目标**

通过构建因子多空组合并计算多空组合净值曲线，分析选股因子的有效性，具体要求如下：

每月末将全部A股根据 barra.csv中动量因子（MOMENTUM）由高到低排序，平均分为五组，第一组作为因子的多头组合，第五组作为因子的空头组合，等权配置，每月调仓，分别计算两个组合自2010年以来的每月净值收益率。将多头组合与空头组合每月净值收益率之差进行累计，生成并绘制市值因子的多空组合净值曲线。

构建**行业中性**的动量因子多空组合，即在每月末每个中信一级行业板块中，将全体A股按Momentum因子值由高到低排序，平均分为五组，选出行业内的多头与空头组合股票组，再合并成为全市场的多头与空头组合（注意多头组合与空头组合的行业分布相同），等权配置，每月调仓，分别计算两个组合自2010年以来的每月净值收益率。将多头组合与空头组合每月净值收益率之差进行累计，生成并绘制市值因子在行业中性条件下的多空组合净值曲线。

构建投资约束下的**行业中性**的动量因子非银多空组合，即在剔除银行业以及股票数小于60的行业后，在剩余中信一级行业板块中，将全体A股按Momentum因子值由高到低排序，平均分为十组，形成行业内的多头与空头组合，再合并成为全市场的多头与空头组合（注意多头组合与空头组合的行业分布相同），等权配置，每季度末调仓，分别计算两个组合自2010年以来的每月净值收益率。将多头组合与空头组合每月净值收益率之差进行累计，生成并绘制市值因子在行业中性条件下的多空组合净值曲线。

**小提示**

利用 pandas 中的 groupby() 与 qcut() 方法。  
注意应使用股票的复权价格计算多空组合的净值收益率。  
citics.csv的股票代码格式与另外两张表不同，需要转换。  
遇到缺失值可直接删除处理。

**提交要求**

各位同学请在jupyter notebook中完成编码、分析和结果呈现，并适当以Markdown的方式加入部分文字说明、理解分析以及代码注释，将最终的完成版通过(File->Download) 转换成三种不同的格式：(.ipynb),(.html),(.py),分别命名为： 学号\_姓名.ipyn , 学号\_姓名.html , 学号\_姓名.py (例如 SA00000\_张三.ipyn, SA00000\_张三.html, SA00000\_张三.py), 并统一集中提交至一位班级代表处。烦请班级代表将所有同学的ipyn,py,html分类好后对应打包成三个zip文件，命名为ipyn.zip,py.zip,html.zip，于**12月27日前**提交至我的邮箱，希望大家从完成作业的过程中，能有所收获，谢谢大家。

# 附录

复权价格：

股票行情有除权与复权，在计算股票涨跌幅时采用复权价格，这是经常要用到的。系统计算分为以下步骤：

1、每次除权行为日，在登记日计算除权价；

除权价=（除权前一日收盘价+配股价X配股比率－每股派息）/（1+配股比率+送股比率）

2、用登记日的收盘价除以除权价得到单次除权因子；

除权因子=收盘价/除权价

3、将公司上市以来的除权因子连乘积，得到对应每一交易日的除权因子；

4、[向后复权](https://www.baidu.com/s?wd=%E5%90%91%E5%90%8E%E5%A4%8D%E6%9D%83&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)价=股票收盘价（实际交易价）\*每一交易日的除权因子；

5、复权涨幅（不特别说明，涨幅均指复权涨幅）=复权价/前一日复权价-1。

#grouped\_last\_5=washed\_data.groupby('日期').apply(lambda x: x.nsmallest(5,'Momentum',keep='first'))

#grouped\_last\_5

#grouped\_fist\_5=washed\_data.groupby('日期').apply(lambda x: x.nlargest(int(103/5),'Momentum',keep='first'))

#grouped\_fist\_5['多方净值收益率']

#net\_yield.apply(lambda x:x\*100).plot(figsize=(8,6),secondary\_y='close',grid=True)

#temp\_sort=pd.qcut(yield\_data['Momentum'],q=[0,1/5,(1-1/5),1],labels=['min','mid','max'])