华东理工大学 2023 –2024 学年第 2 学期

《大数据与金融计算》实验报告

实验名称　中国股市 FF 三因子资产定价模型的实证检验

实验目的/要求
1、掌握三因子模型参数估计方法和时间序列检验方法（单资产和多资产） 2、掌握三因子模型的横截面检验方法（排序法和 Fama-Macbeth 回归）
实验内容
1. 认真阅读三篇文献资料（自己也可以下载相关文献进行阅读），了解资产定价模型的相关研究，重点关注其中的研究思路和方法。 2. 任意选取七个行业指数，从锐思数据库或 tushare 数据库，下载这些指数的周度数据和三因子的周度数据，估计 FF 三因子模型参数，并逐个行业检验其是否满足 FF 三因子模型；再对七个行业进行 FF 三因子模型的多资产检验。（月度数据不少 10 年，周度数据不少于 5 年） 3. 从锐思数据库或 tushare 数据库，提取中国股市 2001-2022 年所有股票的月收益率，月流通市值数据，月市盈率数据，观察和研究课本 P52 页表格 3-6 的计算方法，计算先按市值再按盈利价格比（市盈率倒数）序贯排序的资产组合月平均收益率（注意：变量排序分成 5 组，不需要分成 10 组），并对结果进行讨论。 4. 从锐思数据库或 tushare 数据库，提取中国股市 2001-2022 年，所有股票的月收益率，月流通市值数据，月市盈率数据，月 CAPM 风险因子 Beta 数据，请利用 Fama-MacBech 回归检验市值、盈利价格比（市盈率倒数）、CAPM 风险因子 Beta 对收益率的解释性。
实验总结 **请提供对本次实验结果的讨论分析，以及实验的心得和体会。包括对知识点的掌握，算法的理解，以及对理论课程和实验课程改进的建议。（不少于 500 字）**

本文对于资产定价进行了实证研究。文中进行了三个实验，第一个实验是对FamaFrench 三因子模型对上证不同行业组合收益率的解释力进行研究，选取上证七个行业的周数据，其中包含周收益率、周无风险收益率、周三因子数据等。单资产线性回归的研究表明除了上证消费之外，其他所有行业都可以被三因子模型中的部分因子解释。对于多资产联合检验的结果表明，三个因子具有一定的解释力，但这种解释能力在一定程度上是脆弱的，对显著性水平的选择较为敏感。

第二个和第三个实验中选取中国 A 股 2001 年到 2022 年 A 股主板正常上市股票数据的月收益率，月流通市值数据，月市盈率数据，研究β，市值和盈利价格比对于 A 股市场的解释能力。通过序贯排序和 FamaMacBeth 回归发现三个因子都有一定的解释能力。其中β的系数是负的，这与理论上所认为的股票超额收益率和系统性风险成正比是不同的。log(Size)市值因子在不同因子组合的情况都显著且为负，说明 A 股市场存在明显的市值效应，小市值股票拥有更高的超额收益。EP 盈利价格比因子在只有自己本身和截距项进行回归时系数为负，和其他因子组合时系数为正，这表明在没有控制其他因子影响的情况下，盈利价格比对于收益率吧是负作用。然而，与其他因子组合回归时系数转为正，这表明 EP 因子在特定因子组合框架下起到正面作用，即高盈利价格比的股票在同时考虑其他因子影响时显示出更高的超额收益。

经过这次实验，我更好地掌握了因子模型，它基于多个因子来解释资产价格的变化，这些因子可以是宏观经济指标、市场行为、公司基本面等，因子的选择也一直是研究的热点。这次实验也让我掌握了序贯排序法以及 FamaMacBeth 回归，序贯排序是一个双变量排序方法，当需要对某两个因素对于另一个因素的影响时可以使用序贯排序直观的看出影响。FamaMacBeth 是一种有效的回归方法，其将 T 个时间序列数据看作 T 个独立样本，在每个时间点进行截面回归，最后进行平均以及统计量构造，是一种有效的面板数据研究方法。

教师批阅：　　实验成绩： 教师签名:　　日期：

实验报告正文：

（每次实验报告均为一篇小论文，因此，统一按照学术论文的要求完成实验报告正文，应包括：题目、摘要、文献综述、模型和方法、结果和讨论、参考文献、附录，具体格式如下：

中国股市三因子定价模型实证检验

摘要：从 CAPM 模型、三因子模型到五因子模型等，资产定价一直是金融实证研究领域的热门研究议题，过去有大量学者针对股票市场超额收益的决定因子和模型进行研究，本文选取中国 A 股 2001 年到 2022 年 A 股主板正常上市股票数据，对三因子模型在中国 A 股的解释力进行研究。首先对于行业进行研究，选取上证七个不同行业检验三因子模型在不同行业的解释力，随后将七个行业进行联合检验。接着使用序贯排序法直观地展示股票超额收益随着市值和盈利价格比因子变化的规律。最后采取 FamaMacBeth 回归对于 β，市值和盈利价格比对于超额收益率的解释能力进行进一步研究。结果表明 FamaFrench 三因子模型对于不同行业具有一定的解释能力，同时 β，市值和盈利价格比对于 A 股市场具有一定解释力。

1 文献综述

资产定价一直是金融研究领域的热门议题。1952 年 Markowitz 发表的 Portfolio Selection 一文首次从数学角度探讨了如何寻找最优资产组合的问题[1]。随后研究者们在此方法的基础上不断深入优化。1964 年 Sharpe, William F 提出了著名的 CAPM 模型，其中指出在满足 CAPM 假设的情况下，资产的期望超额收益仅市场因素决定，使用了线性模型描述了资产超额收益和系统性风险的关系[2]。然而随着实证资产定价研究的发展，也有学者发现在实证结果中出现了许多违背 CAPM 模型的市场异象，Ball R 在 1978 年的研究中提出了不满足 CAPM 模型的系统性异象[3]。Banz 等在 1981 年的研究中发现小市值股票相对于大市值股票有更高的风险调整后收益[4]，这一发现后来被称为“规模效应”。Chan L K C 等在 1991 年对于日本股票市场的实证研究中发现了股票的平均收益率与其账面市值比存在显著的相关性[5]。这些研究对资本资产定价模型提出了挑战，因为它表明除了市场风险之外，还有其他因素能够解释股票收益的差异。一部分学者将各种可能影响股票收益率的因子加入模型中试图构造更为合理的定价模型，也被称为多因子定价模型。在众多模型中，最为出名的是 FamaFrench 三因子模型。Fama, E.F. and French. K.R.首先在 1992 年的研究中否定了 CAPM 模型所提出的股票超额收益完全由市场因素决定的结果，又提出了账面市值比和市值两个因素对股票收益率有较好的解释能力[6]。随后在 1993 年的研究中正式提出了三因子模型[7]，其在论文中构造了市场因子，规模溢价和账面市值比这三个因子，通过对美国股票市场的实证研究发现这三个因子的线性回归模型可以解释不同股票的收益率。三因子模型的提出对金融界产生了深远的影响，

它为资产定价这一研究议题提供了新视角，此后越来越多的有解释力的因子模型被发现和提出。Carhart 在 1997 年的研究中提出了四因子模型，在市场因子，规模因子和账面市值比因子的基础上添加了动量因子，动量因子来源于动量效应，即过去表现良好的股票在未来一段时间内往往会继续表现良好，而过去表现不佳的股票往往会继续表现不佳。Carhart 通过将股票数据按过去的累积收益分组构成组合后计算动量因子，通过线性回归证明了该因子的解释力[8]。Fama, E. F., & French, K. R.在 2015 的研究中又对三因子模型进行了改良，增加了盈利能力因子和投资因子以解释股票收益率的横截面差异，盈利能力因子来源于高盈利能力的公司股票预期回报高于低盈利能力的公司股票，投资因子基于公司的投资行为，高投资的公司股票预期回报低于低投资的公司股票[9]。

然而上述研究都是基于美国市场和日本市场等发达国家成熟市场，和中国市场的情况并不一致，为此需要针对中国市场的因子模型的有效性实证研究。张宏亮,赵雅娜在 2014 的研究中选取了 2001-2011 年 A 股上市的股票并剔除 ST，数据缺失和金融行业的股票作为研究数据进行研究，选取了 β 系数、规模、账面市值比、市盈率倒数、财务杠杆这 5 个因子，研究结果表明这五个因子对于 A 股市场都有解释作用，其中规模、账面市值比、市盈率倒数是负作用，而其他两个因子是正作用[10]。赵胜民,闫红蕾,张凯在 2016 年的研究中利用 1995 年至 2014 年股票市场组合收益率和 1 年期国债利率数据进行研究吗，采用三种不同的分组方法对三因子模型和五因子模型在 A 股市场的解释能力进行比较，实证结果指出三因子模型比五因子模型更适合 A 股市场，市值效应和价值效应明显[11]。李志冰,杨光艺,冯永昌,等对五因子模型在 A 股市场中的解释力进行研究，重点关注 2005 年之前股改之前数据和 2005 年股改之后的结果差异，证明了五因子模型具有较强的解释力，并且提出股权分置改革后 A 股市场更具有研究价值，股改前的数据会对实证研究造成扰动[12]。Jianan Liu and Robert F. Stambaugh and Yu Yuan.在 2019 年的研究中基于中国股市构建了适合中国市场的三因子模型 CH-3，并探讨了这些因子在中国股市中的表现和解释力，CH-3 三因子模型包括市场因子，规模因子以及基于盈利价格比的价值因子，证明了该模型在中国市场有很好的解释力[13]。Jennifer N. Carpenter and Fangzhou Lu and Robert F. Whitelaw.在 2020 年的研究中指出中国股市的股票价格已经能够提供关于未来利润的信息，表明市场正在整合信息并为管理者提供有用的信号，这为定价模型在股市的研究的有效性提供了又一个基础[14]。

2 模型和方法

2.1 多因子模型

多因子模型是用于评估证券投资组合风险和回报的一种线性模型。它基于多个因子来解释资产价格的变化，这些因子可以是宏观经济指标、市场行为、公司基本面等。在多因子模型中最为著名的是 FamaFrench 三因子模型。Fama 和 French 选取了三个因子

解释股票超额收益率，分别是

1) 市场风险溢酬因子 RMRF：

该因子和 CAPM 模型的市场因素 R_m-R_f 类似，是市场组合的超额收益，在实际中往往采用某一具有代表性的指数的超额收益替代。其代表了市场因素对于股票收益率的影响。

2) 市值因子 SMB：

市值的计算方法是先将股票数据进行 2×3 分组，即每年 6 月底将有效的股票数据按市值分为前 50%和后 50%，记为 S 和 B，在每一部分中又将股票数据按账面市值比以 30%，40%，30%的比例分为三组，记为 L，M，H。这样就得到了 SL, SM, SH, BL, BM, BH 这六组股票，随后可以通过公式 1 计算 SMB。

$$SMB=(SL+SM+SH)/3-(BL+BM+BH)/3 \quad (1)$$

式 3 中，SL 等就是对应分组投资组合的收益率。

市值因子计算方式是有小市值公司收益率减去大市值公司收益率，反应了市值效应。

3) 账面市值比因子 HML：

在上述股票分组的基础上，账面市值比的计算如公式 2 所示：

$$HML=(BH+SH)/2-(BL+SL)/2 \quad (2)$$

式 2 中，BH 等就是对应分组投资组合的收益率。

账面市值比因子反应了反映了投资者对公司未来盈利能力的预期。一个较低的 BM 比率通常表明市场预期公司未来的盈利能力和成长性较好，因此愿意为其支付较高的价格。

上述是 FamaFrench 三因子模型的介绍。在中国市场中，Jianan Liu and Robert F. Stambaugh and Yu Yuan.构建了适合中国市场的三因子模型 CH-3。在 CH-3 三因子模型中，首先剔除壳公司(市值位于后 30%)的股票，使用市场因子 mkt，市值因子 smb，以及由盈利价格比构建的成长因子 vmg 作为三个因子，能够较好的解释中国股市收益率。

2.2 GRS 检验

在对三因子模型的研究中采用多资产联合回归，即对每一个资产应用公式 3 形式的线性回归：

$$R_{it}=\alpha_i+\beta_i' f_t+\varepsilon_{it},\ t=1,2,\ldots,T, i=1,2,\ldots,N \quad (3)$$

如果多资产满足三因子模型，则多个资产的截距项α_i应该同时为 0，将此条件作为原假设进行假设检验。GRS 检验就是其中一种检验方法，检验所需要计算的统计量为：

$$GRS=\frac{T-N-K}{N}[1+E_T(f_t)'\hat{\Omega}^{-1}E_T(f_t)]^{-1}\hat{\alpha}\hat{\Sigma}^{-1}\hat{\alpha}\sim F(N,T-N-K) \text{ under } H_0. \quad (4)$$

式 4 中，T 为时间序列长度，N 为资产个数，K 为因子数，$E_T(f_t)'$是三因子数据的

均值，$\hat{\Omega}^{-1}$是三因子数据减去均值后与自己转置矩阵相乘后的均值，$\hat{\alpha}$是线性回归得到系数矩阵的截距向量。$\hat{\Sigma}^{-1}$是残差协方差矩阵的逆矩阵的均值。

GRS 统计量满足自由度为(N, T-N-K)的 F 分布，可以通过统计量判断原假设是否成立。

2.3 FamaMacBeth 回归

FamaMacBeth 回归是金融实证研究中常用的方法，由 Fama 和 MacBeth 在 1973 年的研究中提出[15]。在实证资产定价的研究，其常常被用于检验因子对于收益率的解释力。其核心步骤为：

1) t 时刻个体资产的横截面回归：

针对每个时间点 t，使用一组共同的风险因子（如市场因子、规模因子、账面市值比因子等）对资产的收益率进行横截面回归分析，以每个因子的系数。得到每个时刻各因子的截面回归系数。

2) 构造系数和统计量：

通过第一步得到各个因子的 t 个截面回归系数，对于这些因子取平均值，接着可以构造统计量对系数进行显著性检验：

$$t = \frac{\overline{\gamma_i}}{s_{i/\sqrt{T}}} \longrightarrow N(0,1) \tag{5}$$

其中，$\overline{\gamma_i} = \frac{1}{T}\sum_{t=1}^{T} \hat{\gamma}_{it}$ ，$s_i = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T} (\hat{\gamma}_{it} - \overline{\gamma_i})^2}$ 。

FamaMacBeth 回归将每个时间点的数据作为一个独立样本更容易得到回归的残差，其将多个时间点的回归数据取平均后，可以通过对每个时间点的方差平方和除以时间长度 T×(T-1)得到平均的误差，使得 FamaMacBeth 回归可以非常容易地进行显著性检验。

2.4 所使用的 python 代码介绍

为了实现对于因子模型的实证研究需要用到以下 python 代码。

1) 线性回归：

线性回归代码

```
1:  import statsmodels.api as sm
2:  x = sm.add_constant(x)
3:  model = sm.OLS(y,x)
4:  fitresult = model.fit()
5:  print(fitresult.summary())
```

通过上述 statsmodels.api 中的线性回归方法就可以实现资产收益率对于因子的线性回归。

2) 显著性检验求 P 值：

显著性检验求 P 值

```
from scipy.stats import f
pvalue = 1-f.cdf(GRS,N,T-N-K)
print(pvalue)
```

上述代码展示的是 GRS 检验中通过 F 分布的概率分布函数求 P 值的过程，通过 scipy.stats 可以获取不同概率的对象，其中的 cdf 方法只要传入值和自由度就可以获得对应的概率，使得显著性检验非常方便。

3) 对于日期的处理：

日期格式转换

```
stock['ym'] = stock['date'].dt.strftime("%Y%m")
```

上述代码展示的是日期格式的转换，在研究中由于一部分数据选取的是月数据，而原始数据中日期为年月日格式，所以需要对于日期格式进行转换，转换为年月格式。

3 结果与讨论

3.1 不同行业三因子实证检验

本研究旨在检验不同行业组合的超额收益率是否满足三因子模型，行业组合数据选取 2017 年 1 月 1 日到 2022 年 12 月 29 日上证 7 个行业的周数据，三因子数据选取对应日期的 RMRf 市场溢价因子，SMB 市值因子和 HML 账面市值比因子。在研究中使用行业组合的超额收益作为因变量，三因子数据以及截距项作为自变量进行单资产线性回归，回归结果如表 1 所示：

表 1　上证 7 行业三因子模型检验结果

行业	Const	RMRF	SMB	HML	R-squared
上证信息	0.0012	1.0543***	0.5750***	-0.9089***	0.626
上证公用	-0.0005	0.5845***	0.1556***	0.3080***	0.429
上证医药	0.0002	0.8829***	-0.2636***	-0.8560***	0.613
上证可选	-0.0003	1.0626***	-0.2415***	-0.3923***	0.758
上证工业	-0.0007	1.0408***	-0.1902***	0.0619	0.748
上证材料	0.0013	1.1793***	-0.0087	0.2214***	0.623
上证消费	0.0023**	0.9806***	-0.4620***	-0.8074***	0.696

注：Const 是常数项，rmrf，smb，hml 分别是 rmrf 市场因子，smb 市值因子，hml 账面市值比因子的系数。*表示在 10%水平下拒绝原假设，**表示表示在 5%水平下拒绝原假设，***表示表示在 1%水平下拒绝原假设。

从表 1 中可以看出对于上证信息行业，截距项系数不显著，而三个因子皆拒绝原假设，且 R-squared 为 0.626，表明三因子模型在上证信息行业中具有解释力。在上证公用行业中结果和上证信息一样，只有 R-squared 有所下降，说明三个因子都有解释力，但是拟合优度不如上证信息行业。在上证医药行业中结果和上证信息基本一致，三因子模型具有较强的解释力。在上证可选中显著性检验结果一致，拟合优度也提升很多，说明三因子模型解释力更为强大。在上证工业中 HML 的系数不显著，无法解释该行业组合的收益率，但是在其他两个因子的解释下拟合优度达到了 0.748，具有较好的解释能力。在上证材料行业中，SMB 因子无法解释，其他两个因子具有较好的解释能力。在上证消费中，截距项显著，说明三因子模型无法完全解释上证消费行业收益率。

接着对于这多个行业进行联合检验，使用 GRS 检验，结果如表 2 所示：

表 2　上证 7 行业三因子模型联合检验结果

行业	GRS 统计量	P 值
上证 7 行业	1.737	0.071*

注：*表示在 10%水平下拒绝原假设，**表示表示在 5%水平下拒绝原假设，***表示表示在 1%水平下拒绝原假设。

如表 2 所示，在 7 行业联合检验中 GRS 统计量的 P 值为 0.071，在 10%显著性水平下可以拒绝原假设，但是在 5%显著性水平下无法拒绝。在更为严格的显著性水平接受原假设，表明收益率可以被三因子解释，但是在更为宽松的显著性水平下三因子模型无法解释。总的来说可以认为模型具有一定的解释力，但这种解释能力在一定程度上是脆弱的，对显著性水平的选择较为敏感。

3.2 排序法检验

排序法是金融实证研究中的常用研究方法。在本研究中采用序贯排序法，具体方法是将股票数据先按第一因子进行排序并且分成多组，然后再将每一组的数据按照第二因子进行排序再分成多组，通过表格形式可以明显地看出因变量随着两个因子变化的关系。

在本研究中对于收益率与市值和盈利价格比之间的关系的进行研究，选取上证指数 2001 年到 2022 年所有在主板正常上市(剔除 ST)的股票数据进行排序，先将股票数据按照市值从低到高排成五组，然后将每一组数据内部按照盈利价格比从低到高排成五组，计算对应组内投资组合的收益率，结果如表 3 所示。

表 3　排序检验结果

	EP1	EP2	EP3	EP4	EP5
SIZE1	0.02002	0.01606	0.01561	0.01565	0.02150
SIZE2	0.00791	0.00948	0.00864	0.00995	0.01317
SIZE3	0.00353	0.00499	0.00775	0.00867	0.01001
SIZE4	0.00403	0.00338	0.00558	0.00798	0.00859

SIZE5	0.00124	0.00315	0.00424	0.00718	0.00828

注：EP 从 1 到 5 增大，SIZE 从 1 到 5 增大。

从表 3 中可以看出，随着市值 size 逐渐增大，股票的收益率逐渐变小，这说明 A 股市场存在明显的市值效应，小市值股票的收益率高于大市值股票。而在盈利价格比的研究中，在 SIZE3，SIZE4 和 SIZE5 分组中存在明显的正向关系，即收益率随盈利价格比增大而增大，而在市值较小的 SIZE1 和 SIZE2 分组这种关系不明显。

3.3 Fama-MacBech 回归实证研究

在 3.2 中使用序贯排序法研究了 A 股市场股票收益率与市值与盈利价格比之间的关系，然而从排序法中只能看出收益率随着市值、盈利价格比变化的趋势，并不能明确具体的函数关系，因此本研究采用 FamaMacBeth 回归方法对于 A 股市场超额收益与市值、盈利价格比和系统性风险 β 之间的关系进行考察，FamaMacBeth 的原理在 2.3 中已经给出。研究中仍然选用 2001 年到 2022 年所有在主板正常上市(剔除 ST)的股票数据，研究结果如表 4 所示：

表 4　FamaMacBeth 回归结果

因子	(1)	(2)	(3)	(4)	(5)	(6)	(7)
截距项	0.014	0.098	0.011	0.099	0.083	0.084	0.086
	(6.065)	(7.648)	(5.011)	(7.686)	(8.013)	(8.013)	(8.159)
β	-0.003			-0.002	-0.003		-0.002
	(-5.958)			(-3.851)	(-7.767)		(-4.226)
log(Size)		-0.003		-0.003		-0.003	-0.003
		(-7.362)		(-7.308)		(-7.767)	(-7.713)
EP			-0.012		0.011	0.012	0.012
			(-1.649)		(1.811)	(1.812)	(1.874)

由表 4 中的 FamaMacBeth 回归可以看出，不论是什么因子组合，截距项、β 、log(Size)和 EP 都有解释力。截距项始终有解释力，说明除了这三个因子之外仍有其他因子可以解释超额收益。对于 β 来说不论什么因子组合都具有较强的解释力，但是和 CAPM 模型和 FamaFrench 三因子模型相反，β 的系数是负的，这与理论上所认为的股票超额收益率和系统性风险成正比是不同的。log(Size)市值因子在不同因子组合的情况都为负，且系数值非常稳定，说明 A 股市场存在明显的市值效应，小市值股票拥有更高的超额收益。EP 盈利价格比因子在只有自己本身和截距项进行回归时系数为负，和其他因子组合时系数为正，这表明在没有控制其他因子影响的情况下，较高的盈利价格比似乎与较低的超额收益相关联。然而，与其他因子组合回归时系数转为正，这表明 EP 因子在特定因子组合框架下起到正面作用，即高盈利价格比的股票在同时考虑其他因子影响时显示出更高的超额收益。

4 参考文献

[1] Markowitz, H.M. (March 1952)."Portfolio Selection".The Journal of Finance.7(1): 77–91.

[2] Sharpe, William F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk, Journal of Finance, 19 (3), 425-442

[3] Ball R .Anomalies in relationships between securities' yields and yield-surrogates[J].Journal of Financial Economics, 1978, 6(2):103-126.DOI:10.1016/0304-405X(78)90026-0.

[4] Banz, R.W. (1981). The relationship between return and market value of common stocks.Journal of Financial Economics, Vol. 9(1), 3 – 18.

[5] Chan L K C , Hamao Y , Lakonishok J .Fundamentals and Stock Returns in Japan[J].Center on Japanese Economy and Business, Graduate School of Business, Columbia University, 1991(5).

[6] Fama, E.F. and French. K.R. (1992) The Cross-Section of Expected Stock Returns. Journal of Finance, 47, 427-465.

[7] Fama, E. F., and Kenneth, R. (1993). French, 1993, Common risk factors in the returns on stocks and bonds. Journal of Financial Economics, 33(1), 3-56.

[8] Carhart M M .On Persistence in Mutual Fund Performance[J].Social Science Electronic Publishing, 1997, 52(1):57-82.

[9] Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. Journal of financial economics, 116(1), 1-22.

[10] 张宏亮, 赵雅娜. FF 三因子模型风险因子的有效性检验——基于 2001-2011 年我国资本市场数据[J]. 财会通讯, 2014(30): 113-116+124+129.

[11] 赵胜民, 闫红蕾, 张凯. Fama-French 五因子模型比三因子模型更胜一筹吗——来自中国 A 股市场的经验证据[J]. 南开经济研究, 2016(02):41-59.

[12] 李志冰, 杨光艺, 冯永昌,等. Fama-French 五因子模型在中国股票市场的实证检验[J]. 金融研究, 2017(06):191-206.

[13] Jianan Liu and Robert F. Stambaugh and Yu Yuan. Size and value in China[J]. Journal of Financial Economics, 2019, 134(1) : 48-69.

[14] Jennifer N. Carpenter and Fangzhou Lu and Robert F. Whitelaw. The real value of China’s stock market[J]. Journal of Financial Economics, 2020, 139(3) : 679-696.

[15] Fama E F , Macbeth J D .Risk, Return, and Equilibrium: Empirical Tests[J].Journal of Political Economy, 1973, 81(3):607-636.

5 附录

```python
import numpy as np
import pandas as pd
import statsmodels.api as sm
from scipy.stats import f
from GRStest import grstest

data_factors = pd.read_excel("数据/三因子.xls",usecols=[2,6,7,8])
data_factors.columns = ['date','mkt','smb','hml']
data_factors['date'] = pd.to_datetime(data_factors['date'])



data_index = pd.read_excel("数据\指数周收益率.xls",usecols=[1,2,4])
data_index.columns = ['idxname','date','return']
data_index['date'] = pd.to_datetime(data_index['date'])



idxname = np.unique(data_index['idxname'].values)
datas = []

for i in range(len(idxname)):
    data = data_index.loc[data_index['idxname']==idxname[i]]
    datas.append(data)


data_rf = pd.read_excel("数据\指数周收益率.xls",usecols=[2,5])
data_rf.dropna(inplace=True)
data_rf.columns = ['date','return']
data_rf['date'] = pd.to_datetime(data_rf['date'])


data_matrix = data_rf[['return','date']]

cols = ['return','date']

for i in range(len(datas)):
    data_matrix = pd.merge(left=data_matrix,right=datas[i][['date','return']],on = 'date',how='inner')

    cols.append(idxname[i])
    data_matrix.columns = cols


data_matrix.dropna(inplace=True)
data_matrix = pd.merge(left=data_matrix,right=data_factors,on='date')

x = data_matrix.loc[:,'mkt':]
x = sm.add_constant(x)
y = data_matrix['上证信息'] - data_matrix['return']
model = sm.OLS(y,x)
fitresult = model.fit()

ym = data_matrix[['上证信息',    '上证公用',    '上证医药',    '上证可选',    '上证工业',      '上证材料',    '上证消费']]
xm = x
grstest(xm,ym)


```

```python
import numpy as np
import pandas as pd
import pickle
import statsmodels.api as sm
root_name = "数据\RESSET_MRESSTK_"
#第一个数据
stock = pd.read_excel(root_name+"1.xls")
for i in range(2,11):
    file_name = root_name+str(i)+".xls"
    stock = pd.concat((stock,pd.read_excel(file_name)),axis=0)
stock.columns = ['code','date','price','amount','lamount','return','rf','pe']
#处理日期
stock['date'] = pd.to_datetime(stock['date'])
stock['ym'] = stock['date'].dt.strftime("%Y%m")
stock = stock[['code','ym','price','lamount','return','rf','pe']]
stock.dropna(inplace=True)
#市值
stock['size'] = np.log(stock['price'] * stock['lamount'])

#盈利价格比
stock['ep'] = 1 / stock['pe']
#超额收益
stock['R'] = stock['return'] - stock['rf']

stock = stock[['code','ym','size','ep','R']]
stock = stock.sort_values(by='code')
#拆分个股
stock_list = []
codes = np.unique(stock['code'])
codes = list(codes)
for code in codes:
    stk = stock.loc[stock['code']==code]
    stk = stk.sort_values(by='ym')
    stock_list.append(stk)
#计算平均市值
size_list = {}
for i in range(len(stock_list)):
    code = np.unique(stock_list[i]['code']).item()
    size_list[code] = np.mean(stock_list[i]['size'])
#排序
size_list = sorted(size_list.items(),key=lambda x:x[1],reverse=True)
#去除壳公司
codes = []
for i in range(int(len(size_list)*0.7)):
    codes.append(size_list[i][0])
used_stocks = []
for stock in stock_list:
    if np.unique(stock['code']).item() in codes:
        used_stocks.append(stock)
print(len(used_stocks))

f = open("stock_list.pkl",'wb')
pickle.dump(used_stocks,f)
f.close()

root_name = "数据\个股beta"
#第一个数据
b = pd.read_excel(root_name+"1.xls")
for i in range(2,11):
    file_name = root_name+str(i)+".xls"
    b = pd.concat((b,pd.read_excel(file_name)),axis=0)
b.columns = ['code','date','beta']
#处理日期
b['date'] = pd.to_datetime(b['date'])
b['ym'] = b['date'].dt.strftime("%Y%m")
b['ym'] = pd.to_datetime(b['ym'],format="%Y%m")
b= b[['code','ym','beta']]
b.dropna(inplace=True)
b = b.sort_values(by='code')
b_list = []
codes = np.unique(b['code'])
codes = list(codes)
for code in codes:
    stk = b.loc[b['code']==code]
    stk.sort_values(by='ym')
    stk = stk.loc[(stk['ym']>=pd.to_datetime('200101',format=("%Y%m"))) & (stk['ym']<=pd.to_datetime('202201',format=("%Y%m")))]
    stk['ym'] = stk['ym'].dt.strftime("%Y%m")
    b_list.append(stk)

f = open("b_list.pkl",'wb')
pickle.dump(b_list,f)
f.close()
with open('stock_list.pkl','rb') as f:
    stock_list = pickle.load(f)
with open("b_list.pkl",'rb') as f:
    b_list = pickle.load(f)

final_stocks = []

#组合股票
for i in range(len(stock_list)):
    cods = np.unique(stock_list[i]['code'])
    s = stock_list[i].loc[stock_list[i]['code']==cods[0]]
    b = b_list[i].loc[b_list[i]['code']==cods[0]]

    stock = pd.merge(left=stock_list[i],right=b_list[i][['ym','beta']],on='ym',how='inner')

    r_np = np.array(stock['R'])
    r_year = []
    for i in range(len(r_np)-12):
        r_year.append(np.mean(r_np[i:i+12]))
    stock['R'] = pd.Series(r_year)

    # stock['R'] = stock['R'].shift(1)
    stock.dropna(inplace=True)
    final_stocks.append(stock)
    print(stock)




f = open("final_list.pkl",'wb')
pickle.dump(final_stocks,f)
f.close()
```

```python
import numpy as np
import pandas as pd
from scipy.stats import f

#ym是所有资产矩阵，即因变量
#xm是自变量
def grstest(xm,ym):
    T = len(ym)#时间序列长度
    N = 10#资产数Y
    K = 3#因子数3
    #xm是(306,4)包含截距项的自变量矩阵，ym是因变量矩阵为(306,7)，表示7个行业
    #xmT*xm = (4,4)为相关系数矩阵
    #xmT*ym = (4,306) * (306,7) = (4,7)可以理解为xm和ym的相关性
    xmTxm = np.dot(np.transpose(xm),xm)
    xmTym = np.dot(np.transpose(xm),ym)
    #AB_hat = (4,4) * (4,7) = (4,7) 是线性回归估计的系数矩阵
    AB_hat = np.dot(np.linalg.inv(xmTxm),xmTym)
    #ALPHA是系数矩阵中的截距项
    ALPHA = AB_hat[0]

    #残差
    RESD = ym-np.dot(xm,AB_hat)
    #残差相关系数
    COV = np.dot(np.transpose(RESD),RESD)/T
    invCOV = np.linalg.inv(COV)
    #fs是剩下的三个变量
    fs = xm.loc[:,['mkt','smb','hml']]
    fs  =np.array(fs)
    #均值
    muhat = np.mean(fs,axis=0).reshape((3,1))
    fs = fs - np.mean(fs,axis=0)
    omeghat = np.dot(np.transpose(fs),fs)/T
    invOMG = np.linalg.inv(omeghat)
    xxx = np.dot(np.dot(np.transpose(muhat),invOMG),muhat)

    yyy = np.dot(np.dot(ALPHA,invCOV),np.transpose(ALPHA))


    GRS = (T-N-K)/N*(1+xxx[0][0])*yyy
    print(GRS)
    pvalue = 1-f.cdf(GRS,N,T-N-K)

    print(pvalue)
```

```python
import numpy as np
import pandas as pd
import pickle
import statsmodels.api as sm


with open('final_list.pkl','rb') as f:
    stock_list = pickle.load(f)
ym_nums = np.unique(stock_list[0]['ym'])
print(stock_list[0])


params = pd.DataFrame({"const":[],"size":[],"ep":[],"beta":[]})
for ym in ym_nums:
    #组合截面数据
    sym = stock_list[0].loc[stock_list[0]['ym']==ym]
    for i in range(1,len(stock_list)):
        s = stock_list[i].loc[stock_list[i]['ym']==ym]

        if s.empty:
            continue
        sym = pd.concat((sym,s),axis=0)
    #回归
    y = sym['R']

    x = sym[['size','ep','beta']]
    # x = sym[['size','ep']]
    # x = sym[['size','beta']]
    # x = sym[['ep','beta']]
    # x = sym[['size']]
    # x = sym[['ep']]
    # x = sym[['beta']]

    x = sm.add_constant(x)
    model = sm.OLS(y,x)
    fit_model = model.fit()
    param = fit_model.params
    p = pd.DataFrame({"const":[param['const']],
                      "size":[param['size']],
                      "ep":[param['ep']],
                      "beta":[param['beta']]})
    # p = pd.DataFrame({"const":[param['const']],
    #                   "ep":[param['ep']],
    #                   "beta":[param['beta']]})
    # p = pd.DataFrame({"const":[param['const']],
    #                   "size":[param['size']],
    #                   "beta":[param['beta']]})
    # p = pd.DataFrame({"const":[param['const']],
    #                   "size":[param['size']],
    #                   "ep":[param['ep']],
    #                   })
    # p = pd.DataFrame({"const":[param['const']],
    #                   "size":[param['size']],
    #                   })
    # p = pd.DataFrame({"const":[param['const']],
    #                   "ep":[param['ep']],
    #                   })
    # p = pd.DataFrame({"const":[param['const']],
    #                   "beta":[param['beta']]
    #                   })

    params = pd.concat((params,p),axis=0)


params.to_hdf("params.h5",key='df',mode='w')

```

```python
import numpy as np
import pandas as pd
import statsmodels.api as sm
from scipy.stats import norm,t


def calT(avg_y,y):
    T = len(y)
    sigma = 0.0
    for i in range(len(y)):
        mse = (y[i]-avg_y)**2
        sigma+=mse
    si = np.sqrt(1/(T-1)*sigma)
    t = avg_y*np.sqrt(T)/si
    return t


params = pd.read_hdf('params.h5',key='df')
avg_const = np.mean(params['const'])
avg_size = np.mean(params['size'])
avg_ep = np.mean(params['ep'])
avg_beta = np.mean(params['beta'])

tc = calT(avg_const,np.array(params['const']))
print("const")
print("统计量",tc)
print("系数",avg_const)
print("P值",1-norm.cdf(abs(tc),0,1))

tc = calT(avg_size,np.array(params['size']))
print("size")
print("统计量",tc)
print("系数",avg_size)
print("P值",1-norm.cdf(abs(tc),0,1))

tc = calT(avg_ep,np.array(params['ep']))
print("ep")
print("统计量",tc)
print("系数",avg_ep)
print("P值",1-norm.cdf(abs(tc),0,1))

tc = calT(avg_beta,np.array(params['beta']))
print("beta")
print("统计量",tc)
print("系数",avg_beta)
print("P值",1-norm.cdf(abs(tc),0,1))
```