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　中国股市单因子资产定价模型的实证检验

实验目的/要求
1、掌握单因子资产定价模型的时间序列估计与检验； 2、掌握单因子资产定价模型的横截面检验。
实验内容
1. 认真阅读 3 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. 尝试找出以 0 或者 6 开头，最后两位数字和自己学号最后两位数字一样的股票代码，并从中任意选取 8 只股票下载 2000-2022 年的日收益和月收益数据，进行如下实验：（1）利用股票日收益数据对 CAPM 模型做单资产时间序列检验；（2）利用股票月收益数据对 CAPM 模型做多资产时间序列检验。 3. 请利用 2001-2022 年中国 A 股月度数据实证检验中国 A 股市场是具有**惯性效应**，还是**反转效应**？（数据请从锐思数据库或者 Tushare 下载） 研究思路（参考课本《金融计量学》P48）：在每个月，计算过去 N 个月的累积收益率，再根据此累积收益率由低到高排序，构造 5 个等权重投资组合，并计算持有这些投资组合 M 个月的累积收益率，追踪投资组合收益率序列，判断中国 A 股市场存在惯性效应还是反转效应。 N = 1, 3, 6, 12；M = 1, 3, 6, 12
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）

在本次研究中，我们对 A 股市场部分个股和满足条件的所有股票进行了实证分析，旨在探讨资本资产定价模型（CAPM）在该市场中的实际表现和适用性。研究分为三个部分，每部分都旨在从不同角度验证 CAPM 模型的有效性。首先，我们采用了单资产 CAPM 模型的公式形式对选取的 8 只股票进行了实证检验，发现除了一只股票之外，其余 7 只股票都满足单资产 CAPM 模型。随后，对于多资产 CAPM 模型进行联合检验，8 只股票作为一个整体的表现同样符合 CAPM 模型。然而，在第三部分的研究中，选取了所有 2001 年之前上市并在 2022 年之前保持正常交易状态的 A 股股票。通过对这些股票的收益率进行分析，发现了反转效应，这表明股票的历史表现对其未来收益有一定的预测能力。这一现象与 CAPM 模型中关于完美市场的基本假设相悖。这说明在 A 股市场中有部分个股满足 CAPM 模型，但是整体市场与 CAPM 偏离。

通过这次对 A 股市场 CAPM 模型的实证研究，不仅增强了我的 python 编程能力，也让我完整的经历了金融实证研究的过程。我学会了如何寻找实验需要的数据、相关参考文献、利用 python 完成实证研究以及撰写论文的能力。

经过这个实验，我掌握了

（1）python 编程知识，必须利用 python 进行线性回归。

（2）完整的金融实证研究的过程。我学会了如何寻找实验需要的数据、相关参考文献、利用 python 完成实证研究以及撰写论文的能力。

（3）wald 检验、似然比检验、拉格朗日乘子检验方法以及排序法。

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：

中国股市单因子资产定价模型的实证检验

摘要：对于股市是否满足 CAPM 模型的检验一直是金融实证领域的热门议题。本文选取对 A 股市场 8 只股票进行了单资产 CAPM 检验，发现其中 7 只股票符合 CAPM 模型形式。接着，在多资产 CAPM 模型的应用中，这 8 只股票作为一个整体同样符合 CAPM 模型的预期。然而，当研究 2001 年之前上市并在 2022 年之前一直正常上市的所有 A 股股票时，我们观察到股票收益率呈现出明显的反转现象，这与 CAPM 模型中对完美市场的基本假设不符，这说明在 A 股市场中有部分个股满足 CAPM 模型，但是整体市场与 CAPM 偏离。

1 文献综述

Sharpe, William F.等在 1964 年提出的资本资产定价模型（Capital Asset Pricing Model, 简称 CAPM）[1]为现代金融理论奠定了基础，其建立了资产预期收益率与其所承担的系统性风险之间的线性关系。在金融实证领域中，不断有学者对 CAPM 模型在现实市场中的有效性进行研究。Fama，Eugene F，J.MaeBeth 在 1973 年提出了 Fama-MacBeth 回归，其首先通过时间序列回归得到个股的 β 值，再通过横截面回归参数结果取均值得到 CAPM 的截面回归结果，验证了平均股票收益与 β 值之间存在正相关性关系，而且截距大约就等于无风险收益率[2]。

然而，近年来有不少实证研究对 CAPM 的有效性提出了质疑，认为收益率不是只由系统性风险决定，其他因素对收益率也有显著的影响。Ball Ray 在 1978 年的研究中得到股票收益率和股票市值负相关的结论[3]。Bhandari, Laxmi Chand 在 1988 年的研究中发现在控制贝塔系数和公司规模的情况下，普通股预期收益与债务（非普通股负债）与股本的比率呈正相关[4]。Fama, Eugene F. and Kenneth French 在 1992 年的对 size，leverage，BM，β 等多个因子对平均收益率的影响进行研究，证明了 β 在这些因子中对于收益率的解释能力较差[5]。

在国内也有学者针对国内市场是否满足资产定价模型进行了实证研究。靳云汇和刘霖在 2001 年的研究中利用多种方法检验了 CAPM 在中国股票市场上的适用性，其选择了 1997 年 5 月 1 日到 2000 年 4 月 30 日的 496 只股票作为研究对象，以上证指数和深圳成指作为市场指数对标准的 CAPM 模型公式形式进行了最小二乘回归、极大似然回

归和广义矩方法，证明了股票收益率和β之间不是线性关系，而且收益率还和其他因素有关[6]。贾权,陈章武在 2003 年也对类似问题进行了研究，其以上交所和深交所上市的所有 707 支股票(A 股)为样本，通过实证证明了收益率和代表系统性风险的β值呈现反比关系，与 CAPM 模型不符，同时给出了中国市场为何偏离 CAPM 模型的解释：中国股市的非有效性和中国投资者的非理性行为[7]。杨朝军,邢靖在 1998 年对于上海证券市场数据的研究中利用投资组合分散风险后对收益率和系统性风险β的关系进行了研究，发现股票的平均收益率和系统性风险度量单位β并不是如CAPM模型中的线性关系[8]。Shi H L , Jiang Z Q , Zhou W X 利用 1997 年 1 月至 2012 年 12 月期间在上海证券交易所和深圳证券交易所交易的所有股票的月度数据，发现整个样本期内都存在短期和长期反向盈利的证据[9]。

本研究选取 A 股市场的 8 只股票数据进行研究。对每只股票进行单资产检验，研究单一股票收益率是否满足 CAPM 模型，对 8 只股票一起进行多资产检验，联合检验是否满足 CAPM 模型，最后选取 A 股市场上所有 2001 年之前上市且到 2022 年仍正常上市的股票，研究其是否体现了 CAPM 模型无法解释的动量效应或者反转效应。

本文剩余章节安排如下：第二部分为模型和方法，第三部分为实证研究的结果展示以及对应的结果分析。

2 模型和方法

2.1 CAPM 模型

资本资产定价模型(CAPM 模型)是由 Sharpe, William F.等人在 20 世纪 60 年代提出的一种金融学理论[1]。CAPM 模型建立了资产的预期收益率与系统性风险之间的线性关系。模型成立的基本假设为投资者是只根据资产预期收益与收益率标准差作决策的追求效用最大的风险厌恶投资者，不同投资者之间是同质预期的并且可以以无风险利率借贷；资产的收益服从正态分布，可以被无限细分；市场是完美的等待。在满足这些前提假设的情况下，单个资产的风险溢价与市场投资组合的风险溢价成正比，其公式为：

$$E(r_i) - r_f = \frac{Cov(r_i, r_m)}{\sigma_m^2}[E(r_m) - r_f] = \beta_i[E(r_m) - r_f] \tag{1}$$

式(1)中，$E(r_i)$是资产的期望收益率，r_f是无风险利率，$E(r_m)$是市场组合的期望收益率，σ_m^2是市场组合期望收益率的方差。

公式(1)说明了 CAPM 模型中市场组合相对无风险利率的超额收益以及其系数β_i决定了资产的期望收益率。在 CAPM 模型中非系统性风险可以通过多样化投资消除，而

系统性风险（市场风险）则无法通过多样化消除。β 系数衡量的就是某个资产或投资组合相对于整个市场的系统性风险的相关程度。β 值大于 1 表示资产价格波动比市场更大；β 值小于 1 表示资产价格波动小于市场；β 值为 1 则意味着资产价格随市场同幅度变化。

2.2 线性回归

线性回归是一种常用的数据预测与分析方法，其被用来描述因变量与一个或多个自变量之间的线性关系，即因变量可以通过自变量的线性组合来表示。线性回归模型可以表示为：

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \tag{2}$$

式(2)中，x_i为自变量，y是因变量，$w_i (i > 0)$表示各自变量的权重，w_0为常数项，又称为截距。

线性回归的损失函数被定义为真实值和拟合值之间的平方误差之和：

$$L(y, \hat{y}) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3}$$

式(3)中，$\hat{y}$是模型拟合值，y是真实值。

其参数的确定采用最小二乘法(Least Square Method)，即通过最小化这个损失函数来找到最佳的模型参数，使得模型预测值与真实值之间的差异平方和达到最小。

线性回归模型形式简单，容易理解。它假设因变量与自变量之间存在简单线性关系，使得模型易于解释。CAPM 模型是线性形式，适合使用线性回归拟合。

在 python 中使用 statsmodels.api 中的库函数进行线性回归。首先使用 add_constant()函数为自变量 X 添加一个截距项，接着使用 OLS(y,x)构造函数创建一个类进行最小二乘回归并且使用一个变量 model 接收对象，使用 model 对象的 fit()方法进行回归，最后使用 summary()方法获取回归结果。

2.3 Wald 检验

Wald 检验是一种统计学中的假设检验方法，主要用于评估模型参数是否满足某个约束条件或者检验参数值是否等于一个特定的理论值。在计量经济学和其他应用领域中，它被广泛用于验证线性回归模型以及其他类型模型中参数估计的有效性和显著性。其基本原理是股市无约束估计量与约束估计量之间的距离，其计算过程为：

1) 估计模型参数：对无约束的统计模型进行最大似然估计（MLE）或其他估计方法，得到参数的估计值

2) 计算 wald 统计量：

$$W_{\chi^2} = T\left[1 + \frac{\hat{\mu}_m^2}{\hat{\sigma}_m^2}\right]^{-1} \hat{\alpha}^T \hat{\Sigma}^{-1} \hat{\alpha} \sim \chi_N^2 \tag{4}$$

$$W_F = \frac{T-N-1}{N}\left[1 + \frac{\hat{\mu}_m^2}{\hat{\sigma}_m^2}\right]^{-1} \hat{\alpha}^T \hat{\Sigma}^{-1} \hat{\alpha} \sim F(N, T-N-1) \tag{5}$$

式(4)和(5)中，T 是样本总数，N 是模型中待检验参数的数量，$\hat{\alpha}$是待检验的参数向量，即模型中的某些或全部系数估计值，Σ是是协方差矩阵，通常包含了模型参数估计值的协方差和方差信息，反映了参数之间可能存在的相关性。第一个统计量满足卡方分布，第二个统计量满足 F 分布。

在 python 使用 scipy 库的 chi2,f 获取对应分布在 scipy.stats 中的对象，使用其中的 cdf()方法，传入统计量和对应分布的自由度即可得到对应统计量的累计概率，最后用 1 减去这个概率得到 P 值。

2.4 似然比检验

似然比检验的基本思想是如果某一个参数约束是有效的，那么加上约束后不会引起似然函数最大值的的大幅度降低。在似然比检验中，原假设 H_0 为参数约束为 0，备择假设 H_1 为参数约束不为 0，通过比较有某个参数约束时的似然函数函数最大值和无约束时的似然函数最大值，使用其比值作为统计量，该量满足卡方分布。其具体步骤为：

1) 计算似然函数：

$$L(\theta_0) = L(x_1, x_2, x_3, \ldots, x_n; \theta_0) = \prod_{i=1}^{n} p(x_i; \theta_0) \tag{6}$$

$$L(\theta_1) = L(x_1, x_2, x_3, \ldots, x_n; \theta_1) = \prod_{i=1}^{n} p(x_i; \theta_1) \tag{7}$$

式(4)中θ_0是原假设，θ_1是备择假设。

2) 对似然函数取对数：

$$ln\big(L(\theta_0)\big) = ln\big(\prod_{i=1}^{n} p(x_i; \theta_0)\big) \tag{8}$$

$$ln\big(L(\theta_1)\big) = ln\big(\prod_{i=1}^{n} p(x_i; \theta_1)\big) \tag{9}$$

3) 计算似然比：

$$S = [lnL(\theta_1) - lnL(\theta_0)] \sim \chi_n^2 \tag{10}$$

式(8)中，χ_n^2为卡方分布。

通过卡方分布计算求得似然比的 P 值，即可得出是否接收原假设。

2.5 拉格朗日乘子检验

拉格朗日乘子检验的基本思想是，在约束条件下，可以用拉格朗日方法构造目标函数。如果约束有效，则最大化拉格朗日函数所得估计量应位于最大化无约束所得参数估计值附近，其对于有约束公式进行检验。具体的计算过程为：

1) 估计约束模型的参数：

2) 计算无限制模型似然函数的得分向量和信息矩阵:

$$s(\theta)=\frac{\partial logL(\theta)}{\partial \theta}\Big|_{\theta=\{\alpha_i=0,\bar{\theta}\}} \tag{11}$$

$$I(\theta)=-E\left[\frac{\partial^2 logL(\theta)}{\partial\theta\partial\theta'}\right]=-E\left[\frac{\partial s(\theta)}{\partial\theta}\right]\Big|_{\theta=\{\alpha_i=0,\bar{\theta}\}} \tag{12}$$

式(11)是得分向量的计算，式(12)是信息矩阵的计算，其中 θ 是模型中的所有待估计参数，L(θ) 是似然函数。

3 结果与讨论

3.1 CAPM 模型单资产时间序列检验

本研究使用股票代码为 002094 002194 600094 600594 600694 600794 600894 000948 的 8 只股票(注：选择 000948 的原因是以 94 结尾且能满足数据量要求的股票不满 8 只)。选择这 8 只股票都具有交易数据的时间段：2007-12-10 到 2022-12-29，计算对数收益率并减去无风险利率获取每只股票的风险溢价。同时选取对应时间段的上证综指作为市场收益率，减去无风险利率获得市场风险溢价。对 r_i-r_f = α + β (r_m-r_f)进行线性回归，如果该资产满足 CAPM 模型，则常数项 α 应该为 0。为了检验是否满足 CAPM，采用 t 检验，原假设 H_0 为 α =0，备择假设 H_1 $\alpha\neq0$，如果股票收益率满足 CAPM 模型，则在假设检验中常数项 α 的 t 检验结果应接受原假设。

下面表 1 是 8 只股票各自的线性回归假设检验结果。（注： α 是回归参数中的常数项， β 是斜率参数）

表 1. 单资产 CAPM 检验结果

股票代码	回归参数	参数值	t 统计量	P 值	R-squared
000948	α	0.0017	2.723	0.007***	0.193
	β	1.1647	22.636	0.000***	
002094	α	0.0009	1.554	0.120	0.129
	β	0.8980	17.854	0.000***	
002194	α	0.0009	1.589	0.112	0.177
	β	1.0632	21.464	0.000***	

600094	α	0.0005	1.025	0.306	0.154
	β	0.7916	19.777	0.000***	
600594	α	0.0004	0.978	0.328	0.203
	β	0.8980	23.351	0.000***	
600694	α	-0.0002	-0.647	0.518	0.307
	β	0.9372	30.845	0.000***	
600794	α	0.0008	1.709	0.088	0.210
	β	0.9540	23.861	0.000***	
600894	α	0.0004	1.035	0.301	0.307
	β	0.9902	30.827	0.000***	

注：*表示在 10%水平下显著，**表示在 5%水平下显著，***表示在 1%水平下显著。

从表 1 中可以看出除了代码为 000948 的股票外，其他股票的 CAPM 线性回归参数的截距项的 t 检验结果均不显著，即可以接受原假设，α 应为 0，因此在所选的股票中除了代码为 000948 的股票外，其他股票均满足单资产 CAPM 模型。

3.2 CAPM 模型多资产联合检验

将上述 8 个股票资产合并为一个矩阵进行多资产的 CAPM 检验，通过 Wald 检验、似然比检验以及拉格朗日乘子检验法检验多资产是否满足 CAPM。原假设为每个资产的 α 都为 0，即 α_1，α_2，α_3，…，$\alpha_n=0$。检验结果如表 2 所示：

表 2. 多资产 CAPM 检验结果

检验方法	统计量分布	P 值
Wald 检验	卡方分布	0.828
Wald 检验	F 分布	0.859
似然比检验	卡方分布	0.838
拉格朗日乘子检验	卡方分布	0.846

从表 2 中可以看出 Wald 检验，似然比检验和拉格朗日乘子检验统计量的 P 值高于 0.1，即无法拒绝原假设，原假设 α_1，α_2，α_3，…，$\alpha_n=0$ 成立，CAPM 在所选取的资产的多资产估计中成立。

3.3 A 股市场动量效应和反转效应检验

动量效应又称惯性效应，其描述的是过去表现好的股票在未来一段时间内继续表现出色的现象，即过去一段时间内收益率高的股票在未来一段时间内的收益率仍然高于过去收益率低的股票。而反转效应则恰恰相反，其描述的是过去一段时间内表现差的股票在未来的表现会变好，而过去表现好的股票在未来一段内会变差。这些现象与 CAPM 的假设相悖，其假设市场参与者能够迅速、完全地反映所有公开信息，从而使得价格始终处于均衡状态，不存在持续的趋势或惯性。存在动量效应表明市场存在某种形式的信息

处理延迟或者行为偏误，导致价格并非立即达到其基本价值。存在反转效应则表明超额收益和系统性风险β并不是正向线性关系，违背了 CAPM。

在本研究中针对全 A 股市场股票是否存在动量效应或者反转效应进行检验。实验选用 A 股市场 2001 年到 2022 年之间已经上市且不存在退市情况或者 ST 情况的股票，检验过去 1、3、6、12 个月不同表现的股票在未来 1、3、6、12 个月的表现。实验中计算每一个月的过去 1、3、6、12 个月收益率，按照升序排序后的收益率的分位数划分为 Q1、Q2、Q3、Q4、Q5 这 5 个区间，股票过去的表现按照 Q1 到 Q5 的顺序递减。

表 3. 反转效应检验结果

历史月份	五分位	持有期			
		1 个月	**3 个月**	**6 个月**	**12 个月**
1 个月	Q1	0.53%	0.79%	0.19%	-1.11%
	Q2	0.54%	0.87%	0.74%	0.75%
	Q3	0.26%	0.44%	0.50%	0.53%
	Q4	-0.20%	-0.67%	-0.92%	-1.08%
	Q5	-1.493%	-3.00%	-3.90%	-5.29%
3 个月	Q1	0.68%	0.95%	0.24%	-0.60%
	Q2	0.43%	0.82%	0.71%	0.95%
	Q3	0.25%	0.63%	0.74%	1.42%
	Q4	-0.29%	-0.77%	-0.69%	-0.96%
	Q5	-1.58%	-3.25%	-4.35%	-6.41%
6 个月	Q1	0.52%	0.62%	0.26%	0.27%
	Q2	0.49%	0.76%	1.02%	1.79%
	Q3	0.23%	0.62%	1.07%	1.60%
	Q4	-0.29%	-0.51%	-0.30%	-0.71%
	Q5	-1.37%	-2.97%	-4.11%	-7.03%
12 个月	Q1	0.47%	0.80%	1.07%	2.23%
	Q2	0.52%	1.07%	1.74%	3.56%
	Q3	0.32%	0.81%	1.30%	2.25%
	Q4	-0.17%	-0.42%	-0.59%	-0.70%
	Q5	-1.21%	-2.83%	-4.42%	-7.38%

从表 3 中可以看出，A 股市场总体上存在反转效应。过去 1、3、6、12 个月收益率表现最差的 Q1 组合的收益率始终高于过去表现最好的 Q5 组合。这种反转效应在未来一个月中尤其明显，对于过去 1、3、6、12 个月不同表现的股票来说，其在未来第一个月中出现了非常明显的反转效应，Q1 到 Q5 的收益率大体上呈现递减趋势。但是当未来持有期变长后，尽管 Q1 收益率仍然大于 Q5 收益率，但是 Q1 的收益率往往小于 Q2 甚

至 Q3。当未来持有期为 6 个月和 12 个月时，Q2 和 Q3 的表现是最好的，且 Q2，Q3 和 Q1 之间相差较大，说明过去最差的股票在未来并不是表现最好的，过去表现中等偏上的股票在未来表现最好。因此可以证明在未来一个月内中国股市过去表现最差的股票未来表现最好，而在未来 3、6、12 个月过去表现中等偏上的股票表现最好。

3.4 分析总结

本研究中对于 A 股市场进行了三个实证研究。在单资产 CAPM 模型的检验研究中，选取的 8 只股票中有 7 只满足单资产 CAPM 模型。在多资产 CAPM 模型中，对选取的 8 只股票进行联合检验满足 CAPM 模型，然而在对 A 股市场所有 2001 年之前上市且到 2022 年为止仍然正常上市的股票的研究中，验证了这些股票的收益率存在反转效应，不满足 CAPM 基本假设中对于完美市场的假设。造成这种矛盾现象的原因可能是前两个实证检验中选取的股票数目较少，存在特殊性。未来该研究可以选择更多股票数据已得到更准确的实证结果。

4 参考文献

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

[2] Fama, Eugene F, J.MaeBeth, Risk Return and Equilibrium: Empirical tests. Journal of Politieal Eeonomy, 1973, 27

[3] Ball Ray. Anomalies in relationships between securities' yields and yield-surrogates[J]. Journal of Financial Economics, 1978, 6(2-3): 103-126.

[4] Bhandari, Laxmi Chand. Debt, Equity ratio and expected common stock returns: empirical evidence. Journal of Finance, 1988. 507-528

[5] Fama, Eugene F. and Kenneth French. The cross-section in expected stock returns. Journal of Finance, 1992.427-466

[6] 靳云汇，刘霖. 中国股票市场 CAPM 的实证研究[J]. 金融研究, 2001(07): 106-115.

[7] 贾权，陈章武. 中国股市有效性的实证分析[J]. 金融研究, 2003(07): 86-92.

[8] 杨朝军，邢靖. 上海证券市场 CAPM 实证检验[J]. 上海交通大学学报, 1998(03): 61-66.

[9] Shi H L, Jiang Z Q, Zhou W X. Profitability of contrarian strategies in the Chinese stock market[J]. Papers, 2015.

5 附录

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from StockDeal import StockDeal
from Test import TestMethods

stockDeal = StockDeal()
#获取股票df的列表
df_list = stockDeal.getData()
#print(df_list[0])
#获取每个股票拼接后的df
result = stockDeal.sliceStock(df_list)

#数据清洗
for df in result:
    df.dropna(inplace=True)
    df['date']=pd.to_datetime(df['date'])
    df.sort_values(by='date',inplace=True)

result = stockDeal.getReturn(result)

#读取指数数据
index = pd.read_excel('./data/SSEC.xls')
index = index[['交易日期_TrdDt','收盘价(元/点)_ClPr']]
index.columns = ['date','ind_close']
index.dropna(inplace=True)
#计算指数收益率
index['return'] = np.log(index['ind_close'])-np.log(index['ind_close'].shift(1))
index.dropna(inplace=True)
index = index.loc[(index['return']>=-0.1) & (index['return']<=0.1)]
index['date'] = pd.to_datetime(index['date'])

#内连接矩阵
merage_result = index
# merage_result = pd.merge(left=merage_result[['date','return']],right=result[0][['date','return']],on='date')
# merage_result.columns = ['date','ind_return','948_return']
# merage_result = pd.merge(left=merage_result[['date','ind_return','948_return']],right=result[0][['date','return']],on='date')
# merage_result.columns = ['date','ind_return','948_return','2094_return']
# merage_result = pd.merge(left=merage_result[['date','ind_return','948_return','2094_return']],right=result[0][['date','return']],on='date')
# merage_result.columns = ['date','ind_return','948_return','2094_return','2194_return']

cols = ['date','return']
codes = ['948','2094','2194','600094','600594','600694','600794','600894']
for i,df in enumerate(result):
    merage_result = pd.merge(left=merage_result[cols],right=df[['date','return']],on='date')
    cols.append(codes[i])
    merage_result.columns = cols
    print(merage_result)
merage_result.columns = ['date','indR','948R','2094R','2194R','600094R','600594R','600694R','600794R','600894R']
merage_result = pd.merge(left=merage_result,right=result[0][['date','stk_rf']],on='date')
print(merage_result)
#r - rf
columns = merage_result.columns
for col in columns[1:-1]:
    merage_result[col] = merage_result[col]-merage_result['stk_rf']

#线性回归

for col in columns[1:-1]:
    x = sm.add_constant(merage_result['indR'])
    model = sm.OLS(merage_result[col],x)
    result = model.fit()
    print(result.summary())

# test = TestMethods()
# test.waldTest(merage_result)

```

```python
from heapq import merge
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from StockDeal import StockDeal
from Test import TestMethods

data = pd.read_excel('./data/月数据.xls',usecols=[0,2,3,4])
data.columns = ['code','date','r','rf']
data.dropna(inplace=True)
codes = np.unique(data['code'])
df_list = []
for code in codes:
    df = data.loc[data['code']==code]
    df_list.append(df)
for df in df_list:
    df['date'] = pd.to_datetime(df['date'])
    df.sort_values(by='date',inplace=True)
    df.dropna(inplace=True)


#获取月上证指数数据
ssec = pd.read_excel('./data/SSEC月.xls',usecols=[2,3,4])
ssec = ssec.loc[ssec['月末交易日标识_Monflg']==1]
ssec.columns = ['date','close','flag']

ssec['return'] = np.log(ssec['close'])-np.log(ssec['close'].shift(1))
ssec.dropna(inplace=True)
ssec['date']=pd.to_datetime(ssec['date'])

#内连接
codes = ['948','2094','2194','600094','600594','600694','600794','600894']
merge_result = ssec[['date','return']]
cols = ['date','return']
for i,df in enumerate(df_list):
    merge_result = pd.merge(left=merge_result[cols],right=df[['date','r']],on='date')
    cols.append(codes[i])
    merge_result.columns = cols
merge_result.columns = ['date','indR','948R','2094R','2194R','600094R','600594R','600694R','600794R','600894R']
merge_result = pd.merge(left=merge_result,right=df_list[0][['date','rf']],on='date')
columns = merge_result.columns
for col in columns[1:-1]:
    merge_result[col] = merge_result[col]-merge_result['rf']
print(merge_result)
test = TestMethods()
test.waldTest(merge_result)

```

```python
import pandas as pd
import numpy as np

df_list = []
file_list = ['data\data3\RESSET_MRESSTK_20240314144150\RESSET_MRESSTK_1.xls',
             'data\data3\RESSET_MRESSTK_20240314144150\RESSET_MRESSTK_2.xls',
             'data\data3\RESSET_MRESSTK_20240314144402\RESSET_MRESSTK_1.xls',
             'data\data3\RESSET_MRESSTK_20240314144402\RESSET_MRESSTK_2.xls',
             'data\data3\RESSET_MRESSTK_20240314144532\RESSET_MRESSTK_1.xls',
             'data\data3\RESSET_MRESSTK_20240314144532\RESSET_MRESSTK_2.xls',
             'data\data3\RESSET_MRESSTK_20240314144555\RESSET_MRESSTK_1.xls']
for path in file_list:
    df1 = pd.read_excel(path,usecols=[0,2,3])
    df1.columns = ['code','date','close']
    df1['date']=pd.to_datetime(df1['date'])
    df1.dropna(inplace=True)
    df_list.append(df1)

code_list = []
stock_list = []
for df in df_list:
    codes = np.unique(df['code'])
    for code in codes:
        d = df.loc[df['code']==code]
        stock_list.append(d)

del_index = []
for i in range(len(stock_list)):
    stock_list[i] = stock_list[i].loc[(stock_list[i]['date']>='2002-01-01') &(stock_list[i]['date']<='2022-01-01')]
    stock_list[i].dropna(inplace=True)
    if len(stock_list[i]) != len(stock_list[0]):
        del_index.append(i)
for i in del_index:
    stock_list.pop(i)
#计算过去一个月的收益
# for stock in stock_list:
#     stock['return'] = np.log(stock['close'])-np.log(stock['close'].shift(1))
#     stock.dropna(inplace = True)

#重设索引
index = np.arange(0,len(stock_list[0]))
for stock in stock_list:
    stock['index'] = index
    stock.set_index('index',inplace=True)

#参数
last_m = 12   #表示过去月份
next_m = 12   #表示未来月份

totalQ = {0:0,1:0,2:0,3:0,4:0}
countQ = {0:0,1:0,2:0,3:0,4:0}
for month in range(last_m,len(stock_list[0])-next_m):#月份遍历
    qlist = [] #每一个月的收益率
    for stock in stock_list:
        #计算上一个月的收益
        r_last = np.log(stock.loc[month,'close'])-np.log(stock.loc[month-last_m,'close'])

        qlist.append(r_last)
    qlist = np.array(qlist)

    qlist = np.sort(qlist)
    #得到5个分位区间
    q = np.linspace(0,len(stock_list)-1,6).astype('int')
    #print(q)
    Q = {0:0,1:0,2:0,3:0,4:0}
    count = {0:0,1:0,2:0,3:0,4:0}

    for stock in stock_list:
        #计算上一个月的收益
        r_last = np.log(stock.loc[month,'close'])-np.log(stock.loc[month-last_m,'close'])
        for i in range(len(q)-1):
            if r_last>=qlist[q[i]] and r_last<=qlist[q[i+1]]:
                #计算下一个月的收益
                r_next = np.log(stock.loc[month+next_m,'close'])-np.log(stock.loc[month,'close'])
                #算入对应组别投资组合的累计收益率
                Q[i] = Q[i] + r_next
                count[i] = count[i]+1

    for i in range(5):
        Q[i] = Q[i] /count[i]
        totalQ[i] = totalQ[i]+Q[i]
        countQ[i] = countQ[i]+1


for i in range(5):
    totalQ[i] = totalQ[i]/countQ[i]*100
print(totalQ)
```

```python
import pandas as pd
import numpy as np
from scipy.stats import chi2,f

class TestMethods:
    def __init__(self):
        pass
    def waldTest(self,df):
        ret_ind = df['indR']
        res_stocks = df.loc[:,'948R':'600894R']
        T = len(ret_ind)
        N = 8
        mu_market = np.mean(ret_ind)
        sigma_market = np.sum(ret_ind-mu_market**2)/T
        #无限制模型
        x = np.ones((T,2))
        x[:,1] = ret_ind
        y = res_stocks
        xTx = np.dot(np.transpose(x),x)
        xTy = np.dot(np.transpose(x),y)
        AB_hat = np.dot(np.linalg.inv(xTx),xTy)
        alpha = AB_hat[0]
        beta = AB_hat[1]
        resid = y-np.dot(x,AB_hat)
        cov = np.dot(np.transpose(resid),resid)/T
        invCov = np.linalg.inv(cov)
        #限制模型
        xr = np.ones((T,1))
        xr[:,0]=ret_ind
        yr = res_stocks
        xrTxr = np.dot(np.transpose(xr),xr)
        xrTyr = np.dot(np.transpose(xr),yr)
        ABr_hat = np.dot(np.linalg.inv(xrTxr),xrTyr)
        RESDr = yr-np.dot(xr,ABr_hat)
        COVr = np.dot(np.transpose(RESDr),RESDr)/T
        invCOVr = np.linalg.inv(COVr)
        #Wald检验
        trans_ALPHA = np.ones((len(alpha),1))
        trans_ALPHA[:,0]=alpha
        SWChi2 = T*(1/(1+mu_market**2/sigma_market))*np.dot(np.dot(alpha,invCov),trans_ALPHA)
        SWF= (T-N-1)/N*(1/(1+mu_market**2/sigma_market))*np.dot(np.dot(alpha,invCov),trans_ALPHA)
        pvalue_Wchi2 = 1 - chi2.cdf(SWChi2[0],N)
        pvalue_WF = 1-f.cdf(SWF[0],N,T-N-1)
        print("wald检验结果:{0:.3f} {1:.3f}".format(pvalue_Wchi2,pvalue_WF))
        #似然比检验
        SLRchi2 = T*(np.log(np.linalg.det(COVr))-np.log(np.linalg.det(cov)))
        p_SLR = 1 - chi2.cdf(SLRchi2,N)
        print("似然比检验结果:{0:.3f}".format(p_SLR))
        #拉格朗日乘子
        a = np.zeros((8,1))
        a[:,0] = np.sum(RESDr,axis=0)
        salpha = np.dot(invCOVr,a)
        b = np.dot(ret_ind,RESDr)
        sbeta = np.zeros((8,1))
        sbeta[:,0] = np.dot(invCOVr,b)
        score = np.concatenate((salpha,sbeta),axis=0)
        a = np.concatenate((invCOVr*T,invCOVr*np.sum(ret_ind)),axis=1)
        b = np.concatenate((invCOVr*np.sum(ret_ind),invCOVr*np.sum(ret_ind**2)),axis = 1)
        Minfo = np.concatenate((a,b),axis=0)
        SLMchi2 = np.dot(np.dot(np.transpose(score),np.linalg.inv(Minfo)),score)
        pvalue_SLMchi2 = 1 - chi2.cdf(SLMchi2[0][0],N)
        print("拉格朗日检验结果:{0:.3f}".format(pvalue_SLMchi2))
```