基于基本面因子的多因子选股策略研究

冯腾

2022 年 6 月

摘要

多因子选股策略通过选取与股价高度相关的影响因子，可以帮助投资者在实现可观收益的同时控制风险。基于此，本文选取沪深 300 股指的成分股作为多因子模型分析的股票池，通过对候选基本面因子进行回归分析与单因子分析筛选出有效因子，并构造适合一般投资者使用的多因子模型。研究结果表明：6 个月动量、股息率、净资产收益率与营业利润增长率四个指标对股票收益有显著影响；采用简单打分的方法，对这 4 个指标进行赋值，并选出得分靠前的股票进行等额投资，可以获得不错的收益；对比指数与 Alpha 策略，本文构建的多因子模型更具有有效性。

关键词：量化投资；多因子模型；基本面；Alpha 策略

Abstract

Multi-factor stock picking strategies can help investors achieve substantial returns while controlling risk by selecting influencing factors that are highly correlated with stock prices.

This paper selects the constituent stocks of the Based on this, this paper selects the constituent stocks of the CSI 300 stock index as the pool of stocks for multi-factor model analysis, screens out the effective factors through regression analysis and single factor analysis of the candidate fundamental factors, and constructs a multi-factor model suitable for general investors. The results show that the four indicators of 6-month momentum, dividend yield, return on net assets and operating profit growth rate have significant effects on stock returns; the simple scoring method is used to assign values to these four indicators and select the top scoring stocks for equal investment, which can yield good returns; the multi-factor model constructed in this paper is more effective when comparing the index and Alpha strategies.

Keywords: quantitative investment; multi-factor model; fundamentals; Alpha strategy

目录

1. 引言

随着我国资本市场日趋成熟，人们的投资理念日趋科学和理性，投资方式也变得多元化，开始将国外引入的量化投资思想结合中国资本市场的具体情况，而不是依赖于情感和过去的感性经验。这种投资模式的变化使得我国量化投资研究得到迅速发展，逐渐增加的量化基金及其他定量金融产品在投资策略中越来越重要，因此在我国金融市场中，量化投资得到了越来越多的关注。信息不对称使得我国资本市场出现较多的市场失灵现象，往往造成市场非完全有效，与内在价值偏离（定价错误）的股票也较多，因此在这样的市场中，量化策略具有特有的纪律性、分散化持股、套利组合、系统性等优势，量化投资策略的应用前景广阔。

多因子选股模型作为一种应用较为广泛的量化投资策略，基本原理是通过经济逻辑和市场经验，捕获模型信息并采用一系列的因子，选入满足标准的因子，并剔除冗余因子。本文试图基于多因子模型，从基本面分析入手，从众多的候选因子中找出能够有效解释股票收益率且非冗余的因子来构建量化投资组合，并验证其有效性。建立一种基于多因素模型的股票量化选择策略，希望为投资者提供可行的量化投资参考。

2. 文献综述

国内外学者对基本面分析和多因子选股模型的实证研究都已有了较为丰富的

成果，对其在投资预测中的优缺点也有了清楚的认识，形成了一系列实证研究成果。

国外学者方面，许多学者对公司规模、市盈率、市净率、负债率等基本面因素进行了深入研究。Fama & French（1992、1993）提出了著名的 Fama-French 三因素资产定价模型，提出除了风险系数 β 之外，市场规模 SIZE 和账面市值比 BE/ME（即市净率的倒数）都显著解释了股票回报变动现象，研究发现市场规模、账面市值比、负债率等变量单独使用能解释股价回报，而这些变量联合使用时，市场规模和账面市值比对于回报仍具有显著的解释能力且可吸纳盈余价格比与负债率所能解释的股票回报。

其他还有很多学者从股权集中度、换手率、机构投资者比例、投资者情绪、市场关注度等方面对股票收益率进行了研究。Datar，Naik & Radcliff（1998）应用纽约股票交易所非金融类股票从 1963—1991 年间的数据进行实证分析，发现在控制规模、账面市值比以及 β 等因素后，股票收益与换手率具有显著的负相关关系。Piotroski（2000）首次将排序打分法融入多因子选股模型，选出九个财务指标作为判断标准，然后检验得分排序靠前的股票，取得良好效果。Mohanram（2005）从盈利因子、成长因子、稳健因子三方面选取指标对个股进行排序打分，最后建立的投资组合取得了较好的超额收益。Kariya（2012）将传统成长因子与价值性因子结合起来，提出了较有代表性的 GARP 多因子量化策略。Albadvi 和 Norouzi（2013）利用德国市场数据对基本面方面的因子进行了研究，发现不同行业具有不同的有效因子，并且不同因子对收益率的影响程度也不一样。Asness 等（2014）从市净率（P/B）展开讨论，得到衡量股票质量的四个维度——盈利能力、增长能力、安全性和股利发放，并且根据这四个维度的相关指标构建了股票质量因子（QMJ），回溯测试表明该因子能获得显著且稳健的超额收益。Lee 等（2015）发现，基本面因子模型是更好地衡量股票预期收益的指标。同时，基本面因子模型更适用于基本面量化投资。

国内有关研究起步较晚，主要集中在用国外因子模型检验中国市场，同时根据国内股市情况进行修正。陈信元、张田余和陈冬华（2001）对预期股票回报的决定因素进行了横截面分析，结果发现规模和 B/P 表现出显著的解释力，并且这样的结论在不同模型中始终成立；而在不同模型中，β、账面财务杠杆和市盈率始终没

有通过显著性检验。陈守东等（2003）较早证明了方法 FF 三因子选股模型在中国股票市场的适用性。苏宝通、陈炜、陈浪南（2004）对公开信息与股票回报率的相关性进行了研究后发现，公司规模、账面市值比、现金红利率和流通股比例对中国股票回报率有着显著的影响，而资本结构、股票价格、市盈率和前一年持股回报率对中国股票回报率影响不显著。张峥和刘力(2006)研究了换手率与股票收益率的关系，发现换手率可以准确地反映股票的价格行为，且主要由投资者非理性引起的，当股票换手率高时表示投资者高估了股票的市场价格，而股票换手率低时则表示投资者低估了股票市场价格。潘莉、徐建国（2011）研究了 A 股市场的风险与特征因子，发现市场平均回报率、股票市值和市盈率三个因子可以解释回报率变化的90%以上。刘辉、黄建山（2013）的实证研究发现，相较 CAPM 模型，FF 三因子模型能更好地解释 A 股的收益率。江方敏（2013）研究发现只有估值因子中的市净率、市盈率是有效的，同时他还创新加入了现金流因子。朱忆（2014）构建了包含利率风险溢价、换手率等因子在内的多因子模型，得到了可行的策略。

3. 数据与分析方法

3.1 相关理论基础

多因子量化选股模型指的是在选股过程中利用多个对股价走势有显著且有效影响的因子，通过量化不同因子对股票收益率的影响，建立起选股模型。多因子模型的建立主要在 CAPM 模型、APT 模型等理论的基础上逐步演化而来。CAPM 模型公式为:

$$E(r_i) = r_f + \beta_i[E(r_m) - r_f],$$

该模型表明资产的预期超额收益与市场超额收益成正比，股票的价格只与市场风险有关，跟上市公司基本面并没有关系，并且高的股价需要高的β值来支撑。由于 CAPM 模型假设条件过于苛刻，后来的学者们打破原有假设，导出套利定价理论(APT 模型)。模型公式:

$$E(r_i) = r_f + b_{i1}F_1 + b_{i2}F_2 + \cdots + b_{in}F_n,$$

$b_{in}F_n$为证券 i 第 n 个因素的敏感度。该模型比 CAPM 模型的假设更宽松，但是无法从模型中获知哪些因子起到决定性的作用。经过学者们的研究，很多现象不再可以用 CAPM 模型来解释，比如市场中的小市值、价值股表现明显超过市场。故 Fama 及 French 在 CAPM 模型的基础上加入了新的因子来解释资产的收益率。三因素模型表明收益率不仅与市场风险有关，还与账面市值比的模拟组合的收益率、市值因子的模拟组合的收益率有关。研究者后发现市场中的动量现象无法用三因子模型解释，所以在原有三因子的基础上加入动量效应，即某个时段连续上涨或下跌的股票往往会沿着原来的方向继续波动，由此构建四因子模型。先前的模型并没有引入刻画公司资产质量的因子来解释资产收益率，由此加入了代表盈利能力的 RMW 因子和代表投资模式的 CMA 因子，提出了五因子模型。

候选因子的选取

因子选取方面应该尽量考虑可用性、普遍性和较强的差异性。本文通过将量化分析中的常用因子分为价值能力因子、偿债能力因子、成长能力因子、品质能力因子、盈利能力因子、运营能力因子和动量因子七类（表 3-1），并通过有效性检验来筛选因子，构建模型与投资组合。

表 3-1：多因子模型候选因子

价值能力因子	市净率 PB、市盈率 PE、股息率 DP、市盈增长比率 PEG、市现率 PCF、市销率 PS
偿债能力因子	资产负债率 ARL、固定资产比例 FACR
品质能力因子	流通市值 CMC、营业利润/毛利润 Operatingprofit/Grossprofit
成长能力因子	净资产收益率增长率 ROE_g、净利润增长率 $Netprofit_g$、营业利润增长率 $Operatingprofit_g$、营业总收入同比增长率 $Totalrevenue_g$、净资产同比增长率 $Equity_g$、经营活动产生的现金流量净额同比增长率 $Netcashflowsfromoperatingactivities_g$、资产同比增长率 $Assets_g$
盈利能力因子	净资产收益率 ROE、资产报酬率 ROA、投入资本回报率 ROIC、销售净利率 NetProfitMargin、销售毛利率 Grossprofitmargin
运营能力因子	固定资产周转率 Fixedassetsturnover、流动资产周转率 Currentassetsturnover、应收账款周转率 Accountsreceivableturnover

动量因子	6 个月动量 Momentum6、12 个月动量 Momentum12、换手率 Turnoverrate、换手率变动 $Turnoverrate_g$

3.2 数据来源及描述性统计

本文利用 2005 年 5 月-2022 年 4 月沪深 300 成分股作为样本股进行因子的初步筛选。同时考虑到年度数据太少，而月度数据中大多数的财务指标无法体现变化，因此选取季度数据为研究对象。财务数据来自季报、半年报与年报，相应的市净率 PB、市盈率 PE、股息率 DP、市盈增长比率 PEG、市现率 PCF、市销率 PS、流通市值 CMC 等每日行情指标数据为 3 月、6 月、9 月、10 月最后一个交易日数据。股票涨跌幅为季度涨跌幅，分别对应 1 月至 3 月的收益率、4 月至 6 月的收益率、7 月至 9 月的收益率、11 月至 12 月的收益率。股票换手率为季度换手率，分别对应 1 月至 3 月的换手率、4 月至 6 月的换手率、7 月至 9 月的换手率、11 月至 12 月的换手率。所有数据均来自 Tushare 平台。

变量的描述性统计如表 3-2 所示，可以看出，股票季度收益率的平均值为 4.47%，但是最高值高达 1957%，而最低值为-77%；价值能力因子中，市盈率、市盈增长比率与市销率波动较大，最小值与最大值相差较大，其中市盈率均值为 79.79，市销率均值为 16.58，市净率为 4.96，股息率与市现率均值较小，分别为 1.34 和-0.02；偿债能力因子中，资产负债率均值为 52.89，固定资产比例均值为 44.19；品质能力因子中，流通市值均值达到 745 亿元，最小值为 8200 万元，最大值为 25836 亿元；成长能力因子普遍波动较大，其中净利润、营业利润增长率与营业总收入增长率均值均超过了 100%，资产同比增长率均值最小，为 14.79%；盈利能力因子中，净资产收益率、资产报酬率与投入资本回报率均值分别为 9.13、6.55 与 7.50，销售净利率较销售毛利率波动较大，标准差为其 10 倍；运营能力因子中，应收账款周转率均值达到了 71.32；动量因子中，6 个月动量均值为 0.10，12 个月动量均值为 0.23，换手率均值达到了 106%，最大值达到了 1739%，最小值为 0.06%，波动范围较大。

表 3-2：变量因子描述性统计

分类	变量因子	均值	标准差	最小值	最大值	偏度	峰度
因变量	季涨幅	0.044	0.33	-0.77	19.57	15.22	816.48
价值能力因子	市盈率	79.79	487.87	1.68	26570.13	34.01	1388.13
	市盈增长比率	2.28	48.95	-1636.13	3898.12	38.27	3009.23
	市净率	4.96	5.64	0.35	115.31	4.81	47.82
	市销率	16.58	350.31	0.05	23193.78	51.49	2959.54
	股息率	1.34	1.60	0.00	22.02	2.66	14.34
	市现率	-0.02	2.98	-380.31	26.30	-126.51	16133.91
偿债能力因子	资产负债率	52.89	28.61	0.96	1131.36	10.98	367.32
	固定资产比例	44.19	22.04	0.43	97.52	0.32	-0.71
品质能力因子	流通市值（万元）	7451781	16684025	8200.00	258362201	6.65	60.98
	营业利润/毛利润	16.19	196.70	-17769.26	2856.64	-66.41	5264.10
成长能力因子	净利润增长率（%）	101.78	2995.56	-52824.33	320220.09	85.80	8554.37
	营业利润增长率（%）	108.66	3372.93	-56299.85	310215.24	61.23	5008.40
	营业总收入同增长率（%）	100.00	8780.27	-100.00	1121910.77	127.59	16297.84
	净资产收益率增长率（%）	33.13	866.69	-57621.26	68675.63	14.96	3720.33
	资产同比增长率（%）	14.79	38.61	-98.53	1650.20	15.47	476.29
	净资产同比增长率（%）	43.56	1233.18	-14833.81	153750.71	120.91	15073.55
	经营活动产生的现金流量净额同比增长率（%）	80.13	7039.19	-677773.55	327807.26	-44.27	5761.58
盈利能力因子	净资产收益率	9.13	16.40	-1554.47	368.57	-52.07	5072.25
	资产报酬率	6.55	7.05	-146.16	104.57	1.34	29.88
	投入资本回报率	7.50	8.63	-199.15	124.88	-0.14	58.11
	销售净利率	15.25	220.66	-17769.26	10418.21	-40.92	3668.72
	销售毛利率	32.53	22.04	-70.15	192.51	0.84	0.79
运营能力因子	应收账款周转率	71.32	636.52	0.00	43284.66	35.60	1933.40
	流动资产周转率	1.09	1.23	0.00	22.55	4.91	50.02
	固定资产周转率	5.34	11.92	0.00	316.58	9.41	140.64
动量因子	6 个月动量	0.10	0.50	-0.84	21.31	8.69	254.96
	12 个月动量	0.23	0.86	-0.89	20.53	5.73	72.57
	换手率（%）	105.55	123.44	0.06	1738.87	3.66	21.88
	换手率变动	3.92	73.89	-1.00	3642.97	47.91	2352.63

4. 实证分析

4.1 有效因子的选取

本文用沪深 300 中的所有股票涨跌幅对每只股票对应的候选因子进行时间序列回归，公式如下：

$$y_{t+1} = \alpha_{tf} + \beta_t x_t + \varepsilon_t$$

得到每只股票涨跌幅对各因子的回归系数 β_i（i=1，2，3，……，300）后，剔出异常值和缺省值的影响，对有效的所有 β 值在 5%的显著性水平下进行 t 检验（H_0：$\bar{\beta}$=0；H_1：$\bar{\beta}$≠0），结果如表 4-1 所示。

表 4-1：候选因子的有效性检验结果

分　类	因子	样本均值	t 值	Pr>|t|	线性关系	有效性检验
价　值 能　力 因　子	市盈率	79.79	1.15e+01	1.97e-30	正相关	通过
	市盈增长比率	2.28	1.35e+00	1.76e-01	无明显关系	未通过
	市净率	4.96	2.11e+01	4.17e-97	正相关	通过
	市销率	16.58	6.32e+00	2.77e-10	正相关	通过
	股息率	1.34	-1.48e+01	2.21e-49	负相关	通过
	市现率	-0.02	-8.26e-01	4.09e-01	无明显关系	未通过
偿　债 能　力 因　子	资产负债率	52.89	-1.02e+00	3.06e-01	无明显关系	未通过
	固定资产比例	44.19	-2.76e-01	7.83e-01	无明显关系	未通过
品　质 能　力 因　子	流通市值	7451781	3.39e-02	9.73e-01	无明显关系	未通过
	营业利润/毛利润	16.19	-1.36e+00	1.75e-01	无明显关系	未通过
成　长 能　力 因　子	净利润增长率	101.78	1.85e+00	6.44e-02	无明显关系	未通过
	营业利润增长率	108.66	3.03e+00	2.45e-03	正相关	通过
	营业总收入同比增长率	100.00	1.41e-02	9.89e-01	无明显关系	未通过
	净资产收益率增长率	33.13	1.96e+00	4.97e-02	正相关	通过
	资产同比增长率	14.79	9.71e+00	3.25e-22	正相关	通过

	净资产同比增长率	43.56	7.59e+00	3.48e-14	正相关	通过
	经营活动产生的现金流量净额同比增长率	80.13	3.48e-01	7.28e-01	无明显关系	未通过
	净资产收益率	9.13	2.24e+00	2.54e-02	正相关	通过
盈 利	资产报酬率	6.55	1.52e+00	1.28e-01	无明显关系	未通过
能 力	投入资本回报率	7.50	2.14e+00	3.24e-02	正相关	通过
因 子	销售净利率	15.25	-8.55e-01	3.93e-01	无明显关系	未通过
	销售毛利率	32.53	2.12e-01	8.32e-01	无明显关系	未通过
运 营	应收账款周转率	71.32	1.36e+00	1.73e-01	无明显关系	未通过
能 力	流动资产周转率	1.09	4.31e-01	6.67e-01	无明显关系	未通过
因 子	固定资产周转率	5.34	3.11e-01	7.56e-01	无明显关系	未通过
	6 个月动量	0.10	1.35e+02	0.00e+00	正相关	通过
动 量	12 个月动量	0.23	7.19e+01	0.00e+00	正相关	通过
因 子	换手率	105.55	2.31e+01	1.99e-116	正相关	通过
	换手率变动	3.92	8.55e-01	3.93e-01	无明显关系	未通过

可见，较为有效的因子为市净率、市盈率、股息率、市销率、营业利润增长率、净资产收益率增长率、资产同比增长率、净资产同比增长率、净资产收益率、投入资本回报率、6 个月动量、12 个月动量与换手率。从结果可以看出，价值能力因子较为有效，能较好地反映与预测股票收益率，其中股息率因子较为特别，只有它与股票季收益率呈负相关，品质能力因子、偿债能力因子与运营能力因子中的候选因子均未通过有效性检验，说明这三类因子不能有效反映与预测股票收益率。

考虑到价值能力因子、成长能力因子、盈利能力因子与动量因子这四个种类中通过有效性检验的因子较多，为防止冗杂因子的出现，本文在经过有效性检验筛选出上述 13 个因子后，还将通过基于单个因子筛选股票后的收益表现筛选出最终的有效因子。具体操作是针对每一个因子，选择 2005 年 5 月-2022 年 4 月沪深 300 成分股在该因子排名靠前的 100 只，对这 100 只股票数据对应的年化收益率进行数据结果分析。基本数据结果如下表 4-2。

表 4-2：候选因子对样本股票检验结果

因子	年化收益率	因子	年化收益率
6 个月动量	46.90%	市盈率	25.80%
12 个月动量	46.00%	投入资本回报率	25.60%
营业利润增长率	28.80%	市销率	25.40%
股息率	26.80%	市净率	25.30%

净资产收益率	26.20%	资产同比增长率	23.60%
净资产收益率增长率	26.20%	换手率	23.60%
净资产同比增长率	25.90%		

可见，根据动量因子筛选出的股票组合年化收益率明显高于其他因子，其中 6 个月动量的组合收益较高；价值能力因子市净率、市盈率、股息率与市销率中，根据股息率筛选出的股票组合年化收益率最高，达到了 26.8%；盈利能力因子中，根据净资产收益率筛选出的股票组合年化收益率较投入资本回报率的组合更高，达到了 26.2%；成长能力因子营业利润增长率、净资产收益率增长率、资产同比增长率与净资产同比增长率中，根据营业利润增长率筛选出的股票组合年化收益率最高，达到了 28.8%。因此，从四类因子中各选一个因子，最后选择 6 个月动量、股息率、净资产收益率与营业利润增长率一共 4 个因子。

4.2 构建多因子选股模型与投资组合

本文构建的多因子模型如下：

$$E(r_i) = r_f + \beta_1 E(ROE) + \beta_2 E(MTM6) + \beta_2 E(DR) + \beta_2 E\left(OP_g\right) + \varepsilon$$

其中，MTM6 代表 6 个月动量，DR 代表股息率，ROE 代表净资产收益率，OP_g 代表营业利润增长率，从上文的有效性分析可知，股息率与投资组合预期收益存在负相关关系，6 个月动量、净资产收益率与营业利润增长率与投资组合预期收益存在正相关关系。

采用打分法进行投资组合的构建。对于每一时间点的股票按照每个因子值进行排序，正相关的因子按升序排列，负相关的因子按降序排列，并按照该排序给沪深 300 中的每只股票从 1 到 300 给每项因子依次打分，最后将 4 项因子得分的平均成绩按降序排列，股票得分越高表示在该模型上的表现越好。此处打分法中最后的平均得分是基于因子等权重的假设，目的是方便一般投资者能简单运用该模型进行选股策略，同时也为了验证量化选股的强有效性。

本文选取 2005 年 5 月 1 日至 2022 年 4 月 30 日的历史数据对投资组合的收益进行实证分析。选用的基准指数为沪深 300 指数，比较沪深 300 指数收益与多因子模型所构建的资产组合的绩效表现，具体操作为：每 3 个月视为一期，每期选择分数前 10%的股票构建最优投资组合，同时为了检验策略的有效性，每期选择分数后 10%的股票构建最差投资组合，每期采用相同的初始资金，每期初均对投资组合的股票进行等金额的投资，3 个月后清仓并调仓，比较每期投资组合与同期沪深 300 指数的持有至到期收益以及最优组合、最差组合与沪深 300 指数的累计净值，如图 4-1、图 4-2 所示。

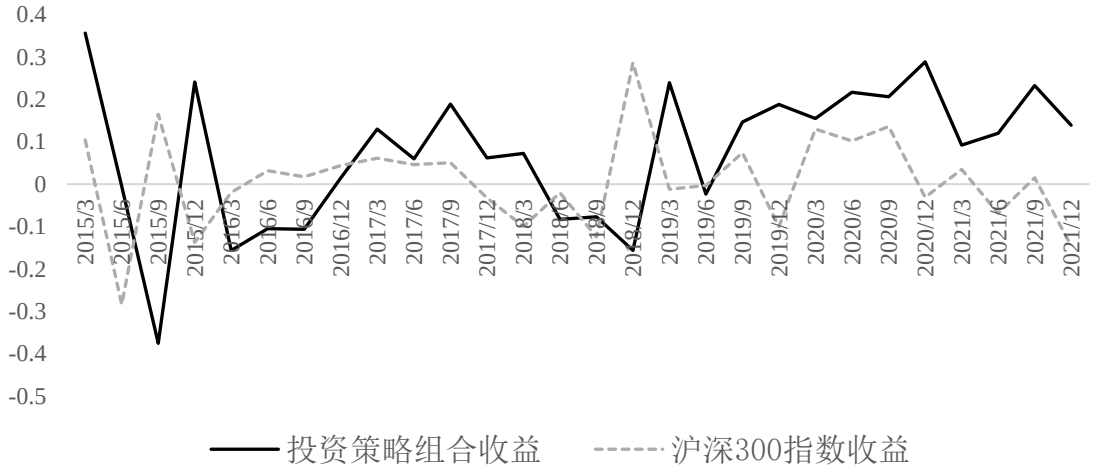

图 4-1：多因子模型组合与沪深 300 指数季度涨跌幅趋势

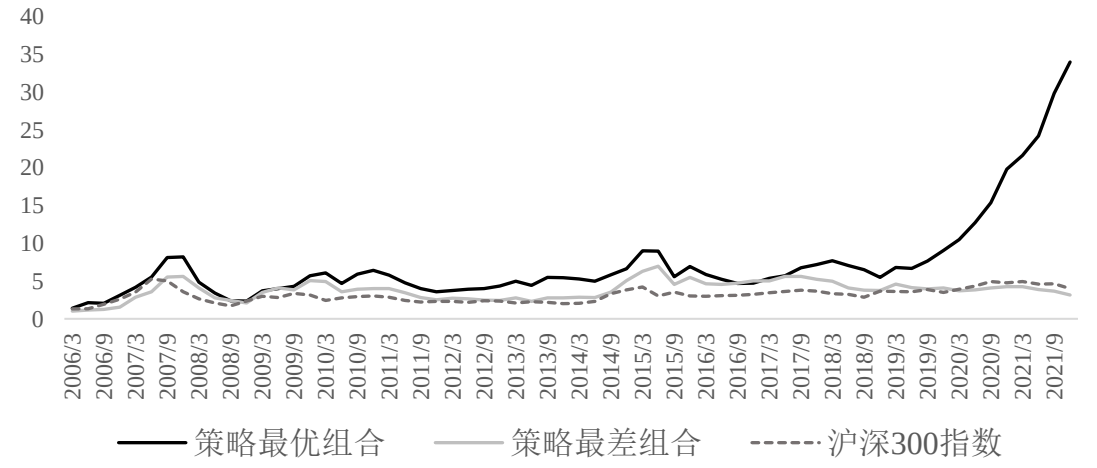

图 4-2：多因子模型组合与沪深 300 指数净值趋势

可见，2015 年至 2022 年中大部分时间投资组合的收益都优于市场表现，只在 2015 年 6 月至 2015 年 9 月、2016 年 3 月至 2016 年 9 月以及 2018 年 6 月至 2019 年 6 月之间有几个月的收益表现差于市场表现，特别是在 2019 年 9 月之后的收益均优于沪深 300 指数。2006 年至 2022 年，策略最优组合、策略最差组合与沪深 300 指数的累计净值的波动幅度相似，策略最优组合的累计净值在 2019 年 9 月之后大幅上升。

投资组合与沪深 300 指数的绩效表现如表 4-3 所示，从结果可知，根据多因子模型构建的投资组合的年化收益率达到了 24.65%，远高于沪深 300 指数的 9.02%，说明本文构建的多因子模型能有效提高投资组合的收益；投资组合的最大回撤与年化波动率均高于沪深 300 指数，说明投资组合的风险水平较大，但投资组合的夏普比率远高于沪深 300 指数。总的来说，本文所构建的多因子选股模型表现较为优异，盈利性较强，虽然风险水平高于市场平均，但收益很高，属于中风险、高收益的投资组合。

表 4-3：投资策略绩效表现

	净值	年化收益率%	最大回撤%	夏普比率	年化波动率
策略最优组合	33.972	24.65	71.36	0.52	41.86
策略最差组合	3.1704	7.48	62.6	0.1	42.84
沪深 300 指数	3.9795	9.02	67.43	0.17	34.68

5. 模型检验与修正

5.1 多重共线性的检验

由相关性检验矩阵（表 5-1）可证明并未有两两因子间出现较高的相关系数，说明该多元回归模型没有多重共线性，也证明最初的多因子模型无冗杂因子。

表 5-1：策略所选因子的相关性检验矩阵

	dv_ratio	mtm6	roe	op_yoy
dv_ratio	1			
mtm6	-0.1616	1		
roe	-0.0199	0.0333	1	
op_yoy	-0.0227	0.0345	0.0400	1

5.2 关于投资组合股票数量的稳健性检验

前文所构建的多因子投资组合中所含股票为候选股票池中前 10%的股票，为了准确反映出多因子模型的稳健性，对选择不同股票数量的投资组合绩效表现进行分析，如表 5-2。

表 5-2：选择不同股票数量的投资组合绩效表现

	净值	年化收益率%	最大回撤%	夏普比率	年化波动率
前 1%	6.5019	12.41	85.15	0.14	66.86
前 5%	56.9785	28.75	69.41	0.54	47.37
前 10%	33.972	24.65	71.36	0.52	41.86
前 20%	29.0665	23.44	65.48	0.52	39.11
前 30%	23.8483	21.92	64.38	0.49	38.33
前 40%	26.3548	22.69	64.39	0.52	37.77
沪深 300 指数	3.9795	9.02	67.43	0.17	34.68

可见，对于所含股票数量不同的多因子模型组合来说，组合的收益表现均优于沪深 300 指数，说明本文构建的多因子模型具有有效性；比较不同股票数量的投资组合，可知选择前 5%的股票所构建的投资组合的收益最好，年化收益率达到了 28.75%，净值为 56.98，夏普比率为 0.54。因此后续选择基于选择前 5%构建的多因子模型组合进行检验。

5.3 与 Alpha 策略对比

为了进一步检验多因子模型的有效性，本文还将进行多因子模型与 Alpha 策略的对比，构建 Alpha 策略组合的具体操作是：确定 Alpha 策略形成期和持仓期两个参数，选择一个交易日作为计算点，对股票池中的组合在计算点之前的形成期内的数据进行计算分析，挑选出超额收益最大的组合进行持有；在持有持仓期之后，再重新计算选择组合进行持有，如此反复进行一直持续到 2022 年 4 月，为了便于计算，选择以 2005 年 5 月第一个计算节点。

为了便于与多因子模型对比，Alpha 策略的持有期选择 3 个月，对于形成期的选择，基于不同形成期构建的 Alpha 策略投资组合的绩效表现如表 5-3。

表 5-3：不同形成期的 Alpha 策略组合绩效表现

	净值	年化收益率%	最大回撤%	夏普比率	年化波动率
3 个月	10.2508	15.4	67.9	0.29	43.45
6 个月	12.8634	17.31	65.76	0.33	43.33
9 个月	5.9055	11.94	71.98	0.2	44.34
12 个月	8.5697	14.87	65.74	0.29	41.1
15 个月	3.6388	8.84	66.76	0.15	39.45
18 个月	1.457	2.54	70.11	-0.01	38.77

可见，形成期为 6 个月的 Alpha 策略的表现最优，年化收益率与夏普比率最大，分别达到了 17.31%和 0.33，因此，Alpha 策略选择 6 个月的形成期。

比较 Alpha 策略组合与多因子模型组合的净值与绩效表现，如图 5-1，表 5-4

所示。

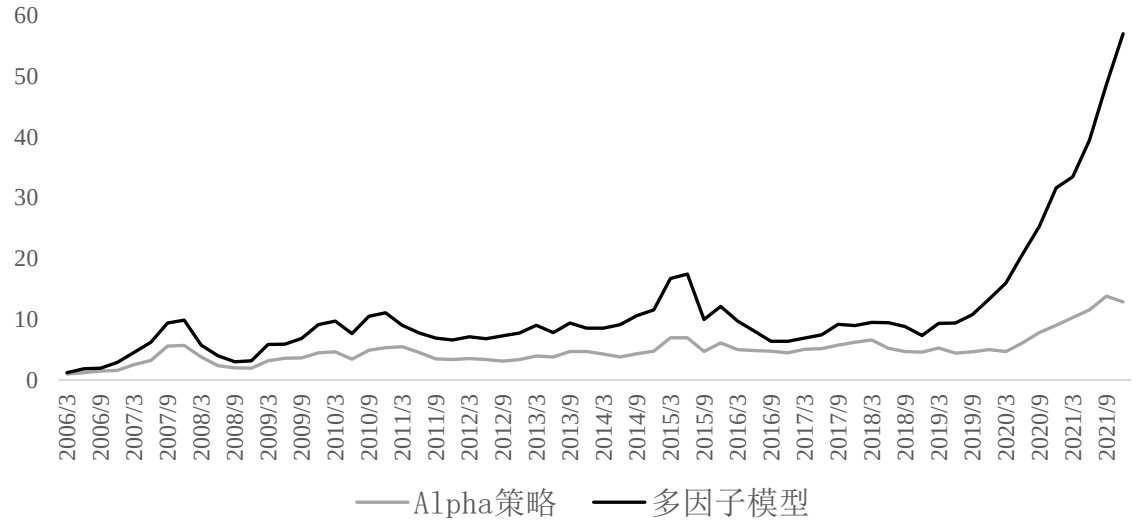

图 5-1：Alpha 策略与多因子模型组合净值趋势

表 5-4：Alpha 策略与多因子模型绩效表现

	净值	年化收益率%	最大回撤%	夏普比率	年化波动率
Alpha 策略	12.8634	17.31	65.76	0.33	43.33
多因子模型	56.9785	28.75	69.41	0.54	47.37

可见，多因子模型组合的净值与年化收益远远高于 Alpha 策略组合，多因子模型的最大回撤与年化波动率略高于 Alpha 策略。说明本文构建的多因子模型的表现优于一般投资策略，相比现存的投资策略，是一种更为有效的投资策略。

6. 结论

本文以量化投资中最常用的多因子模型为基础，从一元回归对因子进行有效性检验开始，筛选出从数据上较为有效的因子，并通过对比以单一因子构建出的投资组合的收益，最后选择了 6 个月动量、股息率、净资产收益率与营业利润增长率一共 4 个因子，构建了基本的多因子模型，采用打分法为广大投资者提供了一个简单可操作、无需太多复杂整理运算的选股策略，并用历史数据验证了该多因子选股模型的投资效果的确高于市场的表现，从而证明了量化选股策略的强有效性。随后进一步检验与修正了多因子模型，通过相关性系数矩阵检验了因子间没有多重共线性，并通过对比选择不同股票数量对于多因子模型组合绩效表现的影响，确定了组合选择的股票数量。最后将多因子模型与 Alpha 策略进行对比，进一步证明了多因子模型的有效性。

总体来看，量化投资依靠较大的数据量与科学的数理统计相关运算，的确能提供寻求超额收益的有效策略，同时也证明了量化投资在中国股票市场具有很强的可行性与市场潜力，多因子模型只是量化投资策略中的简单一种，还有更多丰富的策略等待广大的投资者去验证与修正。肯定的是，量化投资会随着中国资本市场的复杂深化而日渐普及。

7. 参考文献

[1] Eugene F Fama,Kenneth R French. The Crosssection of Expected Stock Returns[J].The Journal of Finance,1992,(2):427-466.

[2] Eugene F Fama,Kenneth R French. Common risk Factors in the Returns on Stocks and Bonds[J]. Journal of Financial Economics,1993,(93):3-56.

[3] Datar V T, N Y Naik, and R Radcliffe. Liquidity and Stock Returns: An Alternative Test[J]. Journal of Financial Markets, 1998, （1）：203-219.

[4] LEE C M C, SO E C. Alphanomics: The informational under pinnings of market efficiency[J]. Foundations and Trends in Accounting, 2015, 9(2-3): 59-258.

[5] ASNESS, FRAZZINI A, ISRAEI R, etal. Fact, fiction, and momentum investing[J]. Journal of Portfolio Management, 2014, 40(1): 75-92.

[6] KARIYA T. Quantitative methods for portfolio analysis: MTV model approach[M]. Springer Science & Business Media, 2012.

[7] PIOTROSKI J D. Value investing: The use of historical financial statement information to separate winners from losers[J]. Journal of Accounting Research, 2000, 38: 1-41.

[8] MOHANRAM. Separating winners from losers among low book-to-market stocks using financial statement analysis[J]. Review of Accounting Studies, 2005, 10(2-3): 133-170.

[9] ALBADVI A, NOROUZI A. Using downside CAPM theory to improve customer lifetime value prediction in setting[J]. Management Science Letters, 2013, 3(12): 3003-3012.

[10]苏宝通，陈炜，陈浪南.公开信息与股票回报率相关性的实证研究[J].管理科学，2004，（12）：67-75.

[11]陈信元，张田余，陈冬华.预期股票收益的横截面多因素分析：来自中国证券市

场的经验证据[J].金融研究，2001，（6）：22-35.

[12]陈守东，孟庆顺，赵云立．中国股票市场ＦＦ多因子模型的比较分析［Ｊ］．吉林大学社会科学学报, 2003（5）：93-98.

[13]张峥，刘力.换手率与股票收益：流动性溢价还是投机性泡沫？[J].经济学（季刊），2006，（4）：871-892.

[14]朱忆．多因子对冲模型的构建［Ｄ］．上海：复旦大学，2014.

[15]刘辉，黄建山．中国Ａ股市场股票收益率风险因素分析：基于 Fama-French 三因素模型［Ｊ］．当代经济科学， 2013，35（4）：27-31.

[16]江方敏．基于多因子量化模型的Ａ股投资组合选股分析［Ｄ］．成都：西南交通大学，2013.

[17]潘莉，徐建国．Ａ股个股回报率的惯性与反转［Ｊ］．金融研究，2011（1）：149-166.

8. 附录

获取数据

```
import tushare as ts
token = 'f484f317c5f86cee18655a75fde19201dff7975219dce79626999828'
ts.set_token(token)
pro = ts.pro_api()
#获取沪深300成分股
indexs = pro.index_basic(market='SSE')
indexs[140:150]
index_stocks = pro.index_weight(index_code='000300.SH',start_date='20050501',
                                end_date='20220401')
index_stocks.to_csv("..\data\index\CSI300_index_weight.csv")

import pandas as pd
stockDF = pd.DataFrame(data = None, columns=['ts_code', 'trade_date', 'open', 'high',
                                             'low', 'close', 'pre_close',
       'change', 'pct_chg', 'vol', 'amount'])

ts_code_list = list(index_stocks.con_code.unique())

count = 0
for ts_code in ts_code_list:
    stock = ts.pro_bar(ts_code=ts_code, adj='qfq', start_date='20050501',
                       end_date='20220401')
    stockDF = pd.concat([stockDF,stock],axis=0)

    count = count + 1
    print("num:{},process:{:.2%}".format(count,count/len(index_stocks.con_code.unique())))

stockDF.to_csv(("..\data\pro_bar_CSI300.csv"))
```

```
#因子计算
ts_code_list = list(index_stocks.con_code.unique())
#读取每日行情指标
import time
import random
Daily_basic = pd.DataFrame(data = None, columns=['ts_code', 'trade_date', 'close',
                                                 'turnover_rate', 'turnover_rate_f',
                                                 'volume_ratio', 'pe', 'pe_ttm', 'pb',
                                                 'ps', 'ps_ttm', 'dv_ratio', 'dv_ttm',
                                                 'total_share', 'float_share',
                                                 'free_share', 'total_mv', 'circ_mv'])

count = 0
for ts_code in ts_code_list:
    stock = pro.daily_basic(ts_code=ts_code, start_date='20050501', end_date='20220401')
    Daily_basic = pd.concat([Daily_basic,stock],axis=0)

    count = count + 1
    print("num:{},process:{:.2%}".format(count,count/len(ts_code_list)))
    if count%10==0:
        t = random.randint(1,10)
        time.sleep(t)

Daily_basic.to_csv(("..\data\daily_basic_CSI300.csv"))
```

```
#读取现金流量表
Cash_flow = pd.DataFrame(data = None, columns=['ts_code','end_date','n_cashflow_act'])

count = 0
for ts_code in ts_code_list:
    stock = pro.cashflow(ts_code=ts_code, start_date='20050501', end_date='20220401')
    Cash_flow = pd.concat([Cash_flow,stock],axis=0)

    count = count + 1
    print("num:{},process:{:.2%}".format(count,count/len(ts_code_list)))
    if count%10==0:
        t = random.randint(1,10)
        time.sleep(t)

Cash_flow = Cash_flow.loc[:,['ts_code','end_date','n_cashflow_act']]
Cash_flow.to_csv(("..\data\cash_flow_CSI300.csv"))
```

```python
#读取财务指标数据表
Fina_indicator = pd.DataFrame(data = None, columns=['ts_code','end_date','debt_to_assets'
                                                    'nca_to_assets','ebit_to_interest',
                                                    'op_to_ebt','ocf_to_profit','roe',
                                                    'roa','roic','op_of_gr',
                                                    'netprofit_margin',
                                                    'grossprofit_margin',
                                                   'netprofit_yoy','op_yoy','tr_yoy',
                                                    'roe_yoy','assets_yoy','equity_yoy',
                                                    'ocf_yoy','inv_turn',
                                                   'ar_turn','ca_turn','fa_turn'])
count = 0
for ts_code in ts_code_list:
    stock = pro.query('fina_indicator',ts_code=ts_code, start_date='20050501', end_date='
    Fina_indicator = pd.concat([Fina_indicator,stock],axis=0)

    count = count + 1
    print("num:{},process:{:.2%}".format(count,count/len(ts_code_list)))
    if count%10==0:
        t = random.randint(1,10)
        time.sleep(t)

Fina_indicator = Fina_indicator.loc[:,['ts_code','end_date','debt_to_assets',
                                       'nca_to_assets','ebit_to_interest','op_to_ebt',
                                       'ocf_to_profit','roe','roa','roic','op_of_gr',
                                       'netprofit_margin','grossprofit_margin',
                                       'netprofit_yoy','op_yoy','tr_yoy','roe_yoy',
                                       'assets_yoy','equity_yoy','ocf_yoy','inv_turn',
                                       'ar_turn','ca_turn','fa_turn']]

Fina_indicator.to_csv(("fina_indicator_CSI300.csv"))
```

```python
# 沪深指数收盘价
market_close = pro.index_daily(ts_code='000300.SH', start_date='20050501', end_date='20220401')
```

```python
market_close.to_csv(("..\data\market300.csv"))
```

读取数据及数据处理

```r
#############读取数据
cash_flow<-read.csv("cash_flow_CSI300.csv",skip=1)
names(cash_flow)<-c("code","date","n_cashflow_act")
cash_flow$yearmonth<-as.integer(substr(cash_flow$date,1,6))
daily_basic<-read.csv("daily_basic_CSI300.csv",skip=1)
names(daily_basic)<-c("code","date","close","turnover_rate","turnover_rate_f"
                      "volume_ratio","pe","pe_ttm","pb","ps","ps_ttm",
                      "dv_ratio","dv_ttm","total_share","float_share",
                      "free_share","total_mv","circ_mv")
fina_indicator<-read.csv("fina_indicator_CSI300.csv",skip=1)
names(fina_indicator)<-c("code","date","debt_to_assets","nca_to_assets",
                         "roe","roa","roic","op_of_gr","netprofit_margin",
                         "grossprofit_margin","netprofit_yoy","op_yoy",
                         "tr_yoy","roe_yoy","assets_yoy","equity_yoy",
                         "ocf_yoy","ar_turn","ca_turn","fa_turn")
fina_indicator$yearmonth<-as.integer(substr(fina_indicator$date,1,6))
```

```
pro_bar<-read.csv("pro_bar_CSI300.csv",skip=1)
names(pro_bar)<-c("code","date","open","high","low","close","pre_close",
                  "change","pct_chg","vol","amount")
csi300_pool<-read.csv("CSI300_index_weight.csv",skip=1)
names(csi300_pool)<-c("code","weight")

###########转换为季度数据
#######取季度最后一天
daily_basic$month<-as.integer(substr(daily_basic$date,5,6))
daily_basic$yearmonth<-as.integer(substr(daily_basic$date,1,6))
daily_basic$day<-as.integer(substr(daily_basic$date,7,8))
a<-aggregate(daily_basic$day,list(daily_basic$yearmonth),max,na.rm=T)
names(a)<-c("yearmonth","day")
a$date<-paste(a$yearmonth,a$day,seq='')
a$date<-gsub(" ","",a$date)
a$month<-as.integer(substr(a$date,5,6))
a$s<-1*(a$month==3|a$month==6|a$month==9|a$month==12)
sdate<-a[a$s==1,]$date
sdate <- as.data.frame(sdate)
names(sdate)<-c("date")
quarterly_basic<-merge(sdate,daily_basic,all.x=T)
quarterly_basic<-quarterly_basic[order(quarterly_basic$code,
                                       quarterly_basic$date),]

#######合并数据
cash_flow <- cash_flow[,-2] ##谨慎运行
fina_indicator <- fina_indicator[,-2]  ##
data<-merge(quarterly_basic,cash_flow,all=FALSE)
data<-merge(data,fina_indicator,all=FALSE)
###计算因子
data$peg<-data$pe_ttm/data$netprofit_yoy
data$pcf<-data$total_mv/data$n_cashflow_act

########动能
#####季度涨跌幅change也是三个月动能
quarterly_basic$lclose<-unlist(tapply(quarterly_basic$close,
                                      list(quarterly_basic$code),
                                      function(x) c(NA,x[-length(x)])))
quarterly_basic$change<-(quarterly_basic$close-
                           quarterly_basic$lclose)/quarterly_basic$lclose
quarterly_basic$l2close<-unlist(tapply(quarterly_basic$lclose,
                                       list(quarterly_basic$code),
                                       function(x) c(NA,x[-length(x)])))
quarterly_basic$l3close<-unlist(tapply(quarterly_basic$l2close,
                                       list(quarterly_basic$code),
                                       function(x) c(NA,x[-length(x)])))
quarterly_basic$l4close<-unlist(tapply(quarterly_basic$l3close,
                                       list(quarterly_basic$code),
                                       function(x) c(NA,x[-length(x)])))

quarterly_basic$mtm6<-(quarterly_basic$close-quarterly_basic$
                         l2close)/quarterly_basic$l2close
quarterly_basic$mtm12<-(quarterly_basic$close-quarterly_basic$
                          l4close)/quarterly_basic$l4close
```

```
######每季度换手率
turnover<-aggregate(daily_basic$turnover_rate,list(daily_basic$code,
                                                    daily_basic$yearmonth),
                    sum,na.rm=T)
names(turnover)<-c("code","yearmonth","turnover_m")
turnover$month<-as.integer(substr(turnover$yearmonth,5,6))
turnover$year<-as.integer(substr(turnover$yearmonth,1,4))
turnover$yearquater1<-3*(turnover$month==1|turnover$month==2|
                           turnover$month==3)
turnover$yearquater2<-6*(turnover$month==4|turnover$month==5|
                           turnover$month==6)
turnover$yearquater3<-9*(turnover$month==7|turnover$month==8|
                           turnover$month==9)
turnover$yearquater4<-12*(turnover$month==10|turnover$month==11|
                            turnover$month==12)
turnover$quater<-turnover$yearquater1+turnover$yearquater2+turnover$
  yearquater3+turnover$yearquater4

turnover$yearquater<-paste(turnover$year,turnover$quater,seq='')
turnover$yearquater<-gsub(" ","",turnover$yearquater)
turnover_q<-aggregate(turnover$turnover_m,list(turnover$code,
                                               turnover$yearquater),
                      sum,na.rm=T)
names(turnover_q)<-c("code","yearquater","turnover_q")
turnover_q$month<-as.integer(substr(turnover_q$yearquater,5,6))
turnover_q$year<-as.integer(substr(turnover_q$yearquater,1,4))
data$year<-as.integer(substr(data$yearmonth,1,4))
data<-merge(data,turnover_q,all.x=T)
######换手率变动
data$lturnover<-unlist(tapply(data$turnover_q,list(data$code,data$yearmonth),
                              function(x) c(NA,x[-length(x)])))
data$trc<-(data$turnover_q-data$lturnover)/data$lturnover
```

有效因子筛选

```
############因子筛选
#一元线性回归
lpb<-lm(change~pb,data)
lpb<-summary(lpb)
lpe<-lm(change~pe,data)
lpe<-summary(lpe)
ldp<-lm(change~dv_ratio,data)
ldp<-summary(ldp)
lpeg<-lm(change~peg,data)
lpeg<-summary(lpeg)
lpcf<-lm(change~pcf,data)
lpcf<-summary(lpcf)
lps<-lm(change~ps,data)
lps<-summary(lps)
larl<-lm(change~debt_to_assets,data)
larl<-summary(larl)
```

```
lfacr<-lm(change~nca_to_assets,data)
lfacr<-summary(lfacr)
lcmc<-lm(change~circ_mv,data)
lcmc<-summary(lcmc)
lop_of_gr<-lm(change~op_of_gr,data)
lop_of_gr<-summary(lop_of_gr)
lroe_yoy<-lm(change~roe_yoy,data)
lroe_yoy<-summary(lroe_yoy)
lnetprofit_yoy<-lm(change~netprofit_yoy,data)
lnetprofit_yoy<-lsummary(lnetprofit_yoy)
lop_yoy<-lm(change~op_yoy,data)
lop_yoy<-summary(lop_yoy)
ltr_yoy<-lm(change~tr_yoy,data)
ltr_yoy<-summary(ltr_yoy)
lassets_yoy<-lm(change~assets_yoy,data)
lassets_yoy<-summary(lassets_yoy)
lequity_yoy<-lm(change~equity_yoy,data)
lequity_yoy<-summary(lequity_yoy)
locf_yoy<-lm(change~ocf_yoy,data)
lroe<-lm(change~roe,data)
lroe<-summary(lroe)
lroa<-lm(change~roa,data)
lroa<-summary(lroa)
lroic<-lm(change~roic,data)
lroic<-summary(lroic)
lnetprofit_margin<-lm(change~netprofit_margin,data)
lnetprofit_margin<-summary(lnetprofit_margin)
lgrossprofit_margin<-lm(change~grossprofit_margin,data)
lgrossprofit_margin<-summary(lgrossprofit_margin)
lfa_turn<-lm(change~fa_turn,data)
lfa_turn<-summary(lfa_turn)
lca_turn<-lm(change~ca_turn,data)
lca_turn<-summary(lca_turn)
lar_turn<-lm(change~ar_turn,data)
lar_turn<-summary(lar_turn)
lmtm6<-lm(change~mtm6,data)
lmtm6<-summary(lmtm6)
lmtm12<-lm(change~mtm12,data)
lmtm12<-summary(lmtm12)
lturnover_q<-lm(change~turnover_q,data)
lturnover_q<-summary(lturnover_q)
ltrc<-lm(change~trc,data)
ltrc<-summary(ltrc)
```

```
factor_dataframe<-c()
lpb_dataframe<-tidy(lpb)
lpb_dataframe$factor<-c("pb")
factor_dataframe<-rbind(factor_dataframe,lpb_dataframe)
lpe_dataframe<-tidy(lpe)
lpe_dataframe$factor<-c("pe")
factor_dataframe<-rbind(factor_dataframe,lpe_dataframe)
ldp_dataframe<-tidy(ldp)
ldp_dataframe$factor<-c("dp")
factor_dataframe<-rbind(factor_dataframe,ldp_dataframe)
ldataframe<-tidy(lpeg)
ldataframe$factor<-c("peg")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(lpcf)
ldataframe$factor<-c("pcf")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(lps)
ldataframe$factor<-c("ps")

factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(larl)
ldataframe$factor<-c("arl")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(lfacr)
ldataframe$factor<-c("facr")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(lcmc)
ldataframe$factor<-c("cmc")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(lop_of_gr)
ldataframe$factor<-c("op_of_gr")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(lroe_yoy)
ldataframe$factor<-c("roe_yoy")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(lnetprofit_yoy)
ldataframe$factor<-c("netprofit_yoy")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
```

```
ldataframe<-tidy(lgrossprofit_margin)
ldataframe$factor<-c("grossprofit_margin")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(lfa_turn)
ldataframe$factor<-c("fa_turn")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(lca_turn)
ldataframe$factor<-c("ca_turn")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(lar_turn)
ldataframe$factor<-c("ar_turn")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(lmtm6)
ldataframe$factor<-c("mtm6")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(lmtm12)
ldataframe$factor<-c("mtm12")
factor_dataframe<-rbind(factor_dataframe,ldataframe)

ldataframe<-tidy(lturnover_q)
ldataframe$factor<-c("turnover_q")
factor_dataframe<-rbind(factor_dataframe,ldataframe)
ldataframe<-tidy(ltrc)
ldataframe$factor<-c("trc")
factor_dataframe<-rbind(factor_dataframe,ldataframe)

###########输出结果为表格到word
###转换为科学记数法
factor_dataframe1<-format.data.frame(factor_dataframe, scientific = T,digits
factordata <- flextable(factor_dataframe1, col_keys = names(factor_dataframe1
# 使用italic函数将指定表格中的元素变成斜体
factordata <- italic(factordata, j = 4:5, part = "header")
# 使用valign函数用于修改表格元素在竖直方向上是居中，顶端对齐还是底部对齐
factordata <- valign(factordata, valign = "center", part = "header")
# 使用align函数对表格元素在水平方向上的对齐方式进行修改，具体参数与valign一致
factordata <- align(factordata, align = "center", part = "all")
# 使用font函数对表格中元素的字体进行修改
factordata <- font(factordata, fontname = "SimSun", part = "all")
# SimSun中文宋体
factordata <- font(factordata, fontname = "Times New Roman", part = "all")
# 然后使用hline_top函数对三线表最上面的线条进行修改，主要是颜色和线条粗细
factordata <- hline_top(factordata, border = fp_border(color="black",
                                                       width = 1.5), part = "
# 使用hline_bottom函数对三线表中最下面的线条进行修改
factordata <- hline_bottom(factordata, border = fp_border(color="black",
                                                          width = 1.5), part
# 然后使用hline函数对三线表中间的那条线进行修改
factordata <- hline(factordata, i = 1, border = fp_border(color="black",
                                                          width = 1),
              part = "header")
# 使用函数set_caption设置三线表的标题
factordata <- set_caption(factordata, "设置三线表标题", autonum = 1,
                          style = "Table Caption")
# 使用autofit函数对生成的三线表进行自动调整
factordata <- autofit(factordata)
# 最后，使用save_as_docx函数将生成的三线表对象导出到word文档中
save_as_docx("三线表结果" = factordata, path = "因子简单回归.docx")
```

```
#############描述性统计
library(psych)
describe<-describe(data)
write.csv(describe,"描述性统计.csv")

#############打分法进一步筛选有效因子
cols_remain<- c("code","yearmonth","year","month","date","close","change",
                "pb","pe","dv_ratio","ps","roe_yoy","op_yoy","equity_yoy",
                "assets_yoy","roe","roic","mtm6","mtm12","turnover_q")
newdata<- data[ ,colnames(data) %in% cols_remain]

meandata<-aggregate(newdata,list(newdata$code),mean,na.rm=T)
dafen<-c()
#pb
meandata<-meandata[order(meandata$pb, decreasing= T), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("pb")
dafen<-rbind(dafen,meanmtm12)
#dv_ratio
meandata<-meandata[order(meandata$dv_ratio), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("dv_ratio")

dafen<-rbind(dafen,meanmtm12)
#pe
meandata<-meandata[order(meandata$pe, decreasing= T), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("pe")
dafen<-rbind(dafen,meanmtm12)
#ps
meandata<-meandata[order(meandata$ps, decreasing= T), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("ps")
dafen<-rbind(dafen,meanmtm12)
#roe_yoy
meandata<-meandata[order(meandata$roe_yoy, decreasing= T), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("roe_yoy")
dafen<-rbind(dafen,meanmtm12)
```

```r
#op_yoy
meandata<-meandata[order(meandata$op_yoy, decreasing= T), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("op_yoy")
dafen<-rbind(dafen,meanmtm12)
#equity_yoy
meandata<-meandata[order(meandata$equity_yoy, decreasing= T), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("equity_yoy")
dafen<-rbind(dafen,meanmtm12)
#assets_yoy
meandata<-meandata[order(meandata$assets_yoy, decreasing= T), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("assets_yoy")
dafen<-rbind(dafen,meanmtm12)
```

```r
#roe
meandata<-meandata[order(meandata$roe, decreasing= T), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("roe")
dafen<-rbind(dafen,meanmtm12)
#roic
meandata<-meandata[order(meandata$roic, decreasing= T), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("roic")
dafen<-rbind(dafen,meanmtm12)
#mtm6
meandata<-meandata[order(meandata$mtm6, decreasing= T), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("mtm6")
dafen<-rbind(dafen,meanmtm12)

#mtm12
meandata<-meandata[order(meandata$mtm12, decreasing= T), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("mtm12")
dafen<-rbind(dafen,meanmtm12)
#turnover_q
meandata<-meandata[order(meandata$turnover_q, decreasing= T), ]
meandata100<-meandata[1:101,]
meanmtm12<-describe(meandata100$change)
row.names(meanmtm12)<-c("turnover_q")
dafen<-rbind(dafen,meanmtm12)
dafen<-round(dafen,4)
dafen<-dafen[order(dafen$mean, decreasing= T), ]
write.csv(dafen,"简单打分法2.csv")
```

```
#####每类因子选一个，最后选择mtm6,op_yoy,roe,dv_ratio
############提取数据
cols_remain<- c("code","yearmonth","year","month","date","close","change",
                "mtm6","op_yoy","roe","dv_ratio")
finaldata1<- data[ ,colnames(data) %in% cols_remain]
write.csv(finaldata1,"选股最初数据.csv")
#######数据处理，删除缺失值，z-score标准化
finaldata<-na.omit(finaldata1)
finaldata$mtm6<-scale(finaldata$mtm6)
finaldata$op_yoy<-scale(finaldata$op_yoy)
finaldata$roe<-scale(finaldata$roe)
finaldata$dv_ratio<-scale(finaldata$dv_ratio)
write.csv(finaldata,"shuju4.csv")
```

构建多因子模型

```
####构建多因子模型
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import time
import matplotlib
from sklearn import linear_model
import statsmodels.api as sm
matplotlib.rcParams['axes.unicode_minus'] = False
plt.style.use("seaborn")
###读取处理过的数据
daily_basic = pd.read_csv("shuju4.csv",index_col=["date"],parse_dates=True)
daily_basic.sort_index(inplace=True)
daily_basic.shape
daily_basic['next quarter ret'] = daily_basic.groupby('code')['change'].shift(-1)
#获取交易日期
trade_date = daily_basic.index.unique()
start = trade_date[0]
end = trade_date[-1]
start,end
# 获取季度日期
datelist = list(trade_date.strftime('%Y%m%d'))
quarter_list = []
for s in range(1,len(datelist)):
    if datelist[s-1][4:6] != datelist[s][4:6]:
        quarter_list.append(datelist[s])
quarter_list = pd.to_datetime(datelist)
```

```
def get_index_stocks(daily_basic,quarter):
    # 获取某个时间点的所有成分股
    stock_pool = daily_basic.loc[quarter]["code"]
    return stock_pool
t1,t2 = quarter_list[0],quarter_list[4]
stock_pool = get_index_stocks(daily_basic,t1)
# 获取因子数据
def get_fundamentals(quarter):
    # 获取季初时间点所有股票的股息率数据
    df = daily_basic.loc[quarter,["code","next quarter ret","mtm6","op_yoy","roe","dv_ratio"]]
    # 更换index
    df.index = df["code"]
    return df
rundf=get_fundamentals(t1).dropna()
```

```
#########循环获取所有的的数据
# 创建数据字典
data_dict = {}
# 获取本次研究所需要的股息率数据
for s in range(len(quarter_list)-1):
    # 当前初始
    quarter = quarter_list[s]
    # 下期截止
    quarter_next = quarter_list[s+1]
    # 动态进度
    print('\r 当前：{} --> 总量：{}'.format(quarter,quarter_list[-1]),end='')
    # 获取quarter开始的股息率
    df = get_fundamentals(quarter).dropna()

    # 储存
    data_dict[quarter] = df
```

```
########基于因子选股并做投资组合
data_ret_df = pd.DataFrame(columns = ['selection high',"selection low"],dtype=object)

# 初始为0
data_ret_df.loc[quarter_list[0]] = 0
# 循环全周期
for s in range(len(quarter_list)-1):
    # 当期初始
    quarter = quarter_list[s]
    # 下期截止
    quarter_next = quarter_list[s+1]
    # 复制数据表data_dict[quarter]
    lsdf = data_dict[quarter].copy()
    lsdf = lsdf.sort_values(by='op_yoy',ascending = True)
    lsdf['op_yoy_fenshu'] = range(len(lsdf))
    lsdf = lsdf.sort_values(by='mtm6',ascending = True)
    lsdf['mtm6_fenshu'] = range(len(lsdf))
    lsdf = lsdf.sort_values(by='roe',ascending = True)
    lsdf['roe_fenshu'] = range(len(lsdf))
    lsdf = lsdf.sort_values(by='dv_ratio',ascending = False)
    lsdf['dv_ratio_fenshu'] = range(len(lsdf))
    lsdf['fenshu'] = lsdf['dv_ratio_fenshu']+lsdf['roe_fenshu']+lsdf['mtm6_fenshu']+lsdf['op_yoy_fenshu']
    # 筛选
    lsdf_high = lsdf.sort_values(by='fenshu',ascending = False).iloc[:int(len(lsdf)*0.05)]
    lsdf_low = lsdf.sort_values(by='fenshu',ascending = False).iloc[int(len(lsdf)*0.95):]

    # 计算收益率均值
    data_ret_df.loc[quarter_next,["selection high"]] = lsdf_high.mean()['next quarter ret']
    # 资金等额分配到选出的股票池中
    data_ret_df.loc[quarter_next,["selection low"]] = lsdf_low.mean()['next quarter ret']
    # 资金等额分配到选出的股票池中
```

```
###基于因子分组的投资组合业绩比较
market_close = pd.read_csv("quatermarket.csv",index_col=["date"],parse_dates=True)
market_close['next quarter ret'] = market_close.groupby('code')['change'].shift(-1)
data_ret_df["market"] = market_close["next quarter ret"]
net_value_df = (data_ret_df+1).cumprod()
#作图
figsize = (10,6)
fontsize=12
data_ret_df.plot()
plt.title('quarter return',fontsize=fontsize)
net_value_df.plot()
plt.title('net values',fontsize=fontsize)
(net_value_df['selection high'] / net_value_df['market']).plot(kind='line',secondary_y=['alpha'],figsize = figsize
plt.title('alpha',fontsize=fontsize)
plt.legend(['alpha'])
```

```
#######对比绩效表现
# 绩效指标
columns = ['净值','年化收益率 % ','最大回撤 % ','夏普率','年化波动率']
# 创建表格
finace_ret_df = pd.DataFrame(columns = columns)
# 再次定义函数：计算最大回撤
def maxdrawdown(arr):
    '''
    输入：净值序列
    输出：最大回撤
    '''
    # 最大回撤结束点
    i = np.argmax((np.maximum.accumulate(arr) - arr)/np.maximum.accumulate(arr))
    # 开始点
    j = np.argmax(arr[:i]) # start of period
    # 输出回撤值
    return (1-arr[i]/arr[j])
```

```
# 设置函数：计算净值曲线的绩效指标
def get_Performance_analysis(T,benchmark,year_day = 4):
    '''
    输入：净值序列 和基准净值序列

    输出：绩效指标
    '''
    # 如果基准净值序列长度为0
    if len(benchmark) == 0:
        # 基准序列等于策略净值序列
        benchmark = T
    net_values = round(T[-1],4)# 获取最终净值
    year_ret_mean = T.pct_change().dropna().mean()*year_day# 计算算术年化收益率
    year_ret_mean = round(year_ret_mean*100,2)
    # 计算几何年化收益率
    year_ret_sqrt = net_values**(year_day/len(T))-1
    year_ret_sqrt = round(year_ret_sqrt*100,2)
    # 计算年化波动率
    volitiy = T.pct_change().dropna().std()*np.sqrt(year_day)
    volitiy = round(volitiy*100,2)
    #计算夏普，无风险收益率记3%
    Sharpe = (year_ret_sqrt - 3)/volitiy
    Sharpe = round(Sharpe,2)
    # 计算超额收益率
    alpha = T[-1]/benchmark[-1]-1
    alpha = round(alpha*100,2)
    # 计算最大回撤
    downlow = maxdrawdown(T)
    downlow = round(downlow*100,2)
    # 输出
    return [net_values,year_ret_sqrt,downlow,Sharpe,volitiy]
```

```
# 调用绩效分析函数
finace_ret_df.loc['selection high'] = get_Performance_analysis(net_value_df['selection high'],[])
finace_ret_df.loc['selection low'] = get_Performance_analysis(net_value_df['selection low'],[])

finace_ret_df.loc['bench mark'] = get_Performance_analysis(net_value_df["market"],[])
finace_ret_df
```

与 Alpha 策略对比

```
#读取数据并处理
data = pd.read_csv("..\data\shuju5.csv",parse_dates=['date'])
market_close = pd.read_csv("quatermarket.csv",parse_dates=['date'])
market_close.drop(['close','lr','Unnamed: 0','code'],axis=1,inplace=True)
market_close.rename(columns={'change':"market"}, inplace = True)
data = pd.merge(data,market_close)
data.set_index(['date'],inplace = True)
```

```
#计算Alpha
grps = data.groupby(['code'])
df_alpha = pd.DataFrame()
for fid,grp in grps:
    rt = grp.loc[:, 'change']
    net_value_rt = pd.DataFrame(rt)
    net_value_rt = net_value_rt.sort_index()
    net_value_rt['1rt'] = net_value_rt['change'].shift(1)
    net_value_rt['2rt'] = net_value_rt['change'].shift(2)
    net_value_rt['3rt'] = net_value_rt['change'].shift(3)
    net_value_rt['4rt'] = net_value_rt['change'].shift(4)
    net_value_rt['5rt'] = net_value_rt['change'].shift(5)
    net_value_rt['6rt'] = net_value_rt['change'].shift(6)
    net_value_rt['net_rt'] = net_value_rt['1rt']+net_value_rt['2rt']+net_value_rt['3rt']+net_value_rt['4rt']
    rm = grp.loc[:, 'market']
    net_value_rm = pd.DataFrame(rm)
    net_value_rm = net_value_rm.sort_index()
    net_value_rm['1rt'] = net_value_rm['market'].shift(1)
    net_value_rm['2rt'] = net_value_rm['market'].shift(2)
    net_value_rm['3rt'] = net_value_rm['market'].shift(3)
    net_value_rm['4rt'] = net_value_rm['market'].shift(4)
    net_value_rm['5rt'] = net_value_rm['market'].shift(5)
    net_value_rm['6rt'] = net_value_rm['market'].shift(6)
    net_value_rm['net_rm'] = net_value_rm['1rt']+net_value_rm['2rt']+net_value_rm['3rt']+net_value_rm['4rt']
    alpha = net_value_rt['net_rt']/net_value_rm['net_rm']
    alpha = pd.DataFrame(alpha)
    alpha['code'] = fid
    df_alpha = df_alpha.append(alpha)
df_alpha.rename(columns={0:"alpha"}, inplace = True)
```

```
#构建Alpha策略
daily_basic = pd.merge(data,df_alpha,on=['code','date'],how='inner')
daily_basic.set_index(['date'],inplace = True)
daily_basic.dropna(inplace = True)

data_ret_df = pd.DataFrame(columns = ['selection high',"selection low"],dtype=object)

# 初始为0
data_ret_df.loc[quarter_list[0]] = 0
# 循环全周期
for s in range(len(quarter_list)-1):
    # 当期初始
    quarter = quarter_list[s]
    # 下期截止
    quarter_next = quarter_list[s+1]

    # 复制数据表data_dict[quarter]
    lsdf = data_dict[quarter].copy()
    lsdf = lsdf.sort_index()
    # 筛选
    lsdf_high = lsdf.sort_values(by='alpha',ascending = False).iloc[:int(len(lsdf)*0.1)]
    lsdf_low = lsdf.sort_values(by='alpha',ascending = False).iloc[int(len(lsdf)*0.9):]

    # 计算收益率均值
    data_ret_df.loc[quarter_next,["selection high"]] = lsdf_high.mean()['next quarter ret']
    # 资金等额分配到选出的股票池中
    data_ret_df.loc[quarter_next,["selection low"]] = lsdf_low.mean()['next quarter ret']
    # 资金等额分配到选出的股票池中
```

9. 致谢

在论文完成之际，我要特别感谢我的指导教师，在我作品完成的过程中，无论是在论文的选题、构思和资料的收集方面，还是在论文的研究方法以及成文定稿方面，我都得到了老师悉心细致的帮助和教诲，特别是他深厚的学术素养、严谨的治学态度和一丝不苟的敬业精神让我终身受益。即使在排满课的情况下，也能耐心的抽时间给予我很多的建议和帮助。从而能一次次的帮我审阅并提出建议，从而使成文得以展现。再次谢谢我的指导老师。

在接下来的日子，我会继续完善我的专业素养，提高自身实践能力，开拓思路，锐意进取，独立创新。在影视制作方面我的能力尚浅，还有许多不完善之处，恳请各位批评指正。