上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

BACHELOR'S THESIS

论文题目：算法交易对金融市场微观结构的影响研究

学生姓名: 王舒

学生学号: 5131209068

专　　业: 金融学

指导教师: 周志中

学院(系): 安泰经济与管理学院

算法交易对金融市场的微观结构影响研究

摘要

本课题通过使用事件研究法借助一次在中国金融市场中影响较大的事件对算法交易者在金融市场中扮演的地位以及其对金融市场微观结构的影响进行了深入研究，使用了针对金融市场微观结构的金融计量模型对事件前后的市场质量和诸多微观性质进行了建模分析，同时结合了宏观上对投资者成分和其行为变化特征的讨论，得出了关于在当时的市场状况下算法交易者对股指期货市场和股票市场微观结构影响的结论。

从 2015 年 9 月中国金融期货交易所对股指期货交易进行严格限制出发，我们首先明确了政策的内容和对相应的受影响的投资者性质和行为特征变化进行分析，认为主要受影响的投资者主体为使用算法交易技术的阿尔法套利者。在定性分析中对市场质量的诸多特征包括价格发现功能、价格有效性和稳定性、市场流动性和潜在波动风险等的变化进行了初步的推断，并在此基础上使用针对高频交易数据特性设计的金融计量模型，包括一系列价格变动模型和动态久期模型，以及已实现波动率分析方法对当时所有股指期货主力合约和一部分股票的高频交易数据进行了建模分析，得到了和预期一致的结果，支持了宏观定性分析的结论，认为算法交易者在金融市场中尤其是在股指期货市场中的行为对市场的价格发现功能以及市场的有效性和稳定性均有正面作用，在算法交易者受限离场后，市场质量受到了明显的损害，同时在期现市场均产生了更大的价格波动风险。

关键词：股指期货 市场微观结构 价格变动模型 条件自回归久期模型 算法交易

RESEARCH ON INFLUENCES OF ALGORITHMIC TRADING ON MARKET MICROSTRUCTURE OF CHINESE FINANCIAL MARKETS

ABSTRACT

This project studies the influences of algorithmic trading on market microstructure of Chinese capital markets in depth based on a seires of significant restriction policy events that happened in August and September, 2015. We employed special financial econometrical models and time series analysis approaches especially designed for studying high-frequency data, and analyzed the changes of characteristics of market microstructure based on the results generated from the modeling process. Started with the restrictions on algorithmic traders' behavior in the stock index futures' market, we first identified the details of the policy changes, the formations of the active investors and the patterns of their behavior, the possible influences and the particular models we should use for the scenario, then, based on the results of price changes models, dynamic duration models and the analysis of realized volatilities, we proved that algorithmic traders have positive influences on markets' qualities, stability, effectiveness and function of price discovering, after their departure, the stock futures market's quality was deeply impaired and more risks from abnormal price concussions were triggered in both stock index futures market and stock market.

Key words: Stock Index Futures, Market Microstructure, Price Changes Model, ACD Model, Algorithmic Trading

目 录

第一章 绪论 ----1
第二章 文献综述 ----3
第三章 高频数据模型设计 ----5
3.1 高频数据特性和数据处理 ----5
3.2 基本变量设定 ----6
3.3 价格变动模型 ----7
3.3.1 顺序概率值模型 ----7
3.3.2 ADS 价格变动分解模型 ----9
3.3.3 已实现波动率分析 ----11
3.4 久期模型 ----13
3.4.1 日模式的调整 ----14
3.4.2 ACD 自回归条件久期模型 ----16
3.5 本章小结 ----17
第四章 实证研究 ----18
4.1 市场功能、投资者成分和事件分析 ----19
4.1.1 股指期货市场的功能 ----19
4.1.2 金融市场参与者组成成分分析 ----19
4.1.3 股指期货市场限制政策 ----21
4.2 研究框架 ----22
4.2.1 研究标的和时间窗的选取 ----22
4.2.2 主要研究方案 ----22
4.3 股指期货市场微观结构的变化 ----23
4.3.1 基本微观结构描述 ----23
4.3.2 已实现波动率平面和隐含价格波动条件概率 ----28
4.3.3 价格变化分解分析 ----33
4.3.4 久期动态特性变化----37
4.4 股票现货市场微观结构的变化 ----41
4.5 本章小结 ----44
第五章 结语 ----45
参考文献 ----47
附录：部分核心代码 ----48
preprocessors.R ----48
models.R ----56
plotting.R ----60
fetch_stocks.py ----62
test.R ----63
谢辞 ----66

第一章 绪论

算法交易在 20 世纪末在国外金融市场中已开始广泛使用，近年来在中国金融市场上的应用也开始了快速发展，尤其在各类期货市场上的占有率逐年增加。2014～2015 年的牛市启动吸引了许多海外归国的量化金融人才，带来了许多量化交易和算法交易的技术和理念，越来越多的机构投资者、共同基金、私募基金、对冲基金以及许多证券公司的自营部门都开始使用量化交易平台进行程序化交易，同时随着计算机科技的日益进步、国内金融市场的稳定发展和金融产品种类和功能的不断完善，使用算法交易支持投资策略实施的市场参与者比例在近两年达到了非常高的水平，他们对市场运行和市场质量也造成了越来越大的影响。在这样的背景下，研究算法交易者对金融市场的影响对监管部门和广大投资者都有较大的实际意义。

通常来说，计算机程序化或自动化交易分为交易决策生成和交易需求执行两部分，其中交易决策生成部分主要是由投资公司的量化研究部门对市场规律进行挖掘，并基于研究所得的量化交易策略得到，主要提供的是决定要交易的标的金融资产和其目标权重等信息，这部分通常称为量化交易；而交易需求执行则是在从量化交易策略得到标的金融资产的交易需求后，使用计算机中特定的算法对交易需求进行执行和实现，同时力求在尽可能短的时间内、使交易成本尽可能小的要求下成功进行调仓操作，这部分则通常称为算法交易。可以看出，量化交易主要决定哪些资产称为投资标的，而真正对市场微观结构造成影响的则为执行交易的算法交易技术。算法交易的根本目的在于尽可能降低交易成本，主要方式是通过对市场微观结构性质的把握使用特定算法计算出最优的执行方式，通过订单拆分的方法进行交易。此外，算法交易还有隐藏交易目的和意图等其他作用。总而言之，从算法交易技术的实施目的和实施方法上可以看出，其广泛使用对市场运行必定有较大的影响，但究竟如何认识他们在市场中扮演的角色是当前市场监管者需要解答的问题。

在股指期货的推出后，投资者们有了对现货市场的系统性风险进行对冲的工具，同时也有了新的投机标的，于是短短的几年里股指期货市场的发展非常迅速，总持仓量和成交量均逐年增大。在股指期货投资者中有套利者、套期保值者和纯投机者。其中套利者又分为期现套利者和主动套利者两种，而后者往往被称为阿尔法套利者。期现套利者的行为模式较简单，主要是借助期现价差回归的市场特性，对股票指数和股指期货的期现价差进行套利，即在对应合约交割日期前，利用价格差做空高价资产做多低价资产，在价格回归时予以平仓获利。由于期现套利者在现货市场主要是通过构建和股票指数尽可能相近的投资组合以减少价格回归的不确定性，其投资风格相对消极，又称为贝塔套利者。相比之下，另一种套利者则更加主动，即阿尔法套利者，主要通过量化投资策略的研究以及投资者本身对市场的认识来构建独特的超额收益资产组合，同时使用股指期货来对冲市场的系统性风险，以此赚取稳定的超额收益。套期保值者的交易手法和套利者相似却不完全同，尽管都是同时持有现货和期货，但其主要投资标的为现货资产，在期货市场中参与交易是为了对冲现货资产的系统性风险暴露，其交易频率和强度相比起套利者而言均较低。纯投机者为脱离现货仓位进行期货的投资，仅仅依据其对市场走势的判断对期货合约进行交易，由于期货市场的杠杆较高，此类投资者的系统性风险暴露巨大。综合以上讨论，在股指期货市场中，主要进行量化交易和算法交易的即为阿尔法套利者，而贝塔套利者虽然少有使用量化交易进行选股构建投资组合，也会使

用算法交易来执行其频繁的调仓需求，这两类交易者的存在是股指期货市场投资者的主要力量，也是 2015 年 9 月中金所对股指期货交易进行限制后受限最大的群体，因此，他们的交易行为是本课题的主要研究对象之一。

本课题的研究目的是在市场的微观层面通过对高频交易数据的挖掘和分析来明确算法交易者对金融市场的影响。一直以来，人们对算法交易和计算机程序化交易行为对市场的影响争议颇多，我们尝试通过金融计量模型的定量分析回答以下问题：算法交易者的频繁交易是否给市场带来了不必要的震荡和额外的波动风险？抑或是恰巧相反，通过频繁的交易减少了不必要的震荡并使得价格变化更加均匀平缓？监管层对股指期货市场进行交易限制对市场本身的稳定有何作用？限制双边市场套利者的行为会对期货市场和现货市场分别造成哪些冲击？如果有，造成冲击的范围是什么？应该如何认识使用量化交易和算法交易技术的套利者在市场中的作用和性质？我们认为，在很好地回答了上述问题后，监管部门能够在未来对不同市场参与者的行为模式以及其在市场中扮演的角色来相应地制定监管政策，以此来维护金融市场的稳定发展，在降低市场异动风险、提高市场定价机制和效率的同时避免对市场质量造成损害。

本课题主要研究的变量是算法交易者在市场中的参与程度，突破口在于 2015 年 9 月中国金融期货交易所对股指期货交易进行的严格限制，由于在限制政策颁布前后算法交易者的市场参与度和行为模式在短时间内发生了较大改变，使得控制其他变量在较短的时间区间内研究市场微观结构变化并和算法交易者联系起来成为可能。在对高频交易价格变动量时间序列和价格变动久期时间序列使用顺序概率值模型、ADS 价格变动分解模型、ACD 自回归条件久期模型和基于不同时间间隔的实际波动率进行建模和分析后，发现了所有股指期货市场中交易最活跃的主力合约和股票现货市场中一部分的指数成分股和非成分股的市场质量均在股指期货交易受限后发生了明显变化，表现为交易活跃度降低、交易成本增加、市场局部稳定性降低、市场微结构噪声增加、价格变动幅度整体波动增加和市场价格有效性降低等。在研究过程中，我们首先对模型的变量设定、针对的高频交易数据特性、模型假设和参数估计等方法进行了深入讨论，然后在实证研究中针对已有数据的特性和技术限制对模型进行结构上的小幅改动，使用模型进行拟合后得到了非常显著的参数和有效的模型，模型的有效性通过对拟合后的残差进行检验来确定，在对有效模型的结构和参数进行分析和解释后，进一步利用拟合的模型计算了市场微观结构的描述变量，例如价格变化的条件概率等。在事件研究的过程中，考虑到在 2015 年 8、9 月颁布了两次对股指期货交易的限制政策，而两次政策力度不同，对市场造成的影响也不同，将事件前后 10 天左右的时间和事件当中的时间整合成三个子时间窗，并且合称全事件时间窗，分别对三个子时间窗中市场微观结构的性质进行建模研究和深入讨论。我们发现，在算法交易者受限后，市场的局部震荡加剧、价格发现功能受损、市场效率降低、价格变动的集束性增加，同时股指期货市场和现货市场均受到影响。从中我们得出结论，算法交易者是沟通期货市场和现货市场的桥梁，其交易行为和在市场中的活跃参与是期货市场价格发现功能的重要支持，对市场价格有效性和价格变化稳定性有正面作用，同时也是市场上流动性的主要提供者。在现货市场中算法交易者虽然由于现货市场交易额巨大，其行为变化的影响不如其对期货市场的影响明显，却也扮演着提供市场流动性和稳定价格的重要角色，其离场和交易模式变化的确对一部分股票现货造成了明显的影响。

第二章 文献综述

在金融计量学领域已有许多研究人员对针对高频价格变化和交易久期数据设计的模型进行了深入讨论。使用特定的模型对高频交易数据进行分析，相比起简单地利用每日数据的简单统计量对数据特性进行描述而言，可以更充分利用日内数据的信息，更加精确地描述目标序列在交易日内不同时刻和不通时段中的行为表现，同时捕捉微观层面上目标序列值和诸多滞后值之间的相关关系，能够更好反映出高频交易数据的诸如价格波动性、价格变动集束性、价格回转等性质微观层面上的表现。

Easley (1996) 、Diamond 和 Verrecchia (1987)、 Hasbrouch (1998) 和 O'Hara (1995) 等在研究市场微观结构时指出，价格变化和逐笔成交之间的价格持续时间长度序列对挖掘金融市场信息流密度和投资者行为特点具有关键作用。为了使用交易久期数据来支持市场微观结构性质的研究，Engle 和 Russell (1998) 首次提出了自回归条件久期模型(Autoregressive Conditional Duration Model, ACD)，获得了很好的效果。此后许多研究者对ACD模型进行了若干的调整和改善，包括使用了不同的假设等，并将其应用在其他高频时间序列的研究上，也取得了很好的效果。ACD模型能够很好捕捉新信息出现之间的久期的隐含动态关系，从信息流密度和交易密度的微观层面描述市场的微观状态。此外，有许多针对价格变动量的高频数据的模型，例如 Hauseman (1992) 在研究高频交易数据时提出的顺序概率值模型(Ordered Probit Model) 以及由 McCulloch 和 Tsay (2000) 提出的ADS分解模型，可以分别对每一次观测到的数据的隐含价格变化和价格变化的各种条件概率进行估计和预测，充分利用和反映了逐笔交易数据中的信息。此外，还可以使用已实现波动率计算在不同的时间间隔下高频数据的波动率，从而在不同的时间尺度上认识市场的波动以及市场微观结构噪声的变化。总而言之，使用上述几个模型能够帮助我们认识到高频交易数据中价格变动在时间空间上的集束性、回转性、局部和整体波动性和市场微结构噪声等性质，

在使用高频交易数据模型对中国金融市场进行实证分析方面，国内相关研究并不是特别多。刘向丽和成思危 (2012) 使用ACD模型对中国期货市场的波动性进行了研究，他们使用了中国商品期货市场合约的1分钟交易数据进行分析，得到了久期和价格的波动性之间存在负相关关系等结论。陈敏和王国明 (2003) 使用了ACD-GARCH模型对中国证券市场的久期特性进行了实证分析，使用的是单一股票的1分钟交易数据，研究了波动率的动态特性 。对于股指期货的量化高频交易数据研究目前国内还较少，王春玲 (2006) 对股指期货市场的基本功能和其与股票现货市场的关系进行了详细讨论，王宇超和李心丹 (2014) 使用仿真建模的方法研究了算法交易者的交易行为对市场的影响，即首先对算法交易者使用的策略进行设定，然后通过模拟的方法研究市场运行是如何受算法交易者影响的。在股指期货受限后，有许多学者表示，这样的严格限制对市场功能会造成一定损害，因为套利者的成交意愿降低后，期货市场和现货市场的联系便难以维持，但使用模型对市场微观结构特性进行分析来支持此类结论的研究则非常少。

相比起过去的研究，本课题的不同和创新之处在于：

(1) 使用了更加细颗粒的高频交易数据，得以对市场微观结构性质进行更细致的刻画，同时使用了有针对性的高频数据模型，对包括市场波动性、局部震荡、市场价格有效性、市场微观结构噪声等特性均进行了详细的讨论。在保证真实性的同时，还选

取了股指期货主力合约作为主要研究对象之一，有很好的代表性，使得模型可以充分反映出算法交易者对市场整体的影响。

(2) 相比起使用仿真建模和基于许多对算法交易者行为模式和市场性质的假设进行的研究，本课题在分析算法交易对市场微观结构的影响时，使用的是在市场中大事件发生时的真实交易数据，同时对算法交易者具体采取的成交策略不进行任何假设，从宏观上分析算法交易者整体的行为对市场的影响。2015年严格限制政策的实施给我们提供了非常好的条件和契机，使得我们得以用真实数据对算法交易者在整个市场中的地位，尤其是其交易行为对市场微观结构的影响进行深入研究。

(3) 对于数据抓取的技术限制，本课题对模型的设定和假设根据具体使用背景进行了合理的修改，成功将针对逐笔交易数据的计量模型应用到中国的金融市场实证研究中，拟合出的参数显著且模型有效。

总而言之，本课题结合了微观结构上高频交易的实证建模研究和宏观上的事件影响分析，能够帮助投资者和监管部门进一步认识算法交易者在市场中扮演的角色，尤其对监管部门监管量化交易和程序化交易者有很好的参考价值，能够帮助其更好地通过合理监管来保证金融市场的基本功能和促进金融市场的稳定发展。

第三章 高频数据模型设计

本课题主要从市场微观结构性质出发通过事件研究法探究算法交易投资者在市场中扮演的角色和地位，使用的模型主要用于处理高频交易数据。本章主要讨论本课题在研究过程中使用到的应用于高频数据分析的金融计量模型。内容包括高频数据特性、基本的变量设定、各模型的基本设定、假设和参数估计方法等。在针对实际数据使用时，考虑到数据收集的技术限制，具体的变量含义会根据实际情况的不同而有所调整，详见实证研究一章。

3.1 高频数据特性和数据处理

高频交易数据在研究微观结构、交易竞价过程与价格发现机制及其有效性中非常重要，其具有许多与普通低频交易数据不同的特殊性质。通常来说，高频交易数据指细颗粒的逐笔成交交易数据，或是数据快照获取间隔低至几秒甚至几百毫秒的市场价格、成交量、成交额、交易价差、价格变动、市场宽度深度以及相关的度量方式的数据。此类数据和每日或每小时统计的低频交易数据相比通常具有如下特征：

(1) **不等距，具有动态特性的久期**：和普通的每日或每小时数据不同，逐笔交易数据通常不是等距的，由于交易发生之间的间隔不相同，交易久期（即逐笔交易之间的时间间隔）的动态特性可以对研究市场微观结构特性提供有用信息。

(2) **价格变动久期存在自相关关系**：在实证研究中我们发现价格变动久期呈现出强烈且持久的自相关关系，较长的价格变化时间间隔更可能接着一个或几个较长的时间间隔。这样的性质和价格变化的离散性以及集束性紧密相关。它们所包含的信息共同反映出了信息流密度和交易密度等市场微观结构的动态特性。

(3) **价格变动久期显著的日模式**： 通常而言，由于交易时间不连续性，在市场关闭时和股价的有用信息仍在积累，同时又有由于休市期间价格不能及时反应信息而产生的风险在全球大多数股票和期货交易市场中，开盘后和收盘前的交易强度要明显大于午休时段前后的交易强度，从而标的金融资产的价格变化、成交量、波动率以及价格变化与交易久期等量均表现出明显的日周期模式，即数据呈现出“U”形或倒“U”形。

(4) **价格变化值的离散性和集束性**：受限于交易系统的设计和报价机制，投资者通常观测到的是离散的价格变化，但实际上标的金融资产的实际价格可能是连续的。在 NYSE 中，交易价格变动的最小单位为十六分之一美元；在上海证券交易所，最小报价一般为 0.01 人民币；在股票指数期货交易市场，期指标的价格变动的最小单位为 0.2 人民币。同时，在实证研究中发现，高频交易数据的价格变化值、波动性以及久期都呈现出集束特性，即较大的值往往接连发生，意味着价格变化间复杂的相关性。

(5) **数据截取间隔内多重交易**：美国的 TAQ 数据能够提供毫秒级的逐笔交易成交信息，但通常而言获取的高频交易数据无法达到如此高的精度。本课题使用的已经是中国金融市场中抓取频率最高的数据，在研究股指期货市场时所采用数据的抓取间隔是 0.5 秒，在研究股票现货市场时数据快照截取间隔为 3 秒。受限于技术实现，在市场交易极度活跃时，3 秒甚至是 500 毫秒的间隔的细度显然还不够，故在数据截取

间隔中会出现多笔交易，这些交易大部分是在同一价格完成，同时也有一部分交易在不同价格完成同时改变了资产价格，对此类问题会在后面详细讨论。

(6) **数据量巨大**：高频交易数据量巨大，通常一个产品一天的高频交易数据信息可以达到几千至几万条记录，在选取含有数日甚至数周的窗口数据进行分析时，需要一次性使用金融计量模型处理几十万条信息。

承上讨论，设计用于处理高频交易数据的时间计量模型应该充分考虑到高频数据的特征，同时在进行数据预处理的时候应该结合数据特征和模型需求来进行相应的处理或使用特殊的假定进行建模分析和计算优化，着重挖掘数据中隐含的能够对描述市场微观结构性质有帮助的信息。

本课题数据来源为金数源金融高频数据库以及基于 Python 的 Tushare 金融财经数据包，数据抓取主要使用 R 和 Python 的批处理脚本实现，搜集到的数据有各股指期货合约的高频价格数据和实时市场快照数据，有实时的买卖盘五档上的价格和挂单量、实时交易量、交易额快照和以秒为单位的时间戳等。数据清理、整理和变量生成等操作函数已封装在专门为本课题设计编写的 R 包中，主要实现功能有处理异常值和断点值识别和修复、时间索引的重新计算、对于交易时段和午休时段的调整、价格变化量的计算、买卖价差的计算、价格变化分类值的识别计算、模型变量的生成和整理、大型高频交易数据集的整合和管理等。对于具体的数据处理流程此处不予赘述，在之后的实证研究中，使用已处理的数据不需要考虑异常值和不连续交易时间的问题，同时用于拟合量化模型的变量也均批量生成完毕，可以直接支持模型使用。数据分析和建模等工作主要在 R Studio 平台进行，R 包中的 R 脚本和 Python 脚本部分核心代码具体实现见附录。

3.2 基本变量设定

首先设定 t_i 为从午夜 12 点开始计算的标的金融资产第 i 次交易发生的时间，或第 i 次取市场快照的时间，或设定为市场上第 i 次有显著的新信息出现的时间，以秒为单位进行计算。显著新信息出现时刻定义为使得价格发生显著变化的时刻。具体对 i 的设定在应用模型时根据数据特性和模型特征来决定。进行以上设定的原因在于国内金融市场的数据截取能力还比较有限，搜集到的数据的颗粒细度不够，无法完全描述市场上所有交易的信息（期货市场数据截取间隔为 0.5 秒，股票市场数据截取间隔为 3 秒）。在市场交易非常活跃的时期，这样的数据颗粒显然无法呈现每一笔交易的具体细节，有大量的交易信息无法直接被观测到，而这里面又有很大一部分交易是在同一价格成交。因此，本课题选取使得价格发生相对大变动的时刻（一般将阈值设定为 2 个最小价格变化单位），用于代表新信息出现的时刻进行分析。由于原本的逐笔交易模型也是将新交易视为新信息出现，进行改进后的数据与国外的逐笔交易数据呈现出相似的特性，和逐笔交易数据相比，变化在于减少了市场微观结构的极其细微噪声以及在所观测的时间节点存在不大于 0.5 秒的误差（对于股指期货市场而言，对于股票市场则为 3 秒，但股票市场单只股票的交易久期常常大于 3 秒，故一般没有上述问题），相当于进行了一次高频数据截取时间间隔内的平滑，这个调整是由于当前市场快照数据的技术限制导致的，由于时间跨度很小，可以假设对模型没有影响，即本课题使用手头的数据来对真实市场微观结构进行描述和分析。

进一步地，设定金融资产在时间 t_i 时刻的价格为 P_{t_i}，价格变动为 $pch_{t_i} := \Delta P_{t_i} = P_{t_i} - P_{t_{i-1}}, i \in \mathbb{N}$，相应的价格变动久期为 $dur_{t_i} := \Delta t_i = t_i - t_{i-1},, i \in \mathbb{N}$。一般来说应该设定 t_i 为第 i 次交易发生的时间，故 pch_{t_i} 和 dur_{t_i} 中的下标可以统一。但考虑到前述的

国内高频交易数据格式的不完善和颗粒度不够细的客观条件限制，在分析价格变动序列时 t_i 应设定为每次观测价格的时间，即在价格变化频繁时间距非常小的等距时间，在价格保持不动或者没有交易的时候则为非等距时间，此时的间隔可视为误差非常小的久期。在价格变化频繁时，由于在最大观测频率下仍然有可能有多笔交易在观测间隔中发生，在分析久期序列时便无法获取交易久期，则只能使用价格变动久期代替，此时价格变动久期序列 dur_{t_i} 中的下标 t_i 为市场中价格发生显著变化的时间点，此时序列 t_i 在绝大多数情况下是不等距的，这是高频交易数据处理和低频交易数据处理的主要不同之处，也是模型设定要考虑的最重要的因素之一。本课题使用的高频交易数据模型即主要讨论对两个核心的离散时间随机过程 $dur_{t_i}, i \in \mathbb{N}$ 和 $pch_{t_i}, i \in \mathbb{N}$ 的分析建模，需要注意两个过程的下标 i 含义不同，我们在进行数据处理时已进行区分。

3.3 价格变动模型

本节主要讨论描述标的金融资产盘中高频价格变动特性的金融计量模型。作为过程 $pch_{t_i}, i \in \mathbb{N}$ 的观测值，采集到的高频价格变动数据拥有前述的诸多性质。模型主要通过从细颗粒价格变动数据中挖掘市场微观结构的特性来描述金融市场某一产品交易的活跃程度和具体的价格变化特性。本课题主要采用的模型有由 Hauseman 在 1992 年研究交易数据的价格变化时提出的顺序概率值模型（Ordered Probit Model）和由 McCulloch 和 Tsay 在 2000 年提出的 ADS 分解模型（本章讨论的是 Rydberg 和 Shephard 在 2003 年提出的模型对变量设定进行简化后的版本），同时还对价格变化序列在不同的时间尺度之下计算的已实现波动率进行了分析和研究。OPM 顺序概率值模型通过对隐含价格变化条件概率的拟合估计，利用数据隐含的信息得到对下一个交易价格变化的估计概率值，可以帮助投资者认识市场价格变化的可能性和概率的诸如均值、方差、波动性和集束性等性质。由于其可以对每一次观测的价格变动的隐含价格变动概率进行估计，我们可以很详细看出高频交易数据在日内的价格变化动态性具体有何种性质以及经历了怎么样的变化。ADS 价格分解模型则是从另一个角度分析价格变化时间序列，将价格变化分解后分别拟合各变量的条件概率。由于模型针对的条件概率比顺序概率值模型更加具体，可以分别对某种特定的条件概率进行更准确的估计，例如通过对价格变化与否，变化方向与变化幅度的条件概率进行估计，可以从中获取关于价格变动集束性和回转性的信息。此外，对已实现波动率的分析能够帮助投资者在微观层面理解市场的波动情况和特点。虽然在计算已实现波动率时是针对一整天的价格变动序列进行的，没有对日内盘中不同时刻的波动情况进行分析，但可以呈现出在不同的时间尺度下市场的波动情况，从另一方面提供关于市场微观结构性质的有用信息。

3.3.1 顺序概率值模型

考虑到交易系统设计的限制带来的价格变化的离散性以及市场噪音对价格的影响，顺序概率值模型引入了标的金融资产的隐含价格变化并设定了观测到的价格变化和隐含价格变化之间的联系，从而通过估计隐含价格变化的条件概率来预测不同价格变化分类发生的概率。通过对隐含值和观测值的分离，顺序概率值模型能够跨越客观条件限制，在统计层面捕捉到价格变化深层次的信息。令标的金融资产在时刻 t_i 的隐含价格为 $P_{t_i}^*$ ，注意此处将无法观测到的隐含价格假设为连续随机变量，可摆脱观测值的离散性质限制。进一步有隐含的连续价格变化为 $pch_{t_i}^* := \Delta P_{t_i}^* = P_{t_i}^* - P_{t_{i-1}}^*, i \in \mathbb{N}$。设定 $pch_{t_i}^*$ 在 t_i 时刻的解释变量为 $\mathbf{x}_{t_{i-1}} = (x_{1,t_{i-1}}, x_{2,t_{i-1}}, \cdots, x_{n,t_{i-1}})^T$，即在 t_{i-1} 时刻能够获取的信息集合，也可表为域流 $\mathcal{F}_{t_{i-1}}$，

向量中 $x_{m,t_{i-1}}$ 为在 t_{i-1} 时刻第 m 个解释变量的值；和解释变量对应参数的 $n\times 1$ 向量则可设定为 $\beta=(\beta_1,\beta_2,\dots,\beta_n)^T$。 将隐含价格变化过程 $pch_{t_i}^*, i\in\mathbb{N}$ 的动态特性描述如下：

$$pch_{t_i}^* = \beta^T \mathbf{x}_{t_{i-1}} + \epsilon_{t_i} \tag{3-1}$$

其中 ϵ_{t_i} 为 t_i 时刻的误差项，对于其条件分布性质，我们假设：

$$\begin{aligned} \mathbb{E}[\epsilon_{t_i}|\mathcal{F}_{t_{i-1}}] &= \mathbb{E}[\epsilon_{t_i}|\mathbf{x}_{t_{i-1}}] = 0 \\ Var(\epsilon_{t_i}|\mathcal{F}_{t_{i-1}}) &= Var(\epsilon_{t_i}|\mathbf{x}_{t_{i-1}}) = \sigma_{t_i}^2 \\ Cov(\epsilon_{t_i},\epsilon_{t_j}) &= 0, \text{for } i\neq j \end{aligned} \tag{3-2}$$

即认为误差具有零条件均值、时变条件方差以及零自相关性。进一步地，为了研究时变条件方差的动态特性，引入用于解释条件方差函数的k维变量 $\mathbf{w_{t_{i-1}}}=(w_{1,t_{i-1}},w_{2,t_{i-1}},\cdots,w_{k,t_{i-1}})^T$ 并设定非负函数 $g(\cdot)$，有 $\sigma_{t_i}=g(\mathbf{w}_{t_{i-1}})$。注意解释变量 $\mathbf{w}_{t_{i-1}}$ 和 $\mathbf{x}_{t_{i-1}}$ 一样可包含截至 t_{i-1} 时获取的其他变量的观测值。为了建模方便，不妨将误差项 ϵ_{t_i} 假设为服从正态分布，于是有误差的条件分布：

$$\epsilon_{t_i}|\mathbf{w}_{t_{i-1}},\mathbf{x}_{t_{i-1}} \sim \mathcal{N}(0,g^2(\mathbf{w}_{t_{i-1}})) \tag{3-3}$$

在对隐含价格建模完毕后，进一步假定观测到的价格变化有 k 个对应的可能取值。设 k 为有限整数，可将价格变化的离散值与价格变化的区间统一划归为 k 个值，有 $\{s_i,i=1,\cdots,k\}$，例如变化在第 l 个区间即取值为 l。为了将隐含价格变化和实际价格变化相联系， pch_{t_i} 和 $pch_{t_i}^*$ 的关系可描述为：

$$pch_{t_i}=s_j \text{ if } \alpha_{j-1}<pch_{t_i}^*\le\alpha_j, j=1,\cdots,k \tag{3-4}$$

其中 $\alpha_j\in\mathbb{R}, j=1,\cdots,k$ ，并有 $-\infty=\alpha_0<\alpha_1<\cdots<\alpha_{k-1}<\alpha_k=\infty$ ，在正态的条件分布下有:

$$\begin{aligned} \mathbb{P}\{pch_{t_i}=s_j|\mathbf{x}_{t_{i-1}},\mathbf{w}_{t_{i-1}}\} &= \mathbb{P}\{\alpha_{j-1}<\beta^T\mathbf{x}_{t_{i-1}}+\epsilon_{t_i}\le\alpha_j|\mathbf{x}_{t_{i-1}},\mathbf{w}_{t_{i-1}}\} \\ &= \begin{cases} \Phi(\frac{\alpha_j-\beta^T\mathbf{x}_{t_{i-1}}}{\sigma_{t_i}}) & j=1 \\ \Phi(\frac{\alpha_j-\beta^T\mathbf{x}_{t_{i-1}}}{\sigma_{t_i}})-\Phi(\frac{\alpha_{j-1}-\beta^T\mathbf{x}_{t_{i-1}}}{\sigma_{t_i}}) & j=2,\cdots,k-1 \\ 1-\Phi(\frac{\alpha_j-1-\beta^T\mathbf{x}_{t_{i-1}}}{\sigma_{t_i}}) & j=k \end{cases} \end{aligned} \tag{3-5}$$

其中 $\Phi(\cdot)$ 为标准正态分布的累积分布函数。顺序概率值模型所包含的需要拟合的参数有 $\beta,\alpha_j,j=1,\cdots,k$ 以及条件方差函数 $\sigma_{t_i}=g(\mathbf{w}_{t_{i-1}})$ 中的参数。在使用极大似然估计法估计时，我们先设定指示变量 $1_{t_i,j}$ ，使得当 $pch_{t_i}=s_j$ 时 $1_{t_i,j}=1$ ，否则为 0。假设一共有 n 次观测，给定解释变量观测值 $\mathbf{x}=(x_{t_{i-1}}^m,i=1,\cdots,n,m=1,\cdots,k)$， 实际价格变动 pch_{t_i} 的观测值 $pch=(pch_{t_1},pch_{t_2},\cdots,pch_{t_n})^T$ 所对应的似然函数为：

$$\begin{aligned} \mathcal{L}(pch|\mathbf{x}) &= \sum_{j=1}^n\sum_{k=1}^m\{1_{t_i,j}\cdot log[\Phi(z_j)-\Phi(z_j-1)\} \\ z_j &= \begin{cases} 0, & j=0 \\ \frac{\alpha_i-\beta^T\mathbf{x}_{t_{t-1}}}{\sigma_{t_i}}, & j=1,\cdots,m-1 \\ 1, & j=m \end{cases} \end{aligned} \tag{3-6}$$

拟合模型后，我们可以根据模型参数对每个观测到的值在当时价格上涨、下跌和保持不变的条件概率进行估计，从概率分布和整体水平可以对市场的价格变动在微观层面的特性进行分析，包括价格变动概率的平均水平和概率的变化幅度等等，可以体现出市场交易的活跃性、价格发现的有效性、反应新信息的及时性以及流动性的变化。

3.3.2 ADS 价格变动分解模型

ADS 模型首先由 McCulloch 和 Tsay 在 2000 年提出，又称为价格变动分解模型，我们使用的模型设定从 Rydberg 和 Shephard 在 2003 年研究的价格变动分解模型简化而来。主要通过将价格变动分解为 3 个分离却不独立的部分来对价格变化的概率和动态特性进行分析。模型主要估计的是用于描述每一个部分在给定分解后的其他变量后的条件概率的函数中的参数。我们可以通过分析资产价格变动的某几种特定的条件概率分布，例如给定上一次观测有变化当前价格发生变化的概率、在上一次价格上涨时当前价格继续上涨的概率以及前一次或几次价格变化的幅度得到当前价格变化的幅度概率分布等，对价格变化的动态特性进行高频细颗粒层面的分析。

具体而言，在第 i 次交易或对交易数据取快照时中的价格变动 pch_{t_i} 可分解为变量 $A_{t_i}, D_{t_i}, S_{t_i}, i \in \mathbb{N}$ 的乘积形式：

$$pch_{t_i} = A_{t_i} D_{t_i} S_{t_i} \tag{3-7}$$

注意三个分解的变量并非相互独立。如之前提到的，在实际应用过程中，由于交易的活跃度过高，数据抓取的密度不及系统成交的密度，完整的逐笔交易数据可能难以获取，故此处将下标 i 标记的时刻设定为第 i 次对交易数据取快照的时刻，从而忽略掉在同一价格成交的数笔交易。在交易高度活跃时期由于交易密度较大，逐笔交易之间的间隔小于数据截取间隔，故 t_i 近似为等距时间，在逐笔交易之间的间隔大于数据截取间隔时，t_i 可视为交易发生的时间，误差可控制在数据截取间隔之内。A_{t_i} 为代表价格变动与否的二元变量，取 1 时为有价格变动； D_{t_i} 为代表价格变动方向的二元变量，取 1 为价格上升，取 -1 为价格下降； S_{t_i} 为价格变动幅度，其仅在价格变动存在时有取值，即 $A_{t_i} = 1$ 时，且将上升下降的两种情况分开进行拟合，故在价格变动方向变量 D_{t_i} 取不同值时，S_{t_i} 的条件概率分布不同。 设定 $\mathcal{F}_i$ 为截止至时间 t_i 的域流，即在 t_i 时刻获取的价格变化信息。

$$\begin{aligned} &\mathbb{P}\{pch_{t_i} | \mathcal{F}_{i-1}\} = \mathbb{P}\{A_{t_i} D_{t_i} S_{t_i} | \mathcal{F}_{i-1}\} \\ &= \mathbb{P}\{S_{t_i} | A_{t_i}, D_{t_i}, \mathcal{F}_{i-1}\} \mathbb{P}\{D_{t_i} | A_{t_i}, \mathcal{F}_{i-1}\} \mathbb{P}\{A_{t_i} | \mathcal{F}_{i-1}\} \end{aligned} \tag{3-8}$$

进一步地，我们对每一个分解出的部分设定相应的条件分布并指定参数：

$$\begin{aligned} A_{t_i} &\sim Bin(p_i),\ p_i &&:= \mathbb{P}\{A_{t_i} = 1\}, \\ \frac{1}{2}(D_{t_i} + 1) | A_{t_i} = 1 &\sim Bin(\delta_i),\ \delta_i &&:= \mathbb{P}\{D_{t_i} = 1 | A_{t_i} = 1\}, \\ S_{t_i} | D_{t_i} = 1, A_{t_i} = 1 &&&\sim g(\lambda_{u,i}) + 1, \\ S_{t_i} | D_{t_i} = -1, A_{t_i} = 1 &&&\sim g(\lambda_{d,i}) + 1 \end{aligned} \tag{3-9}$$

在价格变动时，D_{t_i} 取值为 1 和 -1 而不是通常的两点分布。

模型设定条件概率可表示为：

$$\begin{aligned} ln(\frac{p_i}{1-p_i}) &= \beta^T \mathrm{x}_{t_i}, \\ ln(\frac{\delta_i}{1-\delta_i}) &= \gamma^T \mathrm{z}_{t_i}, \\ ln(\frac{\lambda_{u,i}}{1-\lambda_{u,i}}) &= \theta_u^T \mathrm{w}_{t_i}, \\ ln(\frac{\lambda_{d,i}}{1-\lambda_{d,i}}) &= \theta_d^T \mathrm{w}_{t_i} \end{aligned} \tag{3-10}$$

其中 $\mathrm{x}_{t_i}, \mathrm{z}_{t_i}, \mathrm{w}_{t_i}$ 均为回归的解释变量向量，包含了 $\mathcal{F}_{i-1}$ 中所有的信息，$\beta, \gamma, \theta_u, \theta_d$ 均为需要估计的参数向量。由于 S_{t_i} 不是二元变量，并不像 A_{t_i}, D_{t_i} 一样服从两点条件分布，而是服从参数为 λ 的几何分布，即 $\mathbb{P}\{x = m\} = \lambda(1-\lambda)^m, m = 0,1,2,\cdots$。 $g(\lambda)$ 表示的即为几何分布，加 1 是为了让其取值大于等于 1。同时，为了保证分布参数 $p_i, \delta_i, \lambda_{u,i}, \lambda_{d,i} \in [0,1]$，我们对其进行了 logistic 变换后方才引入回归方程，在模型拟合完毕后使用参数可以计算出拟合的条件概率。

综上，观测到的价格变化 pch_{t_i} 可分为以下 3 类：

(1) 价格不变，即 $A_{t_i} = 0$ ，概率为 $(1-p_i)$ 。
(2) 价格上升，即在 $A_{t_i} = 1$ 的条件下，$D_{t_i} = 1$，概率为 $p_i\delta_i$ 。价格振幅由调整的几何分布决定，分布参数为 $\lambda_{u,i}$。
(3) 价格下降，即在 $A_{t_i} = 1$ 的条件下，$D_{t_i} = -1$，概率为 $p_i(1-\delta_i)$ 。振幅由调整的几何分布决定，分布参数为 $\lambda_{d,i}$。

使用似然估计法估计模型参数时，可先设定指示变量 1_j 代表如上第 j 种情况，其中 $j = 1,2,3$ 。对于价格变动序列的观测值 $pch = (pch_{t_1}, pch_{t_2}, \cdots, pch_{t_n})^T$ 似然函数有

$$\begin{aligned} \mathcal{L}(pch|\mathcal{F}_0) &= \sum_{i=1}^{n} log\ \mathbb{P}\{pch_{t_i}|\mathcal{F}_{i-1}\} \\ log\ \mathbb{P}\{pch_{t_i}|\mathcal{F}_{i-1}\} &= 1_1 log(1-p_i) + 1_2 log(p_i\delta_i\lambda_{u,i}(1-\lambda_{u,i})^{S_i-1}) \\ &\quad + 1_3 log(p_i(1-\delta_i)\lambda_{d,i}(1-\lambda_{d,i})^{S_i-1}) \end{aligned} \tag{3-11}$$

其中 $1_i, i = 1,2,3$ 分别为对应于上述三种情况的指示函数。待估参数为$\beta, \gamma, \theta_u, \theta_d$。

在本课题的实证研究中，我们使用具有如下设定的模型：

$$\begin{aligned} ln(\frac{p_i}{1-p_i}) &= \beta_0 + \beta_1 A_{t_{i-1}} \\ ln(\frac{\delta_i}{1-\delta_i}) &= \gamma_0 + \gamma_1 D_{t_{i-1}} \\ ln(\frac{\lambda_{u,i}}{1-\lambda_{u,i}}) &= \theta_{u,0} + \theta_{u,1} S_{t_{i-1}} \\ ln(\frac{\lambda_{d,i}}{1-\lambda_{d,i}}) &= \theta_{d,0} + \theta_{d,1} S_{t_{i-1}} \end{aligned} \tag{3-12}$$

我们在应用 ADS 价格变动分解模型时仅仅考虑一阶滞后项，实证研究表明在分析股指期货的价格变动时，在数据量不大的情况下，例如几天的几万条数据，使用一阶滞后的模型效果已经较好，参数也均比较显著。通过对条件分布参数即条件概率的拟合结果，可以获取关于价格变化密度、价格变化集束性强弱、价格逆转效应以及价格变动幅度与集束性关系的相关信息。

3.3.3 已实现波动率分析

本节主要讨论描述高频数据已实现波动率的研究方法。通过一系列直观的数据处理方法，可以从不同的时间跨度出发，对标的金融资产高频交易数据进行处理得到已实现波动率，通过分析其波动率序列的变化情况来判断市场波动率特性产生的变化。在计算资产价格变动月化波动率时，可以通过该资产的日收益率序列入手。设资产的日对数收益率序列为 $\{r_{i,k}, i = 1,\cdots,n\}$，即 $r_{i,k} = log(P_{i,k}/P_{i-1,k})$ 且 $P_{i,k}$ 为第 i 个交易日的价格，讨论的月份有 n 个交易日，下标中的 k 指计算的时刻在第 k 个月。此时，对于该资产的月化对数收益率 r_k，有 $r_k = \sum_i r_{i,k}$。此时，假设月对数收益率序列 $\{r_k, k = 1,2,\cdots\}$ 的关于域流 $\{\mathcal{F}_k, k = 1,2,\cdots\}$ 的条件方差和条件协方差存在，有：

$$Var(r_k|\mathcal{F}_{k-1}) = \sum_{i=1}^{n} V\,ar(r_{i,k}|\mathcal{F}_{k-1}) + 2\sum_{i<j} C\,ov(r_{i,k}, r_{j,k}|\mathcal{F}_{k-1}) \tag{3-13}$$

为方便建模，不妨假设 $\{r_{i,k}, i = 1,\cdots,n\}$ 为 $i.i.d.$ 序列，设 $\mathbb{E}r_{i,k} = \mu_k < \infty, Var(r_{i,k}) = \sigma_k^2 < \infty$ 且，即其为一个白噪声序列，有：

$$Var(r_k|\mathcal{F}_{k-1}) = \sum_i V\,ar(r_{i,k}) = n \cdot Var(r_{i,k}) = n \cdot \sigma_k^2 \tag{3-14}$$

其中 σ_k^2 可用如下估计量估计：

$$\hat{\sigma_k^2} = \frac{\sum_{i=1}^{n}(r_{i,k} - \overline{r_k})^2}{n-1},\ \overline{r_k} = \frac{1}{n}\sum_{i=1}^{n} r_{i,k} \tag{3-15}$$

于是第 k 个月的月化波动率 σ^2 的估计值即为：

$$\hat{\sigma^2} = \frac{n}{n-1}\sum_{i=1}^{n}(r_{i,k} - \overline{r_k})^2 \tag{3-16}$$

将如上讨论的计算方法推广到日化（日内）波动率的计算中，需要先设定计算对数收益率的时间间隔，实际上这个间隔可以随意选取以反映在不同时间跨度下资产价格波动性的动态特性。本课题在计算时使用的间隔有：逐笔、1 秒、5 秒、10 秒、20 秒、1 分钟、2 分钟、5 分钟、10 分钟、20 分钟和 30 分钟。一般而言对于本课题使用的高频交易数据，基于 5、10、20、30 分钟计算的时间间隔的已实现波动率值已保持稳定、差别较小，变化均在百分之十以下。继续使用上述的标记方式，将月化波动率改为日化，日区间改为自定义的时间间隔，进一步假设 $\overline{r_k} = 0$，即有近似地 $\hat{\sigma^2} = \frac{n}{n-1}\sum_{i=1}^{n}(r_{i,k})^2 \approx \sum_{i=1}^{n}(r_{i,k})^2$，在时间间隔数量逐渐增大时有该假设近似成立。注意此处 k 指当前在第 k 天计算，i 为计算价格对数收益率的时间间隔下标，n 为计算的时间间隔总数。紧接着前述的关系，可设第 k 天的已实现波动率 $RV_k = \sum_{i=1}^{n}(r_{i,k})^2$，即其为每个时间间隔中对数收益率的平方和。

我们进一步寻求如上设定和计算方式在常用的金融数学建模设定上的一致性。通常我们使用伊藤过程来描述金融资产价格序列，假设标的金融资产的价格过程 $S_t, t \geq 0$， 对数价格过程 $s_t := ln(S_t), t \geq 0$，进而有如下随机微分方程描述随机过程的变化特性：

$$ds_t = \mu_t dt + \sigma_t dW_t \tag{3-17}$$

其中 $\mu_t, \sigma_t, t \geq 0$ 分别为关于域流 $\mathcal{F}_t, t \geq 0$ 可测的适应性均值过程和标准差过程，有时也将 $\mu_t dt$ 和 $\sigma_t dW_t$ 分别称为 s_t 的漂移项和扩散项，而 $W_t, t \geq 0$ 为标准布朗运动，其增量 $dW_t = \epsilon_{dt} \sim \mathcal{N}(0, dt)$ ，同时有布朗运动的二次变差 $[W, W](dt) = dt$。具体来说，对于通常用于描述价格序列的几何布朗运动而言，我们有如下的随机微分方程：

$$\begin{aligned} dS_t &= \mu S_t dt + \sigma S_t dW_t, \\ dS_t/St &= \mu dt + \sigma dW_t, \\ ds_t = dln(S_t) &= (\mu - \frac{1}{2}\sigma^2)dt + \sigma dW_t \end{aligned} \tag{3-18}$$

可以看出几何布朗运动也是一种特殊的伊藤过程，其均值函数和波动函数均不随时间变化。采用伊藤过程的设定研究已实现波动率时，主要关注资产对数价格序列的波动率函数 σ_t 。进一步将某一日内的交易时间区间记为 $[0, T]$ ，则 $dt = T/n$ 。同样对时间间隔做划分 $0 = t_1 < t_2 < \cdots < t_n < t_{n+1} = T$，对每一个区间计算对数收益率 $r_i = s_{t_{i+1}} - s_{t_i}$ 后求平方和 $\sum_{i=1}^{n} r_i^2$ ，其在间隔数 $n \to \infty$ 时依概率收敛于对数价格过程 $\{s_t, t \geq 0\}$ 在时间区间 $[0, T]$ 上积累的二次变差，此结果适用于所有伊藤过程。由于 $dtdt = 0, dtdW_t = 0$ ，我们有：

$$\begin{aligned} \lim_{n\to\infty} \sum_{i=1}^{n} r_i^2 \xrightarrow{p} [s, s](T) &= \int_0^T (ds_t)^2 \\ &= \int_0^T \mu_t^2 \, dtdt + \int_0^T 2\,\mu_t \sigma_t dtdW_t + \int_0^T \sigma_t^2 \, dW_t dW_t \\ &= \int_0^T \sigma_t^2 \, dW_t dW_t \\ &= \int_0^T \sigma_t^2 \, [W, W](dt) \\ &= \int_0^T \sigma_t^2 \, dt \end{aligned} \tag{3-19}$$

亦即在真实概率测度 $\mathbb{P}$ 下，对任意 $\epsilon > 0$ ，均有：

$$\lim_{n\to\infty} \mathbb{P}\left\{ \left| \sum_{i=1}^{n} r_i^2 - \int_0^T \sigma_t^2 \, dt \right| < \epsilon \right\} = 1 \tag{3-20}$$

所有满足 $\lim_{n\to\infty} \sum_{i=1}^{n} r_i^2 = \int_0^T \sigma_t^2 \, dt$ 的路径下具有真实概率测度 1。对收益率使用独立同分布随机变量序列的假设即意味着波动率函数为常数，则对数价格过程在时间区间 $[0, T]$ 上累积的二次变差的速率为 σ^2。在讨论日已实现波动率时，有 $T = 1$ ，则对数收益率平方和为日已实现波动率的估计值。

如上所述，由于二次变差依概率收敛于时变方差（假设为连续函数，实证研究已证实波动率过程条件方差一般不会带跳）对时间的黎曼积分，在对数收益率序列的白噪声假设下，通过提高时间间隔的数量可以用二次变差估计市场对数价格变动的已实现波动率，时间区间间隔越短，误差越小。直观上看在研究的时候应该仅仅研究时间颗粒度最细的数据，以便将收敛性彻底发挥使结果达到更好的精度。然而现实有许多地方不满足模型的理想情况。一方面，由于在时间跨度小到一定程度后，价格变动序列会产生一定的自相关性，导致非零的自协方差使得计算的已实现波动率上升。另一方面由于买卖价差弹性（隐含价格保持不变，由

于交易达成引发的价格变动）和非同步交易（由于资产交易强度不同导致的信息延迟）这两个价格形成过程的特征的存在，资产价格变化很大程度受其带来的市场微观结构噪声影响。此外，市场微观结构噪声还会由市场交易系统的设计、报价最小变动以及投资者之间信息不对称、行为模式特点和恐慌情绪等因素造成。总之，在计算对数收益率的时间间隔长度小到一定程度后，其计算出的已实现波动率会相比起较大时间间隔有明显上升。Zhang 在 2001 年曾经对市场微观结构噪声进行建模分析与深入讨论，认为高频交易数据对数收益率的方差在考虑市场微观结构噪声时，在时间间隔数量趋向于无穷时趋近于结构噪声的方差而不是预想的波动率函数对时间的积分。因此，在使用高频交易数据计算分析已实现波动率的时候，需要在不被市场微观结构噪声干扰的情况下得到尽可能准确的估计。

值得一提的是，本课题在对标的金融资产的分析过程中重心并不是对已实现波动率进行准确建模和预测，而是通过对已实现波动率数值和走势分布的分析来从微观层面研究价格变化特性。故如何选取最优的时间间隔来计算实现并不是最重要的。实际上，本课题在对股指期货高频交易数据的研究中也发现了市场微观结构噪声存在的显著证据，并认为对股指期货的限制进而造成的市场投资者成分和行为模式的变化也对市场微观结构噪声在无论是短期还是长期都产生了很明显的影响。于是，从不同时间跨度分析已实现波动率，无论其是否含有微观结构噪声的污染，都是本课题研究的重心之一。

尽管我们可以接受市场微观结构噪声的存在，但无论如何都应该把其存在带来的误差降到最低。其中一种可以降低结构噪声的方法是，对于同一个计算已实现波动率的时间间隔中，从间隔包含的时间序列中使用不同函数来选取用于计算的价格而不是仅仅取端点值，例如取中位数、最大最小值等，然后对选取出的多个价格取平均，或对计算出来的多个已实现波动率取平均值。本课题使用的是最大最小值取平均的方法。

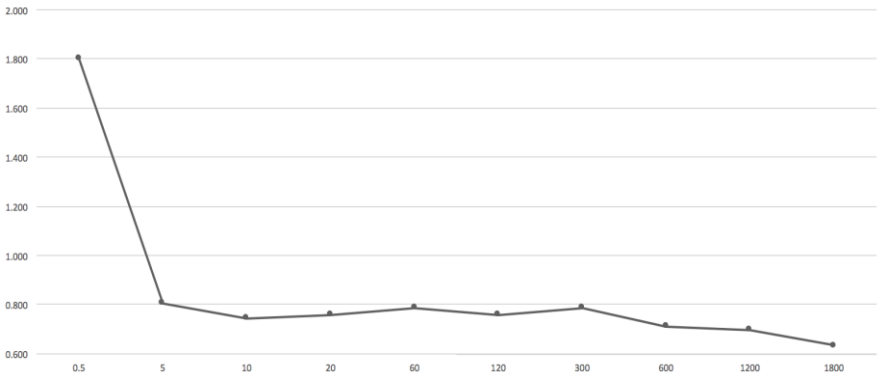

图 3-1 IF1509 在 2015 年 9 月 7 日基于不同时间间隔计算的隐含波动率（单位：秒）

3.4 久期模型

本节讨论针对价格变动久期的高频数据模型。由于高频数据交易以及影响标的金融资产价格变化的显著新息的非等间隔性，对久期的建模能够从另一个角度反映出市场的活跃程度以及市场参与者的普遍行为特性。价格变动久期和交易久期的差别之前已有提及，考虑到数据抓取间隔限制的客观因素，本课题计算的久期为价格变动久期，而不是交易久期，因为在市场活跃时，逐笔交易的信息由于技术限制无法获取，目前市场上最精细的数据也无法满足要求。尽管如此，在不计无价格变动的交易和造成价格在最小波动单位上下波动的交易后，

本课题计算的市场价格较大变化之间（可理解为对标的金融资产价格的新信息之间）的久期呈现出和逐笔交易数据同样的性质，即显著的日模式、自相关性和集束性等等。直观上看，较长的价格变动久期代表了一段没有新的显著消息的时期，抑或是新的消息没有反应在股价上的时期，故对久期的动态特性进行研究有助于认识市场价格变动的特性、市场价格的有效性以及投资者的行为等，进而从微观角度来细致理解市场价格的波动。

研究高频数据久期的金融计量模型中，最常用的即为自回归条件久期模型（Autoregressive Conditional Duration Model, ACD），首先由 Engle 和 Russell 于 1998 年提出。ACD 模型利用类似于条件异方差模型（Autoregressive Conditional Heteroskedasticity Model, ARCH）的思想对久期的条件均值进行建模分析。在他们的研究基础上，有许多人对 ACD 模型的假设、变量设定以及针对实际应用环境下数据的特点进行了一系列的改进和推广，提出了许多更加一般化的模型。本课题在使用 ACD 模型时，尝试了几种不同的变量分布假设，并根据实际情况进行了选择。

3.4.1 日模式的调整

通过观察，我们发现价格变动久期呈现出和波动率、交易量和价格波动等相似的显著的日模式。具体表现为在开盘和收盘的时间段市场交易更加频繁，一方面是为了释放市场关闭时累积的新信息，另一方面是出于对市场关闭时段内的风险的规避。由于久期模型研究的是条件均值的解释和预测，此类显著的周期模式会给数据带来较明显的高阶周期性自相关，干扰模型的解释能力。此外，此类日模式很容易捕捉，于是应该在使用久期模型之前予以剔除来最大化模型的解释效力。

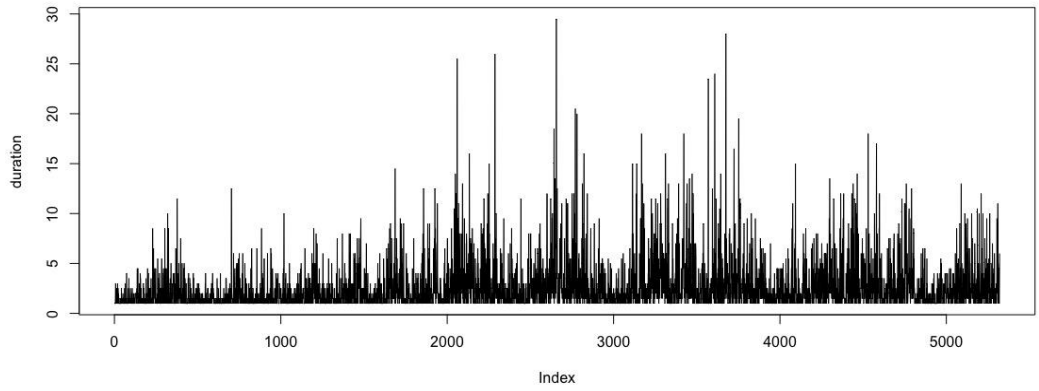

图 3-2 IF1509 在 2015 年 9 月 7 日的价格变动久期原始序列

从图 3-2 中可以观察到在开盘后收盘前的价格变动较频繁，而在盘中价格变动时间较长，即为久期日模式。在生成上述久期时序图的过程中，首先选取了价格变化超过一个最小变动单位的时刻做标记，再计算标记之间的持续期作为价格变化久期，同时将连续两次的标记视为一次，将 0.5 秒（针对期指久期的计算需要，针对股票现货数据计算时不需要，此类情况较少）视作零值予以剔除，仅将大于 0.5 秒的久期计算在内。讨论价格变化久期序列 $dur_{t_i}, i \in \mathbb{N}$，其中 i 为标记的价格发生变化的序号，t_i 为对应的时间戳，在该时刻的久期定义为 $dur_{t_i} := \Delta t_i = t_i - t_{i-1}$，即直到下一次价格发生变化所经历的时间（注意其和价格

变化 $pch_{t_i} = P_t - P_{t-1}$ 的定义方式稍有不同）。假设价格变化久期中含有的日模式函数为 $d(t_i)$ ，简记为 d_{t_i} ，假设其包含了关于久期日模式的所有信息，则剔除日模式后的久期为 $dur_{t_i}^* := dur_{t_i}/d_{t_i}$ 。剔除日模式的方法有许多选择， 本课题采用多项式函数拟合方法来进行数据处理。具体而言，假设 d_{t_i} 有以下成分：

$$\begin{aligned} d_{t_i} &= exp(d'_{t_i}), \\ d'_{t_i} &= \beta^T \mathrm{f}_{t_i}, \\ \beta &= (\beta_0, \beta_2, \cdots, \beta_n)^T \\ \mathrm{f}_{t_i} &= (f_0(t_i), f_1(t_i), \cdots, f_n(t_i))^T \\ f_k(t_i) &= \begin{cases} 1, & k = 0 \\ t_i - c, & k = 1 \\ f_1^k(t_i), & k = 2, \cdots, n \end{cases} \end{aligned} \tag{3-21}$$

观察发现久期的日模式主要呈现倒 "U" 型，为了简化计算，本课题采用 $c = 0, n = 2$ 的简化设定，需要回归的方程为：

$$log\ （dur_{t_i}) = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 \tag{3-22}$$

回归结果为见表 3-1，可见参数均十分显著。从图 3-3 中可发现，调整后的久期时间序列更加均匀平稳，识别出的日模式被很大程度地消减了。考虑到几乎所有的日高频交易数据均呈现出或多或少的日模式，在实证研究中，所有代入模型进行拟合的高频久期时间序列均为经过日模式识别和调整后的序列。因此，对日模式的调整和调整前后的差别在此处讨论后将不加赘述，在解释模型的变动时也不应将日模式的有关因素考虑在内。之后本课题将采用日模式已被完全消除的假设。

表 3-1 IF1509 在 2015 年 9 月 7 日的价格变动久期原始序列日模式多项式拟合参数

	估计值	标准误	t 值	p 值
β_0	2.89E-01	2.42E-02	11.94	<2e-16
β_1	1.15E-04	7.81E-06	14.78	<2e-16
β_2	-5.86E-09	5.04E-10	-11.64	<2e-16

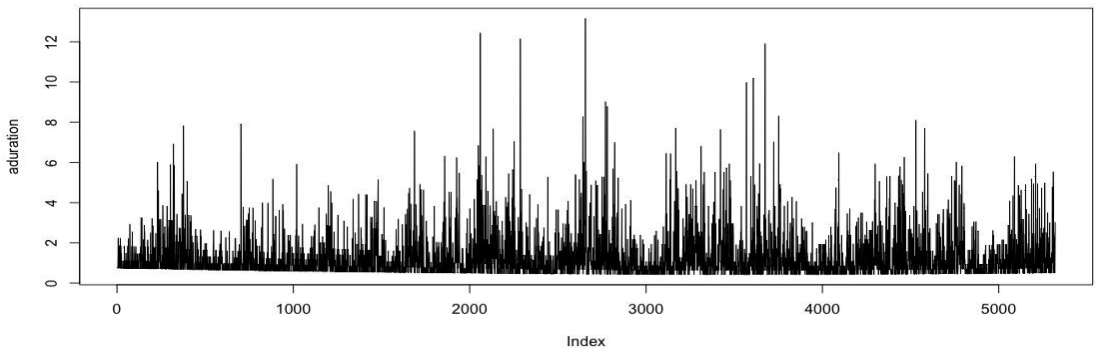

图 3-3 IF1509 在 2015 年 9 月 7 日的价格变动久期调整日模式后序列

3.4.2 ACD 自回归条件久期模型

ACD 模型使用和广义条件异方差模型（GARCH）相似的思想来描述和分析价格变动久期和交易久期的条件均值动态特性。承接上文对价格变动数据限制和应对方案的讨论，讨论的久期设定为价格变动久期。令 $\mu_{t_i} = \mathbb{E}[dur_{t_i}|\mathcal{F}_{i-1}]$ 为久期过程 dur_{t_i} 在域流 $\mathcal{F}_{i-1}$ 下的条件期望，进一步有 $dur_{t_i} = \mu_{t_i}\epsilon_{t_i}$，其中 ϵ_{t_i} 设定为一独立同分布的非负随机变量序列，满足 $\mathbb{E}\epsilon_{t_i} = 1$ 且服从标准韦布尔分布（Standardized Weibull Distribution）或标准指数分布（Standardized Exponential Distribution），此设定由 Engle 和 Russell 在 1998 年进行了讨论。模型假定 ϵ_{t_i} 服从以下表式：

$$\begin{aligned} \mu_{t_i} &= w^T \boldsymbol{\mu}_{\boldsymbol{t_i,s}} + \gamma^T \mathbf{dur}_{t_i,r}, \\ w &= (w_0, w_1, \cdots, w_s)^T, \\ \gamma &= (\gamma_1, \gamma_2, \cdots, \gamma_r)^T, \\ \boldsymbol{\mu}_{\boldsymbol{t_i,s}} &= (1, \mu_{t_i}, \mu_{t_{i-1}}, \cdots, \mu_{t_{i-s}})^T, \\ \mathbf{dur}_{\boldsymbol{t_i,r}} &= (dur_{t_i}, dur_{t_{i-1}}, \cdots, dur_{t_{i-r}})^T, \\ \Rightarrow \epsilon_{t_i} &= w_0 + \sum_{j=1}^{s} w_j\, \mu_{t_{i-j}} + \sum_{j=1}^{r} \gamma_j\, dur_{t_{i-j}} \end{aligned} \tag{3-23}$$

上述表式即为 $ACD(r,s)$ 模型。可以发现其与 $GARCH(r,s)$ 模型有相似之处，但前者是用于解释时序的条件均值的性质，后者则用于描述条件异方差的动态特性。进一步地，ACD 模型的推广可与 $GARCH$ 模型以及 $ARMA$ 模型的不同形式联系起来。在实证分析中，我们会将带有不同的分布假设的模型一并用于分析，然后选取最优。结果证明使用标准化的韦布尔分布和广义伽马分布的模型效果差不多，都远远比标准化的指数分布好。

当设定 ϵ_{t_i} 服从标准指数分布时，上述模型即记为 EACD（Exponential ACD）模型，如果假设标准韦布尔分布，则相应为 WACD（Weibull ACD）模型，指数分布即为一个简化了的韦布尔分布。容易证明，根据上述模型设定求得的残差序列 $e_{t_i} := dur_{t_i} - x_{t_i}$ 为一个鞅差分序列，即 $e_{t_i} = dur_{t_i} - \mathbb{E}[dur_{t_i}|\mathcal{F}_{i-1}]$，同时等式右边两项均为鞅。进一步将 ACD 模型改写为：

$$\begin{aligned} dur_{t_i} - e_{t_i} &= w_0 + \sum_{j=1}^{max(r,s)} (w_j + \gamma_j) \cdot dur_{t_{i-j}} - \sum_{j=1}^{s} w_j\, e_{t_{i-j}} \\ &\begin{cases} \gamma_j &= 0,\ j > r \\ w_j &= 0,\ j > s \end{cases} \end{aligned} \tag{3-24}$$

此为一标准 ARMA 过程表示，进一步地在弱平稳的假定和 $w_0 > 0, \sum_{j=1}^{max(r,s)}(w_j + \gamma_j) < 1$ 的假设下，有：

$$\mathbb{E}dur_{t_i} = \frac{w_0}{\left(1 - \sum_{j=1}^{max(r,s)}(w_j + \gamma_j)\right)} \tag{3-25}$$

在得到能够解释久期强烈持久的自相关性的有效模型之后，可以通过计算久期期望和样本均值刻画其整体平均水平，同时可对模型各部分参数进行分析来解释久期条件期望和滞后久期的关系，从而理解价格变化集束性和价格回转强度等市场微观结构特征。对于 EACD、WACD、GACD 模型的方差分析和条件对数似然函数估计法，具体实现涉及对假设分布的概率密度函数的讨论，此处暂不进行讨论。

3.5 本章小结

本章介绍了专门应用于高频交易价格数据的金融计量模型，在基本的数据处理和变量设定的基础上讨论了高频交易数据的特点和需要解决的问题，以及针对价格变动量序列和价格变动久期序列的计量模型，对其模型设定、模型假设、参数估计方法以及结果说明和意义等进行了仔细研究和整理，为之后实证分析打下了理论基础，明确了高频交易数据模型的作用和使用方法。

价格变动模型可以提供关于价格变化量之间动态相依关系的细致描述，例如隐含价格变动的条件概率动态特征、价格变动的频率和幅度动态特征等。已实现波动率分析可以帮助我们判断出不同尺度下市场波动率的变动，从而认识到市场微观结构噪声和自相关性的变化。对价格变动久期序列进行日模式调整和条件均值自回归模型可以提供关于市场交易分布性质以及信息流密度、价格运行特征、价格变化的集束性等方面的有用信息。通过对高频交易数据中最重要的两个时间序列的建模，我们得以对市场微观结构特性进行充分的描述和分析。本章中使用的诸多模型在对资产价格的高频序列的描述上具有整体的一致性，没有假设和设定上的冲突，主要重心放在挖掘价格变动序列和价格变动久期序列的动态特性上。在讨论实际收益率的计算方法时，虽然使用了对数收益率序列独立同分布的较强假定，但在将得到的计算方法应用到实际数据中后，发现了基于较短的时间间隔计算的已实现波动率数值较大，说明了在微观层面存在着价格变动的自相关性，即过去几次的价格变动会影响当前的价格变动，从而将一部分的自协方差注入了已实现波动率，与市场微结构噪声带来的附加方差一起，大大增加了短时间间隔内的已实现波动率。

在已实现波动率分析中，我们假设对数收益率序列具有独立同分布性质，得到了高频价格序列已实现波动率的计算方法，在波动率函数随时间变化时，对数价格过程的二次变差为波动率函数对时间的积分，在波动率函数不随时间变化时，可以直接求得波动率的值。在给出计算方法时我们假设对数收益率序列独立同分布，但在实证研究中，我们认为价格变动序列、收益率序列之间实际上存在自相关的关系，而这部分自相关关系在使用前述的方法计算已实现波动率时，会将自相关关系引发的协方差引入计算的值中，从而在较短时间间隔内获得更大的已实现波动率值，所以在实证研究中观测到的较短时间间隔内的非常大的已实现波动率，可以认为是由自相关性和市场微观结构噪声共同造成的。

在使用工具上，对于上述的建模过程、数据整理、变量计算以及参数拟合等，本课题主要使用 R 和 Python 完成。在专门设计以支持本课题研究的 R 包 FMM 中，整合了常用金融计量包和重写的数据处理与建模函数。关于模型的实证应用、具体设定和结果分析等详见实证研究一章。部分核心代码实现见附录。

第四章 实证研究

在对主要涉及的金融计量模型的诸多基本设定、假设和估计方法等细节进行详尽深入的讨论后，我们使用前述的研究思路和模型对中国期指市场和股票市场的高频交易数据进行实证分析，沿着时间线对多市场多产品展开，在微观数据层面上给出金融市场微观结构在经历算法交易受限前后明显变化的显著证据，帮助投资者和监管部门了解对应的此类市场参与者的行为特征和交易模式经历了何种变化以及其对市场的影响。本章着重讨论对真实市场高频交易数据的实证研究细节，包括模型设定和结果分析等，并根据模型结果得出关于算法交易在金融市场中扮演的角色的结论。

算法交易近年随着中国金融市场的发展和完善一同进步成长，其主要的投资风格和行为模式、参与市场的程度和对市场造成的影响都在不断地缓慢变化。从市场的总体情况上可以直观看出，在 2014 到 2015 年牛市启动和指数上升期，市场的交易量、交易活跃程度和波动性都有增加，这其中有一定程度是因为国内量化交易投资者的参与和量化交易与算法交易技术的发展导致的。在 2015 年年中随着乐观情绪的扩散，由不合理上行趋势导致的市场系统性风险逐渐累积，市场价格波动的风险也逐渐增加，没有得到充分的释放。有许多人，包括监管层认为，使用算法交易技术进行投资的投资者们会增加额外的不必要的市场价格变动风险，尤其是在价格剧烈波动的时候会带来更强的波动性，损害市场正常运行机制，其在市场中的投资行为也会造成资产价格的不合理波动。本课题根据高频数据模型分析得到的结果表示对此有不同意见，我们认为，算法交易者主要是做两边市场套利，而做纯投机的比例相对比较小，因为风险太大，同时真正投机的主力是在现货市场而不是期货市场。对期货市场进行严格限制对抑制投机行为和稳定市场波动作用不大，反而因为使得众多算法交易投资者离场而损坏了期货市场的正常功能和市场质量，还带来了更多的波动风险。

本课题尝试通过基于细颗粒的交易数据和金融计量模型的实证研究来回答以下问题：在股指期货交易受到严格限制、市场中的交易者成分和行为均受到极大影响后，市场的微观结构性质发生了哪些改变？算法交易者在市场中扮演的角色是什么？对市场微结构有哪些影响？是给市场提供了充足流动性、提高了价格对信息的反应能力和有效性、增加了市场合理的波动性并给市场的平稳运行做出积极影响，还是已经形成了过度活跃的交易，给市场带来了额外的价格波动风险和不必要的多余波动率？在对股指期货的交易进行严格限制的时候，监管部门认为股票市场的剧烈波动是由于市场中的过度投机行为所引起，并认为对股指期货交易进行限制能有效抑制投机行为，使得市场运行更加平稳。但许多学者则认为当时量化交易的发展和广泛使用实际上给金融市场的发展和稳定，尤其是给期货市场的价格发现功能带来了积极的影响和推动，使得期指市场和股票市场的关系更加紧密，从而提高了期现货市场的市场质量。本课题实证研究的重点即为对上述问题进行探讨，在数据分析建模的客观结果基础之上经过独立思考给出答案，对监管部门将来实施的调控监管政策有实际参考意义。

4.1 市场功能、投资者成分和事件分析

4.1.1 股指期货市场的功能

在整个金融市场体系中，股指期货市场拥有非常重要的地位，并且对金融市场和实体经济的稳定发展有非常大的促进和稳定作用。股指期货合约作为将股票指数作为标的资产设计的衍生品，其交易市场和股票市场有着非常紧密的联系。期指市场的主要功能和期现市场的关系有以下几点：

(1) 股指期货能够给投资者提供规避市场整体风险的手段。由于期指和股票都是针对同样的标的金融资产，受同样的宏观经济因素以及整体的供求关系的影响，他们的价格变动方向和运行趋势基本趋于一致，并且期现价差会随着合约到期日的临近而收敛至零。期指市场的存在给广大投资者们提供了进行套期保值和风险管理的有效手段，在做多股票市场的同时，投资者可以通过做空特定的期指合约进行反向交易来对冲股票多头对市场整体趋势风险的暴露。股指期货的存在对跨市场套利者和拥有市场整体多头头寸的套期保值者而言都是不可或缺的。

(2) 股指期货拥有价格发现功能即前瞻性，可以反应投资者们对未来股票现货市场整体趋势的预期。一个有效的股指期货市场中，在多空双方众多投资者的博弈下，市场价格理应能够体现在当前宏观经济趋势下市场参与者对资产的合理预期价格。由于期货市场和现货市场之间的双边套利者的存在和活跃交易，两个市场之间的联系才能得以维持。

(3) 股指期货还能提供投资者做空市场的渠道，使得多空双方能够均衡力量，使得金融资产定价机制更加合理，市场运行更加灵活。拥有不同需求的投资者均可以使用股指期货参与到股票市场的交易中并从中获利。

4.1.2 金融市场参与者组成成分分析

本小节主要对金融市场参与者进行分析，以此来支持我们认为在股指期货交易受限时算法交易者大范围受限并离场同时对市场造成影响的基本分析假设。站在投资者的角度分析，由于众多金融机构投资者整体持仓量庞大，直接对仓位进行大幅度调整不切实际，在产生巨额直接交易成本的同时还会对市场价格产生冲击，形成成交价差成本。因此，诸如券商、保险、公募与私募基金等金融机构通常对市场整体系统风险的暴露非常巨大，急切需要股指期货来对冲。相比之下，散户投资者持仓量小、受期指交易的门槛限制、日常交易频繁且方便、交易成本小，同时风险暴露多为行业和个股风险，故在股指期货市场的参与度较小。此外，相比起股票现货市场，股指期货市场的市场准入门槛相对较高，其对投资者的总体资金量和交易主体都有严格要求。综上，我们认为，股指期货市场和股票现货市场之间联系的桥梁就在于众多金融机构的行为模式上，股指期货市场的交易限制政策对散户影响不大，从而在分析市场微观结构的变化时我们选择忽略散户的行为变化。由于金融机构资金体量大、专业性高、获取信息方便快速、在市场参与程度高且交易频繁，其交易风格以及对市场的期望和判断都对金融市场的稳定有至关重要的作用。同时，股指期货市场中投资者采用算法交易的比例也较高，尤其是进行套利活动的阿尔法套利者和贝塔套利者，实证研究发现仅在 2015 年国内有接近 50%的期货合约是计算机自动化成交，其市场的微观性质和量化交易的活跃程度有很强的关联。

除去散户投资者和非量化的金融机构，绝大多数量化金融机构均使用算法交易方法进行交易需求的执行，根据投资风格可将其分为套利者和投机者两种。在使用量化交易和算法交易的投资者中，套利者占绝大多数，因为借助股指期货进行跨市场套利的收益稳定且风险暴露低，是许多金融机构和基金公司的首选策略。套利者根据具体的行为模式和投资风格具体可以分为阿尔法套利和贝塔套利。阿尔法套利者是通过积极寻找市场中定价有偏差的投资标的来构建表现超越大盘的组合，同时通过做空期货指数来对冲系统整体风险获利。贝塔套利者即为传统的期现套利，通过被动地复制股指期货的标的指数并持有，同时做空期指来赚取价差回归的收益。本课题在研究中认为在 2015 年 8 月底利用期现价差回归性质进行波段交易的贝塔套利者多数均已离场，由于当时现货市场大幅度下跌，市场贴水较深，现货市场难以做空，简单的期现套利已无利可图。反之阿尔法套利者仍然留在市场上，在期现市场间积极寻找套利的机会，由于其投资风格更加主动，只要能够建立起拥有足够超额收益的资产组合并通过股指期货对冲掉市场风险，在当时仍然有机会获得收益，且风险较低。事实上，根据多个基金公司和资产管理公司公布的投资产品净值，他们在市场存在大幅度贴水的情况下，仍积极在期现货市场交易并获利。

另一方面，投机的量化交易者通常使用量化择时策略来发出信号来指导交易。他们主要专注于现货市场，在股指期货市场的参与度较低，远小于套利者。由于本课题考察的时间区间很短，进一步剔除了中长线投资者的行为变化对市场的影响。于是我们认为，在股指期货交易受限的时候，主要受到限制而不得不离场或改变投资策略的，即为一众阿尔法套利者。对于这部分投资者，由于他们整体持仓量很大，风险暴露较大，多使用量化选股的投资策略，交易频率较高，大多使用算法交易的执行方式来完成调仓，参与到市场交易中，以此来尽可能减少交易成本和市场冲击。

值得一提的是，对于采用阿尔法套利者而言，使用期指对冲系统性风险本属于套期保值的目的，但由于中金所对套保行为，尤其是通过空头套保的交易行为监管非常严格，他们之中很多在投资者分类中没有被划归为套保者而是被视为投机者（在之后说明的股指期货市场限制政策中受到了直接的严格限制），另一小部分虽有套期保值的权限，但中金所对其持仓和每日交易的额度限制也非常严格，难以灵活根据其投资策略进行交易，基本无法正常参与市场进行投资活动。同时，另有中金所对期现货严格匹配和对套保交易频率的限制制度，而使用阿尔法超额收益策略的投资者并不是只在对应的指数成分股中而是在全市场中选取标的资产，同时交易频率相比起限制标准要高许多。因此，多数阿尔法超额收益策略的投资者在对股指期货交易进行限制后，正常的投资交易行为已经难以继续。于是，在对投机者的打击中，本来担任着连接调节期现市场关系重任的阿尔法套利投资者则被驱逐，不得不退出股指期货市场。进一步地，许多具有股票现货风险暴露的基金公司和机构投资者失去了对冲系统风险的途径，由于其整体风险暴露巨大，需要迅速降低风险暴露。同时，其投资风格将趋于保守，抑或是直接退出市场休息，抑或是将重心放在量化择时交易等偏向于对股票现货投机的交易策略上，行为模式将向普通散户靠拢。这些变化都将对股票现货市场带来更大的下行压力和波动空间，引发潜在的价格波动风险。

我们认为，频繁使用算法交易的阿尔法套利投资者由于其目的是为了发掘现货市场中被错误定价的股票，其频繁交易行为对市场的有效定价首先是有益的，同时其在期货市场中进行对冲的需求也与其在现货市场中的多头头寸紧密相关，他们的存在首先能提高市场流动性，让期现市场关系得以维持，保持期货市场的基本功能。同时，由于他们广泛使用交易算法进行高频率的下单，其在市场上的操作行为又和普通的投资者或投机者有所不同，由于其本身使用交易算法就是为了减少市场冲击和降低交易成本，这样的行为对于整个市场运行而言也可以使价格变动平稳、让市场价格能及时有效地反映并通过稳健的细微波动消化掉市场上的

新信息，进一步缩小买卖交易价差、完善市场的价格形成机制、减少价格变动的突发性和集束性。反之，如果此类交易者无法继续参与市场，则期货市场在经历成交量急跌的同时，必然会产生买卖价差增加、买卖深度扩大、交易成本剧增、价格变动迟缓且效率降低等变化，在引发价格局部震荡加剧的同时，使得期货市场的基本功能无法实施，升贴水会更加严重；对于现货市场而言，他们在期货市场中对冲风险的工具失效后必然会对现货市场的风险暴露进行大幅度的调整和管理，对市场造成额外的抛压和波动，短期内导致市场多空双方能力的失衡，长期则会降低现货市场的流动性和提高交易成本，这样的变化对于金融市场的发展无疑是有害的。

4.1.3 股指期货市场限制政策

在 2015 年 8 月 26 日与 9 月 7 日，中金所对股指期货交易两次实施了限制性政策，对使用算法交易的股指期货投资者，尤其是阿尔法套利者而言是非常大的打击。具体而言，在 2015 年 8 月 26 日，实施的新规定内容为：沪深 300、上证 50、中证 500 股指期货合约（后称三大期指合约）保证金和平今仓交易手续费标准提高，并调整三大期指合约日内开仓限制标准。在本课题的研究中，称此次规定实施为“事件 A”。在 2015 年 9 月 7 日，再次实施的限制规定内容为：

(1) 三大合约非套保投资者在单个产品单日开仓交易量超过 10 手构成“日内开仓交易量较大”的异常交易行为。

(2) 三大合约非套保持仓保证金标准从 30%提高至 40%。

(3) 三大合约平今仓交易手续费继续从成交金额的 0.0115%提高到 0.23%。

(4) 更加严格的交易账户限制。

同样地，称此次规定实施为 “事件 B”。所有政策限制可以总结为对交易量的限制和对交易成本的提高，本身中金所认为过度投机交易导致了市场异动，于是主要想通过限制交易和降低杠杆的政策限制来抑制过度投机。但在抑制投机行为的政策下，使用算法交易的套利者，尤其是阿尔法套利者已经无法继续参与交易。由于其投资标的的选取范围不符合中金所要求以及调仓操作过于频繁，他们并没有被归类为套期保值者而是属于投机者，因此在交易额度和交易成本的双重限制下，量化选股策略的实施已经困难重重。对于期货市场中没有现货头寸暴露的纯投机者而言，对交易成本的提高也的确抑制了其交易行为，但本课题认为在事件 AB 前后受影响最大且对市场冲击最大的是使用算法交易的阿尔法套利投资者。

4.2 研究框架

本节讨论实证研究的整体框架，包括研究思路、标的选取、具体的实施方案等。

4.2.1 研究标的和时间窗的选取

实证研究的主要研究标的为股指期货市场中当月的主力交易合约，股票市场中各行业的代表性股票。在研究股票指数期货市场的微观结构特性时，研究标的为市场上交易活跃的IC、IF 和 IH 合约，标的分别为中证 500 指数、沪深 300 指数和上证 50 指数。考虑到整体事件的研究窗为 2015 年 8、9 月，尤其是 8 月中下旬到 9 月的中上旬，时间跨度为一个月左右，为了保证事件前后数据和模型的一致性和对比研究的有效性，本课题一律选取 2015 年 09 月 18 日交割的合约进行，例如 IF201509 股指期货合约。

由于本课题从事件研究出发，故设定主要事件发生时间点为 2015 年 8 月 26 日以及 2015 年 9 月 7 日，选取时间点前后一周到 10 天的时间作为主要研究的时间段。事件 A 发生前的一周为第一时间窗，为 8 月 19 日到 8 月 26 日，事件 A 发生后事件 B 发生前的时间段为第二时间窗，为 8 月 26 日到 9 月 7 日，事件 B 发生后的 10 天为第三时间窗，为 9 月 7 日到 9 月 16 日。全事件时间窗即为 8 月 19 日到 9 月 16 日，在接近一个月的大时间窗里，实证研究重心放在分析市场微观结构表征数据以及建模分析结果经历了何种变化上。时间窗中交易日总计 19 日，分别为 8 月 19 日～21 日、8 月 24～28 日、8 月 31 日、9 月 1 日～2 日、9 月 7 日～11 日、9 月 14 日～16 日。

在对数据的研究中发现，在整个研究针对的时间窗中，选取的合约均为当时的主力合约，即交易量和强度最高的合约，能够充分反映出市场运行的特性和微观结构性质。此外幸运的是，在时间窗中主力合约的交易强度和交易量也未因为交割日的临近而产生明显变化，在整个时间窗中，除去由于事件本身造成的交易活跃度的大幅度降低以外，交易数据的量基本保持稳定，因此本课题在研究假设中排除由于距离交割日期的变化而引起的市场性质变化。

4.2.2 主要研究方案

本课题以事件研究法为基础，主要讨论的事件是证监会在 8 月 26 日和 9 月 7 日出台的对股指期货交易，尤其是对算法交易与程序化高频交易等量化交易投资者投资行为的严格限制。本课题把包含这些事件的时间窗口中的数据和事件前后的数据分开单独分析。我们首先研究股指期货市场，将原始高频交易数据进行清洗和整理，并将同一合约在三个子时间窗中的数据进行整合，同时生成可供量化模型使用的一系列变量，使用金融计量模型对其分别进行建模分析，比较子时间窗建模结果之间的差别，从而支持关于市场流动性、价格发现能力、波动性和有效性等性质变化的结论。主要使用的计量分析方法包括对微观结构进行整体描述比较、对价格变动量序列和对价格变动久期序列进行建模等。每个模型和分析方法都针对着某些特定的市场微观结构特性来进行挖掘，对充分认识市场微观性质很有必要。

4.3 股指期货市场微观结构的变化

本节主要讨论期指合约市场微观结构和市场质量的实证研究结果与解释。主要关注的股指期货合约为 IC1509，IF1509 以及 IH1509，分别对应的资产标的为中证 500 股票指数，沪深 300 股票指数和上证 50 股票指数。三个股指期货合约的高频交易数据均表现出极其相似的变化特征，本节选取当时持仓量和总交易量均最大的 IF1509 合约作为主要讨论对象，对该合约的高频交易数据进行建模并对结果进行分析。所有建模过程和图片输出均在 R Studio 平台上实现。

4.3.1 基本微观结构描述

本节从股指期货市场高频交易数据出发，讨论基本的微观结构特征。主要使用交易价格变化量序列、交易价格变动久期序列、买卖价差序列等的基本统计量来初步描述市场微观结构在事件前后，尤其是事件 B 前后的变化。图 4-1 和图 4-2 为 IF1509 合约的逐笔成交量和成交额的在事件前后 3 个时间窗中日内盘中变化，从左到右从上到下依次是，事件 A 前后（8 月 21 日与 8 月 27 日），事件 B 后即刻（9 月 7 日）与一周后（9 月 14 日）。

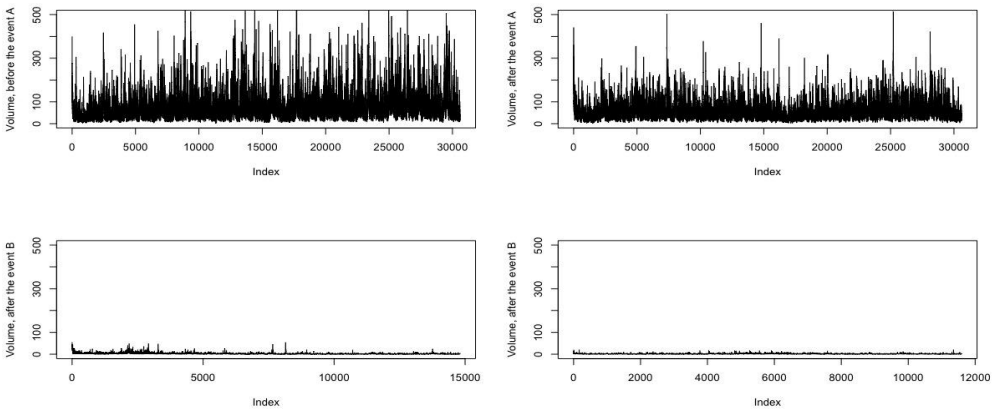

图 4-1 IF1509 在事件 AB 前后的日内盘中交易量变化

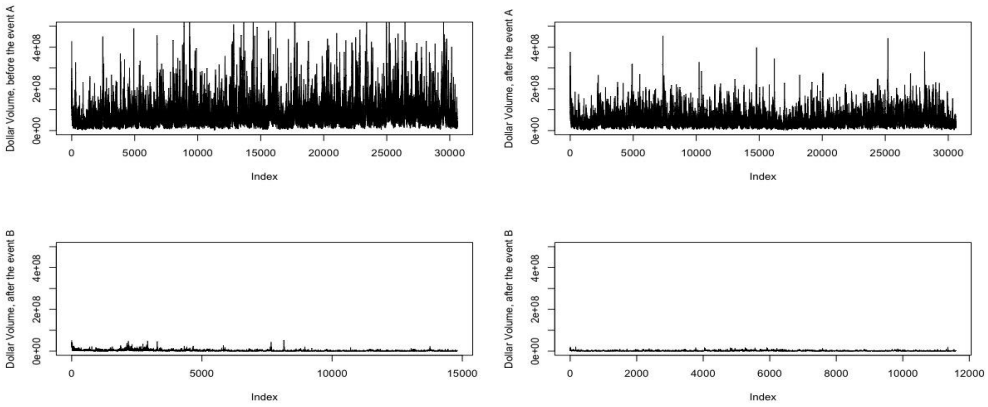

图 4-2 IF1509 在事件 AB 前后的日内盘中交易额变化

从高频交易快照数据可以看到在事件 A 发生后市场整体交易强度和活跃度的较大变化，交易量和交易额均明显萎缩，在事件 B 之后，股指期货的交易者的交易意愿更是受到重创，日内盘中交易量和交易额降低了数十倍，交易量的平均值从 8 月 21 日的 74.145 手降低到了 9 月 14 日的 1.726 手，这和对股指期货交易的交易量限制以及其导致的大量算法交易者离场有直接关系。在盘中交易价格变化方面，价格变化量序列 $\boldsymbol{pch}_{t_i}, \boldsymbol{i} \in \mathbb{N}$ 呈现出明显的尖峰厚尾特性，见图 4-3。

进一步地，为了更直观描述价格变化，我们将价格变动分为 7 个类别，具体分类方法为将整个价格变动区间取分位点后等距划分。例如总区间为 $[-\mathbf{10}, \mathbf{10}]$ 时，则对应的 7 个类别区间即为：$(-\infty, -\mathbf{10},)[-\mathbf{10}, -\mathbf{5}), [-\mathbf{5}, \mathbf{0}), \mathbf{0}, (\mathbf{0}, \mathbf{5}], (\mathbf{5}, \mathbf{10}], (\mathbf{10}, \infty)$ 。表 4-1 为对每天的高频交易价格变化数据进行统计后得到的百分比数值。

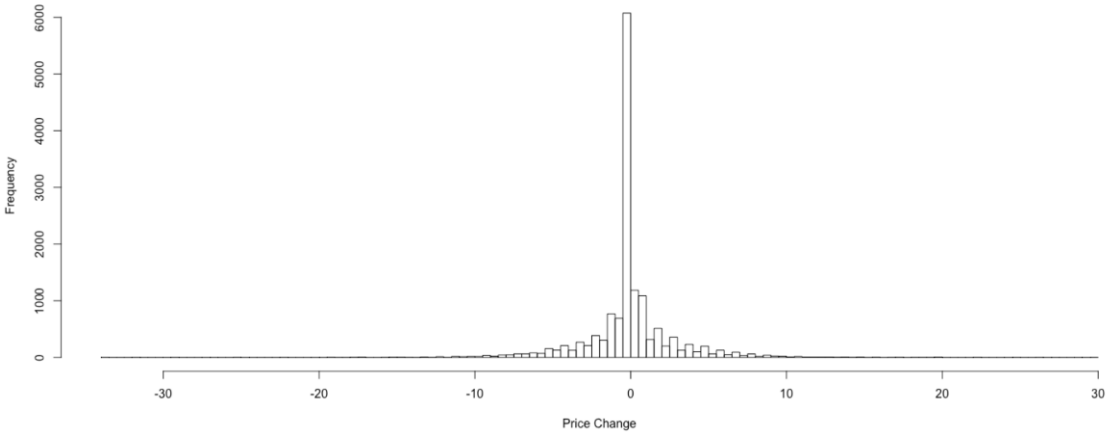

图 4-3 IF1509 在 2015 年 9 月 7 日的盘中价格变化频率直方图

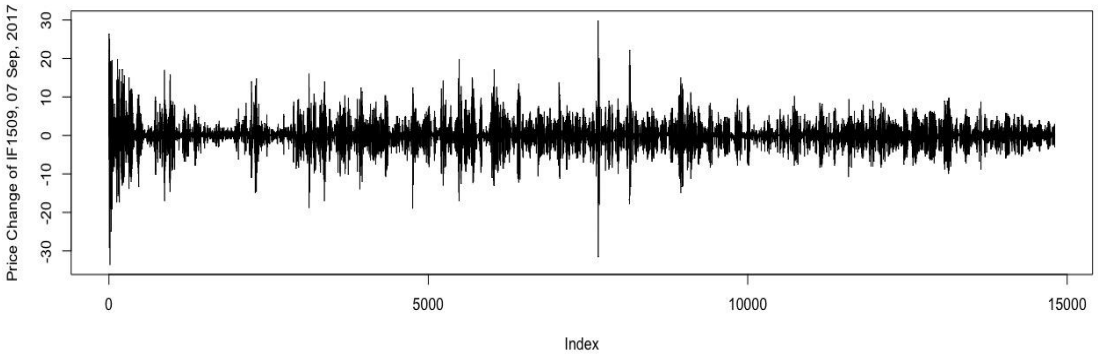

图 4-4 IF1509 在 2015 年 9 月 7 日日内盘中价格变化序列

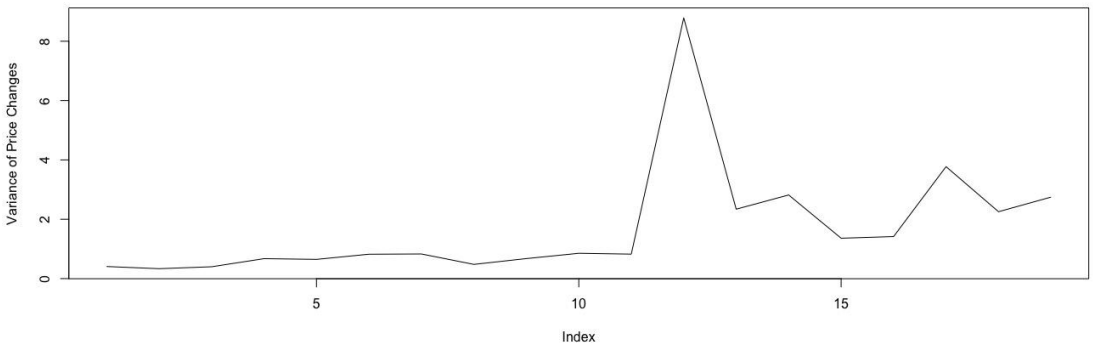

图 4-5 IF1509 在全事件窗口期的日内盘中价格变化的样本方差变化

表 4-1 IF1509 在全事件窗口期的价格变化分类频数百分比统计

日期	1	2	3	4	5	6	7
8 月 19 日	4.03%	7.11%	28.08%	34.17%	19.75%	4.21%	2.64%
8 月 20 日	5.26%	11.45%	23.31%	34.02%	16.75%	6.30%	2.92%
8 月 21 日	3.85%	7.22%	17.15%	43.77%	16.98%	7.36%	3.68%
8 月 24 日	3.46%	7.09%	18.79%	42.36%	18.45%	6.40%	3.46%
8 月 25 日	3.53%	7.31%	20.00%	39.01%	19.66%	7.28%	3.21%
8 月 26 日，事件 A	4.68%	9.44%	17.90%	42.95%	15.75%	6.27%	3.02%
8 月 27 日	3.14%	6.66%	22.56%	35.24%	22.60%	6.47%	3.32%
8 月 28 日	3.00%	10.74%	15.29%	41.78%	15.05%	11.11%	3.02%
8 月 31 日	3.71%	8.00%	17.97%	40.94%	17.98%	7.75%	3.65%
9 月 01 日	3.55%	6.31%	21.38%	38.26%	20.66%	6.36%	3.48%
9 月 02 日	3.15%	6.45%	14.26%	45.17%	17.22%	8.98%	4.77%
9 月 07 日，事件 B	2.63%	4.38%	9.81%	66.34%	9.99%	4.21%	2.62%
9 月 08 日	2.84%	5.29%	9.87%	62.82%	11.28%	5.00%	2.90%
9 月 09 日	3.11%	3.94%	12.53%	60.93%	12.55%	4.13%	2.81%
9 月 10 日	2.89%	5.35%	12.94%	58.30%	11.87%	5.44%	3.21%
9 月 11 日	3.32%	5.66%	11.95%	57.99%	12.12%	5.68%	3.28%
9 月 14 日	2.13%	4.04%	9.02%	65.92%	12.10%	4.30%	2.49%
9 月 15 日	3.71%	5.60%	13.24%	58.42%	10.92%	5.00%	3.10%
9 月 16 日	2.41%	5.25%	13.69%	59.69%	12.30%	4.43%	2.23%

在股指期货严格受限即事件 B 发生后，价格变化的最小值和最大值均分别下降和升高，整体价格幅度扩大，但价格变化的整体数量和各个价格变化分类的绝对数量都有显著降低，表示市场单次价格变化的总数有所下降，但这并不意味着市场稳定性增加。从表 4-1 中可以看出，在每一个价格变化所占的比例分布中，价格不变的比例（分类 4）虽然大幅度增加，同时价格发生小幅和中度波动（即分类 2，3，5，6）的比例明显减少，而价格发生较大波动（即分类 1，7）的比例却没有明显的变化，说明尽管市场整体的交易强度有明显的下降，且价格不变的比例的确有所升高，市场看似获得了更好的平稳性，但实际上在价格变化整体比例降低的同时，较大的价格变化比例没有发生变化。事实上由于在事件 B 发生后价格变动幅度加大，价格类别 1 和 7 在事件 B 发生后意味着更大幅度的价格变动，同样的标准下事件 B 发生后的较大价格变动比例甚至更大。同时，价格变动的方差有着非常明显且持久的上升，平均值从事件 B 之前的 7.57 跃升到事件 B 之后的 27.47。于是，本课题认为，高频交易数据此类特殊情况预示着市场微观结构的隐含问题，即整体波动变小不意味着市场的价格变动风险降低，仅从单次价格变动的高频数据上看（不考虑价格变动的条件分布和时间序列滞后项之间的相关关系），较大的价格变动值依然存在，市场价格要么保持不变，要么就发生较大的价格变动，预示着更高的价格波动风险和市场活动的低效率。

进一步分析实时买卖双方价差数据。图 4-6 是通过在三个时间窗中将交易价差序列数据整合后进行均匀重抽样得到的，可以看出其波动程度和极差分布在事件 B 发生后有显著的扩大，且在之后的 10 天内都保持在较高的水平上，意味着市场中流动性的枯竭和极高的交易成本。从表 4-2 可看出，在事件 A 发生后，价差的中位数、均值都有显著增加，在事件 B 发生后，均值更是从事件 A 后的 0.7 左右上升到 2 左右，中位数和第三分位数均有增加。进一步地，逐日计算交易价差变化的样本方差，从图 4-7 中可发现样本方差在事件 B 发生后增加非常明显，其变化模式和交易价格变化量的变化模式非常相似，共同反映出窗口期日内盘中市场运行情况的显著变化。

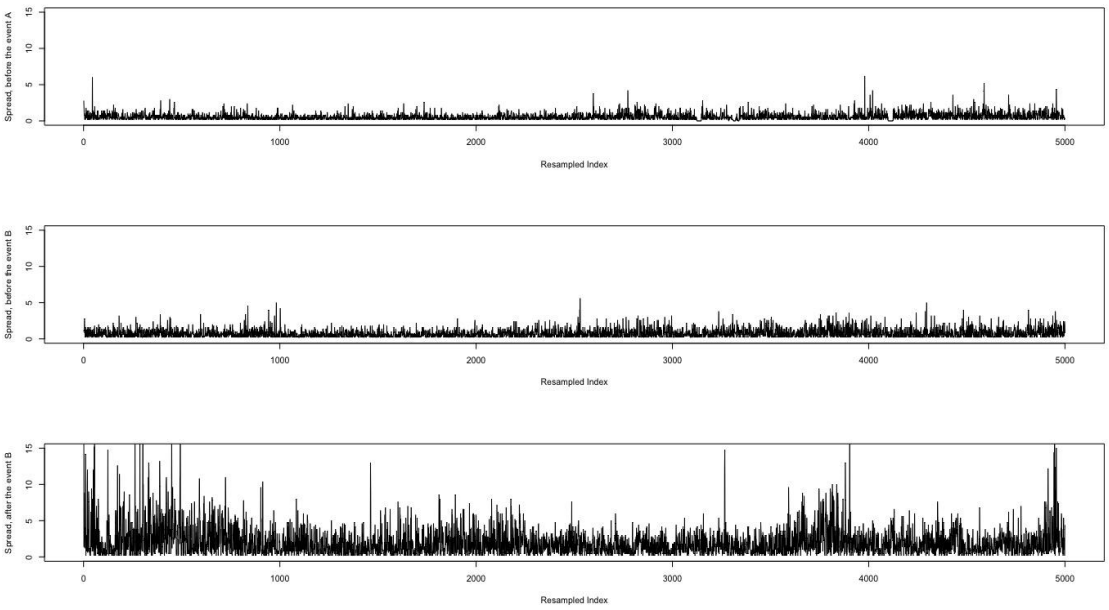

图 4-6 IF1509 在三个子事件窗中的买卖价差序列

表 4-2 IF1509 在全事件窗口期的每日买卖交易价差基本描述统计量

日期	最小值	第一分位数	中位数	均值	第三分位数	最大值
8 月 19 日	0.2	0.2	0.4	0.5716	0.8	6.8
8 月 20 日	0.2	0.2	0.4	0.4735	0.6	4.2
8 月 21 日	0.2	0.2	0.4	0.493	0.6	5.8
8 月 24 日	0	0.2	0.4	0.5874	0.8	9.8
8 月 25 日	0	0.2	0.4	0.5784	0.8	11.2
8 月 26 日，事件 A	0.2	0.4	0.6	0.75	1	8.8
8 月 27 日	0.2	0.4	0.6	0.7493	1	7.8
8 月 28 日	0.2	0.2	0.6	0.6095	0.8	7.8
8 月 31 日	0.2	0.4	0.6	0.7965	1	7.4
9 月 01 日	0.2	0.4	0.8	0.8763	1.2	15.6
9 月 02 日	0.2	0.4	0.8	0.8596	1.2	9.6
9 月 07 日，事件 B	0.2	0.8	2.2	3.077	4.6	57.2
9 月 08 日	0.2	0.6	1.4	1.797	2.6	17.8
9 月 09 日	0.2	0.6	1.6	2.052	3	13.2
9 月 10 日	0.2	0.6	1.4	1.631	2.4	9.8
9 月 11 日	0.2	0.6	1.4	1.599	2.4	9.2
9 月 14 日	0.2	0.8	1.6	2.282	3.2	17
9 月 15 日	0.2	0.8	1.6	1.933	2.8	19
9 月 16 日	0.2	0.6	1.4	1.893	2.6	24.8

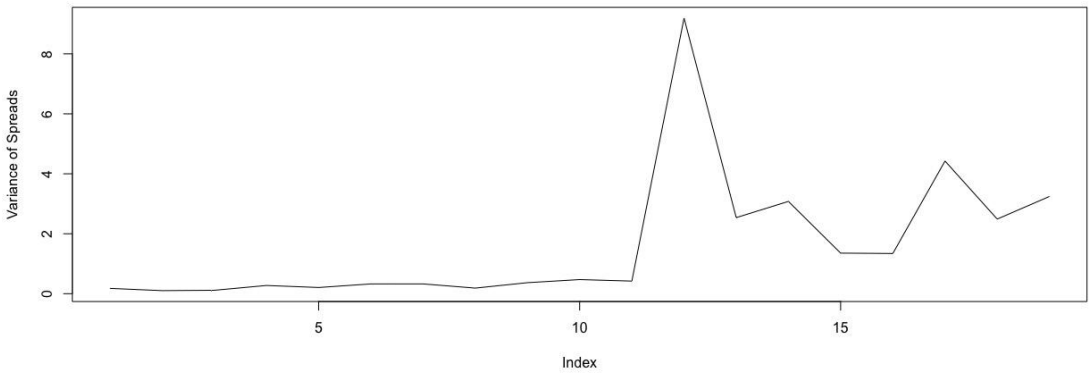

图 4-7 IF1509 在全事件窗口期的日内盘中买卖交易价差的样本方差变化

4.3.2 已实现波动率和价格波动概率分析

本小节使用 3 个和价格变动有关的模型对当时交易最活跃的沪深 300 股指期货在 2015 年 8、9 月的价格变动序列 $pch_{t_i}, i \in \mathbb{N}$ 进行详细研究，发现了在政策限制前中后三个时间段，价格变动动态特性发生了非常显著的变化，反映出了算法交易者的离场和行为模式的变化对股指期货市场的基本功能和动态特征的巨大影响。为了更好地描述高频交易数据的波动特征，首先通过设定一系列重采样区间（0.5 秒，5 秒，10 秒，20 秒，1 分钟，2 分钟，5 分钟，10 分钟等）来计算出不同的时间间隔下的已实现波动率表格。在进行重采样时，可以通过设定不同的函数来确定每个区间中如何选取代表价格来形成价格序列以支持已实现波动率的计算，本课题中使用的函数为最大值和最小值的平均值。使用了不同内部采样函数计算的已实现波动率再进行平均可以一定程度上消除市场微观结构噪声的干扰。对于 IF1509 合约，已实现波动率表格见表 4-3，对每日的不同间隔的波动率画图见图 4-8。

表 4-3 IF1509 在全事件窗口期的不同时间间隔下的已实现波动率

日期	0.5 秒	5 秒	10 秒	20 秒	1 分钟	2 分钟	5 分钟	10 分钟	20 分钟	30 分钟
8 月 19 日	0.488	0.397	0.429	0.464	0.424	0.422	0.428	0.448	0.461	0.529
8 月 20 日	0.442	0.389	0.393	0.395	0.390	0.385	0.422	0.378	0.291	0.267
8 月 21 日	0.499	0.479	0.495	0.493	0.512	0.527	0.541	0.559	0.540	0.449
8 月 24 日	0.641	0.660	0.646	0.643	0.648	0.597	0.583	0.636	0.644	0.785
8 月 25 日	0.707	0.740	0.749	0.761	0.749	0.727	0.687	0.652	0.685	0.768
8 月 26 日	0.877	0.875	0.898	0.904	0.884	0.902	0.953	1.017	0.972	0.846
8 月 27 日	0.865	0.860	0.884	0.928	0.895	0.856	0.835	0.788	0.847	0.602
8 月 28 日	0.626	0.546	0.535	0.521	0.506	0.508	0.394	0.373	0.320	0.258
8 月 31 日	0.745	0.537	0.566	0.592	0.584	0.595	0.569	0.517	0.499	0.471
9 月 01 日	0.849	0.691	0.685	0.727	0.721	0.758	0.819	0.753	0.715	0.758
9 月 02 日	0.827	0.721	0.756	0.775	0.724	0.746	0.735	0.603	0.668	0.722
9 月 07 日	1.804	0.805	0.743	0.758	0.788	0.758	0.787	0.712	0.697	0.634
9 月 08 日	0.902	0.457	0.406	0.383	0.406	0.408	0.431	0.389	0.425	0.395
9 月 09 日	0.881	0.458	0.378	0.372	0.386	0.396	0.359	0.300	0.232	0.208
9 月 10 日	0.541	0.316	0.258	0.242	0.240	0.243	0.230	0.224	0.177	0.123
9 月 11 日	0.549	0.323	0.245	0.214	0.215	0.219	0.222	0.226	0.196	0.167
9 月 14 日	1.042	0.512	0.410	0.397	0.452	0.511	0.561	0.593	0.629	0.583
9 月 15 日	0.766	0.408	0.327	0.319	0.355	0.384	0.347	0.313	0.229	0.301
9 月 16 日	0.804	0.437	0.376	0.333	0.347	0.373	0.445	0.513	0.328	0.264

我们认为，在市场交易活跃度大大降低，流动性大大减弱的同时，市场的稳定性并没有得到好转，反而更加难以及时准确地在价格上反映出新信息；量化交易者或计算机程序化下单没有在微观结构上给资产交易带来过度的额外的噪音和风险，在其受限离场后市场反而表现出较差的稳定性和更多的价格波动风险。观察还发现，在事件 A 和事件 B 发生后，市场微观结构噪声和价格变动序列的自相关性有所增加，带来了更多的额外的误差方差以及自协方差，导致 0.5 秒时间间隔的已实现波动率大幅度上升，同时其他时间间隔计算的已实现波动率也有较大幅度上升，这些波动率在事件发生后数日虽然逐渐下降但仍保持在较高位置，高于或等于交易受限之前的值。在价格序列已经在更长的时间段上保持不变、市场交易活跃大幅度降低、成交量和成交额均显著降低的情况下，这样的数据变化特征说明了价格一旦变化，其变化量序列对整体波动率的影响是巨大的。在更长的时间间隔上，已实现波动率在事件发生之后均出现了即刻的较大增加（可以用市场的恐慌情绪以及机构投资者和基金公司对其头寸进行调整解释），但之后逐渐降低到事件发生前的水平，没有进一步降低。这些较长区间计算的波动率在损失了一部分市场微观结构信息的情况下也避免了市场微观结构噪声的干扰。可以看出，较长区间上计算的已实现波动率在稳定后并没有出现与价差序列、价格变化和交易量交易额序列所呈现的形态相似的巨大差别，说明市场整体波动依旧，同时量化交易者并没有在分钟尺度上给市场带来额外的波动率。

我们进一步使用 OPM 顺序概率值模型对价格变动序列进行建模研究，着重分析日内盘中价格变化的动态相依性。顺序概率值模型通过对隐含价格变动进行建模估计，得到隐含价格和观测到的价格之间的对应关系并对价格变化的条件概率进行预测。从模型的结构中可以看出价格变动和隐含价格变动之间的深层次关系，我们可以利用这些关系对拟合的价格变化条件概率进行分析，进一步认识微观层面上价格变动的波动性和强度，以及其滞后项对隐含价格变动的影响。在使用顺序概率值模型时，本课题先选取某一天的高频交易价格变动数据进行建模，在事件前后发现拟合的价格变化条件概率发生明显变化后，进一步对三个时期的所有价格变动数据分别进行统一建模。

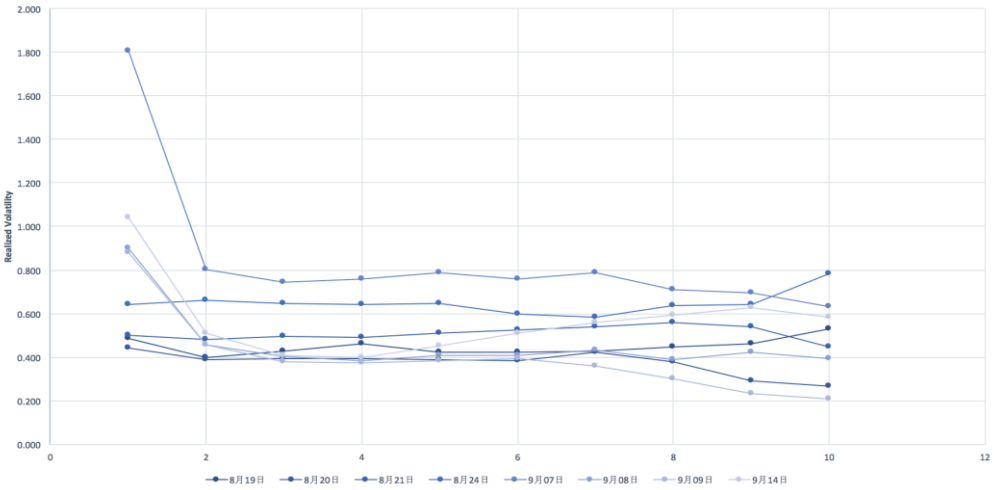

图 4-8 IF1509 在全事件窗口期的不同时间间隔下的已实现波动率

模型使用的变量包括了价格改变的类别、价格改变量以及成交量的当期值和滞后值，滞后阶数将在模型中进行筛选，价格变动类别数量设定为 7 已经足够支持模型。设价格变动类别变量序列 $ctg_{t_i}, i \in \mathbb{N}$ ，变量取值在 1~7 之间，用于标记价格变化的类别，可以理解为“大幅下降”、“中度下降”、“略微下降”、“价格不变”、“略微上升”、“中度上升”和“大幅上升”，其值由对整体价格波动区间的分位点进行等距划分后和价格变动值序列 $pch_{t_i}, i \in \mathbb{N}$ 一同决定。对于股指期货交易市场而言，由于价格变动通常不止在 6 个最小波动单位内浮动，同时市场微观结构噪音的成分相比起最小价格波动单位而言影响较大，在使用顺序概率值模型对股指期货价格变动序列建模时，使用分类值来代替价格变动实际值作为被解释变量，这样有助于充分捕捉滞后值和当期值，即域流 $\mathcal{F}_{i-1}$ 中的信息，而尽可能避免市场微观结构噪声的影响。我们认为，挖掘出对价格变动类别的条件概率分布信息已足够支持本课题的论点。尽管如此，在解释变量中仍然会加入价格变量值序列 $pch_{t_i}, i \in \mathbb{N}$ ，同时由于分类值为因子变量，在使用价格变动类别变量滞后值作为解释变量时需要根据 7 个类别分别定义相应的哑变量用于回归，设 l 为滞后阶数，有：

$$ctg_{l,j} = \begin{cases} 1, & ctg_{t_{i-l}} = s_j \\ 0, & otherwise \end{cases} \tag{4-1}$$

经过比对不同的模型拟合结果，最终对 IF1509 合约建模使用的模型设定为：

$$\begin{aligned} pch^*_{t_i} &= \beta^T \mathrm{x}_{t_i} + \epsilon_{t_i} \\ \beta^T \mathrm{x}_{t_i} &= \sum_{j=2}^{7} \gamma_{1,j}\, ctg_{1,j} + \sum_{j=2}^{7} \gamma_{2,j}\, ctg_{2,j} \\ &\quad + \sum_{j=2}^{7} \gamma_{3,j}\, ctg_{3,j} + \sum_{l=1}^{2} \beta_l\, pch_{t_{i-l}} + \beta_3 vol_{t_{i-1}} \end{aligned} \tag{4-2}$$

其中 $vol_{t_{i-1}}$ 为交易量的一阶滞后项。为简化模型，对于残差项 ϵ_{t_i} 的条件方差函数设定为常量 σ^2 。对于隐含价格 $pch^*_{t_i}$ 的区间边界分割值 $\alpha_j, j = 1, \cdots, k-1$ 可见顺序概率值模型的具体讨论。在使用模型时，我们对交易数据按照三个子时间窗进行分段整合后分别进行拟合（第一时间窗为事件 A 发生前一周，第二时间窗为事件 AB 发生之间 10 天，第三时间窗为事件 B 发生后 10 天）。

表 4-4 IF1509 在三个子时间窗中顺序概率值模型的边界分割值估计结果

	α_1	α_2	α_3	α_4	α_5	α_6
第一时间窗估计值	-1.626	-1.038	-0.303	0.749	1.473	2.029
标准误	0.0320	0.0318	0.0318	0.0318	0.0319	0.0322
t 值	-50.78	-32.58	-9.51	23.53	46.16	63.10
第二时间窗估计值	-2.087	-1.460	-0.757	0.330	1.026	1.652
标准误	0.0407	0.0405	0.0404	0.0404	0.0404	0.0406
t 值	-51.30	-36.10	-18.75	8.18	25.38	40.70
第三时间窗估计值	-4.453	-3.885	-3.261	-1.346	-0.710	-0.150
标准误	0.0579	0.0573	0.0571	0.0567	0.0566	0.0564
t 值	-76.95	-67.74	-57.11	-23.76	-12.56	-2.66

表 4-5 IF1509 在三个子时间窗中顺序概率值模型的解释变量参数估计结果

	$\gamma_{1,2}$	$\gamma_{1,3}$	$\gamma_{1,4}$	$\gamma_{1,5}$	$\gamma_{1,6}$	$\gamma_{1,7}$	$\gamma_{2,2}$
第一时间窗估计值	-0.0936	-0.1105	-0.0399	0.0151	0.0467	-0.0697	-0.0961
标准误	1.68E-02	1.86E-02	2.20E-02	2.70E-02	3.25E-02	4.10E-02	1.68E-02
t 值	-5.57	-5.95	-1.81	0.56	1.44	-1.70	-5.72
第二时间窗估计值	-0.1099	-0.2459	-0.2864	-0.3242	-0.4228	-0.5763	-0.1072
标准误	2.00E-02	2.23E-02	2.68E-02	3.25E-02	3.84E-02	4.82E-02	2.00E-02
t 值	-5.50	-11.01	-10.70	-9.98	-11.02	-11.97	-5.36
第三时间窗估计值	-0.3831	-0.7915	-1.2115	-1.6073	-2.0605	-2.4142	-0.2700
标准误	0.030	0.031	0.034	0.040	0.048	0.059	0.031
t 值	-12.76	-25.37	-35.92	-40.36	-43.37	-41.14	-8.84
	$\gamma_{2,3}$	$\gamma_{2,4}$	$\gamma_{2,5}$	$\gamma_{2,6}$	$\gamma_{2,7}$	$\gamma_{3,2}$	$\gamma_{3,3}$
第一时间窗估计值	-0.0474	0.1098	0.2306	0.2673	0.1681	-0.0620	-0.0156
标准误	1.85E-02	2.20E-02	2.69E-02	3.24E-02	4.09E-02	1.49E-02	1.34E-02
t 值	-2.56	5.00	8.57	8.25	4.11	-4.16	-1.16
第二时间窗估计值	-0.1674	-0.0331	0.1085	0.0614	-0.0281	-0.0066	-0.0181
标准误	2.23E-02	2.68E-02	3.25E-02	3.83E-02	4.81E-02	1.75E-02	1.60E-02
t 值	-7.50	-1.24	3.34	1.60	-0.58	-0.38	-1.13
第三时间窗估计值	-0.5695	-0.6677	-0.8105	-1.0507	-1.3102	-0.1993	-0.3491
标准误	0.032	0.034	0.040	0.048	0.059	0.028	0.025
t 值	-17.99	-19.61	-20.24	-21.97	-22.18	-7.12	-13.73
	$\gamma_{3,4}$	$\gamma_{3,5}$	$\gamma_{3,6}$	$\gamma_{3,7}$	β_1	β_2	β_3
第一时间窗估计值	0.0775	0.1564	0.1563	0.1088	-0.0200	0.0166	0.0004
标准误	1.30E-02	1.36E-02	1.57E-02	1.86E-02	2.04E-03	2.03E-03	4.89E-05
t 值	5.98	11.47	9.94	5.86	-9.78	8.16	9.08
第二时间窗估计值	0.1172	0.2414	0.3021	0.2880	-0.0196	0.0137	-0.0002
标准误	1.54E-02	1.60E-02	1.75E-02	2.03E-02	2.08E-03	2.08E-03	8.84E-05
t 值	7.64	15.08	17.30	14.17	-9.45	6.58	-2.13
第三时间窗估计值	-0.4273	-0.5009	-0.6625	-0.9430	-0.0136	-0.0113	0.0015
标准误	0.024	0.026	0.029	0.033	0.001	0.001	0.002
t 值	-18.13	-19.47	-22.71	-28.50	-14.68	-12.26	0.75

除去个别哑变量外，大多数参数均在通常的 5%水平下是显著的，尤其是边界分割值的估计值 t 值较显著，估计结果较好。从估计的边界分割值中首先可以看出，估计的分割区间长度基本是对称的，但第三时间窗的估计值总区间长度（中间 5 个区间的总长度）为 4.6 左右，大于第一第二时间窗的 3.6 左右的估计值区间长度，说明事件 B 发生后，对应的隐含价格变化幅度有所增加。其次，估计的区间分割值逐渐降低，说明隐含价格有下降的趋势。进一步地，使用拟合的参数计算三个时间窗中价格变化的条件概率估计值，各价格变动分类值的估计条件概率值均值和方差分布见表 4-6，其在三个时间窗中的序列值见图 4-9。图 4-9 中从上到下分别是事件 A 前、事件 A 后事件 B 前和事件 B 后的拟合概率值。

表 4-6 IF1509 在三个时间窗中各价格变动分类值的估计条件概率值均值方差分布

	均值			方差		
	第一时间窗	第二时间窗	第三时间窗	第一时间窗	第二时间窗	第三时间窗
1	0.0419	0.0331	0.0276	2.86E-04	3.54E-04	1.81E-03
2	0.0830	0.0764	0.0506	4.73E-04	7.41E-04	1.55E-03
3	0.2092	0.1831	0.1163	8.84E-04	1.37E-03	2.91E-03
4	0.3931	0.4024	0.6117	1.01E-04	2.81E-04	7.07E-03
5	0.1787	0.1871	0.1176	9.46E-04	1.30E-03	3.02E-03
6	0.0628	0.0815	0.0490	3.38E-04	8.02E-04	1.48E-03
7	0.0313	0.0365	0.0272	1.86E-04	4.28E-04	1.77E-03

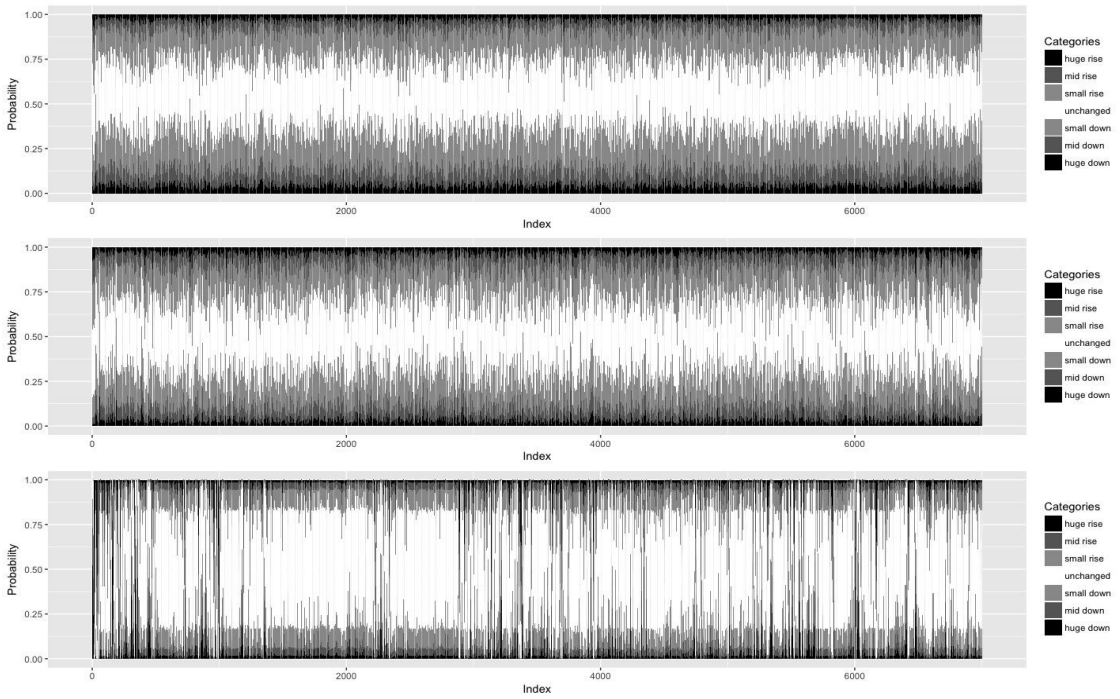

图 4-9 IF1509 在三个时间窗中各价格变动分类值的估计条件概率值

从条件概率值的均值方差可以看出，在事件 A 和事件 B 发生后，价格在统计意义上更有可能保持不变，但较大变动的可能性并没有显著降低，尽管价格不变的可能性显著升高。此外在第二和第三时间窗，拟合的价格变化条件概率值方差相比起第一时间窗有显著的提升，尤其是在第三时间窗提升明显。以上这些现象和从图 4-9 中观察到的拟合条件概率序列相符。在事件 A 发生后，市场价格波动隐含条件概率值已经有了较大波动，说明市场价格稳定性降低，这可以用市场的恐慌情绪以及金融机构和基金公司等投资者在为规避风险进行调仓解释。在事件 B 发生后，即算法交易者受限离场后，市场微观结构发生了非常明显的变化，价格变化不再活跃，在价格保持不变的概率整体增加的同时，一旦价格变动的概率变化，即价格有波动的趋势，则变化概率较大，且幅度也较大，可以从大幅度波动的拟合概率值平均值仍在较高水平看出。我们进而认为，在量化交易受限后，市场的流动性枯竭、投资者成交意愿降至极低的位置时，市场看似获得了稳定，但实际上仍聚集着隐含的价格波动风险。市场上有关价格的新信息无法及时准确地反映到资产价格上，而一旦积累到一定程度或一段时间后方才予以释放，此时估计出的价格变动概率非常高，且较大可能为大幅度的价格变动，这对投资者显然意味着更高的价格波动风险。相比之下，在股指期货交易受限之前，市场运行平稳，价格发生波动的概率和幅度均较稳定，说明市场能够很好地及时有效消化信息。从顺序概率值模型拟合结果观察到的现象和从已实现波动率平面得到的结论一致。

4.3.3 价格变化分解分析

为了挖掘并精确地描述和价格变动条件概率分布有关的详细信息，本节进一步使用价格变化分解模型对标的资产高频价格变化进行建模分析。使用的是 ADS 价格分解模型，模型主要用来估计某几种特定的关于价格变化方向和幅度的条件概率，从实验结果中可以观察到价格变化的集束性以及其在股指期货交易受限前后的变化。模型将价格变化分解为三个分离但不独立的随机变量序列并对其条件分布进行估计。对于第 i 次观测到的 pch_{t_i} ，我们将其分解为 $pch_{t_i} = A_{t_i} D_{t_i} S_{t_i}, i \in \mathbb{N}$ ，实证研究中条件分布和参数设定为：

$$
\begin{aligned}
A_{t_i} \sim Bin(p_i),\ p_i \quad &:= \mathbb{P}\{A_{t_i}=1|A_{t_{i-1}}\}, & p_i &= \frac{e^{\beta_0+\beta_1 A_{t_{i-1}}}}{1+e^{\beta_0+\beta_1 A_{t_{i-1}}}} \\
(\frac{1}{2}(D_{t_i}+1)|A_{t_i}=1) \sim Bin(\delta_i),\ \delta_i \quad &:= \mathbb{P}\{D_{t_i}=1|D_{t_{i-1}},A_{t_i}=1\}, & \delta_i &= \frac{e^{\gamma_0+\gamma_1 D_{t_{i-1}}}}{1+e^{\gamma_0+\gamma_1 D_{t_{i-1}}}} \\
(S_{t_i}|S_{t_{i-1}},D_{t_i}=1,A_{t_i}=1) \quad &\sim g(\lambda_{u,i})+1, & \lambda_{u,i} &= \frac{e^{\theta_{u,0}+\theta_{u,1}S_{t_{i-1}}}}{1+e^{\theta_{u,0}+\theta_{u,1}S_{t_{i-1}}}} \\
(S_{t_i}|S_{t_{i-1}},D_{t_i}=-1,A_{t_i}=1) \quad &\sim g(\lambda_{d,i})+1, & \lambda_{d,i} &= \frac{e^{\theta_{d,0}+\theta_{d,1}S_{t_{i-1}}}}{1+e^{\theta_{d,0}+\theta_{d,1}S_{t_{i-1}}}}
\end{aligned}
\tag{4-3}
$$

对全事件窗口期所有交易日的高频交易价格变化进行建模，拟合参数见表 4-7。

表 4-7 IF1509 在全事件窗口期的 ADS 模型估计参数

	β_0	β_1	γ_0	γ_1	$\theta_{u,0}$	$\theta_{u,1}$	$\theta_{u,0}$	$\theta_{u,1}$
8 月 19 日	1.179	0.231	0.016	-0.429	1.374	-0.298	1.200	-0.253
8 月 20 日	1.157	0.210	-0.022	-0.222	0.904	-0.125	0.753	-0.143
8 月 21 日	1.249	0.187	-0.036	-0.115	0.786	-0.143	0.771	-0.137
8 月 24 日	1.646	0.138	-0.031	0.441	0.974	-0.207	0.972	-0.218
8 月 25 日	1.529	0.196	0.019	0.288	0.964	-0.168	0.889	-0.126
8 月 26 日	1.367	0.182	0.003	-0.038	0.817	-0.106	0.629	-0.104
8 月 27 日	1.514	0.177	0.023	0.120	1.033	-0.130	1.089	-0.171
8 月 28 日	1.374	0.144	-0.027	-0.186	0.623	-0.090	0.634	-0.085
8 月 31 日	1.090	0.317	-0.021	-0.513	0.940	-0.248	0.903	-0.233
9 月 01 日	1.258	0.246	-0.079	-0.366	1.072	-0.234	1.106	-0.251
9 月 02 日	1.249	0.234	0.055	-0.268	0.636	-0.129	0.825	-0.186
9 月 07 日	0.471	0.389	0.000	-0.559	0.892	-0.329	0.874	-0.321
9 月 08 日	0.348	0.460	0.081	-0.630	0.925	-0.344	0.815	-0.306
9 月 09 日	0.437	0.421	0.011	-0.568	1.058	-0.368	1.019	-0.368
9 月 10 日	0.438	0.402	-0.064	-0.508	0.820	-0.278	0.981	-0.337
9 月 11 日	0.521	0.359	0.012	-0.567	0.853	-0.302	0.788	-0.264

9 月 14 日	0.496	0.402	-0.076	-0.550	1.106	-0.417	0.913	-0.313
9 月 15 日	0.539	0.372	-0.055	-0.547	0.841	-0.292	0.885	-0.342
9 月 16 日	0.545	0.349	0.026	-0.583	1.186	-0.407	1.150	-0.415

所有的拟合参数均在 0.1%显著水平下显著。通过以上拟合的参数可以相应计算出每天高频交易数据表现出来的价格变化条件概率，由于采用了价格分解的方法，此模型相较于 OPM 顺序概率值模型能够更好更准确反映出关于价格集束性、价格回转稳定性以及价格变化幅度之间的关系的信息。设定条件概率为：

$$\begin{aligned} p_{0,i} &= \mathbb{P}\{A_{t_i}=0|A_{t_{i-1}}=0\} \\ p_{1,i} &= \mathbb{P}\{A_{t_i}=1|A_{t_{i-1}}=1\} \\ \delta_{k,i} &= \mathbb{P}\{D_{t_i}=1|D_{t_{i-1}}=k, A_{t_{i-1}}=1\} \\ \rho_{k,j,i} &= \mathbb{P}\{S_{t_i}=k|S_{t_{i-1}}=j, D_{t_{i-1}}=1, A_{t_{i-1}}=1\} \\ \varrho_{k,j,i} &= \mathbb{P}\{S_{t_i}=k|S_{t_{i-1}}=j, D_{t_{i-1}}=-1, A_{t_{i-1}}=1\} \end{aligned} \tag{4-4}$$

计算出的全事件窗口期相关的概率见表 4-8 和表 4-9。

表 4-8 IF1509 在全事件窗口期逐日计算的 A_{t_i} 和 D_{t_i} 的相关条件概率

拟合概率	p_0	p_1	δ_1	δ_{-1}	δ_0
8 月 19 日	0.235	0.804	0.398	0.609	0.504
8 月 20 日	0.239	0.797	0.440	0.550	0.495
8 月 21 日	0.223	0.808	0.462	0.520	0.491
8 月 24 日	0.162	0.856	0.601	0.384	0.492
8 月 25 日	0.178	0.849	0.576	0.433	0.505
8 月 26 日	0.203	0.825	0.491	0.510	0.501
8 月 27 日	0.180	0.844	0.536	0.476	0.506
8 月 28 日	0.202	0.820	0.447	0.540	0.493
8 月 31 日	0.252	0.803	0.370	0.621	0.495
9 月 01 日	0.221	0.818	0.391	0.571	0.480
9 月 02 日	0.223	0.815	0.447	0.580	0.514
9 月 07 日	0.384	0.703	0.364	0.636	0.500
9 月 08 日	0.414	0.692	0.366	0.671	0.520
9 月 09 日	0.392	0.702	0.364	0.641	0.503
9 月 10 日	0.392	0.698	0.361	0.609	0.484
9 月 11 日	0.373	0.707	0.365	0.641	0.503

9 月 14 日	0.378	0.711	0.348	0.616	0.481
9 月 15 日	0.368	0.713	0.354	0.620	0.486
9 月 16 日	0.367	0.710	0.364	0.648	0.507

表 4-9 IF1509 在全事件窗口期逐日计算的 S_{t_i} 的相关条件概率

拟合概率	ρ_{31}	ρ_{32}	ρ_{33}	ϱ_{31}	ϱ_{32}	ϱ_{33}
8 月 19 日	0.0670	0.1109	0.1431	0.0798	0.1229	0.1471
8 月 20 日	0.1025	0.1389	0.1470	0.1136	0.1442	0.1438
8 月 21 日	0.1112	0.1433	0.1447	0.1123	0.1437	0.1443
8 月 24 日	0.0971	0.1358	0.1478	0.0973	0.1358	0.1478
8 月 25 日	0.0979	0.1362	0.1477	0.1036	0.1396	0.1468
8 月 26 日	0.1090	0.1423	0.1454	0.1221	0.1469	0.1398
8 月 27 日	0.0926	0.1328	0.1481	0.0883	0.1297	0.1481
8 月 28 日	0.1225	0.1470	0.1396	0.1218	0.1469	0.1400
8 月 31 日	0.0997	0.1373	0.1475	0.1026	0.1390	0.1470
9 月 01 日	0.0896	0.1306	0.1481	0.0870	0.1287	0.1480
9 月 02 日	0.1216	0.1468	0.1401	0.1084	0.1420	0.1456
9 月 07 日	0.1034	0.1394	0.1468	0.1047	0.1401	0.1465
9 月 08 日	0.1009	0.1380	0.1473	0.1091	0.1423	0.1454
9 月 09 日	0.0907	0.1314	0.1481	0.0937	0.1335	0.1481
9 月 10 日	0.1087	0.1421	0.1455	0.0966	0.1354	0.1479
9 月 11 日	0.1063	0.1410	0.1462	0.1111	0.1432	0.1447

9 月 14 日	0.0870	0.1287	0.1480	0.1018	0.1386	0.1471
9 月 15 日	0.1072	0.1414	0.1459	0.1039	0.1397	0.1467
9 月 16 日	0.0808	0.1238	0.1473	0.0837	0.1261	0.1477

观察发现， p_0 在事件 B 后有显著增加，即在前一个观察时价格没有变化的前提下，当前价格继续保持不变的概率明显增加，此现象和之前使用模型得到的结果相符合。同时，p_1 在事件 B 后也有增加，说明之前如果价格发生了变化，则当前价格更有可能发生变化，体现了增强的价格变动集束性。同时，在 δ_1 略为下降的同时， δ_{-1} 明显地逐渐上升，即给定在上一次观测到的价格变化方向，当前发生反方向价格变化的概率增加，反映出了高频交易数据中价格回转现象的增强，说明了增强的市场微观结构噪声以及价格波动的自相关性。对于众多 $\rho_{k,j}$ 和 $\varrho_{k,j}$ 代表的是在之前发生了不同幅度的价格变动时，当前发生较大价格变动的条件概率，这两个估计概率序列在事件 B 前后变化不大。模型仅仅对非零 S_{t_i} 序列进行拟合，还将正负变化分离开，捕捉的仅仅是相邻两次价格变化，模型结果反映出一个较大价格变化的概率和之前价格变化幅度的关系不大。

4.3.4 久期动态特性变化

本节着重研究另一个关键时间序列-价格变化久期序列 $dur_{t_i}, i \in \mathbb{N}$ ，此处的下标定义为观测到价格发生变化的时刻序号。主要使用自回归条件久期模型进行研究，模型主要用于分析交易之间（或价格或价格变动的交易之间）时长的动态特性。在处理久期序列时，考虑到市场微观结构噪声的影响，我们采用设置价格变化阈值的方式来过滤微小的价格变动（0.2 元最小单位，阈值一般设定为 2 个单位），得到的价格变动久期序列能充分反应信息出现的密度，并且发现了显著的日模式和自相关关系。对于日模式，使用项式函数的拟合方法对其进行显著识别并剔除，调整前后的久期序列见图 4-10。在对调整后久期时间序列的研究中，使用 ACF 自相关函数发现了序列中非常强的高阶自相关性，使用带有标准韦布尔分布假设的 WACD 模型和带有广义伽马分布的 GACD 模型在获得非常显著的参数的同时很好拟合了久期中存在的自相关性，进一步通过 ACF 函数分析残差序列的自相关性来确定模型的有效性。在全事件窗口期中，事件 B 发生后久期的自相关关系发生了显著的增强和更强的持久性，同时拟合的 WACD 模型结构有显著变化。

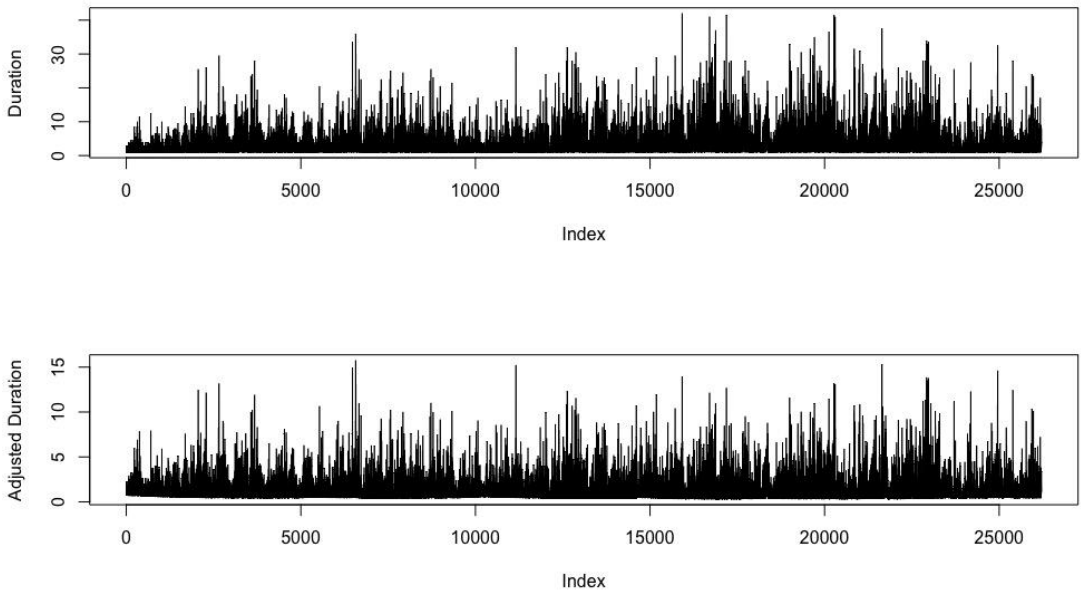

图 4-10 IF1509 在股指期货交易受限事件 B 之后的原始价格变动久期序列和剔除日模式后的价格变动久期序列

对三个时间窗中的数据进行整合后，发现在事件 A 前后序列有强烈的自相关性，阶数约 20 到 30 阶。在事件 B 发生后，自相关性显著增加，且滞后阶数到 40 阶仍有显著自相关。即在事件 B 后一周内的数据呈现出了非常显著且持久的自相关关系。由于此处价格变化久期中的日模式已经被显著识别且消除，自相关关系则不是由于日模式引起，而是由于价格变动久期本身的性质造成。自相关函数见图 4-11。对于股指期货合约高频交易价格变动久期序列，使用指数分布进行拟合效果不佳，使用基于标准韦布尔分布的 $\boldsymbol{WACD(1,2)}$ 和基于广义伽马分布的 $\boldsymbol{GACD(1,2)}$ 模型均能获得十分显著的参数。拟合参数及相关信息见表 4-10 和表 4-11。模型设定如下：

$$\begin{aligned} dur_{t_i} &= \mu_{t_i}\epsilon_{t_i}, \\ \mu_{t_i} &= \mathbb{E}[dur_{t_i}|\mathcal{F}_{i-1}], \\ \mu_{t_i} &= \omega + \alpha_1 dur_{t_{i-1}} + \beta_1\mu_{t_{i-1}} + \beta_2\mu_{t_{i-2}} \end{aligned} \tag{4-5}$$

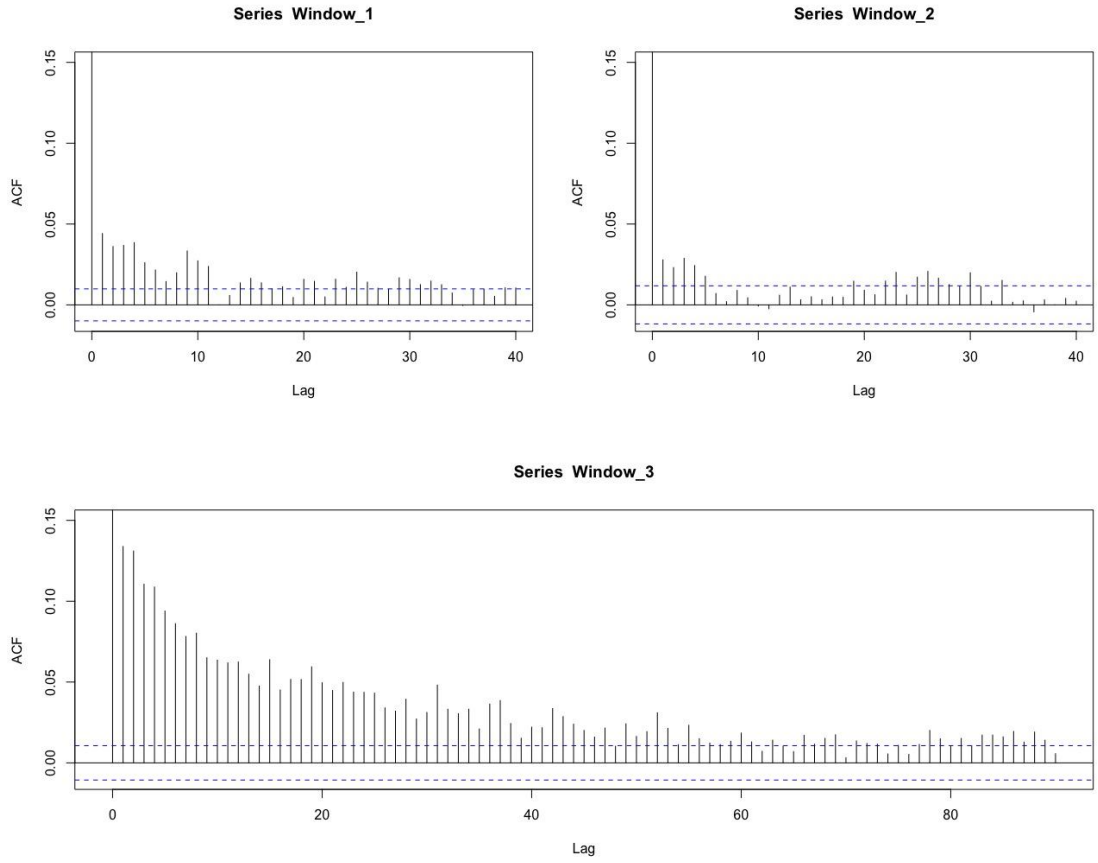

图 4-11 IF1509 在三个子时间窗中的久期序列自相关函数

表 4-10 IF1509 在三个子时间窗中的价格变动久期序列 WACD(1,2)模型估计参数

时间窗	参数	估计值	标准误	t 值	p 值
第一时间窗	ω	0.00240	0.0016	1.49	1.35E-01
	α_1	0.02350	0.0026	8.90	< 2.22E-16
	β_1	0.41558	0.1221	3.40	6.65E-04
	β_2	0.55885	0.1203	4.65	3.39E-06
第二时间窗	ω	0.01422	0.0057	2.51	1.22E-02
	α_1	0.02236	0.0039	5.79	7.07E-09
	β_1	0.39846	0.1847	2.16	3.10E-02

	β_2	0.56609	0.1822	3.11	1.90E-03
第三时间窗	ω	0.06340	0.0050	12.76	< 2.00E-16
	α_1	0.10105	0.0051	19.95	< 2.00E-16
	β_1	0.83126	0.0507	16.41	< 2.00E-16
	β_2	0.02359	0.0460	0.51	6.08E-01

表 4-11 IF1509 在三个子时间窗中的价格变动久期序列 GACD(1,2)模型估计参数

时间窗	参数	估计值	标准误	t 值	p 值
第一时间窗	ω	0.07886	0.0214	3.68	2.32E-04
	α_1	0.02849	0.0043	6.68	2.45E-11
	β_1	0.60472	0.1466	4.12	3.72E-05
	β_2	0.29399	0.1483	1.98	4.75E-02
第二时间窗	ω	0.04001	0.0163	2.46	1.40E-02
	α_1	0.01452	0.0036	4.00	6.40E-05
	β_1	0.42721	0.2351	1.82	6.92E-02
	β_2	0.52096	0.2299	2.27	2.35E-02
第三时间窗	ω	0.10204	0.0072	14.22	2.22E-16
	α_1	0.08371	0.0046	18.27	2.22E-16
	β_1	0.82852	0.0556	14.91	2.22E-16
	β_2	0.00823	0.0499	0.17	8.69E-01

表 4-12 IF1509 在三个子时间窗中基于 ACD 模型拟合参数计算的相关描述值

	参数和	期望值	样本均值
第一时间窗	0.9979	1.1568	1.0934
第二时间窗	0.9869	1.0866	1.0804
第三时间窗	0.9623	1.4347	1.3495

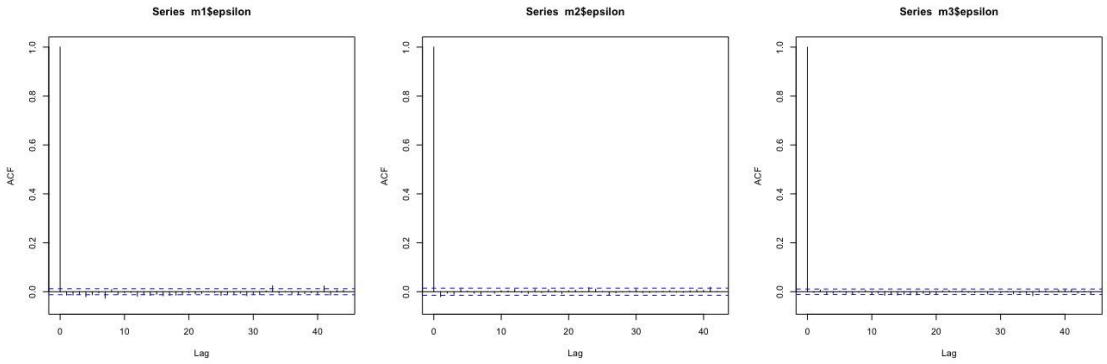

图 4-12 IF1509 在三个子时间窗中 ACD 模型残差自相关函数

进一步检查残差的自相关关系，可以发现模型很好地拟合了价格变动久期的条件期望和滞后项的关系，自相关关系已被消除，见图 4-12。将参数加和得到的值非常接近 1，说明消除日模式后的调整价格变化久期序列中存在特定的持续性，进一步计算模型给出的期望值，和样本均值十分接近，同时其在事件 B 发生后提高了约百分之 30。在模型拟合的各系数中发现了 alpha 系数大大增加，说明了久期序列增强的自相关性和集束性。

综合以上讨论，在识别消除了价格变动久期的显著的日模式并得到基于不同假设分布的有效的 ACD 模型后，发现在股指期货交易受限即事件 B 发生后，价格变动久期的自相关性的强度和阶数都有大幅度增加，两个一阶滞后项的参数也有显著增加，说明了滞后项的值对价格变动久期的条件期望有更大的影响，反映了增强的价格变化的集束性。同时整体久期水平变长，反映了交易活跃程度以及价格变动频率整体下降。实际上在第一第二时间窗中交易价格变动久期应该比计算值小得多，即整体久期平均值在事件 B 发生后有非常大的上升，是因为在计算久期时忽略了仅仅在最小价格变动区间上运行的价格变动。从另一个角度可以反映出，在较大价格变化久期的整体水平上，第三时间窗的均值和第一第二时间窗的均值的关系和交易量的变动是不匹配的，说明了仍有一定比例的价格变动幅度较大，这和之前的分析结果相一致。

4.4 股票现货市场微观结构的变化

本节主要阐述关于股票现货交易市场微观结构的实证分析研究的结果和对结果的解释分析。在股票市场方面，本课题主要关注对应期指合约的指数成分股和所研究的指数成分股性质类似的非成分股在事件前中后三个阶段市场微观结构的变化差别。由于前述的事件研究主要关于在 2015 年 8、9 月股指期货市场中总成交量和持仓量最大的 IF1509 合约展开，其标的为沪深 300 股票指数，故本课题从期现市场的关系入手，首先考虑沪深 300 指数成份股，主要关注金融、能源、房地产、建筑和食品等几个板块。我们认为，IF 合约的异动首先会

对成分股造成影响，但考虑到阿尔法套利者的选股范围不仅限于成分股，我们进一步在流动性好的高市值非成分股中选取标的研究，发现了一部分非成分股也受到和成分股类似的影响。说明了阿尔法套利者的投资选取标的不仅仅限制在成分股范围内。

在对金融行业的研究中，发现了一部分的高市值、高流动性、低换手率的股票，在事件 B 前后呈现出显著的变化。例如，600016 民生银行与 600000 浦发银行，其均为市值较大流动性较好的标的，由于其个股风险暴露低，交易成本低，容易成为阿尔法套利者的投资标的。在算法交易者离场后，这些股票均出现了流动性降低、价格波动增加、价格稳定性降低、局部震荡加剧、价格变动集束性增加等现象。通过建模研究发现其已实现波动率和利用顺序概率值模型计算出来的价格变动条件概率值均有明显变化，同时对价格变动久期序列使用 WACD(1,1)模型进行建模后发现模型结构在事件 B 发生后有显著变化，在绝大多数参数均在 0.01 的显著性水平下显著且序列自相关性被模型充分拟合的情况下，观察到一阶滞后变量的参数有所增加，即一阶滞后的久期值对当前久期的条件期望值的贡献和影响增加。同时，通过模型拟合结果计算的久期期望值也有所增加，和拟合参数一同说明了价格变动集束性的上升以及整体交易活跃度的降低。在对其价格变动序列使用顺序概率值建模时，根据模型拟合参数进行价格变动条件概率的估计图可看出，在事件 B 发生后，股票的交易强度明显降低，价格不变概率整体增加，但大幅度价格变动的概率整体没有降低，同时概率的波动值非常大，即出现许多较大的大幅度价格波动概率，此现象在股指期货市场中已观察到。

表 4-13 600016 民生银行在三个子时间窗中的价格变动久期序列 WACD(1,1)模型估计参数

时间窗	参数	拟合值	标准误	t 值	p 值
第一时间窗	ω	0.21800	0.0533	4.09	4.34E-05
	α	0.08628	0.0142	6.09	1.15E-09
	β	0.81661	0.0327	24.98	2.22E-16
第二时间窗	ω	0.31763	0.1075	2.95	3.13E-03
	α	0.06199	0.0162	3.83	1.28E-04
	β	0.81428	0.0503	16.19	2.22E-16
第三时间窗	ω	1.74868	0.3678	4.75	1.99E-06
	α	0.10929	0.0325	3.36	7.70E-04
	β	0.30112	0.1269	2.37	1.77E-02

表 4-14 600016 民生银行在三个子时间窗中基于 WACD 模型拟合参数计算的相关描述值

时间窗	参数和	期望值	样本均值
第一时间窗	0.9028891	2.24482	2.36136
第二时间窗	0.876265	2.566988	2.776472
第三时间窗	0.4104111	2.965927	3.256702

表 4-15 600000 浦发银行在三个子时间窗中的价格变动久期序列 WACD(1,1)模型估计参数

时间窗	参数	拟合值	标准误	t 值	p 值
第一时间窗	ω	0.24105	0.0692	3.48	4.93E-04
	α	0.06291	0.0155	4.05	5.08E-05
	β	0.84029	0.0347	24.23	2.22E-16
第二时间窗	ω	0.23235	0.0853	2.72	6.46E-03
	α	0.04668	0.0152	3.07	2.13E-03
	β	0.85600	0.0412	20.79	2.22E-16
第三时间窗	ω	0.62335	0.1530	4.07	4.60E-05
	α	0.15278	0.0317	4.82	1.45E-06
	β	0.57149	0.0822	6.95	3.59E-12

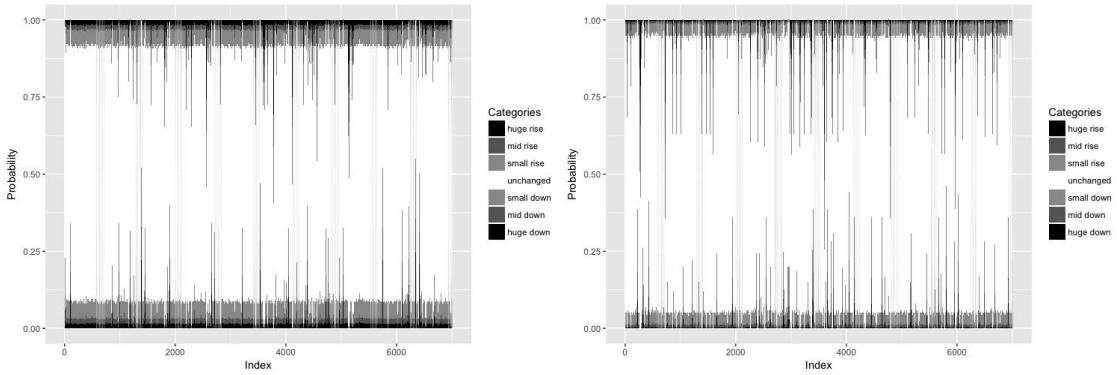

图 4-13 600016 民生银行在股指期货交易受限前后中各价格变动分类值的估计条件概率值

从期现市场的关系看，本课题认为，在股指期货交易受限后，之前市场中投资者的风格和行为模式发生变化，尤其是算法交易者的离场和转型是造成此类现象的原因。一众依赖股指期货通过超额收益投资组合盈利的私募对冲基金和使用量化交易进行投资行为的金融机构的行为模式发生了质的变化，抑或是直接平仓退场，抑或是转型成直接多头的量化择时投资者，不仅仅对期货市场的功能造成了很大的负面影响，还将风险压入了股票现货市场。此外，考虑到此类投资者的投资风格和投资标的特性，尽管他们都使用股指期货来对冲系统性风险，但在采用量化选股算法构建投资组合的时候，依然会在全市场进行投资标的选取。在此类投资者退场后，作为其投资标的的非成分股同样也会受到影响。在对股票市场进行研究的过程中，发现不同行业中的成分股和非成分股比例不同，例如对于金融行业和房地产行业，

大部分的高市值公司均被纳入指数计算范围内，而使用换手率低、交易量小、流通市值低的非成分股进行比较会产生其他因素造成的误差。而在能源行业，成分股和非成分股中都存在整体体量较大且流动性充分的标的，故本课题选取能源行业中和成分股性质相近的非成分股对其高频交易数据进行深入分析，发现了算法交易投资者的退场对其造成的影响，图 4-15 和表 4-16 分别展示了使用 WACD(1,1)模型和基于顺序概率值模型的建模结果计算的价格变动条件概率值，同样可以发现价格不变的条件概率在增加的同时，价格发生较大变化的条件概率实际上仍存在较大的值，说明股票价格有效性降低、局部波动增加和流动性的降低。

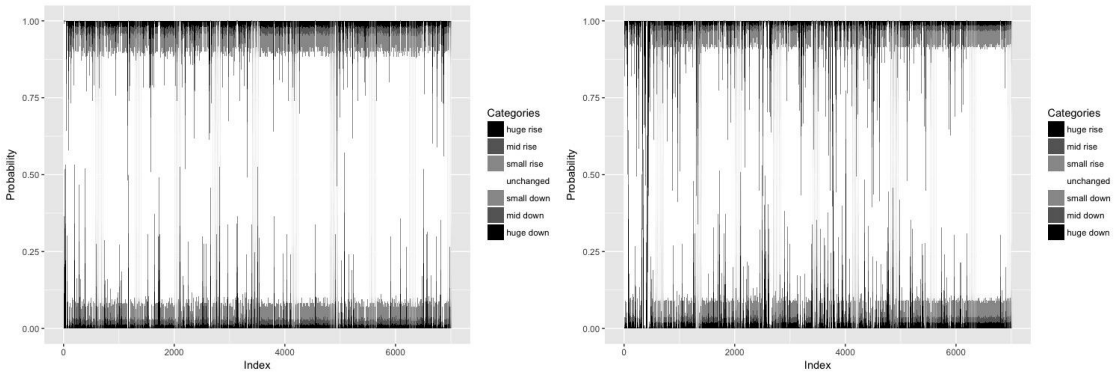

图 4-14 600000 浦发银行在股指期货交易受限前后中各价格变动分类值的估计条件概率值

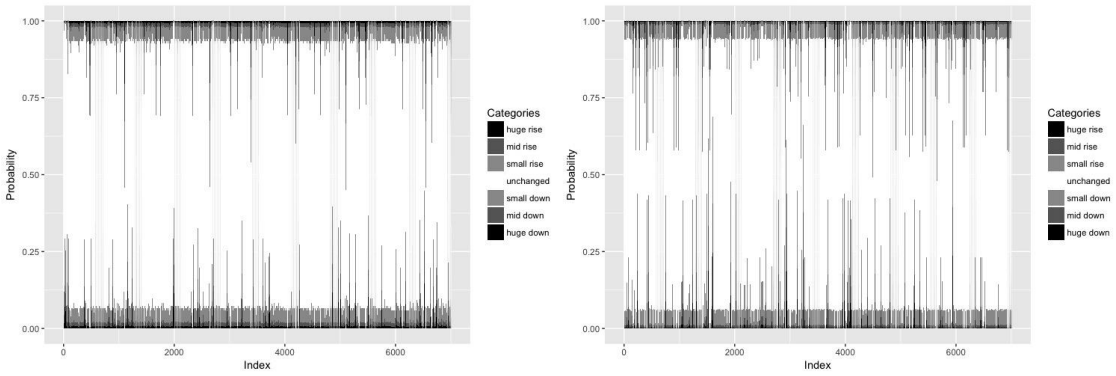

图 4-15 600027 华电国际在股指期货交易受限前后中各价格变动分类值的估计条件概率值

表 4-16 600027 华电国际在三个子时间窗中的价格变动久期序列 WACD(1,1)模型估计参数

时间窗	参数	拟合值	标准误	t 值	p 值
第一时间窗	ω	0.6685	0.3003	2.23	2.60E-02
	α	0.1271	0.0435	2.92	3.45E-03
	β	0.6131	0.1414	4.34	1.45E-05

第二时间窗	ω	2.2800	0.1000	22.81	2.22E-16
	α	0.1978	0.0684	2.89	3.84E-03
	β	0.0000	NA	NA	NA
第三时间窗	ω	0.6371	0.2306	2.76	5.73E-03
	α	0.1236	0.0432	2.86	4.20E-03
	β	0.6633	0.0966	6.87	6.55E-12

4.5 本章小结

本章讨论了实证研究部分，使用了一系列针对高频交易数据的计量模型对期指合约和部分股票进行了建模，并对结果进行了具有实证参考价值的解释。基于真实的交易数据我们发现了在股指期货交易严重受限后，使用算法交易的特定成分投资者的离场给市场微观结构带来的显著变化，从而可以在整个市场的宏观层面帮助投资者和监管部门认识到当时的算法交易者对市场质量和功能和微结构的诸多特性造成的影响以及其在市场运行中扮演的角色。在对投资者组成成分和行为模式的分析中，本课题结合现实情况，认为股指期货交易的限制在当时主要限制了使用统计套利策略的量化交易投资者，包括许多量化私募基金和资产管理公司等，他们的离场对股指期货市场有很明显的负面作用，还对股票现货市场造成了一定的非正常波动和流动型降低。在期指市场的分析中，我们对三大合约的高频交易数据进行了全方位的深入研究，并选取 IF1509 作为例证支持我们的观点。在对股票市场的交易数据分析中，由于涉及的股票数量众多、特征不尽相同、当时的事件窗口期内发生的事件细节也无法详尽获得，并没有发现某一行业受到统一的明显变化，但我们仍然发现在成分股和非成分股中均存在一部分股票受到了事件影响，在算法交易受限后表现出交易萎缩和波动增加等特征。总的来说，实证分析能够有效结合模型的特点和实际数据的特征，在对模型设定进行少许改动的情况下发挥出了模型强大的解释能力，在微观层面细致地分析了统计套利者对市场的贡献和扮演的角色，对投资者和监管部门有很好的参考价值。

第五章 结语

本课题主要研究算法交易者在 2015 年 9 月在股指期货交易受限而离场后，期指和股票等金融市场的市场效率和市场质量受到的影响和所经历的变化。通过对真实高频交易数据

使用特定的金融计量模型进行建模分析后，我们发现了股指期货市场的市场功能受到损害的明显现象，同时观察到了金融市场中整体流动性大幅度降低、市场宽度放大、价格有效性降低、价格波动风险增加、市场微观结构噪声增加、交易成本增加等明显变化。通过实证分析得出的量化表示的结果和对当期市场环境下对投资者组成成分和投资风格的变化分析得到的定性判断相符合。我们认为：算法交易投资者们，尤其是众多关注于在股指期货市场和股票现货市场中实现套利的阿尔法套利者们，在当时的情况下，其在市场中的参与水平处在一个合理的范围内，在给市场的正常交易提供流动性的同时，保障了资产价格能够快速有效反映出新的信息，同时维护了期指市场的稳定运行。他们的频繁交易虽然的确会给市场带来一定的波动性，但这样高频均匀且稳定的波动是有助于提高市场质量和实现价格发现功能的。

我们对当时监管层希望直接通过强制性限制股指期货交易的方法来抑制过度投机从而缓和市场下跌的做法表示怀疑。算法交易者的行为没有给市场带来过度的波动或是下跌风险，市场的整体运行方向和波动不应该归咎于他们，在通过提高交易成本和对交易频率进行严格监管后，实际上作为套保者的阿尔法套利者被视作是投机者被驱逐离场，但股指期货市场和股票现货市场并没有趋于稳定，反而出现了更大的价格变动和更强的价格变动集束性和回转性，说明了市场质量的降低和受损，而不是市场重新获得了稳定和较低的波动率。直接对期货市场进行限制没有很好地真正抑制投机力量，因为投机者主力多在现货市场而不是有更大风险暴露的期货市场。

为了支持以上结论，我们首先对股指期货市场的功能、投资者成分、套利者行为模式以及具体的政策限制进行充分讨论，支持了在事件发生前后主要离场者为算法交易者的前提，并认为对他们的行为变化会给现货市场带来附加的负面影响，即更大的抛压和波动风险。为了证实我们从宏观角度对整个市场性质变化的分析，我们使用了真实的高频交易数据，并采用了针对高频交易数据的金融计量模型进行建模分析，从市场宽度、市场价格的有效性、即时性和稳定性、交易成本、微结构噪声和已实现波动率等方面入手去描述微结构的诸多特性，发现了和预想一致的现象。

具体而言，在对市场微观结构进行基本描述时，我们首先发现了股指期货市场在算法交易受限后成交量和成交额急剧萎缩，价格变化的区间加大，较大的价格比例在价格不变比例明显增大时仍保持不变，价格变动和买卖价差的样本方差在算法交易受限后先大幅度增加然后在较高水平保持，这些现象都能直接说明市场活跃度在急剧下降后并没有实现真正的稳定，虽然整体上看价格波动减少，但大幅度的局部震荡却变得更猛烈和频繁，价格变动变得难以预测，价格对市场新信息的敏感度也有所下降。紧接着，使用顺序概率值模型时，在捕捉到了滞后项和隐含价格变动的显著关系后进一步从价格变动条件概率的分布中发现了市场波动风险增加的证据，即事件发生后价格变动的集束性和反转性都加剧了，说明了更强的价格震荡。在使用价格变化分解模型和已实现波动率的计算对高频价格序列进行分析发现，市场微观结构噪声在算法交易者离场后增加，而整体的市场波动率没有显著降低，说明了市场价格发现功能的受损，市场无法继续及时有效稳定地将新信息反映在价格上，导致了局部价格震荡加剧。同时，将市场整体剧烈波动归咎于量化交易者是不妥的。量化交易者的高频交易给市场带来了稳定的微观波动，有助于价格平稳运行到有效位置，并将价格异动风险控制在一定范围内。最后，我们使用自回归价格久期模型对价格变动久期序列增强的自相关关系进行了分析，模型结果显示出在时间跨度上增强的局部价格变化集束性，即在价格变动久期平均值增加的同时，一阶滞后系数也增加，说明了在长期价格波动极本维持和成交量明显缩减的情况下加剧的局部短期震荡。

基于建模结果和深入的分析，我们认为，股指期货市场和现货市场的桥梁为使用算法交易和量化交易技术的阿尔法套利者，他们的离场和投资风格的转变在使得股指期货市场基本功能受到致命性打击的同时，也促进了投机资金在不同市场间进行转移，将风险引入股票现货市场或商品期货等市场，在一定的时间内造成了价格的过度波动以及流动性的缺失，给金融市场甚至实体经济的平稳运行带来进一步的风险和下行压力。研究中我们结合了宏观上大事件的契机和微观结构的细致研究，对算法交易者在市场中的地位进行了细致的剖析。宏观事件首先提供了良好的研究环境和宝贵的真实第一手数据，较大力度的政策限制使得在很短的时间内算法交易者迅速被迫离场，使得我们能够很好地在控制变量的同时对真实高频数据进行研究，通过分析能够一定程度上代表市场整体的、交易量和交易额都非常巨大的股指期货主力合约，从微观层面准确捕捉算法交易者对整个市场的影响。相信本课题的研究结论对监管部门在未来对股指期货市场进行合理监管有着很好的参考价值，在对投资者行为模式以及它们的交易对市场的影响有了深入认识后，对金融市场的监管才能真正达到促进市场稳定发展的长远目的。

参考文献

[1] Engle R F, and Russell J R. Forecasting the Frequency of Changes in Quoted Foreign Exchange Prices with the Autoregressive Conditional Duration Model[J]. Journal of Empirical Finance. 1997(4):187-212.
[2] Engle R F. Autoregressive conditional duration: A new model for irregular spaced transaction data[J]. Econometrica. 1998,66(5):1127-1162.
[3] Ghysels E J. GARCH for irregularly spaced financial data: the ACD-GARCH model[J]. Studies in Nonlinear Dynamics & Econometrics, 1998,2(4):133—149.
[4] Engle R F. The econometrics of ultra-high frequency data[J]. Econometric, 2000, 68(4): 1-22
[5] Zhang M Y, Jeffrey R, Russell, and Ruey S. Tsay. A nonlinear autoregressive conditional duration model with application to financial transaction data[J]. Journal of Econometrics. 2001(104): 179-207.
[6] Gramming J W M. Modeling of ultra-high frequency data[J]. Econometric. 2000, 68(4): 1-22.
[7] Ruey S. Tsay. Analysis of Financial Time Series, Third Edition (Wiley Series in Probability and Statistics). John Wiley & Sons. 2010.
[8] Hauseman and MacKinlay. An ordered probit analysis of transaction stock prices. Journal of Financial Economics. 1992(31): 319-379.
[9] Rydberg T H and Shephard N. Dynamics of trade-by-trade price movements: Decomposition and models. Journal of Financial Econometrics. 2003(1): 2-25.
[10] McCulloch, R. E. and Tsay, R. S.. Nonlinearity in high frequency data and hierarchical models. Studies in Nonlinear Dynamics and Econometrics. 2000(5): 1-17.
[11] 陈敏，王国明. 中国证券市场的 ACD－GARCH 模型及其研究[J]. 统计研究，2003(11): 60-62。
[12] 蒋学雷、陈敏、王国明. 股票市场的流动性度量的动态 ACD 模型[J]. 统计研究，2004,119(2): 381-412
[13] 郭兴义、杜本峰、何龙灿. 超高频数据分析与建模[J]. 统计研究，2002(11)
[14] 刘向丽、成思危. 基于 ACD 模型的中国期货市场波动性[J]. 系统工程理论与实践，2012(2)。
[15] 刘向丽、程刚、成思危等. 中国期货市场日内效应分析[J]. 系统工程理论与实践，2008, 28(8): 63-80。
[16] 王亚楠、张燕、吴祈宗等. 基于 ACD 模型的成交量建模研究[J]. 数学的实践与认识，2013(11)。
[17] 王桂堂、闫盼盼. 金融市场中的高频交易与监管[J]. 金融教学与研究，2013(5).
[18] 李风雨. 高频交易对证券市场的影响及监管对策[J]. 上海金融，2012(9): 48-52.
[19] 周小川. 金融市场交易的频率特性[J]. 经济导刊，2010(4): 4-10

附录

本项目 R 与 Python 脚本部分核心代码实现。

preprocessors.R

```
##### Stocks #####
### Main Preprocessors for Stocks
#' Get the window grand datalist for Stocks
#' @description Windows settings are the same as which of `FMM:get_window`,
#'   requirements of models: $price & $tsec required by RV analysis; $ctgs &
#'   $pch & $vol required by OPM; $ADS required by ADS models; $pch & $tsec
#'   required by ACD models
#' @param product
#' @param month1 the previous month, format "201508"
#' @param month2 the following month, format "201509"
#' @return Store the window data in `./data` folder
get_stock_window <- function(product = "600016", month1="201508", month2="201509
") {
  days1 <- c('19', '20', '21', '24', '25', '26', '27', '28', '31')
  days2 <- c('01', '02', '07', '08', '09', '10', '11', '14')
  datalist <- list()
  for (i in 1:length(days1)) {
    file_path <- get_extdata_path(product, month1, days1[i])
    df <- read.table(file_path, header=F, sep=' ')
    df <- df[,1:5]
    df$date <- paste(month1, days1[i], sep = '')
    names(df) <- c("tsec", "price", "pch", "vol", "dvol", "date")
    datalist[[length(datalist)+1]] <- df
  } # august
  for (i in 1:length(days2)) {
    file_path <- get_extdata_path(product, month2, days2[i])
    df <- read.table(file_path, header=F, sep=' ')
    # i.e. './inst/extdata/600016_201508/600016_20150819.txt'
    df <- df[,1:5]
    df$date <- paste(month2, days2[i], sep = '')
    names(df) <- c("tsec", "price", "pch", "vol", "dvol", "date")
    datalist[[length(datalist)+1]] <- df
```

```
  } # september
  for (i in 1:length(datalist)) {
    df <- datalist[[i]]
    df <- add_pch(df)
    df <- adj_lunch(df)
    row.names(df) <- 1:nrow(df)
    # plot(df$tsec,type='l') # check the time indexes
    df$ctgs <- as.integer(df$pch/0.01) + 4
    df$ctgs[df$ctgs < 1] <- 1
    df$ctgs[df$ctgs > 7] <- 7
    df$ctgs <- as.factor(df$ctgs)
    df <- add_ADS(df)
    # assign(paste(code,i,sep='_'), df)
    #   no need for assigning the names after assembling dfs into a list
    datalist[[i]] <- df
  } # reformatting the grand window datalist
  target_name <- paste("./data/",
                       paste(product, "win", sep = "_"),
                       ".RData", sep = "")
  save(datalist, file = target_name)
  # store the stock datalist in `./data` folder, ready for the models
}

##### SIF #####
### Main Prepocessors for Stock Index Futures
#' Gets the window grand datalist for Index Futures, after FMM::prebatch
#' @description Windows settings: previous: 08 - 19 20 21 24 25 26; middle: 08 -
#'   27 28 31 ; 09 - 1 2; after: 09 - 7 8 9 10 11 14. The function first runs
#'   `FMM::prebatch` to get the raw `.RData` files in `./data-raw` and trim
#'   them, then assembles data of corresponding dates and makes a grand datalist
#'   of the target window.
#' @param product
#' @param month1 the previous month, format "201508"
#' @param month2 the following month, format "201509"
#' @return Nothing but put the window data in `./data` folder
get_window <- function(product="IF1509", month1="201508", month2="201509") {
  source_name1 <- paste("./data/",
                        paste(product, month1, sep = "_"),
                        ".RData", sep = "")
  source_name2 <- paste("./data/",
                        paste(product, month2, sep = "_"),
                        ".RData", sep = "")
  target_name <- paste("./data/",
                       paste(product, "win", sep = "_"),
```

```
                    ".RData", sep = "")
  # Define the names
  prebatch(product, month1)
  prebatch(product, month2)
  load(source_name1); I0908 <- datalist
  load(source_name2); I0909 <- datalist
  # Get the trimmed data (stored in `./data/`) in the environment awaiting
  win09 <- list(I0908[[6]],I0908[[7]],I0908[[8]],I0908[[9]],I0908[[10]],
                I0908[[11]],I0908[[12]],I0908[[13]],I0908[[14]],I0909[[1]],I0909
[[2]],
                I0909[[3]],I0909[[4]],I0909[[5]],I0909[[6]],I0909[[7]],I0909[[8]],
I0909[[9]],I0909[[10]])
  datalist <- win09
  save(datalist, file = target_name)
  # save the new window data in the `./data`
}
#' Gets the monthly trimmed datalist from `./txt` in `./inst/extdata` to
#' `./RData` in `./data`
#' @description This is the MOTHER FUNCTIONS of trimming raw .txt files and
#'   getting them in the `./data` folder. In the first part, we get the UTF-8
#'   encoded .txt files into `./data-raw/` and store them as `.RData` files with
#'   raw grand datalists, then we trim the datalists in `./data-raw/` and get
#'   them in `./data/`, after that, the data is ready for the models. The
#'   function `FMM::prebatch` could also be called by `FMM::get_window` inorder
#'   to first get the raw data trimmed and form the grand datalists for the
#'   event windows.
prebatch <- function(product, month) {
  # Default wd: .../FMM
  # test AFTER clearing the ./data and ./data-raw folders
  # Change the name here
  fetch_data(product, month) # ONLY RUN ONCE to get the raw .RData saved in ./dat
a-raw
  ### What follows is the trimming process for files in ./data-raw
  source_name <- paste("./data-raw/",
                       paste(product, month, sep = "_"),
                       ".RData", sep = '')
  load(source_name) # get the datalist
  datalist <- reformat_list(datalist)   # $time, $bids, $asks required
  datalist <- trim_hnt_list(datalist)   # $tsec
  datalist <- adj_lunch_list(datalist)  # $tsec
  datalist <- reorder_list(datalist)
  datalist <- add_pch_list(datalist)  # $price + $pch
  datalist <- add_pchctgs_list(datalist)  # $pch + $ctgs
  datalist <- add_ADS_list(datalist)  # $pch $ctgs + $A $D $S
```

```
  # datalist <- add_duration_list(datalist)
  target_name <- paste("./data/",
                       paste(product, month, sep = "_"),
                       ".RData", sep = '')
  save(datalist, file = target_name) # ./data
  str(datalist[[1]])
  View(datalist[[1]])
  # Check the outputs
  # NAMESPACE export all: exportPattern("^[[:alpha:]]+")
  return(datalist)
}
#' Fetches data from external folder before data cleaning
#' @description Fetch raw \code{.txt} data stored in \code{./inst/extdata} and
#'   store \code{.RData} file in \code{./data-raw}
#' @param product the name of the target product
#' @param month the month of the data
#' @return the list consists of dataframes for data in every trading day
#' @examples
#' datalist <- fetch_data("IC1509", "201509")
#' datalist[[1]]
fetch_data <- function(product, month) {
  folder <- paste('./inst/extdata/',
                  paste(product, month, sep = "_"),
                  sep = '')
  file_names <- dir(folder)
  datalist   <- list()
  for (i in 1:length(file_names)) {
    file_path <- paste(folder, file_names[i], sep = '/')
    raw_input <- read.table(file_path, header = TRUE, sep = ',')
    name      <- strsplit(file_names[i], ".txt")[[1]][1]
    datalist[[length(datalist)+1]] <- data.frame(raw_input[,c(2,3,4,5,7,8,13:2
2)])
    assign(name, datalist[[length(datalist)]])
  }
  save(datalist,
       file = paste("./data-raw/",
                    paste(product, month, sep = "_"),
                    ".RData", sep = ''))
  return(datalist)
}
#' Reformats the dataframe
#' @description Reformats the raw dataframes read from \code{.txt} files. The
#'   reformatting process includes renaming, splitting the \code{$time} column
#'   into formatted \code{$date}, \code{$hour}, \code{$min} and \code{$sec},
```

```
#'   generating raw \code{$tsec} and \code{$spread}, which are representing
#'   "total seconds" and "bid-ask spread". Then returns the reordered dataframe.
#' @param df the dataframe to be reformatted.
#' @return  the reformatted dataframe, the \code{$bid} and \code{ask} columns
#'   are removed.
#' @example
#' df <- reformat(datalist[[1]])
reformat <- function(df) {
  colnames(df) <- c("code", "time", "price", "position",
                    "dvol", "vol",
                    "bid1", "bid2", "bid3", "bid4", "bid5",
                    "ask1", "ask2", "ask3", "ask4", "ask5")
  df$time <- as.character(df$time)
  df$date <- as.Date(substring(df$time, 1, 10), "%Y-%m-%d")
  df$hour <- as.numeric(substring(df$time, 12, 13))
  df$min <- as.numeric(substring(df$time, 15, 16))
  df$sec <- as.numeric(substring(df$time, 18, 21))
  df$tsec <- df$hour*3600 + df$min*60 + df$sec
  bid_df <- data.frame(df$bid1, df$bid2, df$bid3, df$bid4, df$bid5)
  ask_df <- data.frame(df$ask1, df$ask2, df$ask3, df$ask4, df$ask5)
  hi_bid <- get_m_series(bid_df, max) # in case of first bid-ask data missing
  lo_ask <- get_m_series(ask_df, min)
  hi_bid[hi_bid==0] <- lo_ask[hi_bid==0]
  lo_ask[lo_ask==0] <- hi_bid[lo_ask==0]
  df$spread <- lo_ask - hi_bid
  df <- df[, c(17:21,3:6,22)] # remove code, time, bid, ask
  return(df)
}

##### Assistants #####
### Auxiliary functions for main prepocessors
### if any functions seem to vague in preps. pls check here
#' Gets the path of a particular raw data file in `./inst/extdata/` folder
get_extdata_path <- function(product, month, day) {
  folder_name <- paste(product, month, sep="_") # 600016_201508
  file_name   <- paste(paste(folder_name, day, sep=""), ".txt", sep="")
  folder_path <- paste('./inst/extdata', folder_name, sep='/')
  file_path <- paste(folder_path, file_name, sep='/')
  # ./inst/extdata/600016_201508/600016_20150819.txt
  return(file_path)
}
#' Gets Max or Min series
#' @description Returns a series consists of max(min) of each row of the
#'   dataframe.
```

```
#' @param df the dataframe base on which the series will be calculated.
#' @param func the function for choosing the element in a single row of the
#'   dataframe.
#' @return the series
get_m_series <- function(df, func) {
  series <- c(1:nrow(df))
  for (i in 1:nrow(df)) {
    series[i] <- func(df[i,])
  }
  return(series)
}
#' Transfers time string to total seconds
#' @description Transfers the time string to total seconds starting midnight.
#'   Will be used for supporting \code{FMM::trim_hnt()} and
#'   \code{FMM::trim.lunch()}.
#' @param char_time time string, i.e. "15:00".
#' @return time in total seconds starting midnight.
#' @example
#' timestmp <- time_in_sec("15:00") # 54000
time_in_sec <- function(char_time) {
  hour_in_sec <- as.numeric(strsplit(char_time, ":")[[1]][1]) * 3600
  min_in_sec  <- as.numeric(strsplit(char_time, ":")[[1]][2]) * 60
  return (hour_in_sec + min_in_sec)
}

##### Trim Timespans #####
#' Trims the heads and tails
#' @description Trims the heads and tails of a dataframe. Add no column. This
#'   function should not be called after \code{FMM::adj_lunch()}.
#' @param df the dataframe to be trimmed, with column \code{$tsec}.
#' @return the trimmed dataframe without info before 9:15 and after 15:00.
trim_hnt <- function(df) {
  start_stmp  <- time_in_sec("9:15") + 3 # get rid of the starting abnomalies
  end_stmp    <- time_in_sec("15:00")
  return(df[(df$tsec > start_stmp) & (df$tsec <= end_stmp),])
}
#' Adjusts the time in seconds according to the lunch break
#' @description Adjust the \code{$tsec} according to the lunch break. This
#'   function should be called after \code{FMM::trim_hnt}
#' @param df the dataframe to be trimmed, with column \code{$tsec} after \code{F
MM::trim_hnt}.
#' @return the trimmed dataframe with adjusted \code{$tsec}.
adj_lunch <- function(df) {
  # required: $tsec after trim_hnt
```

```
  lunch_stmp <- time_in_sec("11:30") + 10 # get rid of the lagged abnomal trades
  lunch_time <- time_in_sec("1:30")
  # get the index for data requiring adjustment
  tmpindex <- c(1:dim(df)[1])[df$tsec>lunch_stmp]
  adj_lunch <- df$tsec[tmpindex]-lunch_time
  # catenate the adjusted lower part and the original upperpart
  df$tsec <- c(df$tsec[-tmpindex],adj_lunch)
  # rearrange the dataframe's columns
  # df <- df[,c(1,2,3,4,5,13,6,7,8,9,10,11,12)]
  return(df)
}

##### Add Columns #####
#' Adds Price Changes
#' @description Generates the differentiated seiries of the price for futher
#'   uses.
#' @param df the dataframe with trimmed price (after \code{FMM::trim_hnt()} and
#'   \code{FMM::adjust_lunch()}), the dataframe should be from the \code{./data}
#'   folder.
#' @return the dataframe with new column \code{$pch}.
add_pch <- function(df) {
  pch <- numeric(length(df$price))
  pch[2:length(pch)] <- diff(df$price)
  df$pch <- pch
  return(df)
}
#' Adds Price Changes' Categories
#' @description Generates the categories for price changes to support the
#'   ordered probit model.
#' @param df the dataframe with \code{$pch}.
#' @param num the number of categories, default 7.
#' @param cut the percentile of price changes which should be ignored as
#'   abnormal values, default 1%, the function will cut the head 1% and tail 1%
#'   before generating the ranges for categorizing. Note that the OLD VERSION of
#'   categorizing was based on \code{rg <- range(df$pch); ctgs <- seq(rg[1],
#'   rg[2], (rg[2]-rg[1])/7)}.
#' @return the dataframe with \code{$ctgs}, factors
add_pchctgs <- function(df, num = 7, cut = 0.01) {
  qt <- quantile(df$pch, probs = seq(0, 1, cut))
  start <- qt[2]; end <- qt[length(qt)-1]
  ctgs <- seq(start, end, (end-start)/7)
  df$ctgs <- ifelse(df$pch <= ctgs[2], 1,
                    ifelse(df$pch <= ctgs[3], 2,
                           ifelse(df$pch <= ctgs[4], 3,
```

```
                                ifelse(df$pch <= ctgs[5], 4,
                                       ifelse(df$pch <= ctgs[6], 5,
                                              ifelse(df$pch <= ctgs[7], 6, 7))))))
  df$ctgs <- as.factor(df$ctgs)
  return(df)
}
#' Adds ADS Columns for ADS Model
#' @description Adds ADS columns to the original dataframe to support the ADS mo
del.
#'   A: if there is a price change
#'   D: the direction of the change if there has any
#'   S: the amplitude of the price change described by categories' absolute valu
e
#' @param df the dataframe with \code{$pch} and \code{$ctgs}.
#' @return the dataframe with \code{$A}, \code{$D} and \code{$S}
add_ADS <-function(df) {
  df$A <- ifelse(df$pch == 0, 0, 1)
  df$D <- ifelse(df$pch > 0, 1, ifelse(df$pch < 0, -1, 0))
  df$S <- abs(as.numeric(df$ctgs)-4)
  return(df)
}
#' Adds Duration for Duration Models
#' @description Adds durations to support the duration models.
#' @param df the original dataframe with adjusted \code{$tsec}.
#' @return the dataframe with \code{$duration}
add_duration <-
  function(df) {
    datasize <- dim(df)[1]
    duration <- numeric(datasize)
    duration[3:datasize] <- diff(df$tsec[-1])
    df$duration <- duration
    return(df)
  }
```

models.R

```
##### Var Preparations #####
#' Preparation for ADS Models
#' @description Returns the grand list VARS consists of several dataframe of
#'   variables supporting the ADS model, \code{$A}, \code{$D}, \code{$S}
#' @param df the dataframe from \code{datalist} in \code{./data} folder
#' @return the grand list VARS
prepare_ads <- function(df) {
  # ADS have been added by add_ADS()
  size <- nrow(df)
  A <- df$A; D <- df$D; S <- df$S
  vars <- data.frame(1:(size-1)) # df for raw vars
  vars$A1 <- A[2:size]
  vars$A2 <- A[1:size-1]
  vars$D1 <- D[2:size]
  vars$D2 <- D[1:size-1]
  vars$S1 <- S[2:size]
  vars$S2 <- S[1:size-1]
  d1 <- vars$D1[vars$A1==1]
  d2 <- vars$D2[vars$A1==1]
  d1 <- (d1+abs(d1))/2 # transfer d1 to binary
  d <- data.frame(d1, d2)
  s1 <- vars$S1[vars$D1==1]
  s2 <- vars$S2[vars$D1==1]
  ss1 <- s1[s1!=0]
  ss2 <- s2[s1!=0] # eliminate the zeros of s1
  # the process above means that we only care about big pches
  s <- data.frame(ss1, ss2)
  ns1 <- vars$S1[vars$D1==-1]
  ns2 <- vars$S2[vars$D1==-1]
  nss1 <- ns1[ns1!=0]
  nss2 <- ns2[ns1!=0]
  ns <- data.frame(nss1, nss2)
  # s and ns cannot be passed into FMM::geo_size, since during the
  #   previous categorizing process, we regard some small changes
  #   as "unchanged", so when we use the condition `D1==1`, there
  #   are still some s are 0, which is not permitted in the geo_size
  #   function. 2 ways to solve this, 1st, use glm to fit the model,
  #   2nd, generate new s and ns arrays.
  VARS <- list(vars=vars, d=d, s=s, ns=ns)
  return(VARS)
}
```

```
#' Preparation for Ordered Probit Model
#' @description Returns the dataframe consists of stand-by variables,
#'   \code{$ctgs}, \code{$pch}, \code{$vol} and \code{$dvol} are required
#' @param df the dataframe from \code{datalist} in \code{./data} folder
#' @return the dataframe of variables for OPM
prepare_opm <- function(df) {
  size <- nrow(df)
  opm_var <- data.frame(1:(size-3))
  opm_var$ctgs <- df$ctgs[4:size]
  opm_var$ctgs1 <- df$ctgs[3:(size-1)]
  opm_var$ctgs2 <- df$ctgs[2:(size-2)]
  opm_var$ctgs3 <- df$ctgs[1:(size-3)]
  opm_var$cp1 <- df$pch[3:(size-1)]/0.2
  opm_var$cp2 <- df$pch[2:(size-2)]/0.2
  opm_var$cp3 <- df$pch[1:(size-3)]/0.2
  # lag 1 volume, lag 2 volume
  opm_var$dvol1 <- df$dvol[3:(size-1)] # lag 1 dollar volume
  opm_var$dvol2 <- df$dvol[2:(size-2)] # lag 2 dollar volume
  opm_var$vol1 <- df$vol[3:(size-1)] # lag 1 dollar volume
  opm_var$vol2 <- df$vol[2:(size-2)] # lag 2 dollar volume
  opm_var$spread1 <- df$spread[3:(size-1)] # lag 1 spread
  opm_var$spread2 <- df$spread[2:(size-2)] # lag 1 spread
  return(opm_var[-1])
}
#' Preparation for Duration Models
#' @description Returns the durations
#' @param df the dataframe from \code{datalist} in \code{./data} folder
#' @return the series of durations
prepare_duration <- function(df, threshold = 1) {
  size <- nrow(df)
  df$bpch <- ifelse(abs(df$pch) > threshold * 0.2, df$pch, 0)
  # eliminate the small changes
  pchsec <- df$tsec[df$bpch!=0]
  duration <- diff(pchsec)
  duration <- ifelse(duration < 20 * mean(duration), duration,
                     c(0, duration[1:(length(duration)-1)]))
  duration <- duration[duration > 0.5]
  # smooth out the abnomalities (very high values and small ones: 0 & 0.5)
  return(duration)
}
#' Preparation for Duration Models (with Day Patterns' Elimination)
#' @description Returns the durations with time index to support the DPE
#' @param df the dataframe from \code{datalist} in \code{./data} folder
#' @return the df with the serie of durations and the original $tsec
```

```
prepare_adj_duration <- function(df, threshold = 1, unit = 0.2) {
  size <- nrow(df)
  df$bpch <- ifelse(abs(df$pch) > threshold * unit, df$pch, 0)
  pchsec <- df$tsec[df$bpch!=0]
  duration <- diff(pchsec)
  duration <- ifelse(duration < 20 * mean(duration), duration,
                     c(0, duration[1:(length(duration)-1)]))
  pchsec <- pchsec[1:length(pchsec)-1] - 33300
  pchsec <- pchsec[duration > 0.5]
  duration <- duration[duration > 0.5]
  adj_vars <- cbind(pchsec, pchsec^2)
  m <- lm(log(duration)~adj_vars)
  fit <- m$fitted.values
  adj_duration <-duration/exp(fit)
  return(adj_duration)
}

##### Realized Volatility #####
#' Generates the dataframe of Realized Volatility
#' @description Returns the RV df from the grand datalist imported from
#'   \code{./data}.
#' @param list the datalist, consists of data, each elements is the product's
#'   data of a trading day
#' @param sec_scales the vector of scales in seconds
#' @param min_scales the vector of scales in minutes
#' @param N the number of groups required
get_rv_df <- function(list, sec_scales, min_scales, N) {
  scales <- c(0, sec_scales, min_scales)
  rv_df <- data.frame(rep(0,length(scales)))
  for (i in 1:N) {
    df <- list[[i]]
    rvs <- get_rvs(df, sec_scales, min_scales) # get a RV vector for the df
    rv_df <- cbind(rv_df, rvs)
  }
  names <- "interval"
  for (i in 1:N)
    names <- append(names, as.character(list[[i]][1,1]))
  colnames(rv_df) <- names
  rv_df$interval <- scales
  return(rv_df)
}
#' Generates the Realized Volatility Vector
#' @description Returns the RV vector of a dataframe (one day data of a
#'   product). The vector contains RVs calculated in different scales
```

```
#' @param df the original dataframe
#' @param sec_scales the scale vector in seconds
#' @param min_scales the scale vector in minutes
#' @return the RV vector
get_rvs <- function(df, sec_scales, min_scales) {
  rvs <- as.vector(get_rv(df$price))
  for (j in sec_scales)
    rvs <- append(rvs, get_rv(reintv_sec(df, j, min)))
  for (k in min_scales)
    rvs <- append(rvs, get_rv(reintv_min(df, k, min)))
  return(rvs)
}
#' Calculates the Realized Volatility
#' @description Calculates the realized volatility based on the price series.
#'   Calculates trade by trade volatility when the \code{df$price} is passed in
#'   directly. (0.5 sec intv)
#' @param prices price series
#' @return the annual volatility of the log return
get_rv <- function(prices) {
  # to get the log return of price series
  rtn <- diff(log(prices))
  return(sqrt(sum(rtn ^ 2) * 252))
}

##### Resetting Intervals #####
#' Generates the price vector by a prescribed scale in seconds
reintv_sec <- function(df, intv = 10, func = max, lag = 0) {
  start_stmp <- time_in_sec("9:15")
  end_stmp <- time_in_sec("15:00")
  lunch_time <- time_in_sec("1:30")
  time_index <- seq(start_stmp, end_stmp - lunch_time, intv)
  # coresponding to tsec(adjusted)
  intv_num <- (end_stmp - start_stmp - lunch_time)/intv
  # total number of intervals
  price <- df$price; adjtsec <- df$tsec
  prices <- NULL # initialize the price vector
  idx <- 1
  while (idx <= intv_num) { # check every interval
    boolidx  <- ((adjtsec >= time_index[idx]) & (adjtsec < time_index[idx+1]))
    if (all(!boolidx)) { # all false - no price matched in this interval
      prices <- append(prices, prices[length(prices)])
    } else {
      price_vector <- price[boolidx]
      if (!lag) { # if no lag was passed in, we use the customized function
```

```
        intv_price <- func(price_vector)
      } else { # if we set a lag
        intv_price <- price_vector[lag] # lag should not be greater than length(p.
v)
      }
      prices <- append(prices, intv_price)
    }
    idx <- idx + 1
  }
  return(prices)
}
#' Generates the price vector by a prescribed scale in minutes
reintv_min <- function(df, intv = 1, func = max) {
  intv <- intv * 60
  return(reintv_sec(df, intv, func))
}
```

plotting.R

```
#' Fitted Probabilities Plotting
#' @description Generates the graph for fitted probabilities
#' @param fp_df RAW Fitted Probability Data Frame
plot_fp <- function(fp) {
  zero_index <- floor(dim(fp)[2]/2) + 1 # for those cases with 5 categories
  unchanged <- fp[,zero_index]
  decline <- apply(fp[,1:zero_index-1], 1, sum)
  rise <- apply(fp[,(zero_index+1):dim(fp)[2]], 1, sum)
  fp <- data.frame(rise = rise,
                   unchanged = unchanged,
                   decline = decline,
                   key = 1:nrow(fp))
  fp_m <- reshape2::melt(fp, id = 4)
  names(fp_m) <- c("Index", "Categories", "Fitted Probability")
  f <- ggplot2::ggplot(fp_m, ggplot2::aes(Index, probabilities))
  f <- f + ggplot2::geom_col(ggplot2::aes(fill = Categories), position = "fill")
  f <- f + ggplot2::scale_fill_manual(values=c("#333333", "#FFFFFF", "#333333"))
  # Color combinations: c("#CC3366", "#FFFFFF", "#339999"); c("#66FF99", "#FFFFFF",
"#FF3366")
  return(f)
}
plot_fp_all <- function(fp) {
```

```
  fp <- data.frame(fp, key = 1:nrow(fp))
  names(fp) <- c('huge down', 'mid down', 'small down', 'unchanged', 'small rise',
'mid rise', 'huge rise', 'key')
  fp <- fp[c('huge rise','mid rise','small rise','unchanged','small down','mid
down','huge down', 'key')]
  fp_m <- reshape2::melt(fp, id = 8)
  names(fp_m) <- c("Index", "Categories", "Probability")
  f <- ggplot2::ggplot(fp_m, ggplot2::aes(Index, Probability))
  f <- f + ggplot2::geom_col(ggplot2::aes(fill = Categories), position = "fill")
  # f <- f +
ggplot2::scale_fill_manual(values=c('#660000','#CC0000','#FF3366','#FFFFFF','#9
9FF99','#009933','#003300'))
  f <- f +
ggplot2::scale_fill_manual(values=c('#000000','#666666','#999999','#FFFFFF','#9
99999','#666666','#000000'))
}
#' Multiple plot function
#' @description ggplot objects can be passed in ..., or to plotlist (as a list
#'   of ggplot objects). If the layout is something like matrix(c(1,2,3,3),
#'   nrow=2, byrow=TRUE), then plot 1 will go in the upper left, 2 will go in
#'   the upper right, and 3 will go all the way across the bottom.
#' @param cols Number of columns in layout
#' @param layout A matrix specifying the layout. If present, 'cols' is ignored.
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)
  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)
  numPlots = length(plots)
  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }
  if (numPlots == 1) {
    print(plots[[1]])
  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))
    # Make each plot, in the correct location
    for (i in 1:numPlots) {
```

```
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))
      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

fetch_stocks.py

```
import pandas as pd
import tushare as ts
import sys # check sys.path
from dateutil.parser import parse
import os

def reformat_time(dataFrame, column):
  for i in range(len(dataFrame)):
      tmp = parse(dataFrame.ix[i, column])
      dataFrame.ix[i, column] = tmp.hour * 3600 + tmp.minute * 60 + tmp.second
  dataReturn = dataFrame.sort_values(by = column)
  return dataReturn

def main():
  codes = ['601318', '600036', '601166', '600016', '601328', '600000', '601288',
 '600300','600837', '601398']# weighted, financial
  dates = ['2015-08-19', '2015-08-20', '2015-08-21', '2015-08-24', '2015-08-25',
 '2015-08-26', '2015-08-27', '2015-08-28', '2015-08-31', '2015-09-01', '2015-09-
02', '2015-09-07', '2015-09-08', '2015-09-09', '2015-09-10', '2015-09-11', '2015
-09-14']
  for i in range(len(codes)):
    for j in range(len(dates)):
        df = ts.get_tick_data(codes[i], date = dates[j])
        df = reformat_time(df, 'time') # get the raw data
        file_name_raw = '_'.join([codes[i], ''.join([dates[j][0:4], dates[j][5:
7], dates[j][8:]])]) # 601099_20150820
        file_name = ''.join([file_name_raw, '.txt'])
        folder_name = '_'.join([codes[i], ''.join([dates[j][0:4],dates[j][5:
7]])])
        folder_path = '/'.join(['..', 'inst', 'extdata', folder_name])
        file_path = '/'.join([folder_path, file_name])
```

```
        # ../inst/extdata/601099_201508/601099_20150820.txt
        if not os.path.exists(folder_path):
            os.mkdir(folder_path)
        df.to_csv(file_path, encoding = 'utf-8', header=None, index=None, sep=' ',
 mode='a')
  return

if __name__ == "__main__":
  main()
```

test.R (partial)

```
library(FMM)
load("../data/IF1509_win.RData")

ctgs_table <- data.frame()
for (i in 1:length(datalist)) {
  ctgs_table <- rbind(ctgs_table,table(datalist[[i]]$ctgs))
}
colnames(ctgs_table) <- c('1', '2', '3', '4', '5', '6', '7')

table2 <- ctgs_table
for (i in 1:nrow(ctgs_table)) {
  table2[i,] <- ctgs_table[i,]/sum(ctgs_table[i,])
}

vars <- vector() # variances of price changes
for (i in 1:length(datalist))
  vars <- c(vars, round(var(datalist[[i]]$pch), 4))
spread_vars <- vector() # variances of price changes
for (i in 1:length(datalist))
  spread_vars <- c(spread_vars, round(var(datalist[[i]]$spread), 4))

spreads_before <- vector()
for (i in 1:6) {
  spreads_before <- c(spreads_before, datalist[[i]]$spread)
} # assemble the spreads before the event
# resampling
res_num    <- 5000 # number of selected data
res_before <- spreads_before[seq(1, length(spreads_before), length.out = res_nu
m)]
```

```
par(mfrow = c(3,1))
plot(res_before,
     xlab = 'Resampled Index',
     ylab = 'Spread, before the event A',
     ylim = c(0,15), type='l')

sec_scales <- c(5, 10, 20); min_scales <- c(1, 2, 5, 10, 20, 30)
rv_df <- get_rv_df(datalist, sec_scales, min_scales, length(datalist))
# plot_rv_df(rv_df) # not implemented yet
rv_df[,c("2015-08-19","2015-08-20","2015-08-21","2015-08-24","2015-08-25",
         "2015-09-07","2015-09-08","2015-09-09","2015-09-10","2015-09-11")]

before <- data.frame()
for (i in 1:6) {
  df <- datalist[[i]]
  df <- df[,c("ctgs","pch","vol","dvol")]
  before <- rbind(before, df)
}
middle <- data.frame()
for (i in 7:11) {
  df <- datalist[[i]]
  df <- df[,c("ctgs","pch","vol","dvol")]
  middle <- rbind(middle, df)
}
after <- data.frame()
for (i in 12:19) {
  df <- datalist[[i]]
  df <- df[,c("ctgs","pch","vol","dvol")]
  after <- rbind(after, df)
}
vars <- prepare_opm(before)
opm <- MASS::polr(vars$ctgs ~ vars$ctgs1 + vars$ctgs2 + vars$ctgs3
                  + vars$cp1 + vars$cp2 + vars$vol2, method="probit") # SIFs
summary(opm)
raw_probs <- opm$fitted.values
probs <- raw_probs[seq(1,dim(raw_probs)[1],1),][1:7000,]
plot1 <- plot_fp(probs)
fplot1 <- plot_fp_all(probs)
multiplot(fplot1, fplot2, fplot3, cols = 1)

before <- data.frame()
for (i in 1:6) {
  df <- datalist[[i]]
  df <- df[,c("A","D","S")]
```

```
  before <- rbind(before, df)
}

before <- vector()
for (i in 1:6) {
  df <- datalist[[i]]
  # duration <- prepare_adj_duration(df, threshold = 2, unit = 0.2) # for SIFs
  duration <- prepare_adj_duration(df, threshold = 2, unit = 0.01) # for stocks
  before <- c(before, duration)
}
acf(before,ylim = c(-0.01, 0.2))
plot(before, type='l', xlab = "Time Index", ylab = "Adjusted and Trimmed Duratio
n")
m1 <- fit_acd(before, order=c(1,1), cond.dist="weibull") # try other distributio
ns
sum(m$estimates[2:3]); m$estimates[1]/(1-sum(m$estimates[2:3])); mean(before)
# IF 0.9979265, 1.156822, 1.093395
acf(m$epsilon, ylim = c(-0.05, 1)); Box.test(m$epsilon, lag=10, type='Ljung')
```

谢辞

在此诚挚感谢安泰经济与管理学院管理信息系统系周志中教授作为我的指导老师在课题研究过程中给予我的巨大帮助、支持和鼓励。

2017 年 5 月 18 日

RESEARCH ON INFLUENCES OF ALGORITHMIC TRADING ON MARKET MICROSTRUCTURE OF CHINESE FINANCIAL MARKETS

This project studies the influence of algorithmic traders on market microstructure of Chinese capital markets in depth based on a series of significant events that happened in August and September, 2015. We employed a series of financial econometrical models and time series analysis approaches especially designed for studying high-frequency data, and analyzed the changes of characteristics of market microstructure after the policy changes based on the results generated from the modeling process.

The quantitative trading and algorithmic trading techniques have developed rapidly and globally since decades ago. The computer based trading strategy development and the order execution system could help investors to perform more effective market research and discovery of underpriced assets and arbitrage opportunities. The highly frequent trading also help markets to be more effective and sensitive to new information appeared in the macro context and investors' new expectation towards the future. In China, an increasing amount of financial institutions, including hedge funds, asset management firms and investment banks, have started to use algorithmic trading to ensure the execution and lower the transaction costs and market impact as few as possible. Easy to find that Chinese markets have experienced significant changes these years, including growing market volatility and trading volumes. Undoubtfully, the rise of algorithmic traders has contributed a lot to this. However, investors and the market supervisors have always had contentions about the roles and the effects of algorithmic trading in the markets. We need to fully understand the influences and the changes brought about by those high-frequency traders to make sure that the markets could keep healthy and stable.

In this project, our purpose is to answer the following questions based on the results from modeling and analyzing the real-world data: Does the algorithmic trading generate more volatility and risk of unexpected price changes? Did the restriction policy work when the supervisors wanted to gain back the stability of the markets? What would happen after the algorithmic traders leave the stock index futures market? What influences did the departure of algorithmic traders have on the stock market?

Started with the strict restrictions on algorithmic traders in the stock index futures market, we first identified the details of the policy changes, the formation of the active investors and the patterns of their behaviors, the possible influences and the model we should use for the special scenario. We think that those quantitative traders using the alpha arbitrage strategy were those who were greatly affected by the policy changes and had the most significant influences on the markets, since the beta arbitragers were already left months ago due to the very high forward discounts in

the stock index futures market. Those alpha arbitragers usually build their own portfolio with high excess return rate and use the stock index futures to hedge out the market's systematic risks to gain the riskless and stable excess return, so even under the pressure of forward discounts they still have chances to gain profit. However, due to the policy changes, which were mainly about trading volumes and frequency, they no long had approaches to operate normally and then had to quit.

Since those arbitragers, mainly quantitative hedge funds and asset management firms, were the major power in the markets and they participated in the markets very frequently based on the computer-based order execution system, we believe that their departure would have negative influences on the stability and the effectiveness of the markets; their absence would increase the transaction costs, the partial vibration of prices, and decrease the volatility. Except for the possible damage on the basic functions of the futures market, restrictions on them would very much likely force them to leave the stock market as well, which would further decrease the volatility and the stability of the stock market, bring about more pressure on the stock prices and cause extra vibration risks of prices.

To gain a clearer perspective on the variation of the market microstructure before and after the policy changes, we employed several quantitative econometrical models to analyze the underlying patterns hidden behind the high-frequency trading data of the major stock index futures contracts and selected representative stocks. For stock index futures market, we studied three main stock index futures contracts and show one as an example in this project. For stock market, we studied dozens of stocks with huge market cap and show several weighted and unweighted stocks in finance and energy industry to illustrate the point that the influences were market-wide instead of just weighted stocks. For price changes series, we used the Ordered Probit Model to obtain the estimated conditional probabilities of possible price changes of underlying prices in each half second, and found that the prices had more severe partial vibrations after the target event; we also used the ADS decomposition model to estimate the particular conditional probabilities of the price changes series, and we discovered more clustering effects and reversion effects of prices, indicating that when there're some new information appeared, the prices would experience more vibrations and intense changes. Besides, based on the calculated realized volatilities in different time scales, we found that after the event, the volatilities in seconds had increased a lot, showing stronger microstructure noises and higher autocorrelation in small scales. Based on the results, we concluded that the algorithmic traders have positive influences on markets qualities, stability, effectiveness and the development of the pricing function, after their departure, the stock futures market quality was deeply impaired and more risks from abnormal price concussions were triggered in both stock index futures market and stock market. Majority of weighted stocks and unweighted stocks were effected similarly as the stock index futures contracts after the departure of the algorithmic traders.

What's different from the previous research in this project is that we took advantage of the macro event and used the models to study the highly representative financial products. Hence, we obtained the changes of characteristics in micro perspective representing the whole market. The events had very strong impact on the investors within a very short period, so we could control the variables effectively. Based on the conclusions of this project, investors would have better comprehensions on algorithmic traders' roles in the markets and the influences of their frequent

trading on markets with respect to market microstructure, and the market supervisors could make the proper policies to maintain the stable evolvement of the financial markets.