对外经济贸易大学
University of International Business and Economics

蓝莓组 SAS 期末论文一

高频交易数据分析:

交易成本计算与 VAR 分析

组员 1 郑祎格 201520220563

组员 2 卢 吕 201520220687

组员 3 刘楚薇 201520220549

时间 2015 年 12 月 15 日

目录

小组成员的分工情况 1
一、问题阐述 2
二、数据描述 2
三、数据处理（附思维导图） 3
四、结果数据集与分析 5
五、实证结论 16
附件一：SAS 程序 17
1.数据提取 17
2.数据分析 18
附件二：数据集示例 24
附件三：小组两次讨论截图（部分截图） 26

小组成员的分工情况

- **数据收集：**

郑祎格：主要负责数据下载和拷贝

卢吕：主要负责数据的提取

- **程序编写：**

刘楚薇：主要整理和修改《SAS 在金融中的应用》第十章的代码，解释每个步骤并做出必要步骤的翻译，对代码含义进行注释。

卢吕：主要负责所有数据的运行和整理结果。

- **报告撰写：**

刘楚薇：主要负责报告的撰写和相关操作的注释，以及结果分析。

郑祎格：对于报告进行排版和报告内容修改。

签字处：

一、问题阐述

证券市场的的微观结构在过去的20年内已经成为金融研究中的一个热门话题。随着交易数据越来越容易获得，关于证券市场微观结构的实证研究也不断增多。许多问题通过这种方式得到了讨论，最普遍的的就是买卖价差问题。研究已经证实买卖价差的存在，并提出它由三部分组成：订单处理成本、存货成本以及信息不对称成本。另外一个吸引了许多学者注意的问题是机构与其他投资者在证券交易中所面对的交易成本问题。

然而对于交易成本的精确计量是十分困难的，尽管存在着有效价差以及类似的有效推算方法，但是相关数据的缺乏限制了对交易成本有效估计的，Demsetz(1968)将交易成本定义为投资者为立即执行交易而支付的费用,做市商买卖报价之间的差异度量了交易成本的大小。买卖价差实际上是交易成本的一种静态概念,可以在交易发生时观测。另一种方法是估计交易引起的瞬时价格变化,它是交易成本的一种动态概念,与价格变化有关。

与交易成本有关的另一个研究是关于一笔交易对股票价格的影响。关于交易对价格影响的精确估计可以为市场流动性好坏的估计提供信息。最早对股价与交易相关性的分析来自于Hasbrouck(1991)年的研究，他利用了向量自回归模型来估计对价格的影响。本文的主要目的是演示如何利用SAS来进行这种估计。此外，本文还将展示如何计算不同的交易成本的计量方法以及如何将交易分类为买入与卖出交易。国内相关文献中通常假设价格冲击函数为线性, 即假设交易引起的价格变动和交易规模间存在线性关系。但是, 线性的交易成本明显是不现实的 , 它会低估中等规模交易和小规模交易的冲击成本, 而高估大规模交易的冲击成本, 并且价格冲击不会是没有界限的。实际上，Bhattacharya和 Spiegel(1991)， Spiegel和Subrah-manyam(2000)指出价格冲击应该是连续非线性的,并且价格冲击函数是S型的,最重要的是,随着交易规模的增大价格冲击的大小是有限的。

二、数据描述

本次研究分析所使用的数据时沪深 300 成分股在 2005 年与 2006 年的高频交易

数据。分为 05 年成分股交易数据与 06 年成分股交易数据两个文件夹。每个文件夹中包括了沪深 300 的三百只股票的股票代码，交易日期及时间，成交价，成交量，报价的卖价与买价各五档。

由于我们从瑞思上获得的数据是交易与报价合在一起的，也就是说每一个观测对应了这个日期和时间上的交易价格，交易量，买一到买五的量价，卖一到卖五的量价。而且瑞思的数据质量较高，已经包含了计算出来的相对价差和相对有效价差，以及交易方向。运用这些数据已经足够用来分析交易成本，但是我们自己也进行了计算，得到的相对价差和有效价差与瑞思给数据一致。

所有的数据预先要分为成交与报价两个部分来分开处理，因为交易数据有延迟，美国的市场平均延迟 5 秒，但是报价基本没有延迟。这是因为报价的时候，速度是非常重要的，specialists 的下单基本不会有延迟，而成交之后是由交易所的职员对交易进行确认的，这个时候他们会不慌不忙的进行纪录，所以一般都会有延迟。我们需要研究的交易成本是紧跟着 quote 的 trade，因此需要对 trade 的时间做调整。

为了分别处理 trade 和 quote，我们将瑞思的数据先分开处理再合并。比较重要的成交有关的变量包括交易价格、时间戳记（time stamps）以及成交量；比较重要的与报价有关的变量包括买价与卖价以及报价的时间。此外，两部分数据都应该包括表明特殊事件以及错误的变量，以去除数据中错误的部分和法定交易时间之外的交易与报价。

三、数据处理（附思维导图）

最终结果呈现应该是两年，各三百只股票的交易成本分析与向量自回归模型分析，应该将每年的三百支股票合并在一个数据集中，将每个股票进行分析，结果全部在一个 report 里面。但是在初始阶段，本组用股票 stk06_000001 的高频数据进行试错，这样可以减少运算时间，提高效率。

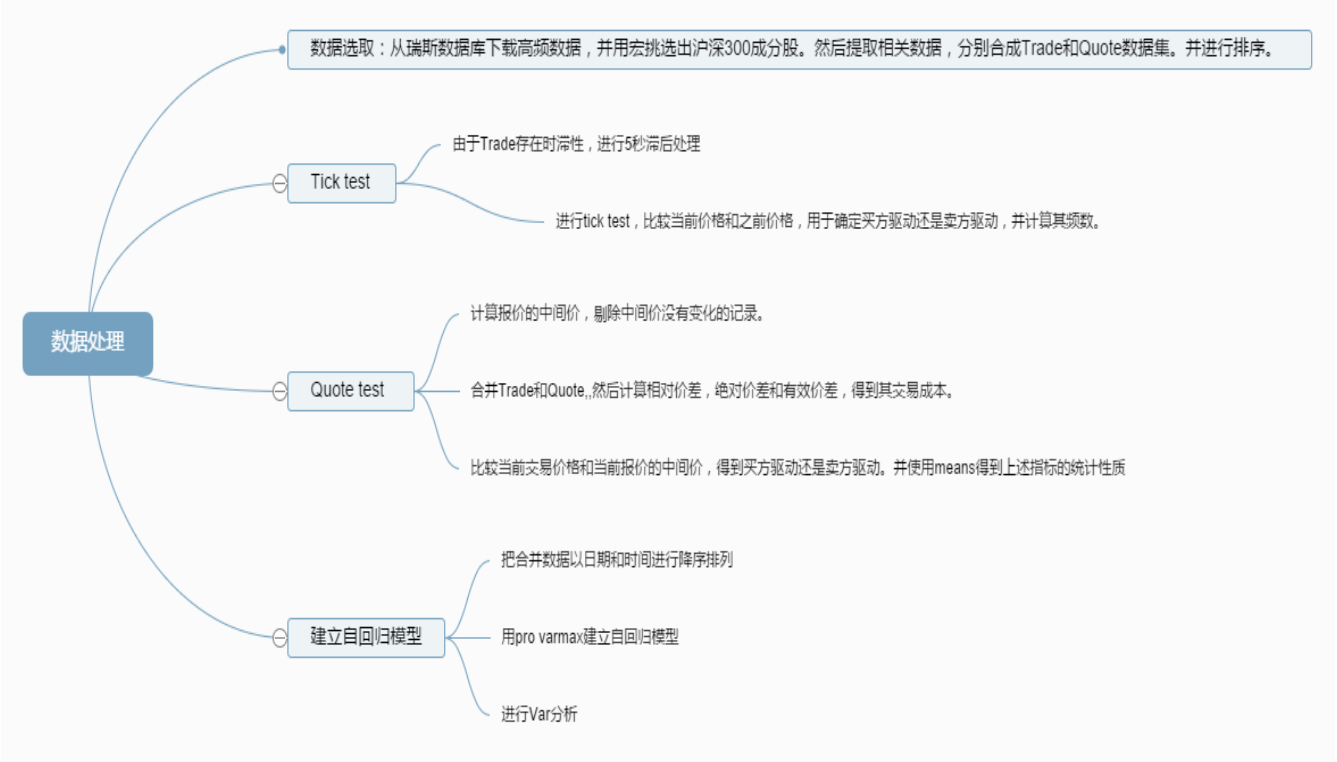
为了分别进行交易成本分析和 VAR 分析，我们用了两种不同的排序方法：交易成本分析：报价在前，交易价在后；VAR 分析：交易价在前，报价在后，以研究交易对报价变化的影响。

1. 从原始数据集中提取股票代码，交易日期，交易时间，成交价格和成交量组成 trade 数据集。提取股票代码，交易日期，交易时间，买一价格与卖一价格组成

quote 数据集。分别将数据集按照日期 date，时间 time 和交易价格 price 排序

2. 由于报价没有时滞，而交易价在纪录的时候有 5 秒时滞，所以将交易价的时间减去五秒。为了模型的稳健性，我们将继续尝试 10 秒与 15 秒时滞，得到的最终结果集区别并不大。
3. 进行 tick test，用于确定一个交易是买方驱动交易和卖方驱动交易，需要比较当前交易价格和之前的价格，若当前交易价格高，则将 tick 标为 1，买方驱动；若当前价格低，则将 tick 标为-1，卖方驱动。如果当前价格与之前价格相等，则比较之前价格和再之前的价格，若再一次相等，则 tick 标为 0。对处理结果进行频数表分析。
4. 计算报价的买卖中间价，由于在计算交易成本时只关心中间价变了的报价对于交易价格的影响，所以剔除掉中间价没有变化的记录。此处的中间价使用的是买一和卖一的算术平均。
5. 每一天开始的时候没有买卖价和中间价，将他们全部设为缺失值。接着计算相对价差，绝对价差和有效价差。这就是我们需要研究的交易成本。通过计算所有股票的交易成本，我们得到有效价差的均值进行分析，以得出最终结论。
6. 用 quote test 将当前交易价格和当前报价的中间价比较，得到一个记录是买方驱动 ordersign＝1 还是卖方驱动 ordersign＝-1，若当前交易价格和当前报价的中间价相等，则使用之前保存的 tick。使用 ordersign 与交易量的乘积计算订单净流入。最后使用 means 过程查看以上指标的统计性质。
7. 建立向量自回归模型之前需要将日期与时间以降序排列，以便得到紧跟着 trade 的 quote 这样的组合，我们计算了低于 15 秒的间隔与低于 30 秒的间隔得到的 VAR 估计参数，发现计算结果没有显著区别。如果有报价变化但是没有实际成交的话，就会把交易方向设为 0。
8. 最终进行 VAR 分析，我们分别做了 VAR（2）和 VAR（5）以说明结论的稳健性。并计算了报价中间价的变化值和净订单流的正交脉冲响应分析，画图以清晰地显示结果。

附：思维导图



四、结果数据集与分析

变量名	注解
Code	证券代码
QDate	交易日期
Time	交易时间
TPrice	成交价
TVolume	成交量
Askpr1	卖一
Bidpr1	买一
Midpoint	买一与卖一的中间价
Qid	报价的 ID 序号
Tid	交易的 ID 序号
Time_real	经过延时调整的交易时间
Lagprice	滞后一个成交价纪录

Tick	Tick test 得到的成交方向
Ordersign	Quote test 得到的成交方向
Effsprd	有效价差 abs(Tprice-(bidpr1+askpr1)/2)*2
Asprd	绝对价差 bidpr1－askpr1
Rsprd	相对价差 asprd/Tprice
Nof	净订单流 ordersign * TVolume
NBid	离成交最近的上一个买一价
NOfr	离成交最近的上一个卖一价
Trade	标识交易记录
Qtime	离成交最近的上一个报价的时间
Qage	交易与最近一次报价的时间差
Mpret	报价中点的变化

为了方便展示，本文截取 stk.000001 深发展的实证结果进行分析。前半部分是交易成本分析，主要目的是计算绝对价差，相对价差和相对有效价差。2005 于 2006 年沪深 300 的所有成分股的结果在附件 html 的文件里。

下图为 tick test 的频数表，是用当前成交价格和上一次成交价格比较，当前成交价格大则 tick＝1，tick 为 1 说明这笔交易是买方驱动，股票价格将向上移动，tick 为-1 说明这笔交易是卖方驱动，股票价格将向下移动。由于 tick 为 0 的交易过多，也会导致 tick test 的结果并不准确，所以我们将用另外一种 quote test 来讨论交易方向。

```
/*调整 trade 的时间延迟  */
data ntrades;
set adjtrades;
time_real =time;
time = time - 5;
tid = _n_;
Label time = 'trade time - 5 secs';
Label time_real = 'reported trade time';
```

```
Format time_real time8.;
/*调整trade的时间延迟  */
lagprice = lag(Tprice);
lag2price = lag2(Tprice);
If Tprice > lagprice then tick = 1;
If Tprice < lagprice then tick = -1;
If Tprice = lagprice then do;
If lagprice > lag2price then tick = 1;
If lagprice < lag2price then tick = -1;
End;
If _n_<3 then tick = 0;
If tick =. Then tick = 0;
Drop time_real lagprice lag2price;
Label tick = 'trade indicator based on tick test';
label tid='trade identifier';
label numtrades='number of aggregated trades';
run;
```

VAR estimation
quote revisions and net order flow

FREQ 过程

Code=000001

trade indicator based on tick test				
tick	频数	百分比	累积频数	累积百分比
-1	72042	31.09	72042	31.09
0	88471	38.18	160513	69.27
1	71196	30.73	231709	100.00

下图是 quote test 的结果，是将当前成交价与报价的中间价比较，若当前成交价大则 ordersign 为 1，说明是买方驱动。但若当前成交价与报价的中间价相等，则比较 tick test 的结果。只有 quote test 和 tick test 的结果都为 0 才将 0 赋给 ordersign 这

样利于之后计算净订单流 Nof（net order flow）。

```
/*计算价差  */
data spread;
set qandt;
by code Qdate;
if first.code or first.Qdate then do;
nbid=.;nofr=.;currentmidpoint=.;
end;
if bidpr1 ne . then nbid=bidpr1;
if askpr1 ne . then nofr=askpr1;
if midpoint ne . then currentmidpoint=midpoint;
effsprd = abs(Tprice - (nbid+nofr)/2)*2;
asprd = nofr - nbid;
rsprd = asprd/Tprice;
/*quote test  进一步计算交易方向  */
If currentmidpoint ne . then do;
If Tprice < currentmidpoint then ordersign = -1;
If Tprice > currentmidpoint then ordersign= 1;
If Tprice = currentmidpoint then do;
/*如果ordersign为0，则用tick test的结果继续分类*/
If tick = 1 then ordersign = 1;
If tick = -1 then ordersign = -1;
If tick = 0 then ordersign = 0;
End;
/*net order flow净订单流=交易方向乘以成交量*/
Nof = ordersign * TVolume;
End;
label nbid = 'last outstanding bid';
label nofr = 'last outstanding ofr';
Label effsprd = 'effective spread';
```

```
Label asprd = 'absolute spread';
Label rsprd = 'relative spread';
Label nof = 'net order flow';
Label ordersign = 'indicator for trade direction';
/*trade和价差对应*/
if trade=1 then output spread;
retain nbid nofr currentmidpoint;
drop bidpr1 askpr1 midpoint qid trade;
Run;
```

Spread estimation

FREQ 过程

indicator for trade direction				
ordersign	频数	百分比	累积频数	累积百分比
-1	118472	51.21	118472	51.21
0	131	0.06	118603	51.26
1	112761	48.74	231364	100.00

频数缺失 = 345

下表展示了价差的描述性统计量。可以看到有效价差的均值为0.047，略大于美国市场，然而价差也和股票自身特征有关。绝对价差和相对价差的均值都为负，绝对价差的均值失真可以从最大值出看出来，这个最大值的产生很可能是一个错误记录，或者可能是缺失值，在有异常值的时候，看中位数要比看均值更加准确。价差的存在造成了市场的摩擦，使得股票价格会偏离有效价格，这个偏离就是价格冲击。

```
/* 交易成本分析*/
Proc means data = spread n mean median min max;
by code;
Var Tprice TVolume effsprd asprd rsprd ordersign nof;
Run;
```

由下表可以看出，有效价差的中位数是 0.01，但是这个例子是 stk.000001 深发展，深发展是大公司，规模较大的公司价差小，0.01 虽然不能代表全市场的平均水平，但是根据我们生成的 html 文件结果集可以看出，绝大多数股票的有效价差的中位数都是 0.01。

Spread estimation

MEANS PROCEDURE

Code=000001

变量	标签	N	均值	中位数	最小值	最大值
TPrice	Trade Price	231709	6.1103834	6.1400000	0	7.3900000
tvolume	Trade Volume	231709	8028.68	2600.00	0	3176243.00
effsprd	effective spread	231364	0.0475668	0.0100000	0	11.1900000
asprd	absolute spread	231364	-0.0215284	0.0100000	-6.7900000	5.9700000
rsprd	relative spread	231363	-0.0034175	0.0016367	-1.0017483	1.0000000
ordersign	indicator for trade direction	231364	-0.0246840	-1.0000000	-1.0000000	1.0000000
Nof	net order flow	231364	-79.4872884	-100.0000000	-3089184.00	3176243.00

下图是建立向量自回归模型之前进行的描述性统计。可以看出深发展在这一段时间里股价向下走的时候多于股价向上走的时候。报价的中间价变化的记录只占所有交易记录不到十分之一。净订单流的均值和中位数都是负的，但是相对于它们的数量级来说，筹码净流出的量还是较小的。

```
/*均值过程  计算描述性统计量*/
Proc means data= result.vardata n mean median min max;
Var mpret tick nof ordersign;
by code;
run;
```

VAR estimation

MEANS PROCEDURE

Code=000001

变量	标签	N	均值	中位数	最小值	最大值
Mpret	change in quote midpoint	19395	-0.000376128	8.881784E-16	-3.2200000	5.2900000
tick	nof sign - tick test	231718	-0.0036510	0	-1.0000000	1.0000000
Nof	net order flow	231373	-79.4841965	-100.0000000	-3089184.00	3176243.00
ordersign	nof sign - combined quote and tick test	231373	-0.0246831	-1.0000000	-1.0000000	1.0000000

下图是向量自回归模型，向量自回归的形式如下。

$$x_t = \alpha_0 + \sum_{i=1}^{p} \alpha_{t-i}^{x} x_{t-i} + \sum_{t=1}^{p} \alpha_{t-i}^{r} r_{t-i} + u^{x}$$

$$r_t = \gamma_0 + \gamma_t^{x} x_t + \sum_{i=1}^{p} \gamma_{t-i}^{x} x_{t-i} + \sum_{t=1}^{p} \gamma_{t-i}^{r} r_{t-i} + u^{r}$$

模型来自 Hasbrouck（1991），其中 x 代表净订单流，r 代表报价中点的变化。误差项 u 均值为 0。这种价格冲击模型是线性的。

VAR estimation
quote revisions and net order flow

VARMAX 过程

观测数	19368
配对缺失数	0

简单汇总统计量

变量	类型	N	均值	标准偏差	最小值	最大值	标签
Nof	Dependent	19368	-158.13104	52422.28695	-3089184.000	3176243.0000	net order flow
Mpret	Dependent	19368	-0.00036	0.05908	-3.22000	5.29000	change in quote midpoint

根据书中模型，给 VAR 定阶为 5，运用最小二乘估计法进行参数估计。此处应进行优化。比如可以使用最大似然估计进行参数估计，或是将 VAR 定阶为 2 再建模，因为一般向量自回归模型滞后 1 阶或 2 阶的情况较为普遍。

VAR estimation
quote revisions and net order flow

VARMAX 过程

模型类型	VAR(5)
估计方法	Least Squares Estimation

模型有两个方程，第一个方程的被解释变量是净订单流 Nof，解释变量包含了 nof 的五阶滞后项和中间价变化值 mpret 的五阶滞后项和一个常数。第二个方程的被解释变量是中间价变化值 mpret，解释变量包含了 nof 的五阶滞后项和中间价变化值 mpret 的五阶滞后项和一个常数。可以看出，滞后 1、2 阶参数的 t 统计量都比较显著。

模型参数估计值						
方程	参数	估计值	标准误差	t 值	Pr > |t|	变量
Nof	CONST1	-198.75309	361.24101	-0.55	0.5822	1
	AR1_1_1	-0.26131	0.00719	-36.36	0.0001	Nof(t-1)
	AR1_1_2	-63883.19768	6118.58041	-10.44	0.0001	Mpret(t-1)
	AR2_1_1	0.04512	0.00743	6.08	0.0001	Nof(t-2)
	AR2_1_2	-22141.48906	6135.03960	-3.61	0.0003	Mpret(t-2)
	AR3_1_1	0.06804	0.00742	9.17	0.0001	Nof(t-3)
	AR3_1_2	6119.03929	6137.10228	1.00	0.3187	Mpret(t-3)
	AR4_1_1	0.03503	0.00743	4.72	0.0001	Nof(t-4)
	AR4_1_2	2052.36016	6135.63962	0.33	0.7380	Mpret(t-4)
	AR5_1_1	0.02135	0.00717	2.98	0.0029	Nof(t-5)
	AR5_1_2	6671.52520	6132.82605	1.09	0.2767	Mpret(t-5)
Mpret	CONST2	-0.00037	0.00042	-0.88	0.3796	1
	AR1_2_1	0.00000	0.00000	2.35	0.0188	Nof(t-1)
	AR1_2_2	-0.00224	0.00719	-0.31	0.7556	Mpret(t-1)
	AR2_2_1	-0.00000	0.00000	-3.66	0.0003	Nof(t-2)
	AR2_2_2	-0.00216	0.00721	-0.30	0.7640	Mpret(t-2)
	AR3_2_1	-0.00000	0.00000	-1.01	0.3104	Nof(t-3)
	AR3_2_2	0.00064	0.00721	0.09	0.9293	Mpret(t-3)
	AR4_2_1	-0.00000	0.00000	-2.48	0.0132	Nof(t-4)
	AR4_2_2	-0.00445	0.00721	-0.62	0.5366	Mpret(t-4)
	AR5_2_1	-0.00000	0.00000	-1.43	0.1534	Nof(t-5)
	AR5_2_2	0.00099	0.00721	0.14	0.8902	Mpret(t-5)

新息的协方差		
变量	Nof	Mpret
Nof	2526096040.2	49.23907
Mpret	49.23907	0.00349

下图是 VAR 模型滞后期的选择，信息准则包括 Akaike（AIC），Schwartz（SBC），Hannan－Quinn（HQC），Final Prediction Error（FPEC）。以下信息准则都比较小，所以我们认为 VAR（5）是合理的。

VAR estimation
quote revisions and net order flow

VARMAX 过程

信息准则	
AICC	15.99217
HQC	15.9951
AIC	15.99217
SBC	16.00111
FPEC	8816761

下图是正交化脉冲响应函数。对于任何一个 VAR 模型都可以表示成为一个无限阶的 MA 模型。

$$\mathbf{Y_{t+s}} = \mathbf{U_{t+s}} + \Psi_1 \mathbf{U_{t+s-1}} + \Psi_2 \mathbf{U_{t+s-2}} + \ldots + \Psi_s \mathbf{U_t} + \ldots \qquad \Psi_s = \partial \mathbf{Y}_{t+s} / \partial \mathbf{U}_t$$

Ψ s中第i行第j列元素表示的是,令其它误差项在任何时期都不变的条件下, 当第j个变量yj t对应的误差项uj t在t期受到一个单位的冲击后,对第i个内生 变量yit在t + s期造成的影响。把Ψ s中第i行第j列元素看作是滞后期s的函数称作脉冲响应函数(impulse-response function)。

$$\partial \mathbf{yi,t+s} / \partial \mathbf{ujt}$$

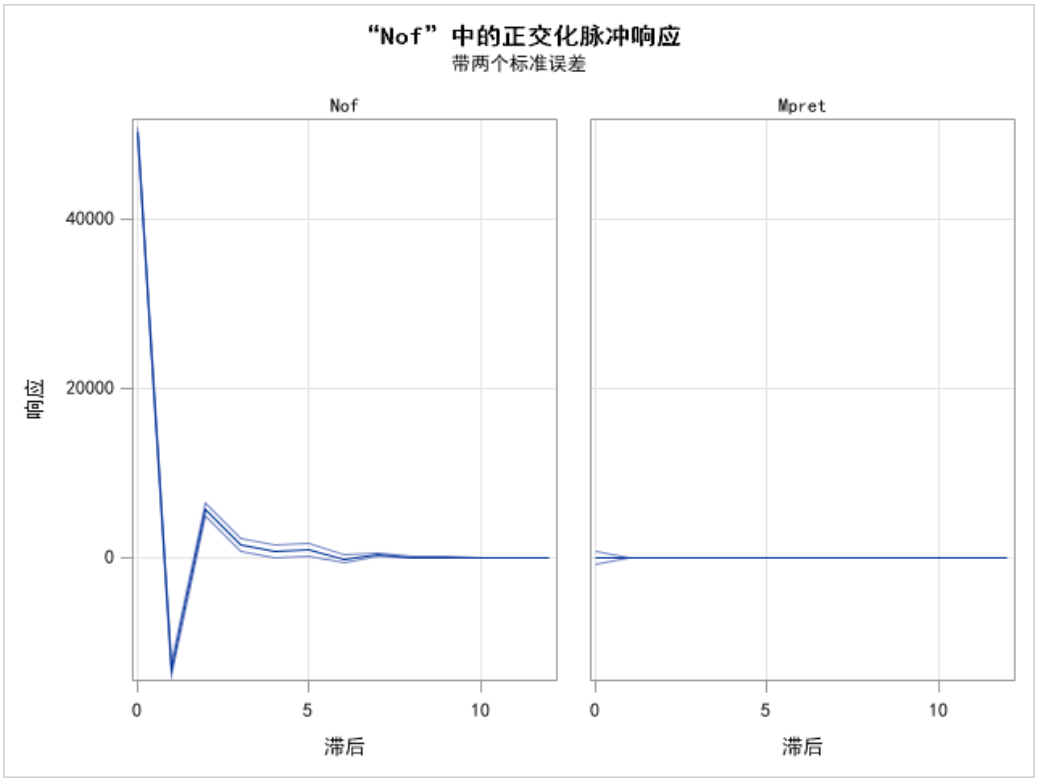
脉冲响应函数描述了其它变量在t期以及以前各期保持不变的前提下,yi, t+s对uj, t时一次冲击的响应过程。下图展示的是净订单流nof和中间价的变化值mpret各种滞后12阶的偏导数。也就是说给定其他变量的冲击在各期都不变的前提下，可以看出，随着阶数的增加，这种一次性冲击的影响是不断衰减的。

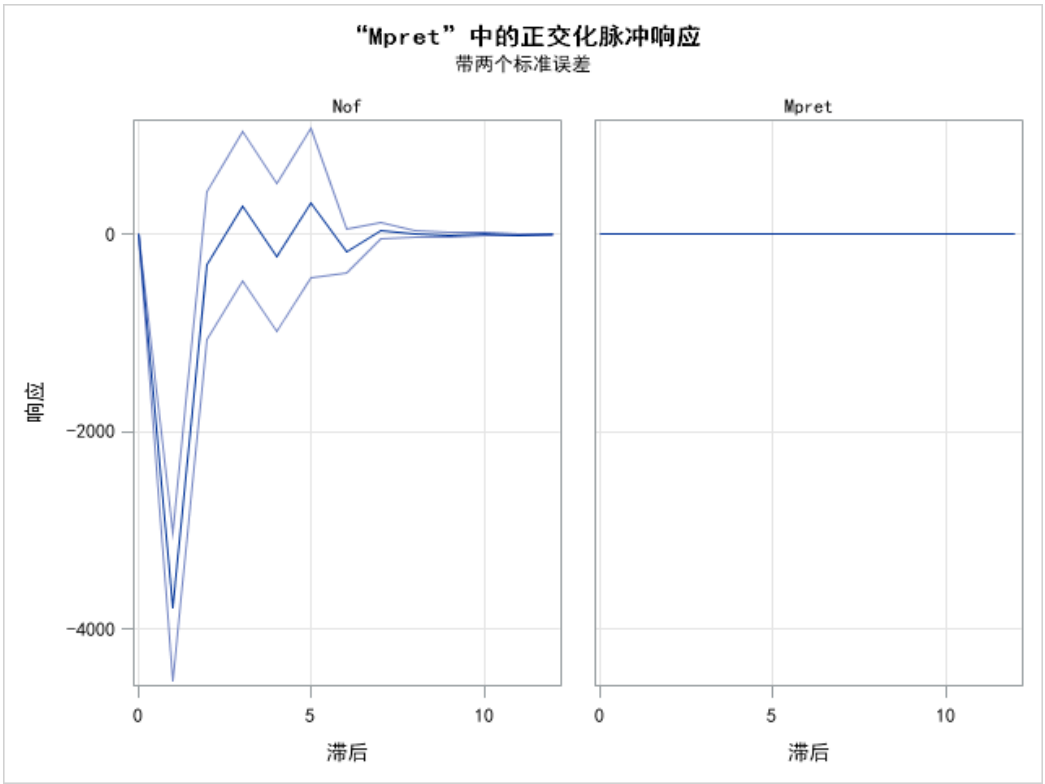
下表的纵向是Nof和Mpret滞后0到12阶变化一单位分别对当前Nof和Mpret的影响。其中滞后0阶的意思就是当前期的数值。

按变量划分的正交化脉冲响应			
变量 响应\脉冲	滞后	Nof	Mpret
Nof	0	50260.28293	0.00000
	1	-13196.23797	-3772.01397
	2	5630.86189	-313.23866
	3	1456.19007	288.81149
	4	779.98704	-231.14203
	5	922.67855	322.68788
	6	-137.50066	-170.54087
	7	289.70426	37.49071
	8	43.43694	3.47581
	9	38.07936	-5.25346
	10	25.99306	6.05915
	11	6.37795	-3.31924
	12	9.17231	1.50114
Mpret	0	0.00098	0.05905
	1	0.00100	-0.00013
	2	-0.00187	-0.00020
	3	0.00009	0.00015
	4	-0.00112	-0.00021
	5	-0.00040	0.00013
	6	0.00003	0.00006
	7	-0.00014	-0.00002
	8	-0.00003	0.00001
	9	-0.00004	-0.00000
	10	-0.00001	-0.00000
	11	-0.00000	0.00000
	12	-0.00001	-0.00000

下图为净订单流（nof）和报价中间价的变化值的正交脉冲响应。横轴为滞后阶数，纵轴为响应的大小。可以看出中国的市场还是较为混乱与无序的，净订单流的增加会导致滞后一阶两阶的净订单流降低，偏导数为负。另外，由于数量级差距过

大，报价中间价造成的影响和被影响的值变化并不大。





五、实证结论

本文借鉴金融中的 SAS 应用，利用沪深 300 指数成分股票的分笔交易数据，分别估计了证券交易成本的相对价差，绝对价差和有效价差。我们通过价格变化的方向来估计交易发生的方向，如果交易是由买方发起的，那么价格会有上升的趋势；如果交易是由卖方发起的，那么价格会有下降的趋势。有效价差的平均水平约为 1%，约等于印花税的水平。对交易成本的估计一方面说明我国股票市场的印花税水平相对较高，另一方面说明买卖价差中还存在其他交易成本成分。但是交易成本还和股票自身的特征有关。

本文的研究，不仅有助于了解中国股票市场交易成本的水平，还为交易对报价的影响分析提供了实证结果。但是本文的向量自回归模型和用来计算价差的方法没有考虑到订单流不平衡和价格冲击的非线性特征，这样会低谷中等规模交易和小规模交易的冲击成本，这也是日后的一个改进方向。最后，做正交脉冲响应函数的时候，其严重依赖于分解次序，也就是说，为了保证模型的稳健型，需要把中间价的变化值放在净订单流的前面也做一遍正交脉冲响应函数，两者的结果相差不大则说明计算结果是稳健的。

附件一：SAS 程序

```
/* Blueberry 刘楚薇 卢吕 郑祎格*/
/*高频数据——交易成本分析及VAR分析 */
```

1.数据提取

```
/*数据提取*/
libname ressethf   'E:\研究生课程\sas\小组作业数据\HFSAS';/*建立拷贝好的高频数据库*/
libname res_hf05   'E:\研究生课程\sas\小组作业数据\05年成分股交易数据';/*电脑上储存数据的位置*/
libname res_hf06   'E:\研究生课程\sas\小组作业数据\05年成分股交易数据';

/*读入沪深300成分股代码 */
proc import out=stkcode(rename=(_COL0=code))
datafile='D:\sas\小组作业高频\code.xls'/*事先下好的名单*/
dbms=excel replace;
getnames=yes;
run;
data _null_;
set stkcode;
file 'D:\sas\小组作业高频\code.txt';
string='%creat_data('||compress(code)||');';
put string $;
run;
/*根据沪深300代码读入数据 */
%macro creat_data(code);
data ressethf.stk05_&code;
set res_hf05.stk05_&code;
run;
%mend creat_data;
%include 'D:\sas\小组作业高频\code.txt';

%macro creat_data(code);
data ressethf.stk06_&code;
set res_hf06.stk06_&code;
run;
%mend creat_data;
%include 'D:\sas\小组作业高频\code.txt';

/*数据合并 */

x dir E:\研究生课程\sas\小组作业数据\05年成分股交易数据\stk*.*/b>D:\sas\小组作业高频\namelist.txt;
```

```
%macro creat_data(code);
data ressethf.all;
set res_hf05.stk05_&code
      res_hf06.stk06_&code ;
run;
%mend creat_data;
%include 'D:\sas\小组作业高频\namelist.txt';
run;
```

2.数据分析

```
/*数据操作 */
libname res_hf05   'C:\SAS高频作业\05年成分股交易数据';
libname result 'D:\sas\小组作业高频';

/* 将trade分出来*/
data result.trades;
set gp.all;
keep code qdate qtime tprice tvolume;
rename qtime=time;
run;

/*将quote分出来 */
data result.quotes;
set gp.all;
keep code qdate qtime Askpr1 Bidpr1;
rename qtime=time;
run;

Proc sort data=result.trades   out=trades ;
By code Qdate time Tprice;
run;

/*均值过程 得到trade描述性统计量 */
Proc means data=trades noprint;
By code Qdate time Tprice;
Output out = adjtrades (rename=(_freq_=numtrades) drop=_type_) sum(tvolume)=tvolume;
run;

/*调整trade的时间延迟 */
data ntrades;
set adjtrades;
time_real =time;
```

```
time = time - 5;
tid = _n_;
Label time = 'trade time - 5 secs';
Label time_real = 'reported trade time';
Format time_real time8.;

/* tick test  得到成交方向*/
lagprice = lag(Tprice);
lag2price = lag2(Tprice);
If Tprice > lagprice then tick = 1;
If Tprice < lagprice then tick = -1;
If Tprice = lagprice then do;
If lagprice > lag2price then tick = 1;
If lagprice < lag2price then tick = -1;
End;
If _n_<3 then tick = 0;
If tick =. Then tick = 0;
Drop time_real lagprice lag2price;
Label tick = 'trade indicator based on tick test';
label tid='trade identifier';
label numtrades='number of aggregated trades';
run;

/*频数过程  得到tick的分布*/
proc freq data = ntrades;
by code;
tables tick;
run;

proc sort data = result.quotes;
by code Qdate time;
run;
/*计算报价中间价的变化  */
data allqchange;
set result.quotes;
by code;
Midpoint = (Bidpr1 + Askpr1)/2;
Oldmp = lag(midpoint);
if first.code then oldmp=.;
qid = _n_;
label qid='quote identifier';
Drop oldmp;
Label midpoint = 'quote midpoint';
If midpoint ne oldmp then output;
```

```
run;
/*将trade和quote合并 */
data qandt;
set allqchange (in=a) ntrades (in=b);
if a then trade=0;
if b then trade=1;
run;
title1 'Spread estimation';
proc sort data=qandt;
by code Qdate time;
run;
/*计算价差 */
data spread;
set qandt;
by code Qdate;
if first.code or first.Qdate then do;
nbid=.;nofr=.;currentmidpoint=.;
end;
if bidpr1 ne . then nbid=bidpr1;
if askpr1 ne . then nofr=askpr1;
if midpoint ne . then currentmidpoint=midpoint;
effsprd = abs(Tprice - (nbid+nofr)/2)*2;
asprd = nofr - nbid;
rsprd = asprd/Tprice;
/*quote test 进一步计算交易方向 */
If currentmidpoint ne . then do;
If Tprice < currentmidpoint then ordersign = -1;
If Tprice > currentmidpoint then ordersign= 1;
If Tprice = currentmidpoint then do;
/*如果ordersign为0，则用tick test的结果继续分类*/
If tick = 1 then ordersign = 1;
If tick = -1 then ordersign = -1;
If tick = 0 then ordersign = 0;
End;
/*net order flow净订单流=交易方向乘以成交量*/
Nof = ordersign * TVolume;
End;
label nbid = 'last outstanding bid';
label nofr = 'last outstanding ofr';
Label effsprd = 'effective spread';
Label asprd = 'absolute spread';
Label rsprd = 'relative spread';
Label nof = 'net order flow';
Label ordersign = 'indicator for trade direction';
```

```
/*trade和价差对应*/
if trade=1 then output spread;
retain nbid nofr currentmidpoint;
drop bidpr1 askpr1 midpoint qid trade;
Run;

/*频数过程 看ordersign的分布 */
proc freq data = spread;
tables ordersign;
run;

/* 交易成本分析*/
Proc means data = spread n mean median min max;
by code;
Var Tprice TVolume effsprd asprd rsprd ordersign nof;
Run;

/*VAR estimation*/
Proc sort data = qandt;
Title 'VAR estimation';
By code descending Qdate descending time descending qid;
Run;
/* 将数据按日期时间倒序重排*/
data tradematch;
set qandt;
by code descending Qdate ;
lagtrade=lag(trade);
lagcode=lag(code);
if first.code and first.Qdate then do;
end;

if trade=0 then do;
nbid=bidpr1;nofr=askpr1;nqid=qid;qtime=time;nmidpoint=midpoint;
end;
/*保留距离quote最近的前一个trade，间隔30秒最多 */
if trade=1 and lagtrade=0 and code=lagcode then do;
qage=qtime-time;
if qage<=30 then do;
bid=bidpr1;
ofr=askpr1;
qid=nqid;
midpoint=nmidpoint;
end;
format qtime time8.;
```

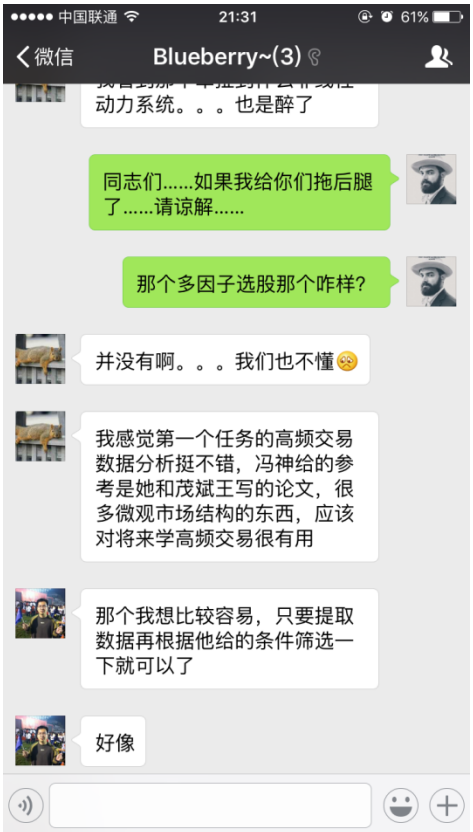
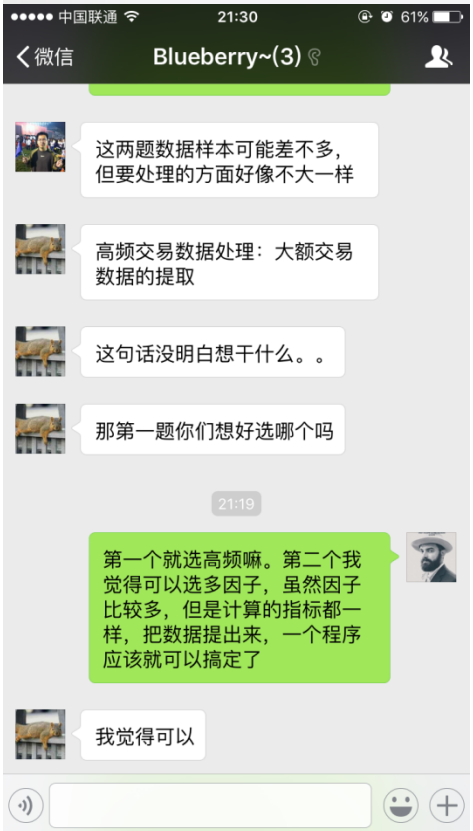
```
end;
label nbid     = 'last outstanding bid';
label nofr     = 'last outstanding ofr';
label gtime = 'time of last quote';
label qage    = 'delay between trade and quote';
retain nbid nofr qtime nmidpoint nqid;
drop nmidpoint nqid lagcode lagtrade trade;
if trade=1 then output tradematch;
run;

proc sort data=tradematch (keep=qid qage where=(qage ne . and qage<=30))
out=qids nodupkey;
by qid;
run;
/*横向合并所有报价变化，保留不能匹配的报价 */
data quotematch;
merge allqchange (in=a) qids (in=b drop=qage);
by qid;
if not b;
run;

/*将不能匹配的报价纵向合并进去 */
data qnspread;
set tradematch (drop=bid ofr Tprice numtrades qtime) quotematch (in=new);
unmatchedquote=new;
run;
proc sort data=qnspread;
by code Qdate time qid;
run;

/*计算中间价变化值 */
data allqncspread;
set qnspread;
by code;
if not first.code then do;
if midpoint ne . then currentmidpoint=midpoint;
Mpret = midpoint - lag(midpoint);
end;
label Mpret='change in quote midpoint';
retain currentmidpoint;
drop currentmidpoint;
run;

Proc sort data = allqncspread;
```

```
by tid;
run;

/*把成交方向变量合并进去 */
data result.vardata;
merge allqncspread (keep=code Qdate time tid unmatchedquote mpret)
      spread (keep=tid nof ordersign tick);
by tid;
/*没有匹配上的quote的净订单流和交易方向都设为0*/
if unmatchedquote then do;
nof=0;
tick=0;
ordersign=0;
end;
label mpret = 'change in quote midpoint';
label nof = 'net order flow';
label ordersign = 'nof sign - combined quote and tick test';
label tick = 'nof sign - tick test';
drop unmatchedquote;
run;

proc sort data=result.vardata;
by code Qdate time;
run;
/*均值过程 计算描述性统计量*/
Proc means data= result.vardata n mean median min max;
Var mpret tick nof ordersign;
by code;
run;

/*VAR参数估计与正交脉冲响应函数 */
Proc varmax data=result.vardata (where=(mpret ne . and nof ne .));
by code;
Title2 'quote revisions and net order flow';
Model nof, mpret / p=5 method=ls printform=univariate print=(impulse=(orth));
Output lead=0;
run;

*ods printer close;
```

附件二：数据集示例

原始数据集：ressethf.all

变量名	注解	观测个数
Code	证券代码	966398
QDate	交易日期	966398
Time	交易时间	966398
TPrice	成交价	966398
TVolume	成交量	966398
Askpr1	卖一	966398
Bidpr1	买一	966398
Midpoint	买一与卖一的中间价	966398
Qid	报价的 ID 序号	966398
Tid	交易的 ID 序号	966398
Time_real	经过延时调整的交易时间	966398
Lagprice	滞后一个成交价纪录	966398
Tick	Tick test 得到的成交方向	966398
Ordersign	Quote test 得到的成交方向	966398
Effsprd	有效价差 abs(Tprice-(bidpr1+askpr1)/2)*2	966398
Asprd	绝对价差 bidpr1－askpr1	966398

Rsprd	相对价差 asprd/Tprice	966398
Nof	净订单流 ordersign * TVolume	966398
NBid	离成交最近的上一个买一价	966398
NOfr	离成交最近的上一个卖一价	966398
Trade	标识交易记录	966398
Qtime	离成交最近的上一个报价的时间	966398
Qage	交易与最近一次报价的时间差	966398
Mpret	报价中点的变化	966398

附件三：小组两次讨论截图（部分截图）