/\* HOLDEN AND JACOBSEN DAILY TAQ PC-BASED CODE 2016-07-21

Our SAS code selects the Daily TAQ (DTAQ) data that you want from WRDS, downloads it to a PC, creates the official complete National Best Bid and Offer (NBBO), and computes standard liquidity measures. It was last updated on July 21, 2016.

It is based on: Holden, C. and S. Jacobsen, 2014, Liquidity Measurement Problems in Fast, Competitive Markets: Expensive and Cheap Solutions, Journal of Finance 69, 1747-1785. Our original research code has been adapted to work with WRDS and updated for the switch to microseconds.

Our code creates the following files in the "project" folder of a PC:

(1) Raw data files containing DTAQ data downloaded from WRDS:

"project.DailyNBBO" contains NBBO data

"project.DailyQuote" contains quote data

"project.DailyTrade" contains trade data

Importantly, the "project.DailyNBBO" file does NOT contain the complete NBBO. When one exchange has both the best bid and best offer it is only noted in the "project.DailyQuote" file, not the

"project.DailyNBBO" file. Our code combines data from both files to construct the official complete NBBO (see file below).

(2) Intermediate data files:

"project.OfficialCompleteNBBO" contains the official complete NBBO

"project.TradesandCorrespondingNBBO" contains trades and the corresponding NBBO from the prior microsecond

"project.BuySellIndicators" adds buy/sell indicators based on three conventions: LR = Lee & Ready (1991), EMO = Ellis, Michaely & O’Hara (2000), CLNV =Chakrabarty, Li, Nguyen, & Van Ness (2006)

(3) Output files containing standard liquidity measures:

"project.QuotedSpreadsandDepths" contains Quoted Spreads and Depths

"project.EffectiveSpreads" contains Effective Spreads

"project.RealizedSpreadsandPriceImpacts" contains Realized Spreads and Price Impacts that are aggregated based three conventions:

Ave = simple average, DW = dollar-weighted, SW = share-weighted

Step-by-step instructions for running this program using WRDS PC-SAS Connect are available at: www.kelley.iu.edu/cholden/instructions.pdf

We welcome any comments or suggestions. We can be reached at:

Craig Holden: cholden@indiana.edu

Stacey Jacobsen: staceyj@cox.smu.edu

We thank Charles Collver, Ruslan Goyenko, and Zhong Zhang for helpful suggestions and corrections. We are solely responsible for any remaining errors. Of course, you use this code at your own risk. \*/

/\* Global settings \*/

libname project 'C:\project\';

options errors=**50**;

**/\* STEP 1: RETRIEVE DAILY TRADE AND QUOTE (DTAQ) FILES FROM WRDS AND DOWNLOAD TO PC \*/**

/\* Connect to WRDS \*/

%let wrds = localhost 4016;

options comamid=TCP remote=WRDS;

signon username=\_prompt\_;

/\* Submit SAS statements to WRDS \*/

rsubmit;

libname nbbo '/wrds/nyse/sasdata/taqms/nbbo';

libname cq '/wrds/nyse/sasdata/taqms/cq';

libname ct '/wrds/nyse/sasdata/taqms/ct';

option msglevel=i mprint source;

/\* Retrieve NBBO data \*/

**data** DailyNBBO;

/\* Enter NBBO file names in YYYYMMDD format for the dates you want \*/

set nbbo.nbbom\_20150727 nbbo.nbbom\_20150728;

/\* Enter company tickers you want \*/

where sym\_root in ('AAPL','IBM') and

/\* Quotes are retrieved prior to market open time to ensure NBBO

Quotes are available for beginning of the day trades \*/

((**"9:00:00.000000"t**) <= time\_m <= (**"16:00:00.000000"t**));

format date date9.;

format time\_m part\_time trf\_time TIME20.6;

**run**;

/\* Retrieve Quote data \*/

**data** DailyQuote;

/\* Enter Quote file names in YYYYMMDD format for the same dates \*/

set cq.cqm\_20150727 cq.cqm\_20150728;

/\* Enter the same company tickers as above \*/

where sym\_root in ('AAPL','IBM') and

/\* Quotes are retrieved prior to market open time to ensure NBBO

Quotes are available for beginning of the day trades\*/

((**"9:00:00.000000"t**) <= time\_m <= (**"16:00:00.000000"t**));

format date date9.;

format time\_m part\_time trf\_time TIME20.6;

**run**;

/\* Retrieve Trade data \*/

**data** DailyTrade;

/\* Enter Trade file names in YYYYMMDD format for the same dates \*/

set ct.ctm\_20150727 ct.ctm\_20150728;

/\* Enter the same company tickers as above \*/

where sym\_root in ('AAPL','IBM') and

/\* Retrieve trades during normal market hours \*/

((**"9:30:00.000000"t**) <= time\_m <= (**"16:00:00.000000"t**));

type='T';

format date date9.;

format time\_m part\_time trf\_time TIME20.6;

**run**;

/\* Download to PC \*/

**proc** **download** data=DailyNBBO out=project.DailyNBBO;

**run**;

**proc** **download** data=DailyQuote out=project.DailyQuote;

**run**;

**proc** **download** data=DailyTrade out=project.DailyTrade;

**run**;

endrsubmit;

**/\* STEP 2: CLEAN THE DTAQ NBBO FILE \*/**

**data** NBBO2;

set project.DailyNBBO;

/\* Quote Condition must be normal (i.e., A,B,H,O,R,W) \*/

if Qu\_Cond not in ('A','B','H','O','R','W') then delete;

/\* If canceled then delete \*/

if Qu\_Cancel='B' then delete;

/\* if both ask and bid are set to 0 or . then delete \*/

if Best\_Ask le **0** and Best\_Bid le **0** then delete;

if Best\_Asksiz le **0** and Best\_Bidsiz le **0** then delete;

if Best\_Ask = **.** and Best\_Bid = **.** then delete;

if Best\_Asksiz = **.** and Best\_Bidsiz = **.** then delete;

/\* Create spread and midpoint \*/

Spread=Best\_Ask-Best\_Bid;

Midpoint=(Best\_Ask+Best\_Bid)/**2**;

/\* If size/price = 0 or . then price/size is set to . \*/

if Best\_Ask le **0** then do;

Best\_Ask=**.**;

Best\_Asksiz=**.**;

end;

if Best\_Ask=**.** then Best\_Asksiz=**.**;

if Best\_Asksiz le **0** then do;

Best\_Ask=**.**;

Best\_Asksiz=**.**;

end;

if Best\_Asksiz=**.** then Best\_Ask=**.**;

if Best\_Bid le **0** then do;

Best\_Bid=**.**;

Best\_Bidsiz=**.**;

end;

if Best\_Bid=**.** then Best\_Bidsiz=**.**;

if Best\_Bidsiz le **0** then do;

Best\_Bid=**.**;

Best\_Bidsiz=**.**;

end;

if Best\_Bidsiz=**.** then Best\_Bid=**.**;

/\* Bid/Ask size are in round lots, replace with new shares variable\*/

Best\_BidSizeShares = Best\_BidSiz \* **100**;

Best\_AskSizeShares = Best\_AskSiz \* **100**;

**run**;

**/\* STEP 3: GET PREVIOUS MIDPOINT \*/**

**proc** **sort**

data=NBBO2 (drop = Best\_BidSiz Best\_AskSiz);

by sym\_root date;

**run**;

**data** NBBO2;

set NBBO2;

by sym\_root date;

lmid=lag(Midpoint);

if first.sym\_root or first.date then lmid=**.**;

lm25=lmid-**2.5**;

lp25=lmid+**2.5**;

**run**;

/\* If the quoted spread is greater than $5.00 and the bid (ask) price is less

(greater) than the previous midpoint - $2.50 (previous midpoint + $2.50),

then the bid (ask) is not considered. \*/

**data** NBBO2;

set NBBO2;

if Spread gt **5** and Best\_Bid lt lm25 then do;

Best\_Bid=**.**;

Best\_BidSizeShares=**.**;

end;

if Spread gt **5** and Best\_Ask gt lp25 then do;

Best\_Ask=**.**;

Best\_AskSizeShares=**.**;

end;

keep date time\_m sym\_root Best\_Bidex Best\_Bid Best\_BidSizeShares Best\_Askex

Best\_Ask Best\_AskSizeShares Qu\_SeqNum;

**run**;

**/\* STEP 4: OUTPUT NEW NBBO RECORDS - IDENTIFY CHANGES IN NBBO RECORDS**

**(CHANGES IN PRICE AND/OR DEPTH) \*/**

**data** NBBO2;

set NBBO2;

if sym\_root ne lag(sym\_root)

or date ne lag(date)

or Best\_Ask ne lag(Best\_Ask)

or Best\_Bid ne lag(Best\_Bid)

or Best\_AskSizeShares ne lag(Best\_AskSizeShares)

or Best\_BidSizeShares ne lag(Best\_BidSizeShares);

**run**;

**/\* STEP 5: CLEAN DTAQ QUOTES DATA \*/**

**data** quoteAB;

set project.DailyQuote;

/\* Create spread and midpoint\*/;

Spread=Ask-Bid;

/\* Delete if abnormal quote conditions \*/

if Qu\_Cond not in ('A','B','H','O','R','W')then delete;

/\* Delete if abnormal crossed markets \*/

if Bid>Ask then delete;

/\* Delete abnormal spreads\*/

if Spread>**5** then delete;

/\* Delete withdrawn Quotes. This is

when an exchange temporarily has no quote, as indicated by quotes

with price or depth fields containing values less than or equal to 0

or equal to '.'. See discussion in Holden and Jacobsen (2014),

page 11. \*/

if Ask le **0** or Ask =**.** then delete;

if Asksiz le **0** or Asksiz =**.** then delete;

if Bid le **0** or Bid =**.** then delete;

if Bidsiz le **0** or Bidsiz =**.** then delete;

drop Sym\_Suffix Bidex Askex Qu\_Cancel Qu\_Source RPI SSR LULD\_BBO\_CQS

LULD\_BBO\_UTP FINRA\_ADF\_MPID SIP\_Message\_ID Part\_Time RRN TRF\_Time

Spread NATL\_BBO\_LULD;

**run**;

**/\* STEP 6: CLEAN DAILY TRADES DATA - DELETE ABNORMAL TRADES \*/**

**data** trade2;

set project.DailyTrade;

where Tr\_Corr eq '00' and price gt **0**;

drop Tr\_Corr Tr\_Source TR\_RF Part\_Time RRN TRF\_Time Sym\_Suffix Tr\_SCond

Tr\_StopInd;

**run**;

**/\* STEP 7: THE NBBO FILE IS INCOMPLETE BY ITSELF (IF A SINGLE EXCHANGE HAS THE BEST BID AND OFFER, THE QUOTE IS INCLUDED IN THE QUOTES FILE, BUT NOT THE NBBO FILE). TO CREATE THE COMPLETE OFFICIAL NBBO, WE NEED TO MERGE WITH THE QUOTES FILE (SEE FOOTNOTE 6 AND 24 IN OUR PAPER) \*/**

**data** quoteAB2 (rename=(Ask=Best\_Ask Bid=Best\_Bid));

set quoteAB;

where NatBBO\_Ind='1' or NASDBBO\_Ind='4';

keep date time\_m sym\_root Qu\_SeqNum Bid Best\_BidSizeShares Ask Best\_AskSizeShares;

/\* Bid/Ask size are in round lots, replace with new shares variable

and rename Best\_BidSizeShares and Best\_AskSizeShares\*/

Best\_BidSizeShares = Bidsiz \* **100**;

Best\_AskSizeShares = Asksiz \* **100**;

**run**;

**proc** **sort** data=NBBO2;

by sym\_root date Qu\_SeqNum;

**run**;

**proc** **sort** data=quoteAB2;

by sym\_root date Qu\_SeqNum;

**run**;

**data** project.OfficialCompleteNBBO (drop=Best\_Askex Best\_Bidex);

set NBBO2 quoteAB2;

by sym\_root date Qu\_SeqNum;

**run**;

/\* If the NBBO Contains two quotes in the exact same microseond, assume

last quotes (based on sequence number) is active one \*/

**proc** **sort** data=project.OfficialCompleteNBBO;

by sym\_root date time\_m descending Qu\_SeqNum;

**run**;

**proc** **sort** data=project.OfficialCompleteNBBO nodupkey;

by sym\_root date time\_m;

**run**;

**/\* STEP 8: INTERLEAVE TRADES WITH NBBO QUOTES. DTAQ TRADES AT MICROSECOND TMMMMMM ARE MATCHED WITH THE DTAQ NBBO QUOTES STILL IN FORCE AT THE MICROSECOND TMMMMM(M-1) \*/;**

**data** project.OfficialCompleteNBBO;

set project.OfficialCompleteNBBO;type='Q';

time\_m=time\_m+**.000001**;

drop Qu\_SeqNum;

**run**;

**proc** **sort** data=project.OfficialCompleteNBBO;

by sym\_root date time\_m;

**run**;

**proc** **sort** data=trade2;

by sym\_root date time\_m Tr\_SeqNum;

**run**;

**data** project.TradesandCorrespondingNBBO;

set project.OfficialCompleteNBBO trade2;

by sym\_root date time\_m type;

**run**;

**data** project.TradesandCorrespondingNBBO

(drop=Best\_Ask Best\_Bid Best\_AskSizeShares Best\_BidSizeShares);

set project.TradesandCorrespondingNBBO;

by sym\_root date;

retain QTime NBO NBB NBOqty NBBqty;

if first.sym\_root or first.date and type='T' then do;

QTime=**.**;

NBO=**.**;

NBB=**.**;

NBOqty=**.**;

NBBqty=**.**;

end;

if type='Q' then Qtime=time\_m;

else Qtime=Qtime;

if type='Q' then NBO=Best\_Ask;

else NBO=NBO;

if type='Q' then NBB=Best\_Bid;

else NBB=NBB;

if type='Q' then NBOqty=Best\_AskSizeShares;

else NBOqty=NBOqty;

if type='Q' then NBBqty=Best\_BidSizeShares;

else NBBqty=NBBqty;

format Qtime TIME20.6;

**run**;

**/\* STEP 9: CLASSIFY TRADES AS "BUYS" OR "SELLS" USING THREE CONVENTIONS: LR = LEE AND READY (1991), EMO = ELLIS, MICHAELY, AND O'HARA (2000)AND CLNV = CHAKRABARTY, LI, NGUYEN, AND VAN NESS (2006); DETERMINE NBBO MIDPOINT AND LOCKED AND CROSSED NBBOs \*/**

**data** project.BuySellIndicators;

set project.TradesandCorrespondingNBBO;

where type='T';

midpoint=(NBO+NBB)/**2**;

if NBO=NBB then lock=**1**;else lock=**0**;

if NBO<NBB then cross=**1**;else cross=**0**;

**run**;

/\* Determine Whether Trade Price is Higher or Lower than Previous Trade

Price, or "Trade Direction" \*/

**data** project.BuySellIndicators;

set project.BuySellIndicators;

by sym\_root date;

retain direction2;

direction=dif(price);

if first.sym\_root or first.date then direction=**.**;

if direction ne **0** then direction2=direction;

else direction2=direction2;

drop direction;

**run**;

/\* First Classification Step: Classify Trades Using Tick Test \*/

**data** project.BuySellIndicators;

set project.BuySellIndicators;

if direction2>**0** then BuySellLR=**1**;

if direction2<**0** then BuySellLR=-**1**;

if direction2=**.** then BuySellLR=**.**;

if direction2>**0** then BuySellEMO=**1**;

if direction2<**0** then BuySellEMO=-**1**;

if direction2=**.** then BuySellEMO=**.**;

if direction2>**0** then BuySellCLNV=**1**;

if direction2<**0** then BuySellCLNV=-**1**;

if direction2=**.** then BuySellCLNV=**.**;

**run**;

/\* Second Classification Step: Update Trade Classification When

Conditions are Met as Specified by LR, EMO, and CLNV \*/

**data** project.BuySellIndicators;

set project.BuySellIndicators;

if lock=**0** and cross=**0** and price gt midpoint then BuySellLR=**1**;

if lock=**0** and cross=**0** and price lt midpoint then BuySellLR=-**1**;

if lock=**0** and cross=**0** and price=NBO then BuySellEMO=**1**;

if lock=**0** and cross=**0** and price=NBB then BuySellEMO=-**1**;

ofr30=NBO-**.3**\*(NBO-NBB);

bid30=NBB+**.3**\*(NBO-NBB);

if lock=**0** and cross=**0** and price le NBO and price ge ofr30

then BuySellCLNV=**1**;

if lock=**0** and cross=**0** and price le bid30 and price ge NBB

then BuySellCLNV=-**1**;

**run**;

**/\* STEP 10: CALCULATE QUOTED SRPEADS AND DEPTHS \*/**

/\* Use Quotes During Normal Market Hours \*/

**data** QSpread1;

set project.OfficialCompleteNBBO;

if time\_m lt (**"9:30:00.000000"t**) then delete;

**run**;

/\* Determine Time Each Quote is In Force - If Last Quote of Day, then Quote

is Inforce Until 4:00 pm \*/

**proc** **sort** data=QSpread1;

by sym\_root date descending time\_m;

**run**;

**data** QSpread1;

set QSpread1;

by sym\_root date;

inforce=abs(dif(time\_m));

if first.sym\_root or first.date

then inforce=max((**"16:00:00.000000"t**-time\_m),**0**);

midpoint=(Best\_Ask+Best\_Bid)/**2**;

**run**;

**proc** **sort** data=QSpread1;

by sym\_root date time\_m;

**run**;

**data** QSpread2;

set QSpread1;

/\* Delete Locked and Crossed Quotes \*/

if Best\_Ask=Best\_Bid or Best\_Ask<Best\_Bid then delete;

/\* Multiply Dollar Quoted Spread, Percent Quoted Spread, Best Dollar

Depth, and Best Share Depth by Time Inforce \*/

wQuotedSpread\_Dollar=(Best\_Ask-Best\_Bid)\*inforce;

wQuotedSpread\_Percent=(log(Best\_Ask)-log(Best\_Bid))\*inforce;

wBestOfrDepth\_Dollar=Best\_Ask\*Best\_AskSizeShares\*inforce;

wBestBidDepth\_Dollar=Best\_Bid\*Best\_BidSizeShares\*inforce;

wBestOfrDepth\_Share=Best\_AskSizeShares\*inforce;

wBestBidDepth\_Share=Best\_BidSizeShares\*inforce;

**run**;

/\* Find Average Across Firm-Day \*/

**proc** **sql**;

create table project.QuotedSpreadsandDepths

as select sym\_root,date,

sum(inforce) as sumtime,

sum(wQuotedSpread\_Dollar) as swQuotedSpread\_Dollar,

sum(wQuotedSpread\_Percent) as swQuotedSpread\_Percent,

sum(wBestOfrDepth\_Dollar) as swBestOfrDepth\_Dollar,

sum(wBestBidDepth\_Dollar) as swBestBidDepth\_Dollar,

sum(wBestOfrDepth\_Share) as swBestOfrDepth\_Share,

sum(wBestBidDepth\_Share) as swBestBidDepth\_Share

from QSpread2

group by sym\_root,date

order by sym\_root,date;

**quit**;

/\* Calculate Time-Weighted Dollar Quoted Spread, Percent Quoted Spread,

Best Dollar Depth, and Best Share Depth \*/

**data** project.QuotedSpreadsandDepths;

set project.QuotedSpreadsandDepths;

QuotedSpread\_Dollar=swQuotedSpread\_Dollar/sumtime;

QuotedSpread\_Percent=swQuotedSpread\_Percent/sumtime;

BestOfrDepth\_Dollar=swBestOfrDepth\_Dollar/sumtime;

BestBidDepth\_Dollar=swBestBidDepth\_Dollar/sumtime;

BestOfrDepth\_Share=swBestOfrDepth\_Share/sumtime;

BestBidDepth\_Share=swBestBidDepth\_Share/sumtime;

drop swQuotedSpread\_Dollar swQuotedSpread\_Percent

swBestOfrDepth\_Dollar swBestBidDepth\_Dollar

swBestOfrDepth\_Share swBestBidDepth\_Share;

**run**;

**/\* STEP 11: CALCULATE EFFECTIVE SPREADS; AGGREGATE BASED ON 3 CONVENTIONS: Ave = SIMPLE AVERAGE, DW = DOLLAR-WEIGHTED, SW = SHARE-WEIGHTED \*/**

**data** project.BuySellIndicators;

set project.BuySellIndicators;

wEffectiveSpread\_Dollar=(abs(price-midpoint))\***2**;

wEffectiveSpread\_Percent=abs(log(price)-log(midpoint))\***2**;

dollar=price\*size;

wEffectiveSpread\_Dollar\_DW=wEffectiveSpread\_Dollar\*dollar;

wEffectiveSpread\_Dollar\_SW=wEffectiveSpread\_Dollar\*size;

wEffectiveSpread\_Percent\_DW=wEffectiveSpread\_Percent\*dollar;

wEffectiveSpread\_Percent\_SW=wEffectiveSpread\_Percent\*size;

**run**;

/\* Delete Trades Associated with Locked or Crossed Best Bids or Best Offers \*/

**data** TSpread2;

set project.BuySellIndicators;

if lock=**1** or cross=**1** then delete;

**run**;

/\* Find average across firm-day \*/

**proc** **sql**;

create table project.EffectiveSpreads

as select sym\_root,date,

sum(dollar) as sumdollar,

sum(size) as sumsize,

mean(wEffectiveSpread\_Dollar) as EffectiveSpread\_Dollar\_Ave,

mean(wEffectiveSpread\_Percent) as EffectiveSpread\_Percent\_Ave,

sum(wEffectiveSpread\_Dollar\_DW) as waEffectiveSpread\_Dollar\_DW,

sum(wEffectiveSpread\_Dollar\_SW) as waEffectiveSpread\_Dollar\_SW,

sum(wEffectiveSpread\_Percent\_DW) as waEffectiveSpread\_Percent\_DW,

sum(wEffectiveSpread\_Percent\_SW) as waEffectiveSpread\_Percent\_SW

from TSpread2

group by sym\_root,date

order by sym\_root,date;

**quit**;

/\* Calculate Dollar-Weighted (DW) and Share-Weighted (SW) Dollar Effective Spreads and Percent Effective Spreads \*/

**data** project.EffectiveSpreads;

set project.EffectiveSpreads;

EffectiveSpread\_Dollar\_DW=waEffectiveSpread\_Dollar\_DW/sumdollar;

EffectiveSpread\_Dollar\_SW=waEffectiveSpread\_Dollar\_SW/sumsize;

EffectiveSpread\_Percent\_DW=waEffectiveSpread\_Percent\_DW/sumdollar;

EffectiveSpread\_Percent\_SW=waEffectiveSpread\_Percent\_SW/sumsize;

drop waEffectiveSpread\_Dollar\_DW waEffectiveSpread\_Dollar\_SW

waEffectiveSpread\_Percent\_DW aEffectiveSpread\_Percent\_SW;

**run**;

**/\* STEP 12: CALCULATE REALIZED SPREADS AND PRICE IMPACTS BASED ON THREE:CONVENTIONS: LR = LEE AND READY (1991), EMO = ELLIS, MICHAELY, AND O'HARA (2000) AND CLNV = CHAKRABARTY, LI, NGUYEN, AND VAN NESS (2006);**

**FIND THE NBBO MIDPOINT 5 MINUTES SUBSEQUENT TO THE TRADE \*/**

/\* Redefine the time variable as 5 minutes earlier (e.g., quotes at 10:00:00 are redefined as occurring at 9:55:00 in order to match to trades occurring at 9:55:00. This way we match trades occurring at time T to NBBO quotes outstanding at T+5). \*/

**data** MidQ (keep=sym\_root date type midpointnew time\_m BEST\_ASKnew

BEST\_BIDnew);

set QSpread1;

midpointnew=midpoint;

time\_m=time\_m-**300**;

Best\_AskNew=Best\_Ask;

Best\_BidNew=Best\_Bid;

**run**;

**data** MidT (keep=sym\_root date time\_M type midpoint price BuySellLR

BuySellEMO BuySellCLNV wEffectiveSpread\_Dollar size dollar);

set project.BuySellIndicators;

**run**;

**proc** **sort** data=MidQ;

by sym\_root date Time\_M type;

**run**;

**proc** **sort** data=MidT;

by sym\_root date Time\_M type;

**run**;

/\* Stack Trades at Time T with NBBO Quotes at Time T+5 \*/

**data** Mid1;

set MidT MidQ;

by sym\_root date Time\_M type;

**run**;

/\* For Each Trade at Time T, Identify the Outstanding NBBO at Time T+5 \*/

**data** Mid1;

set Mid1;

by sym\_root date;

retain midpoint5 Best\_Ask5 Best\_Bid5;

if type='Q' then midpoint5=midpointnew;

else midpoint5=midpoint5;

if type='Q' then Best\_Ask5=Best\_AskNew;else Best\_Ask5=Best\_Ask5;

if type='Q' then Best\_Bid5=Best\_BidNew;else Best\_Bid5=Best\_Bid5;

drop midpointnew Best\_AskNew Best\_BidNew;

**run**;

/\* Delete Trades at T Associated with Locked or Crossed Best Bids or Best Offers at T+5 \*/

**data** Mid1;

set Mid1;

if Best\_Ask5=Best\_Bid5 or Best\_Ask5<Best\_Bid5 then delete;

**run**;

/\* Compute Dollar and Percent Realized Spread and Price Impact for LR, EMO, and CLNV\*/

**data** Mid1;

set Mid1;

where type='T';

/\* Compute Dollar and Percent Realized Spread for LR, EMO, and CLNV \*/

wDollarRealizedSpread\_LR=BuySellLR\*(price-midpoint5)\***2**;

wDollarRealizedSpread\_EMO=BuySellEMO\*(price-midpoint5)\***2**;

wDollarRealizedSpread\_CLNV=BuySellCLNV\*(price-midpoint5)\***2**;

wPercentRealizedSpread\_LR=BuySellLR\*(log(price)-log(midpoint5))\***2**;

wPercentRealizedSpread\_EMO=BuySellEMO\*(log(price)-log(midpoint5))\***2**;

wPercentRealizedSpread\_CLNV=BuySellCLNV\*(log(price)-log(midpoint5))\***2**;

/\* Compute Dollar and Percent Price Impact for LR, EMO, and CLNV \*/

wDollarPriceImpact\_LR=BuySellLR\*(midpoint5-midpoint)\***2**;

wDollarPriceImpact\_EMO=BuySellEMO\*(midpoint5-midpoint)\***2**;

wDollarPriceImpact\_CLNV=BuySellCLNV\*(midpoint5-midpoint)\***2**;

wPercentPriceImpact\_LR=BuySellLR\*(log(midpoint5)-log(midpoint))\***2**;

wPercentPriceImpact\_EMO=BuySellEMO\*(log(midpoint5)-log(midpoint))\***2**;

wPercentPriceImpact\_CLNV=BuySellCLNV\*(log(midpoint5)-log(midpoint))\***2**;

/\* Multiply Realized Spreads and Price Impact by Dollar and Share Size of Trade for LR, EMO, and CLNV \*/

wDollarRealizedSpread\_LR\_SW=wDollarRealizedSpread\_LR\*size;

wDollarRealizedSpread\_LR\_DW=wDollarRealizedSpread\_LR\*dollar;

wPercentRealizedSpread\_LR\_SW=wPercentRealizedSpread\_LR\*size;

wPercentRealizedSpread\_LR\_DW=wPercentRealizedSpread\_LR\*dollar;

wDollarPriceImpact\_LR\_SW=wDollarPriceImpact\_LR\*size;

wDollarPriceImpact\_LR\_DW=wDollarPriceImpact\_LR\*dollar;

wPercentPriceImpact\_LR\_SW=wPercentPriceImpact\_LR\*size;

wPercentPriceImpact\_LR\_DW=wPercentPriceImpact\_LR\*dollar;

wDollarRealizedSpread\_EMO\_SW=wDollarRealizedSpread\_EMO\*size;

wDollarRealizedSpread\_EMO\_DW=wDollarRealizedSpread\_EMO\*dollar;

wPercentRealizedSpread\_EMO\_SW=wPercentRealizedSpread\_EMO\*size;

wPercentRealizedSpread\_EMO\_DW=wPercentRealizedSpread\_EMO\*dollar;

wDollarPriceImpact\_EMO\_SW=wDollarPriceImpact\_EMO\*size;

wDollarPriceImpact\_EMO\_DW=wDollarPriceImpact\_EMO\*dollar;

wPercentPriceImpact\_EMO\_SW=wPercentPriceImpact\_EMO\*size;

wPercentPriceImpact\_EMO\_DW=wPercentPriceImpact\_EMO\*dollar;

wDollarRealizedSpread\_CLNV\_SW=wDollarRealizedSpread\_CLNV\*size;

wDollarRealizedSpread\_CLNV\_DW=wDollarRealizedSpread\_CLNV\*dollar;

wPercentRealizedSpread\_CLNV\_SW=wPercentRealizedSpread\_CLNV\*size;

wPercentRealizedSpread\_CLNV\_DW=wPercentRealizedSpread\_CLNV\*dollar;

wDollarPriceImpact\_CLNV\_SW=wDollarPriceImpact\_CLNV\*size;

wDollarPriceImpact\_CLNV\_DW=wDollarPriceImpact\_CLNV\*dollar;

wPercentPriceImpact\_CLNV\_SW=wPercentPriceImpact\_CLNV\*size;

wPercentPriceImpact\_CLNV\_DW=wPercentPriceImpact\_CLNV\*dollar;

**run**;

/\* Find average across firm-day \*/

**proc** **sql**;

create table project.RealizedSpreadsandPriceImpacts

as select sym\_root,date,

sum(dollar) as sumdollar,

sum(size) as sumsize,

mean(wDollarRealizedSpread\_LR) as DollarRealizedSpread\_LR\_Ave,

mean(wDollarRealizedSpread\_EMO) as DollarRealizedSpread\_EMO\_Ave,

mean(wDollarRealizedSpread\_CLNV) as DollarRealizedSpread\_CLNV\_Ave,

mean(wPercentRealizedSpread\_LR) as PercentRealizedSpread\_LR\_Ave,

mean(wPercentRealizedSpread\_EMO) as PercentRealizedSpread\_EMO\_Ave,

mean(wPercentRealizedSpread\_CLNV) as PercentRealizedSpread\_CLNV\_Ave,

mean(wDollarPriceImpact\_LR) as DollarPriceImpact\_LR\_Ave,

mean(wDollarPriceImpact\_EMO) as DollarPriceImpact\_EMO\_Ave,

mean(wDollarPriceImpact\_CLNV) as DollarPriceImpact\_CLNV\_Ave,

mean(wPercentPriceImpact\_LR) as PercentPriceImpact\_LR\_Ave,

mean(wPercentPriceImpact\_EMO) as PercentPriceImpact\_EMO\_Ave,

mean(wPercentPriceImpact\_CLNV) as PercentPriceImpact\_CLNV\_Ave,

sum(wDollarRealizedSpread\_LR\_SW) as waDollarRealizedSpread\_LR\_SW,

sum(wDollarRealizedSpread\_LR\_DW) as waDollarRealizedSpread\_LR\_DW,

sum(wPercentRealizedSpread\_LR\_SW) as waPercentRealizedSpread\_LR\_SW,

sum(wPercentRealizedSpread\_LR\_DW) as waPercentRealizedSpread\_LR\_DW,

sum(wDollarPriceImpact\_LR\_SW) as waDollarPriceImpact\_LR\_SW,

sum(wDollarPriceImpact\_LR\_DW) as waDollarPriceImpact\_LR\_DW,

sum(wPercentPriceImpact\_LR\_SW) as waPercentPriceImpact\_LR\_SW,

sum(wPercentPriceImpact\_LR\_DW) as waPercentPriceImpact\_LR\_DW,

sum(wDollarRealizedSpread\_EMO\_SW) as waDollarRealizedSpread\_EMO\_SW,

sum(wDollarRealizedSpread\_EMO\_DW) as waDollarRealizedSpread\_EMO\_DW,

sum(wPercentRealizedSpread\_EMO\_SW) as waPercentRealizedSpread\_EMO\_SW,

sum(wPercentRealizedSpread\_EMO\_DW) as waPercentRealizedSpread\_EMO\_DW,

sum(wDollarPriceImpact\_EMO\_SW) as waDollarPriceImpact\_EMO\_SW,

sum(wDollarPriceImpact\_EMO\_DW) as waDollarPriceImpact\_EMO\_DW,

sum(wPercentPriceImpact\_EMO\_SW) as waPercentPriceImpact\_EMO\_SW,

sum(wPercentPriceImpact\_EMO\_DW) as waPercentPriceImpact\_EMO\_DW,

sum(wDollarRealizedSpread\_CLNV\_SW) as waDollarRealizedSpread\_CLNV\_SW,

sum(wDollarRealizedSpread\_CLNV\_DW) as waDollarRealizedSpread\_CLNV\_DW,

sum(wPercentRealizedSpread\_CLNV\_SW) as waPercentRealizedSpread\_CLNV\_SW,

sum(wPercentRealizedSpread\_CLNV\_DW) as waPercentRealizedSpread\_CLNV\_DW,

sum(wDollarPriceImpact\_CLNV\_SW) as waDollarPriceImpact\_CLNV\_SW,

sum(wDollarPriceImpact\_CLNV\_DW) as waDollarPriceImpact\_CLNV\_DW,

sum(wPercentPriceImpact\_CLNV\_SW) as waPercentPriceImpact\_CLNV\_SW,

sum(wPercentPriceImpact\_CLNV\_DW) as waPercentPriceImpact\_CLNV\_DW

from Mid1

group by sym\_root,date

order by sym\_root,date;

**quit**;

/\* Calculate Dollar-Weighted (DW) and Share-Weighted (SW) Realized Spreads

and Price Impact \*/

**data** project.RealizedSpreadsandPriceImpacts;

set project.RealizedSpreadsandPriceImpacts;

DollarRealizedSpread\_LR\_SW=waDollarRealizedSpread\_LR\_SW/sumsize;

DollarRealizedSpread\_LR\_DW=waDollarRealizedSpread\_LR\_DW/sumdollar;

PercentRealizedSpread\_LR\_SW=waPercentRealizedSpread\_LR\_SW/sumsize;

PercentRealizedSpread\_LR\_DW=waPercentRealizedSpread\_LR\_DW/sumdollar;

DollarPriceImpact\_LR\_SW=waDollarPriceImpact\_LR\_SW/sumsize;

DollarPriceImpact\_LR\_DW=waDollarPriceImpact\_LR\_DW/sumdollar;

PercentPriceImpact\_LR\_SW=waPercentPriceImpact\_LR\_SW/sumsize;

PercentPriceImpact\_LR\_DW=waPercentPriceImpact\_LR\_DW/sumdollar;

DollarRealizedSpread\_EMO\_SW=waDollarRealizedSpread\_EMO\_SW/sumsize;

DollarRealizedSpread\_EMO\_DW=waDollarRealizedSpread\_EMO\_DW/sumdollar;

PercentRealizedSpread\_EMO\_SW=waPercentRealizedSpread\_EMO\_SW/sumsize;

PercentRealizedSpread\_EMO\_DW=waPercentRealizedSpread\_EMO\_DW/sumdollar;

DollarPriceImpact\_EMO\_SW=waDollarPriceImpact\_EMO\_SW/sumsize;

DollarPriceImpact\_EMO\_DW=waDollarPriceImpact\_EMO\_DW/sumdollar;

PercentPriceImpact\_EMO\_SW=waPercentPriceImpact\_EMO\_SW/sumsize;

PercentPriceImpact\_EMO\_DW=waPercentPriceImpact\_EMO\_DW/sumdollar;

DollarRealizedSpread\_CLNV\_SW=waDollarRealizedSpread\_CLNV\_SW/sumsize;

DollarRealizedSpread\_CLNV\_DW=waDollarRealizedSpread\_CLNV\_DW/sumdollar;

PercentRealizedSpread\_CLNV\_SW=waPercentRealizedSpread\_CLNV\_SW/sumsize;

PercentRealizedSpread\_CLNV\_DW=waPercentRealizedSpread\_CLNV\_DW/sumdollar;

DollarPriceImpact\_CLNV\_SW=waDollarPriceImpact\_CLNV\_SW/sumsize;

DollarPriceImpact\_CLNV\_DW=waDollarPriceImpact\_CLNV\_DW/sumdollar;

PercentPriceImpact\_CLNV\_SW=waPercentPriceImpact\_CLNV\_SW/sumsize;

PercentPriceImpact\_CLNV\_DW=waPercentPriceImpact\_CLNV\_DW/sumdollar;

drop waDollarRealizedSpread\_LR\_SW waDollarRealizedSpread\_LR\_DW

waPercentRealizedSpread\_LR\_SW waPercentRealizedSpread\_LR\_DW

waDollarPriceImpact\_LR\_SW waDollarPriceImpact\_LR\_DW

waPercentPriceImpact\_LR\_SW waPercentPriceImpact\_LR\_DW

waDollarRealizedSpread\_EMO\_SW waDollarRealizedSpread\_EMO\_DW

waPercentRealizedSpread\_EMO\_SW waPercentRealizedSpread\_EMO\_DW

waDollarPriceImpact\_EMO\_SW waDollarPriceImpact\_EMO\_DW

waPercentPriceImpact\_EMO\_SW waPercentPriceImpact\_EMO\_DW

waDollarRealizedSpread\_CLNV\_SW waDollarRealizedSpread\_CLNV\_DW

waPercentRealizedSpread\_CLNV\_SW waPercentRealizedSpread\_CLNV\_DW

waDollarPriceImpact\_CLNV\_SW waDollarPriceImpact\_CLNV\_DW

waPercentPriceImpact\_CLNV\_SW waPercentPriceImpact\_CLNV\_DW;

**run**;