Stevens Institute of Technology

Market Microstructure - Final Project

Liquidity Measures and Daily Volatility Estimate

Arjun Koshal, Ethan Li,
Skanda Srikkanth, Mark Wong

December 12, 2022

1 The Microstructure Dataset

Cryptocurrencies are a type of digital decentralized currency and not controlled by any government. The history dates back to the 1980s, more precisely 1989, when cryptocurrencies were called cyber currencies. In the early 1990s cryptographic protocols, as well as software, began to be developed that would enable the creation of a truly decentralized digital currency. In 2008, Satoshi Nakamoto published a paper that introduced a system, which would create a digital currency that had no involvement in any third party. One year after, Bitcoin was launched and became the highest market captialization in the crypto market. Ethereum was launched 6 years after Bitcoin in 2015 and became the number two cryptocurrency in terms of market share.

Ethereum is an open-source public service that uses blockchain technology to facilitate smart contracts and cryptocurrency trading securely without a third party. The Ethereum blockchain quickly rose to become the second largest cryptocurrency by market capitalization in January 2018. While being a decentralized and open-source blockchain like many other crypto assets, it stood out from the crowd by using something called Smart Contracts. Smart Contracts can execute automatic actions if certain pre-set conditions are met. These allow the Ethereum blockchain to run an entire ecosystem on its blockchain while also hosting its own native currency: Ether (ETH).

The analysis conducted uses data from crypto assets, which contain crypto futures and perpetual swaps). The site used was `https://public.bitmex.com/`. We created a TAQ dataset `tq_data.RDATA` which covers the quotes and trade data for October 03, 2021. We chose to focus our research on ETHUSD (Ethereum), which contained n = 48,265 trades. We select October 03, 2021, as there were no news of a market crash or any underlying circumstances that would affect our study. We aim to study Ethereum on a normal day.

```
head(tq_data)
```

DATE	TIMESTAMP	SYMBOL	BIDSIZE	BID	OFR	OFRSIZE	SIDE	SIZE	PRICE	TICKDIRECTION	GROSSVALUE
2021-10-03	00:00:00.607983	ETHUSD	150	3389.50	3389.55	23	Buy	12	3389.55	PlusTick	4067460
2021-10-03	00:00:00.607983	ETHUSD	150	3389.50	3389.55	23	Buy	7	3389.55	ZeroPlusTick	2372685
2021-10-03	00:00:01.845111	ETHUSD	298	3388.10	3389.55	127	Sell	8	3389.50	MinusTick	2711600
2021-10-03	00:00:01.845111	ETHUSD	298	3388.10	3389.55	127	Sell	37	3389.50	ZeroMinusTick	12541150
2021-10-03	00:00:01.845111	ETHUSD	298	3388.10	3389.55	127	Sell	1	3389.50	ZeroMinusTick	338950
2021-10-03	00:00:01.845111	ETHUSD	298	3388.10	3389.55	127	Sell	104	3388.70	MinusTick	35242480

2 Study of Liquidity

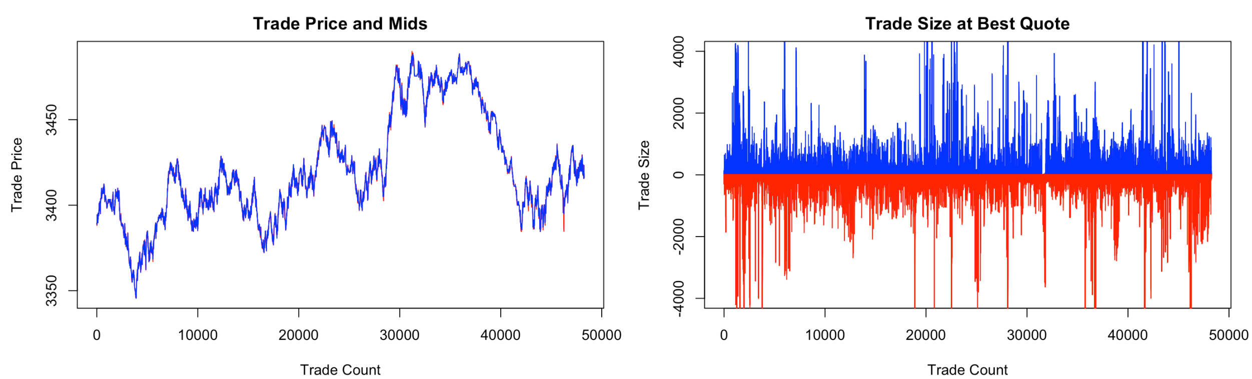
Liquidity is the property of the markets which allows rapid and cheap trade execution. It is a critical characteristic for well-functioning markets. Liquidity has several dimensions, which are time (trade execution), size (can trade large amounts of stock), and cost (transaction costs).

The main liquidity measures used in microstructure are spread-based measures:

- Quoted Spread - Difference between the best ask price and best price bid price.
- Effective Spread - Difference between trading price and midpoint price.
- Realized Spread - Difference between trading price and midpoint price at the next time step.

Using the TAQ dataset that we created, we perform the following analysis on the dataset.

1. Use `getLiquidityMeasures` to estimate the *quotedSpread*, *effectiveSpread*, *realizedSpread*, and *priceImpact*.
2. Study the intra-day liquidity dynamics and compute the quoted and effective spreads for each 1-hour interval of the whole day. Plot these results and comment on the findings.
3. Calculate the Roll's estimate of the bid-ask spread.
4. Use `getTradeDirection` to relax the assumption that the trade signs are uncorrelated and improve the Roll's estimate of the bid-ask spread.
5. Estimate the daily volatility using both the Roll's estimate and sampling at frequency q in trading time (using every q-th trade).



(a) Plot of trade prices (blue) and mid-prices (red) (b) Quote size at best-bid and best-ask prices

Figure 1: Plot of Trade Price and Trade Size

3 Quoted Spread, Effective Spread, and Realized Spread

The three main liquidity measures that our analysis will focus on are Quoted Spread, Effective Spread, and Realized Spread.

Quoted Spread is the simplest type of bid-ask spread, where the spread is taken directly from quotes. The spread is the difference between the lowest ask price, which is the lowest price at which someone will sell and the highest bid price, which is the highest someone will buy.

$$\text{QS} = a_t - b_t$$

Effective Spread is a measure of trading costs, where it is taken as the difference between the price at which a market order is executed and the mid-price. It provides an overall estimate of the cost of trading with the mid-price as a benchmark price. Effective Spread measures the immediate payoff realized by the liquidity provider.

$$\text{ES} = 2 \cdot q_t \cdot (p_t - m_t)$$

Realized Spread is also a measure of trading cost and is similar to Effective Spread, with the exception that Realized Spread has delayed mid-prices. It accounts for mid-price changes when trading, so the mid-price should be taken at a later time, $t + \Delta$, which is known as lagging. Realized Spread measures the actual payoff realized by the liquidity provider because it accounts for the price impact, which is the private information contained in trades, or the amount incurred by the liquidity provider.

$$\text{RS}_\Delta = 2 \cdot q_t \cdot (p_t - m_{t+\Delta})$$
$$\text{Price Impact}_\Delta = 2 \cdot q_t \cdot (m_{t+\Delta} - m_t)$$

Comparing Effective Spread and Realized Spread, from the definitions above, we can state that the Effective Spread will usually be greater than the Realized Spread since the Realized Spread accounts for the price impact. Therefore,

$$\text{ES} = \text{RS}_\Delta + \text{Price Impact}_\Delta$$

The formulas for the mean of the types of bid-ask spreads are shown below:

1. Quoted Spread is defined in terms of quotes, best-ask a_t and best-bid b_t prices:

$$\text{QS} = \mathbb{E}([a_t - b_t]) = \frac{1}{n}\sum_{t=1}^{n} a_t - b_t$$

2. Effective Spread takes into account trade prices, where p_t is the trade price and q_t is the trade indicator ($q_t = \pm$ for buy/sell), and the mid-price $m_t = \frac{1}{2}(a_t + b_t)$:

$$\text{ES} = \mathbb{E}([2 \cdot q_t \cdot (p_t - m_t)]) = \frac{1}{n}\sum_{t=1}^{n} 2 \cdot q_t \cdot (p_t - m_t)$$

3. Realized Spread is defined with delayed mid-prices and takes into account mid-price changes due to trading, where Δ = delay:

$$\text{RS} = \mathbb{E}([2 \cdot q_t \cdot (p_t - m_{t+\Delta})]) = \frac{1}{n}\sum_{t=1}^{n} 2 \cdot q_t \cdot (p_t - m_{t+\Delta})$$

We can use `getLiquidityMeasures` to estimate the Quoted Spread, Effective Spread, Realized Spread (by 300 seconds), and Price Impact for our TAQ dataset.

```
liq.measures <- getLiquidityMeasures(tq_data)

# Quoted Spread
qs <- mean(as.numeric(liq.measures$quotedSpread))
qs

0.704641

# Effective Spread
es <- mean(as.numeric(liq.measures$effectiveSpread))
es

0.5409696

# Realized Spread
rs <- mean(na.omit(as.numeric(liq.measures$realizedSpread)))
rs

0.257715

# Price Impact
price_impact <- mean(na.omit(as.numeric(liq.measures$priceImpact)))
price_impact

0.1418727
```

From `getLiquidityMeasures`, Price Impact = $\frac{\text{Effective Spread}-\text{Realized Spread}}{2}$. We can verify that this holds true as,

$$\frac{0.5409696 - 0.257715}{2} = 0.141.$$

4 Intraday Liquidity Dynamics

To analyze the intraday liquidity, we can compute both the quoted spread and effective spread based on 1-hour intervals. Since cryptocurrency can be traded 24 hours a day by investors around the world, we have to analyze all trades executed from 12 AM until the next day. The data collected was in GMT, so for our study, we converted to EST, which is 5 hours behind GMT.

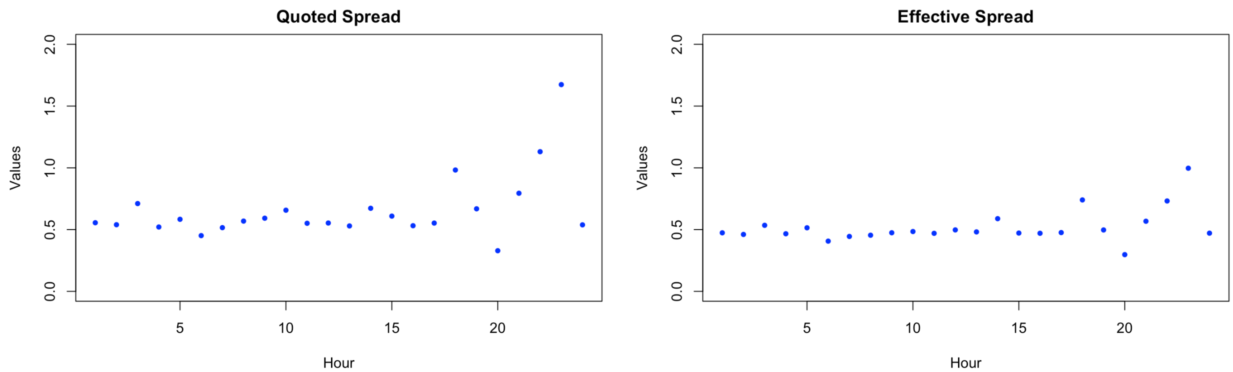


Figure 2: Intra-day Liquidity Dynamics. The plots show the quoted and effective spread each hour for Ethereum.

We can see that ETHUSD's average spread is highest during the end of the day and lies at 0.5 throughout the day. We know that tighter spreads are a sign of greater liquidity, however, we can see that cryptocurrency has wider spreads. This implies that the cryptocurrency market is highly volatile and manual day trading is impossible. Automated cryptocurrency trading has several advantages over manual trading: specifically the fact that bots can work efficiently without rest.

5 Roll Model Estimate

Another bid-ask spread measure is the Roll model estimate. Given market efficiency, the effective bid-ask spread is measured by

$$\text{Spread} = 2\sqrt{-\text{cov}}$$

where cov is the first-order serial covariance of price changes. This implicit measure of the bid-ask spread is derived below.

We first assume the following:

- Buys and sell are equally likely and occur with probability $\frac{1}{2}$: $\mathbb{E}([d_t]) = 0$.
- No autocorrelation in orders: $\mathbb{E}([d_t d_s]) = 0$, $t \neq s$.
- Trading and the efficient price process are uncorrelated: $\mathbb{E}([d_t u_t]) = 0$.
- Constant and zero expected return: $\mathbb{E}([u_t]) = 0$.
- The efficient price is m_t and $m_t = m_{t-1} + u_t$, where u_t are independently and identically distributed zero-mean random variables.
- Dealers incur a cost of c per trade, which reflects costs like clearing fees and per-trade allocations of fixed costs.

The trade prices observed at a micro-scale consist of two components.

- The trade price has a slow-moving component, m_t, which is the efficient price. This is the fundamental security value and embeds information about the future earnings of the stock.
- The trade price has a trade direction indicator which shows if a trade is a buy or a sale. This is "noise" due to trading activity.

The trade prices p_t are expressed as

$$p_t = m_t + cd_t$$

where d_t is the trade direction indicator: $d_t = +1$ if the trade is a buy and $d_t = -1$ if the trade is a sale.

The Roll model has two parameters, which are c and $\sigma_u^2 = \text{var}(\Delta m_t) = \text{var}(u_t)$.

The price changes have MA(1) structure

$$\Delta p_t = p_t - p_{t-1} = u_t + c(d_t - d_{t-1})$$

Since the price changes follow a MA(1) model, all covariances with lag more than 2 for Δp_t vanish. We can plot the autocorrelation of price changes for Ethereum to verify that this holds true.

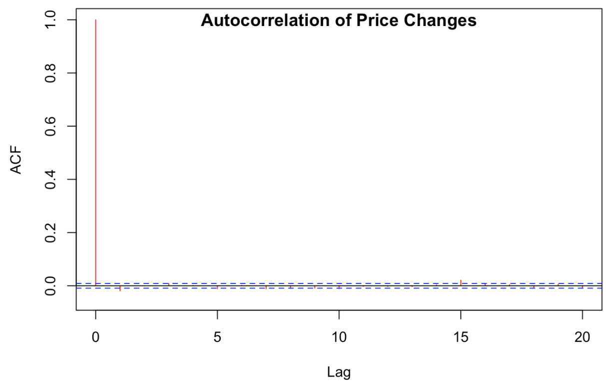


Figure 3: The plot shows the auto-correlation $\rho(k) = \mathrm{corr}(\Delta P_i, \Delta P_{i+k})$ of the successive trade price changes $\Delta P_i = P_i - P_{i-1}$ for ETHUSD. There is a negative autocorrelation at lag 1, and zero at higher lags.

From Figure 3, we can confirm that the price changes have all zero auto-covariances of order 2 or higher.

We can define the auto-covariances of order 0 and order 1.

Variance:

$$\begin{aligned}
\gamma_0 &= \mathrm{var}(\Delta p_t) = \mathbb{E}([(\Delta p_t)^2]) - \mathbb{E}([\Delta p_t])^2 \\
&= \mathbb{E}([(u_t + c(d_t - d_{t-1}))^2]) - \mathbb{E}([u_t + c(d_t - d_{t-1})])^2 \\
&= \mathbb{E}([u_t^2 + c^2 d_t^2 - 2c^2 d_t d_{t-1} + c^2 d_{t-1}^2 - 2d_{t-1} u_t c + 2 d_t u_t c]) \\
&= \mathrm{var}(u_t) + 2c^2 \\
&= \sigma_u^2 + 2c^2.
\end{aligned}$$

First Order Covariance:

$$\begin{aligned}
\gamma_1 &= Cov(\Delta p_{t-1}, \Delta p_t)^2 = \mathbb{E}([\Delta p_{t-1} \Delta p_t]) \\
&= \mathbb{E}([c^2(d_{t-2}d_{t-1} - d_{t-1}^2 - d_{t-2}d_t + d_{t-1}d_t) + c(d_t u_{t-1} - d_{t-1}u_{t-1} + u_t d_{t-1} - u_t d_{t-2})]) \\
&= -c^2
\end{aligned}$$

The model parameters are then:

$$c = \sqrt{-\gamma_1}$$
$$\sigma_u^2 = \gamma_0 + 2\gamma_1$$

We can then calibrate the Roll model and estimate the bid-ask spread.

```
covpr <- acf(dpr, lag.max=20, type="covariance", plot=FALSE)


# Variance
```

```
gamma0 <- sd(dpr)^2
gamma0

0.3291168

# First Order Covariance
gamma1 <- covpr$acf[2]
gamma1

-0.006268989

# c paramter
cparam <- sqrt(-covpr$acf[2])
cparam

0.07917695

# Variance of u_t
sig2u <- gamma0 + 2*gamma1
sigu <- sqrt(sig2u)
sigu

0.5626534
```

From our values obtained above, we can see that $c \approx 0.08$ so the Roll estimate is $2c = 0.16$. We can also plug γ_0 and γ_1 into the equations for our model parameters.

$$
\begin{aligned}
c &= \sqrt{-\gamma_1} & \sigma_u^2 &= \gamma_0 + 2\gamma_1 \\
c &= \sqrt{0.006268989} & \sigma_u^2 &= 0.3291168 + 2 \cdot (-0.006268989) \\
c &= 0.07917 & \sigma_u &= 0.56265
\end{aligned}
$$

6 Improved Roll Model Estimate

The Roll model assumes uncorrelated trade signs, which is often not the case. To account for this assumption, we can improve the Roll model by using the trade signs from the Lee-Ready rule. The Lee-Ready rule uses both trade prices p_t and quotes a_t and b_t. The Lee-Ready rule confirms if a trade is a buy or sale by comparing p_t with the mid-price $m_t = \frac{1}{2}(a_t + b_t)$. If $p_t > m_t$, then the trade direction is +1 (buy), if $p_t < m_t$, then the trade direction is -1 (sale), and if $p_t = m_t$, then we use the tick rule. The tick rule uses only the trade prices. If $p_t > p_{t-1}$ (uptick) or $p_t = p_{t-1} > p_{t-2}$ (zero-uptick), then $d_t = +1$ (buy) and if $p_t < p_{t-1}$ (downtick) or $p_t = p_{t-1} < p_{t-2}$ (zero-downtick), then $d_t = -1$ (sale).

```
tradeSigns <- getTradeDirection(tq_data)
acTS <- acf(tradeSigns, main="ACF trade signs")
```

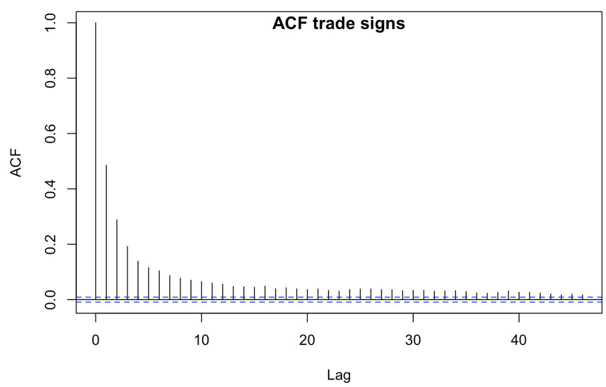


Figure 4: The plot shows the ACF Trade Signs using the Lee-Ready rule.

```
p <- as.numeric(tq_data$PRICE)
dp <- diff(p)
deps <- diff(tradeSigns)
mids <- (as.numeric(tq_data$OFR) + as.numeric(tq_data$BID))/2
dm <- diff(mids)
(fit.lm <- lm(dp ~ dm + deps))
fit.lm$coeff[3]

0.2428599
```

This gives $c = 0.2428599$, therefore the spread ≈ 0.49.

7 Estimation of Daily Volatility

We can draw the signature plot, where the signature plot is a function of the realized variance of the sampling frequency, q.

```
realizedVar <- function(q){rCov(diff(p, lag=q, differences=1))/q}

rv_data <- NULL
for(q in 1:200){
  rv_data <- c(rv_data, realizedVar(q))
}
```

One method of estimating the daily volatility depends on the sampling frequency. The dependence of the volatility estimate on the lag (q) is the signature plot, which is shown below.

```
# q5min is number of trades per 5 mins
# 1440 minutes in 24 hours

q5min <- n.trades*5/1440
rv5 = realizedVar(q5min)
rv5

14325.64

sqrt(rv5)

119.6897
```

We can also use Roll's model estimate to calculate the daily volatility.

```
rvRoll <- sig2u*n.trades
rvRoll

15279.68

sigRoll <- sqrt(sig2u*n.trades)
sigRoll

123.611
```

We can plot both the Roll's estimate and the sampling at frequency q using every q-th trade.

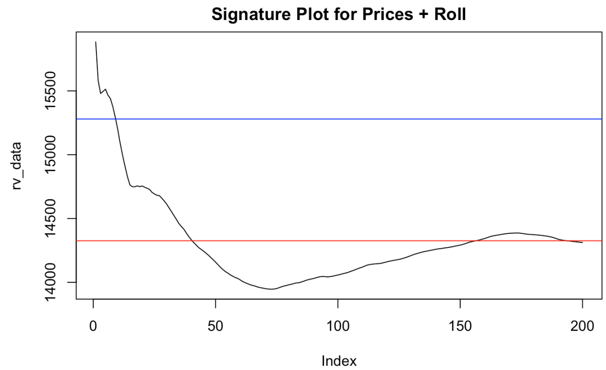


Figure 5: Signature Plot for Prices with Both Volatility Estimates

The volatility for both estimates is greater than 100%, which implies the wild gyrations that ETHUSD follows. This is similar for all cryptocurrencies because of the influence of supply and demand, investor and user sentiments, government regulations, and media hype. All of these factors work together to create a high volatility. The high volatility can also be attributed to the fact that we used the prices of ETHUSD and not the log prices.

8 Conclusions and Discussion

Let us compare the estimates of the bid-ask spread obtained from the different methods used. We can also comment on our findings for the daily volatility estimation and compare them to the general trends of intraday trading with cryptocurrency.

We can create a table of the bid-ask spread as illustrated below.

Method	Spread	Comments
Quoted Spread	0.70 cents	uses quotes
Effective Spread	0.54 cents	takes into account trade prices
Realized Spread	0.26 cents	takes into account mid price changes
Roll model	0.16 cents	assumes uncorrelated trade signs
Improved Roll model	0.49 cents	use trade signs from Lee, Ready

From the table we see that the Roll model with the assumption of uncorrelated trade signs is extremely low compared to the other measures. From Section 7 when we plot the ACF of the trade signs, we see that the trade signs are extremely correlated. By using the trade signs from the Lee-Ready rule, we see that the improved roll model estimate is closer to the effective spread estimate.

We also notice that the realized spread is smaller than both the quoted and effective spread. This is attributed to the fact that the realized spread accounts for the mid price changes. When computing the realized spread, we lagged the time by 300 seconds. This means that after 300 seconds in the trading day, we take the mid-price and calculate the spread. This accounts for the price impact and should naturally be smaller compared to the effective spread, since realized spread measures the actual payoff realized by the liquidity provider. The difference between the effective spread and realized spread is known as the price impact, which we verified the relationship in Section 3.

For the daily volatility estimates, we drew the signature plot and used Roll's model estimate. For the signature plot, we used a sampling frequency every 5 minutes. At the highest sampling frequency (lag = 1), the estimated volatility is the highest. This is due to the bid-ask bounce noise

$$\widehat{\sigma}(q) = \frac{1}{\Delta t}\mathrm{Var}(\epsilon) + \sigma_u^2$$

The bid-ask bounce averages out at larger lags, and at lag q $\approx$ 50, the estimated volatility is due only to fluctuations of the efficient price. Therefore

$$\widehat{\sigma}(q) \approx \sigma_u$$

This is a realistic estimate of the stock price volatility. We can see from Figure 5 that approximately at q = 50, the sampling frequency and Roll's estimate are almost equal.

Overall our results demonstrate the use of liquidity measures and daily volatility estimation. We can validate the fact that Ethereum has a very a volatile price based on the volatility estimation metrics and Ethereum is relatively liquid from our intraday liquidity dynamics study throughout the daytime, as opposed to later times.

9 R code for analysis

```
library(tidyverse)
library(lubridate)
library(xts)
library(highfrequency)

q_data <- read.csv(file = '20211003-quote.csv')
t_data <- read.csv(file = '20211003-trade.csv')

# Data cleaning
q_data$timestamp <- gsub('D', ' ', q_data$timestamp)
t_data$timestamp <- gsub('D', ' ', t_data$timestamp)

names(q_data)[names(q_data) == "bidPrice"] <- "BID"
names(q_data)[names(q_data) == "askPrice"] <- "OFR"
names(q_data)[names(q_data) == "symbol"] <- "SYMBOL"

names(t_data)[names(t_data) == "bidPrice"] <- "BID"
names(t_data)[names(t_data) == "askPrice"] <- "OFR"
names(t_data)[names(t_data) == "symbol"] <- "SYMBOL"

q_data.xts <- xts(q_data[,],
                  order.by=as.POSIXct(q_data$timestamp, format = "%Y-%m-%d %H:%M:%OS"))


t_data.xts <- xts(t_data[,],
                  order.by=as.POSIXct(t_data$timestamp, format = "%Y-%m-%d %H:%M:%OS"))

tq_data <- matchTradesQuotes(t_data.xts, q_data.xts)

tq_data <- tq_data[tq_data$SYMBOL == "ETHUSD", ]

# Number of trades
n.trades <- length(tq_data$SIZE)
n.trades

prices <- as.numeric(tq_data$PRICE)
asks <- as.numeric(tq_data$OFR)
bids <- as.numeric(tq_data$BID)
mids <- 0.5*bids + 0.5*asks

# Plot trade price and mids
plot(prices, type="l", col="red", main="Trade Price and Mids",
     xlab="Trade Count",
     ylab="Trade Price")
lines(mids, type="l", col="blue")

ask.SIZE <- as.numeric(tq_data$ASKSIZE)
bid.SIZE <- -1*as.numeric(tq_data$BIDSIZE)
LOB.imbalance <- ask.SIZE + bid.SIZE

# Plot trade size at best quote
plot(ask.SIZE,col="blue", type="h",
```

```
     ylab="Trade Size",
     xlab="Trade Count", main="Trade Size at Best Quote", ylim=c(-4000,4000))
lines(bid.SIZE, col="red", type="h")

# Liquidity measures for entire TQ dataset
liq.measures <- getLiquidityMeasures(tq_data)

# Quoted spread
qs <- mean(as.numeric(liq.measures$quotedSpread))
qs

# Effective spread
es <- mean(as.numeric(liq.measures$effectiveSpread))
es

# Realized spread
rs <- mean(na.omit(as.numeric(liq.measures$realizedSpread)))
rs

# Price impact
price_impact <- mean(na.omit(as.numeric(liq.measures$priceImpact)))
price_impact

# Slice data by each hour
tq_data_1 <- tq_data["T00:00/T01:00"]
tq_data_2 <- tq_data["T01:00/T02:00"]
tq_data_3 <- tq_data["T02:00/T03:00"]
tq_data_4 <- tq_data["T03:00/T04:00"]
tq_data_5 <- tq_data["T04:00/T05:00"]
tq_data_6 <- tq_data["T05:00/T06:00"]
tq_data_7 <- tq_data["T06:00/T07:00"]
tq_data_8 <- tq_data["T07:00/T08:00"]
tq_data_9 <- tq_data["T08:00/T09:00"]
tq_data_10 <- tq_data["T09:00/T10:00"]
tq_data_11 <- tq_data["T10:00/T11:00"]
tq_data_12 <- tq_data["T11:00/T12:00"]
tq_data_13 <- tq_data["T12:00/T13:00"]
tq_data_14 <- tq_data["T13:00/T14:00"]
tq_data_15 <- tq_data["T14:00/T15:00"]
tq_data_16 <- tq_data["T15:00/T16:00"]
tq_data_17 <- tq_data["T16:00/T17:00"]
tq_data_18 <- tq_data["T17:00/T18:00"]
tq_data_19 <- tq_data["T18:00/T19:00"]
tq_data_20 <- tq_data["T19:00/T20:00"]
tq_data_21 <- tq_data["T20:00/T21:00"]
tq_data_22 <- tq_data["T21:00/T22:00"]
tq_data_23 <- tq_data["T22:00/T23:00"]
tq_data_24 <- tq_data["T23:00/T23:59:45.352204000"]

# Obtain liquidity measures for each hour
liq.1 <- getLiquidityMeasures(tq_data_1)
liq.2 <- getLiquidityMeasures(tq_data_2)
liq.3 <- getLiquidityMeasures(tq_data_3)
liq.4 <- getLiquidityMeasures(tq_data_4)
liq.5 <- getLiquidityMeasures(tq_data_5)
```

```
liq.6 <- getLiquidityMeasures(tq_data_6)
liq.7 <- getLiquidityMeasures(tq_data_7)
liq.8 <- getLiquidityMeasures(tq_data_8)
liq.9 <- getLiquidityMeasures(tq_data_9)
liq.10 <- getLiquidityMeasures(tq_data_10)
liq.11 <- getLiquidityMeasures(tq_data_11)
liq.12 <- getLiquidityMeasures(tq_data_12)
liq.13 <- getLiquidityMeasures(tq_data_13)
liq.14 <- getLiquidityMeasures(tq_data_14)
liq.15 <- getLiquidityMeasures(tq_data_15)
liq.16 <- getLiquidityMeasures(tq_data_16)
liq.17 <- getLiquidityMeasures(tq_data_17)
liq.18 <- getLiquidityMeasures(tq_data_18)
liq.19 <- getLiquidityMeasures(tq_data_19)
liq.20 <- getLiquidityMeasures(tq_data_20)
liq.21 <- getLiquidityMeasures(tq_data_21)
liq.22 <- getLiquidityMeasures(tq_data_22)
liq.23 <- getLiquidityMeasures(tq_data_23)
liq.24 <- getLiquidityMeasures(tq_data_24)

# Quoted spread for each hour
qs.hr <- c()
qs.hr[1] <- mean(as.numeric(liq.1$quotedSpread))
qs.hr[2] <- mean(as.numeric(liq.2$quotedSpread))
qs.hr[3] <- mean(as.numeric(liq.3$quotedSpread))
qs.hr[4] <- mean(as.numeric(liq.4$quotedSpread))
qs.hr[5] <- mean(as.numeric(liq.5$quotedSpread))
qs.hr[6] <- mean(as.numeric(liq.6$quotedSpread))
qs.hr[7] <- mean(as.numeric(liq.7$quotedSpread))
qs.hr[8] <- mean(as.numeric(liq.8$quotedSpread))
qs.hr[9] <- mean(as.numeric(liq.9$quotedSpread))
qs.hr[10] <- mean(as.numeric(liq.10$quotedSpread))
qs.hr[11] <- mean(as.numeric(liq.11$quotedSpread))
qs.hr[12] <- mean(as.numeric(liq.12$quotedSpread))
qs.hr[13] <- mean(as.numeric(liq.13$quotedSpread))
qs.hr[14] <- mean(as.numeric(liq.14$quotedSpread))
qs.hr[15] <- mean(as.numeric(liq.15$quotedSpread))
qs.hr[16] <- mean(as.numeric(liq.16$quotedSpread))
qs.hr[17] <- mean(as.numeric(liq.17$quotedSpread))
qs.hr[18] <- mean(as.numeric(liq.18$quotedSpread))
qs.hr[19] <- mean(as.numeric(liq.19$quotedSpread))
qs.hr[20] <- mean(as.numeric(liq.20$quotedSpread))
qs.hr[21] <- mean(as.numeric(liq.21$quotedSpread))
qs.hr[22] <- mean(as.numeric(liq.22$quotedSpread))
qs.hr[23] <- mean(as.numeric(liq.23$quotedSpread))
qs.hr[24] <- mean(as.numeric(liq.24$quotedSpread))

plot(1:24, qs.hr, type="p", pch=20, col="blue", main="Quoted Spread",
     xlab = "Hour", ylab = "Values", ylim = c(0,2.0))

# Effective spread for each hour
es.hr <- c()
es.hr[1] <- mean(as.numeric(liq.1$effectiveSpread))
es.hr[2] <- mean(as.numeric(liq.2$effectiveSpread))
es.hr[3] <- mean(as.numeric(liq.3$effectiveSpread))
```

```
es.hr[4] <- mean(as.numeric(liq.4$effectiveSpread))
es.hr[5] <- mean(as.numeric(liq.5$effectiveSpread))
es.hr[6] <- mean(as.numeric(liq.6$effectiveSpread))
es.hr[7] <- mean(as.numeric(liq.7$effectiveSpread))
es.hr[8] <- mean(as.numeric(liq.8$effectiveSpread))
es.hr[9] <- mean(as.numeric(liq.9$effectiveSpread))
es.hr[10] <- mean(as.numeric(liq.10$effectiveSpread))
es.hr[11] <- mean(as.numeric(liq.11$effectiveSpread))
es.hr[12] <- mean(as.numeric(liq.12$effectiveSpread))
es.hr[13] <- mean(as.numeric(liq.13$effectiveSpread))
es.hr[14] <- mean(as.numeric(liq.14$effectiveSpread))
es.hr[15] <- mean(as.numeric(liq.15$effectiveSpread))
es.hr[16] <- mean(as.numeric(liq.16$effectiveSpread))
es.hr[17] <- mean(as.numeric(liq.17$effectiveSpread))
es.hr[18] <- mean(as.numeric(liq.18$effectiveSpread))
es.hr[19] <- mean(as.numeric(liq.19$effectiveSpread))
es.hr[20] <- mean(as.numeric(liq.20$effectiveSpread))
es.hr[21] <- mean(as.numeric(liq.21$effectiveSpread))
es.hr[22] <- mean(as.numeric(liq.22$effectiveSpread))
es.hr[23] <- mean(as.numeric(liq.23$effectiveSpread))
es.hr[24] <- mean(as.numeric(liq.24$effectiveSpread))

plot(1:24, es.hr, type="p", pch=20, col="blue", main="Effective Spread",
     xlab = "Hour", ylab = "Values", ylim = c(0,2.0))

# Autocorrelation of price changes
pr <- as.numeric(tq_data$PRICE)
dpr <- diff(pr)
covpr <- acf(dpr, lag.max=20, type="correlation", plot=FALSE)
plot(covpr, col="red", ylim = c(-0.05,1))
title("Autocorrelation of Price Changes",line=-1)

# Roll model parameters
covpr <- acf(dpr, lag.max=20, type="covariance", plot=FALSE)
gamma0 <- sd(dpr)^2
gamma0
gamma1 <- covpr$acf[2]
gamma1
cparam <- sqrt(-covpr$acf[2])
cparam
sig2u <- gamma0 + 2*gamma1
sigu <- sqrt(sig2u)
sigu

# Trade signs from Lee-Ready rule
tradeSigns <- getTradeDirection(tq_data)
acTS <- acf(tradeSigns, main="ACF trade signs")

# Improved Roll model
p <- as.numeric(tq_data$PRICE)
dp <- diff(p)
deps <- diff(tradeSigns)
mids <- (as.numeric(tq_data$OFR) + as.numeric(tq_data$BID))/2
dm <- diff(mids)
(fit.lm <- lm(dp ~ dm + deps))
```

```
fit.lm$coeff[3]
title("ACF trade signs",line=-1)

# Realized variance
realizedVar <- function(q){rCov(diff(p, lag=q, differences=1))/q}
rv_data <- NULL
for(q in 1:200){
  rv_data <- c(rv_data, realizedVar(q))
}

# Daily volatility by frequency
q5min <- n.trades*5/1440

rv5 = realizedVar(q5min)
rv5
sqrt(rv5)

# Daily volatility by Roll model estimate
rvRoll <- sig2u*n.trades
rvRoll
sigRoll <- sqrt(sig2u*n.trades)
sigRoll

# Signature plot
plot(rv_data, type ="l",
     main="Signature Plot for Prices + Roll")
abline(h=rv5,col="red")
abline(h=rvRoll,col="blue")
```