Muhammad Elfayoumi, Kevin Herbst, Abhishek Panchal, Zach Orrico

Professor Pirjol

FE 570

13 December, 2024

Empirical Analysis of Taiwan Semiconductor Manufacturing Company Microstructure Data

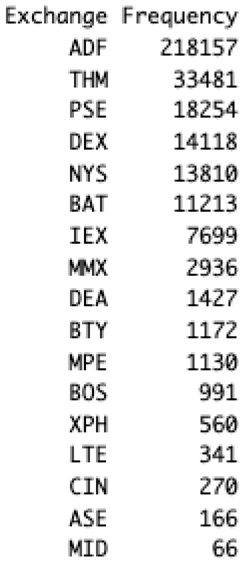
1. Introduction

This project explores the microstructure of Taiwan Semiconductor Manufacturing Company's (TSM) stock. TSM is a global leader in the semiconductor industry and a critical player in the technology supply chain. As market dynamics evolve, it's essential to have insights into liquidity, volatility, and informed trading behavior. This project aims to analyze tick-level data for TSM using a trade and quote (TAQ) dataset to study intraday trading patterns. The primary objectives include calculating quoted, effective, and realized spreads across various time intervals, estimating daily and annual volatility, and assessing the Probability of Informed Trading (PIN). These metrics provide a comprehensive perspective of TSM's market microstructure, contributing to a deeper understanding of the relationship between liquidity and informed trading in one of the most critical companies within the semiconductor industry.

2. Data Analysis

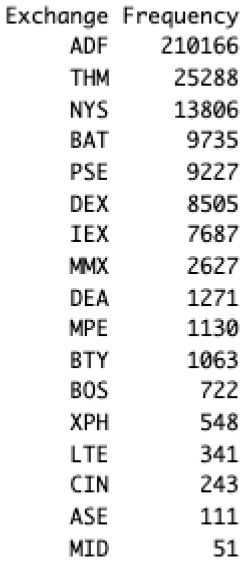
This project utilizes TSM TAQ data from October 17, 2024. This is a notable date because TSM reported its third-quarter earnings for 2024 and TSM exceeded revenue expectations. As a result, its stock experienced a significant increase of 12.57% from the previous day's close to the high on October 17th. This dataset contains a total of 326,120 trades and 292,527 trades occurred during active trading hours (9:30 AM to 4:00 PM EST). Below, two figures show a detailed breakdown of trades by exchange, highlighting activity levels across different platforms. Figure 1 shows the trade breakdown for the entire day and Figure 2 shows the trade breakdown only during active trading hours.

Figure 1



```
Exchange Frequency
     ADF    218157
     THM     33481
     PSE     18254
     DEX     14118
     NYS     13810
     BAT     11213
     IEX      7699
     MMX      2936
     DEA      1427
     BTY      1172
     MPE      1130
     BOS       991
     XPH       560
     LTE       341
     CIN       270
     ASE       166
     MID        66
```

Figure 2



```
Exchange Frequency
     ADF    210166
     THM     25288
     NYS     13806
     BAT      9735
     PSE      9227
     DEX      8505
     IEX      7687
     MMX      2627
     DEA      1271
     MPE      1130
     BTY      1063
     BOS       722
     XPH       548
     LTE       341
     CIN       243
     ASE       111
     MID        51
```

This project examines the ADF (Nasdaq High Cap) exchange for the liquidity analysis and volatility estimate. The NYS (New York Stock) exchange is used for the PIN estimate due to computation time. Using the ADF exchange for that model could be more efficient and have a long run time.

3. Liquidity Analysis

Liquidity in the financial markets is often evaluated using various spread measures that reflect the costs and efficiency of trading. This analysis examined three key liquidity measures: quoted, effective, and realized spreads. The quoted spread provides an instantaneous measure of liquidity based on available quotes. The effective spread measures the actual cost of a trade by comparing the trade price to the mid-price at the time of execution. The realized spread accounts for post-trade movements, indicating the impact of adverse selection on trade execution. Below are the formulas for each spread.

$$a_t = best\ ask\ price$$

$$b_t = best\ bid\ price$$

$$q_t = trade\ indicator\ (\pm 1\ for\ buy/sell)$$

$$m_t = \frac{1}{2}(a_t - b_t)$$

$$QS = \frac{1}{T}\sum_{t=1}^{T} \quad (a_t - b_t)$$

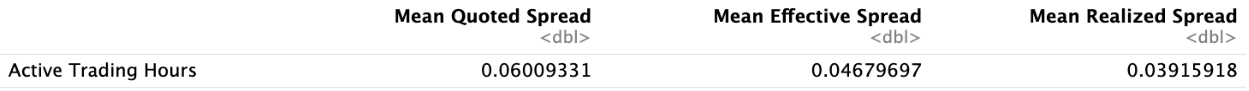
$$ES = \frac{1}{T}\sum_{t=1}^{T} \quad 2q_t(p_t - m_t)$$

$$RS = \frac{1}{T}\sum_{t=1}^{T} \quad 2q_t(p_t - m_{t+\delta})$$

The spreads were calculated for active trading hours in the ADF exchange using the getLiquidityMeasures function. To capture intraday variations, the data was divided into hourly intervals and the average spreads for each measure were computed.

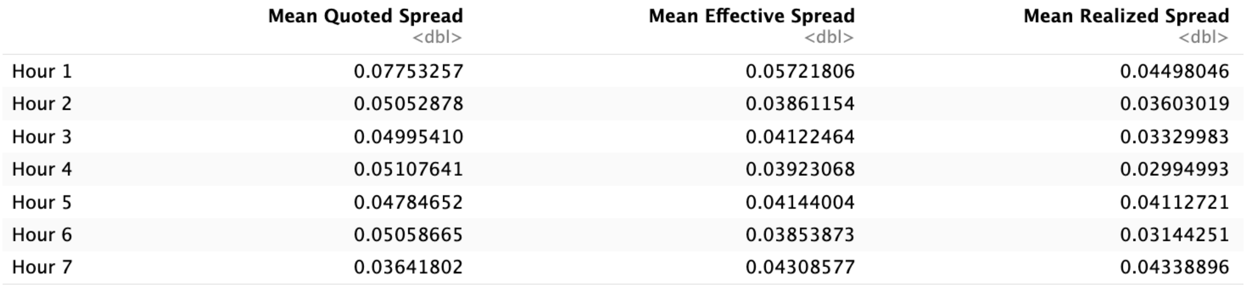
Figure 3 shows the average spread values for the whole active trading session and Figure 4 shows the average spread values for each hour in the active trading session. When analyzing the results, the quoted spread displayed the highest average value for the entire trading session at 0.0601. The effective spread averaged 0.0468, suggesting that some trades benefitted from price improvement. Finally, the realized spread averaged 0.0392, reflecting the cost implications of adverse selection post-trade. Intraday trends revealed a consistent decline in all three spread measures which can be seen in Figures 5, 6, and 7. All average spreads begin high when trading opens but start narrowing as trading continues. The narrowing of spreads over the session indicates increased liquidity and reduced uncertainty later in the day.

Figure 3



	Mean Quoted Spread	Mean Effective Spread	Mean Realized Spread
	<dbl>	<dbl>	<dbl>
Active Trading Hours	0.06009331	0.04679697	0.03915918

Figure 4



	Mean Quoted Spread	Mean Effective Spread	Mean Realized Spread
	<dbl>	<dbl>	<dbl>
Hour 1	0.07753257	0.05721806	0.04498046
Hour 2	0.05052878	0.03861154	0.03603019
Hour 3	0.04995410	0.04122464	0.03329983
Hour 4	0.05107641	0.03923068	0.02994993
Hour 5	0.04784652	0.04144004	0.04112721
Hour 6	0.05058665	0.03853873	0.03144251
Hour 7	0.03641802	0.04308577	0.04338896

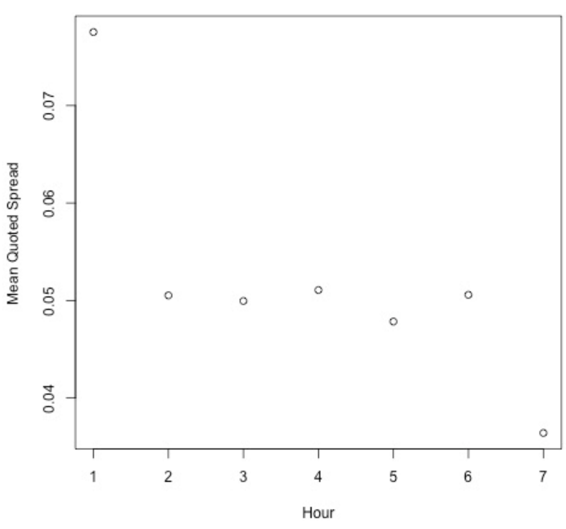


Figure 5: TSM Average Quoted Spread by Hour

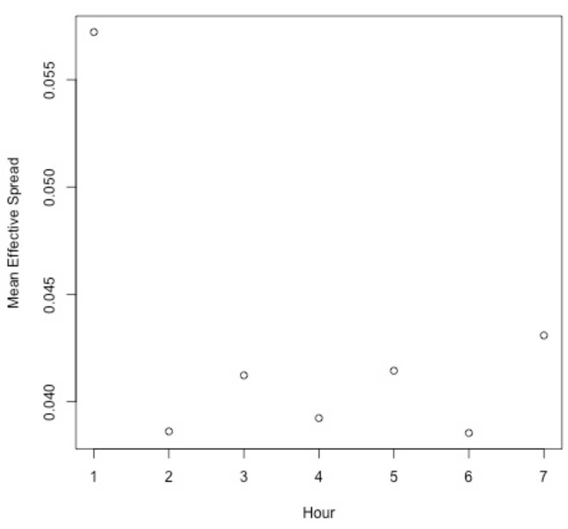


Figure 6: TSM Average Effective Spread by Hour

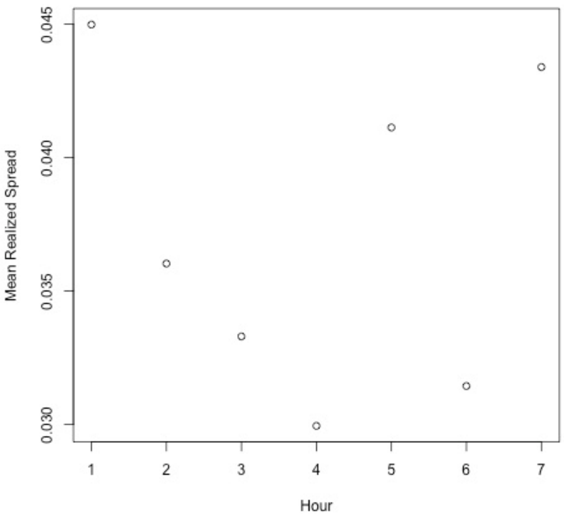


Figure 7: TSM Average Realized Spread by Hour

The results demonstrate that TSM experiences improving liquidity as the trading session progresses, likely due to higher trading volumes and reduced uncertainty. These insights are crucial for intraday traders and investors aiming to minimize trading costs, as they suggest that transactions later in the day are generally more cost-efficient.

4. Volatility Estimate

The Roll model was implemented to estimate trading costs and volatility measures across the different trading intervals. The Roll model uses the covariance structures of price changes to derive parameters like trading costs, efficient price volatility, daily volatility, and annual volatility. This section will discuss implementation, show and interpret the results, and implications for intraday trading dynamics.

To implement the Roll model, the price data was first cleaned to remove missing values, and the first differences were calculated. Using the ACF function, the first lag (γ_1) and the variance (γ_0) were extracted to estimate two key parameters: trading cost (c) and efficient price volatility (σ_u). The trading cost is a proxy for the bid-ask spread and reflects the cost incurred due to market frictions like liquidity shortages or transaction fees. The efficient price volatility captures the inherent randomness of the underlying asset's true value, unaffected by market frictions. Those formulas are provided below:

$$c = \sqrt{-\gamma_1}$$

$$\sigma_u = \sqrt{\gamma_0 + 2\gamma_1}$$

The efficient price volatility was further scaled to daily and annual volatility to help quantify the risk profile of the stock. Those formulas are provided below:

For daily data: Daily Volatility = $\sigma_u\sqrt{ntrades}$,

Annual Volatility = Daily Volatility * $\sqrt{252}$

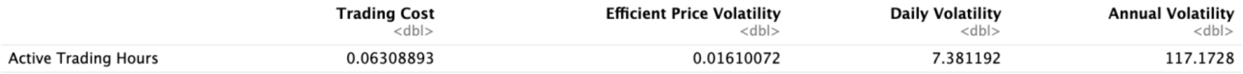
For hourly data: Hourly Volatility = $\sigma_u\sqrt{ntrades}$,

Daily Volatility = Hourly Volatility * $\sqrt{6.5}$,

Annual Volatility = Daily Volatility * $\sqrt{252}$

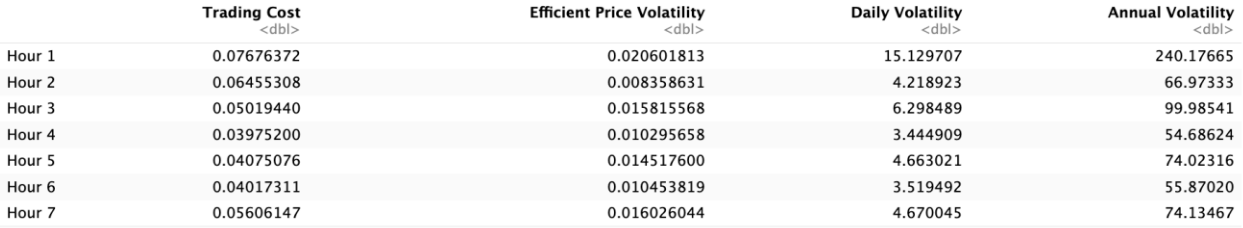
Figure 8 shows the Roll model parameters for the whole active trading session in the ADF exchange and Figure 9 shows the parameters for each hour in the active trading session. Trading costs were highest in hour one, at 0.0768, indicating a period of reduced liquidity or heightened uncertainty like a news announcement. In contrast, hour four exhibited the lowest cost, at 0.0398, suggesting a period of higher liquidity and efficient markets. The efficient price volatility was highest during hour one at 0.0206. This reflects significant price adjustments at the market opening in response to major news and a surge in trading activity. The lowest efficient volatility was in hour two which was 0.0084. This demonstrates a calm period with minimal significant price updates. The annual volatility estimates further underscore the heightened risk during the early trading hours with hour one reaching a peak of 240.18. The lowest annual volatility was seen during hour four which was 54.69. This supports a quieter period with significantly reduced trading activity.

Figure 8



	Trading Cost	Efficient Price Volatility	Daily Volatility	Annual Volatility
	<dbl>	<dbl>	<dbl>	<dbl>
Active Trading Hours	0.06308893	0.01610072	7.381192	117.1728

Figure 9



	Trading Cost	Efficient Price Volatility	Daily Volatility	Annual Volatility
	<dbl>	<dbl>	<dbl>	<dbl>
Hour 1	0.07676372	0.020601813	15.129707	240.17665
Hour 2	0.06455308	0.008358631	4.218923	66.97333
Hour 3	0.05019440	0.015815568	6.298489	99.98541
Hour 4	0.03975200	0.010295658	3.444909	54.68624
Hour 5	0.04075076	0.014517600	4.663021	74.02316
Hour 6	0.04017311	0.010453819	3.519492	55.87020
Hour 7	0.05606147	0.016026044	4.670045	74.13467

The Roll model highlights how trading dynamics vary intraday, which offers valuable insights into market structure, timing strategies, and risk management. These variations emphasize the importance of the temporal distribution of costs and risks for optimizing trading performance and improving market efficiency.

5. PIN Estimate

The PIN model was implemented to calculate the approximate number of informed traders in the market. The PIN model quantifies the likelihood that any given trade is informed by private information, providing valuable insights into market efficiency and the role of information asymmetry. It utilizes the number of buyer or seller initiated trades to calculate five separate parameters: α, δ, μ, ε_b, and ε_s. These five parameters are defined as:

α = intensity of news arrival

δ = probability of bad news

μ = probability of an informed buy or sell

ε_b, = probability of an uninformed buy

ε_s, = probability of an uninformed sell

These five parameters are estimated using maximum likelihood estimation (MLE). To solve for the parameters, two approaches from the InfoTrad package were utilized. The first approach was the Yan and Zhang (YZ) method, and the second was the GAN method. The YZ and GAN functions use either the EHO or LK methods of factorizing the likelihood function. From there, the YZ function uses boundary solutions to find the values of the parameters, while the GAN function uses a clustering approach that sorts the data into good, bad, and no news based on the mean absolute difference in the order imbalance. This is the likelihood equation to estimate the parameters:

$$p(n_b, n_s|\Theta) = (\propto * \,\delta) * \{e^{-\varepsilon_b} * \frac{\varepsilon_b^{n_b}}{n_b!}\} * \{e^{-(\mu + \varepsilon_s)} * \frac{(\mu + \varepsilon_s)^{n_s}}{n_s!}\}$$

$$+ \propto * \,(1 - \delta) * \{e^{-(\mu + \varepsilon_b)} * \frac{(\mu + \varepsilon_b)^{n_b}}{n_b!}\} * \{e^{-\varepsilon_s} * \frac{(\mu + \varepsilon_s)^{n_s}}{n_s!}\} + (1 -$$

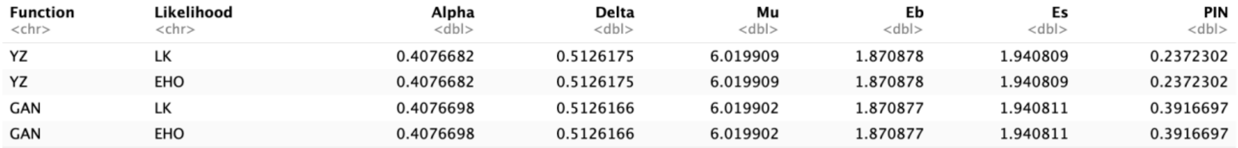
$$\propto) * \{e^{-\varepsilon_b} * \frac{\varepsilon_b^{n_b}}{n_b!}\} * \{e^{-\varepsilon_s} * \frac{\varepsilon_s^{n_s}}{n_s!}\}$$

The value of the PIN can be calculated as:

$$\text{PIN} = \frac{\propto * \mu}{\propto \mu + \varepsilon_b + \varepsilon_s}$$

Four versions of the model were generated, two with the YZ approach and two with the GAN approach. Each approach was run with the EHO factorization and LK factorization. The results can be seen in Figure 10. For both functions, the differences between the results with EHO and the ones found with LK are almost nonexistent. The two algorithms both agree on the values of the five parameters but disagree on the value of the PIN due to the differences in how each algorithm calculates it. When doing manual calculations, it was found that the PIN was more likely closer to 39% as opposed to the 23% that was reported first in the table. The probability of informed trading for TSM on October 17th was between 23% and 40% which is very high considering a normal PIN is around 10%. These elevated levels are probably due to October 17 being TSM's earnings day and many people could have been moving with knowledge of the true reported value.

Figure 10



Function	Likelihood	Alpha	Delta	Mu	Eb	Es	PIN
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
YZ	LK	0.4076682	0.5126175	6.019909	1.870878	1.940809	0.2372302
YZ	EHO	0.4076682	0.5126175	6.019909	1.870878	1.940809	0.2372302
GAN	LK	0.4076698	0.5126166	6.019902	1.870877	1.940811	0.3916697
GAN	EHO	0.4076698	0.5126166	6.019902	1.870877	1.940811	0.3916697

6. Conclusion

All in all, this project investigates the microstructure of TSM with TAQ data. A detailed analysis of their liquidity, volatility, and informed trading was performed by implementing advanced methodologies such as the computation of their spreads, the Roll model, and the PIN measure. The results showed a pattern in their liquidity and volatility, which explains the dynamic nature of trading costs and market uncertainty within the trading session. It was clear that the spreads narrow and volatility decreases as the day progresses which depicts improved market efficiency and better liquidity. From the above-average PIN value, it is evident that TSM's earnings announcement on October 24, 2024, impacted trading as higher levels of informed trading were observed. This analysis serves to prove how important it is to integrate advanced models that study market efficiency and trading behaviors.

Appendix

~~~
# Loading and Cleaning Data
```{r}
library(xts)
library(highfrequency)
library(data.table)

options(digits.secs=3)
Sys.setenv(TZ='America/New_York')

# # Loading Refinitiv data
# data <- read.csv("TSM_TickHistoryTimeandSales_20241017.csv")
# data <- data[-c(1:6),]
#
# # Subsetting into trades and quotes
# tdata <- subset(data, Type == "Trade")
# qdata <- subset(data, Type == "Quote")
#
# # Removing T in dates and pulling columns we want
# tdata2 <- data.frame(DT = gsub("T", " ", tdata$Date.Time, perl = TRUE),
#                      SYMBOL = "TSM",
#                      PRICE = tdata$Price,
#                      SIZE = tdata$Volume,
#                      EX = tdata$Ex.Cntrb.ID)
#
# qdata2 <- data.frame(DT = gsub("T", " ", qdata$Date.Time, perl = TRUE),
#                      SYMBOL = "TSM",
#                      BID = qdata$Bid.Price,
#                      BIDSIZ = qdata$Bid.Size,
#                      OFR = qdata$Ask.Price,
#                      OFRSIZ = qdata$Ask.Size)
#
# tdata3 <- data.frame(DT = gsub("-04", "", tdata2$DT, perl = TRUE),
#                      SYMBOL = "TSM",
#                      PRICE = tdata$Price,
#                      SIZE = tdata$Volume,
#                      EX = tdata$Ex.Cntrb.ID)
#
# qdata3 <- data.frame(DT = gsub("-04", "", qdata2$DT, perl = TRUE),
#                      SYMBOL = "TSM",
#                      BID = qdata$Bid.Price,
~~~

~~~~
#                           BIDSIZ = qdata$Bid.Size,
#                           OFR = qdata$Ask.Price,
#                           OFRSIZ = qdata$Ask.Size)
#
# # Formatting dates
# tdata3$DT <- as.POSIXct(tdata3$DT, format = "%Y-%m-%d %H:%M:%OS", tz = "America/New_York")
# qdata3$DT <- as.POSIXct(qdata3$DT, format = "%Y-%m-%d %H:%M:%OS", tz = "America/New_York")
#
# # Merging times
# tdata4 <- mergeTradesSameTimestamp(as.data.table(tdata3))
# qdata4 <- mergeQuotesSameTimestamp(as.data.table(qdata3))
#
# # Merging trade and quote data
# tqdata <- matchTradesQuotes(as.data.table(tdata4), as.data.table(qdata4))
#
# # Saving as RData file
# save(tqdata, file = "taqdata_TSM_20241017.RData")

load("taqdata_TSM_20241017.RData")
```


# Data Analysis
```{r}
library(dplyr)
library(lubridate)

tqdata$SIDE <- getTradeDirection(tqdata)

# Filtering for active trading hours
tqdata_active <- tqdata %>%
  filter(format(DT, "%H:%M:%S") >= "09:30:00" & format(DT, "%H:%M:%S") <= "16:00:00")

n_trades <- nrow(tqdata)
cat("Total Number of Trades:", n_trades, "\n")
n_trades_active <- nrow(tqdata_active)
cat("Total Number of Trades During Trading Hours:", n_trades_active, "\n")

ex_trades <- as.data.frame(table(tqdata$EX))
ex_trades <- ex_trades[-1,]
ex_trades <- ex_trades[order(-ex_trades$Freq),]
colnames(ex_trades) <- c("Exchange", "Frequency")
print(format(ex_trades, justify = "right"))

ex_trades_active <- as.data.frame(table(tqdata_active$EX))
~~~~

~~~
ex_trades_active <- ex_trades_active[-1,]
ex_trades_active <- ex_trades_active[order(-ex_trades_active$Freq),]
colnames(ex_trades_active) <- c("Exchange", "Frequency")
print(format(ex_trades_active, justify = "right"))
```

# Liquidity Analysis
```{r}
# Filtering for ADF
tqdata_active_adf <- tqdata_active[tqdata_active$EX == "ADF"]

# Filtering for hourly data
hours <- list(hr1 = c("09:30:00", "10:30:00"),
              hr2 = c("10:30:00", "11:30:00"),
              hr3 = c("11:30:00", "12:30:00"),
              hr4 = c("12:30:00", "13:30:00"),
              hr5 = c("13:30:00", "14:30:00"),
              hr6 = c("14:30:00", "15:30:00"),
              hr7 = c("15:30:00", "16:00:00"))

hr_data <- list()
for (name in names(hours)) {
  filtered_data <- tqdata_active_adf %>%
    filter(format(DT, "%H:%M:%S") >= hours[[name]][1] & format(DT, "%H:%M:%S") <
hours[[name]][2])
  hr_data[[name]] <- filtered_data
}
hr_data <- lapply(hr_data, as.data.table)

mean_liquidity_measures <- function(data) {
  liqMeasures <- getLiquidityMeasures(data)
  liqMeas_spreads <- c(mean(liqMeasures$quotedSpread),
                       mean(liqMeasures$effectiveSpread, na.rm = TRUE),
                       mean(liqMeasures$realizedSpread, na.rm = TRUE))
  return(liqMeas_spreads)
}

liqMeas_active <- mean_liquidity_measures(tqdata_active_adf)
liqMeas_active <- data.frame(t(liqMeas_active))
colnames(liqMeas_active) <- c("Mean Quoted Spread", "Mean Effective Spread", "Mean
Realized Spread")
rownames(liqMeas_active) <- "Active Trading Hours"
liqMeas_active

liqMeas_hourly <- lapply(hr_data, mean_liquidity_measures)
liqMeas_hourly <- data.frame(do.call(rbind, liqMeas_hourly))
~~~

~~~
colnames(liqMeas_hourly) <- c("Mean Quoted Spread", "Mean Effective Spread", "Mean
Realized Spread")
rownames(liqMeas_hourly) <- c("Hour 1", "Hour 2", "Hour 3", "Hour 4", "Hour 5", "Hour 6",
"Hour 7")
liqMeas_hourly

#jpeg("rplot1.jpeg")
plot(1:7, liqMeas_hourly$`Mean Quoted Spread`, xlab = "Hour", ylab = "Mean Quoted Spread",
     main = "Figure 5: TSM Average Quoted Spread by Hour")
#dev.off()

#jpeg("rplot2.jpeg")
plot(1:7, liqMeas_hourly$`Mean Effective Spread`, xlab = "Hour", ylab = "Mean Effective
Spread",
     main = "Figure 6: TSM Average Effective Spread by Hour")
#dev.off()

#jpeg("rplot3.jpeg")
plot(1:7, liqMeas_hourly$`Mean Realized Spread`, xlab = "Hour", ylab = "Mean Realized
Spread",
     main = "Figure 7: TSM Average Realized Spread by Hour")
#dev.off()
```

# Volatility Estimate
```{r}
# Function to calculate roll model
roll_model <- function(data, interval) {
  pr <- na.omit(data$PRICE)
  dpr <- diff(pr)

  covdpr <- acf(dpr, lag.max = 10, type = "covariance", plot = FALSE)

  gamma0 <- covdpr$acf[1]
  gamma1 <- covdpr$acf[2]

  cparam <- sqrt(-gamma1)

  sig2u <- gamma0 + 2*gamma1
  sigu <- sqrt(sig2u)

  ntrades <- length(pr)

  if (interval == "day") {
    daily_vol <- sigu*sqrt(ntrades)
    ann_vol <- daily_vol*sqrt(252)
~~~

~~~~
  }
  if (interval == "hour") {
    hourly_vol <- sigu*sqrt(ntrades)
    daily_vol <- hourly_vol*sqrt(6.5)
    ann_vol <- daily_vol*sqrt(252)

  }

  return(c(cparam, sigu, daily_vol, ann_vol))
}


vol_active <- roll_model(tqdata_active_adf, "day")
vol_active <- data.frame(t(vol_active))
colnames(vol_active) <- c("Trading Cost", "Efficient Price Volatility", "Daily Volatility", "Annual Volatility")
rownames(vol_active) <- "Active Trading Hours"
vol_active

vol_hourly <- lapply(hr_data, function(data) roll_model(data, interval = "hour"))
vol_hourly <- data.frame(do.call(rbind, vol_hourly))
colnames(vol_hourly) <- c("Trading Cost", "Efficient Price Volatility", "Daily Volatility", "Annual Volatility")
rownames(vol_hourly) <- c("Hour 1", "Hour 2", "Hour 3", "Hour 4", "Hour 5", "Hour 6", "Hour 7")
vol_hourly
```

# PIN Estimate
```{r}
library(InfoTrad)

# Pulling number of buy and sell trades from NYSE
buys <- tqdata_active$NUMTRADES[tqdata_active$SIDE == 1 & tqdata_active$EX == "NYS"]
buys <- na.omit(buys)
sells <- tqdata_active$NUMTRADES[tqdata_active$SIDE == -1 & tqdata_active$EX == "NYS"]
sells <- na.omit(sells)

data <- cbind(buys, sells[1:6546])

# Calibrating model
par0 <- c(0.5,0.5,300,400,500)
methods <- c("Nelder-Mead", "BFGS", "CG", "SANN")
~~~~

~~~
likelihoods <- c("LK", "EHO")

YZ_greeks <- data.frame()
for (l in likelihoods) {
  model <- YZ(data, likelihood = l)
  alpha <- model$alpha
  delta <- model$delta
  mu <- model$mu
  eb <- model$epsilon_b
  es <- model$epsilon_s
  pin <- model$PIN
  result <- c(alpha, delta, mu, eb, es, pin)

  YZ_greeks <- rbind(YZ_greeks, result)
}

GAN_greeks <- data.frame()
for (l in likelihoods) {
  model <- GAN(data, likelihood = l)
  alpha <- model$alpha
  delta <- model$delta
  mu <- model$mu
  eb <- model$epsilon_b
  es <- model$epsilon_s
  pin <- model$PIN
  result <- c(alpha, delta, mu, eb, es, pin)

  GAN_greeks <- rbind(GAN_greeks, result)
}

fun_methods_YZ <- data.frame(c(rep("YZ", 2)), likelihoods)
YZ_greeks <- cbind(fun_methods_YZ, YZ_greeks)
colnames(YZ_greeks) <- c("Function", "Likelihood", "Alpha", "Delta", "Mu", "Eb", "Es",
"PIN")

fun_methods_GAN <- data.frame(c(rep("GAN", 2)), likelihoods)
GAN_greeks <- cbind(fun_methods_GAN, GAN_greeks)
colnames(GAN_greeks) <- c("Function", "Likelihood", "Alpha", "Delta", "Mu", "Eb", "Es",
"PIN")

greeks_summary <- rbind(YZ_greeks, GAN_greeks)
greeks_summary
```
~~~