is607Project2

The puprpose of this data profile is to introduce an important problem in algorithmic trading.

Is it better to trade smaller but more frequently to reduce the price impact cost associated with wider bid-ask spreads in periods of lower liquidity OR is it better to trade larger to minimize fees paid to brokers and regulatory agencies?

Problem 1:

I chose a dataset that contains 3,472 observations of 16 variables. The dataset is a day's worth of algorithmic stock trading. This file contains multiple types of numerical columns (prices, quantities), categorial columns (sub accounts) and character columns (stock symbols).

```
library(plyr)
library(tidyr)
library(reshape2)
library(ggplot2)
setwd("~/Downloads")
file <- read.csv("account-summary-MRTTRADING1-20140930.csv")
head(file)
```

```
##       Account Security       Order.ID Trade.Date Trade.Time Side Liquidity
## 1 MRTTRADING1      EWZ '146357881511'  9/30/2014   15:58:35    3   Removed
## 2 MRTTRADING1      EWZ '163537750695'  9/30/2014   15:58:35    3   Removed
## 3 MRTTRADING1      EWZ '180717619879'  9/30/2014   15:58:35    3   Removed
## 4 MRTTRADING1      EWZ '945221798567'  9/30/2014   15:58:36    3   Removed
## 5 MRTTRADING1      EWZ '962401667751'  9/30/2014   15:58:36    3   Removed
## 6 MRTTRADING1      EWZ '979581536935'  9/30/2014   15:58:36    3   Removed
##   Route Quantity Price Lime.Fee ECN.Fee ACT.Fee SEC.Fee NASD.Fee
## 1 BATSY      100 43.61     0.08   -0.16       0     0.1     0.02
## 2 EDGAA      100 43.61     0.08   -0.02       0     0.1     0.02
## 3   BSX      100 43.61     0.08   -0.04       0     0.1     0.02
## 4 BATSY      100 43.61     0.08   -0.16       0     0.1     0.02
## 5 EDGAA      100 43.61     0.08   -0.02       0     0.1     0.02
## 6   BSX      100 43.61     0.08   -0.04       0     0.1     0.02
##   Rounded.Order.Commission
## 1                     0.04
## 2                     0.18
## 3                     0.16
## 4                     0.04
## 5                     0.18
## 6                     0.16
```

```
tail(file)
```

```
##          Account Security      Order.ID Trade.Date Trade.Time Side
## 3467 MRTTRADING9      VOD '34688732674'  9/30/2014   15:59:58    1
## 3468 MRTTRADING9      VOD '34688732674'  9/30/2014   15:59:58    1
## 3469 MRTTRADING9      VOD '43278667266'  9/30/2014   15:59:58    1
## 3470 MRTTRADING9      VOD '43278667266'  9/30/2014   15:59:58    1
## 3471 MRTTRADING9      KMB '51868601858'  9/30/2014   15:59:58    1
## 3472 MRTTRADING9      KMB '60458536450'  9/30/2014   15:59:58    1
```

```
##             Liquidity    Route Quantity Price Lime.Fee ECN.Fee ACT.Fee
## 3467 Removed-MidPoint INET-FIX       30  32.9   0.0240  0.0810       0
## 3468 Removed-MidPoint INET-FIX       70  32.9   0.0560  0.1890       0
## 3469 Removed-MidPoint INET-FIX        3  32.9   0.0024  0.0081       0
## 3470 Removed-MidPoint INET-FIX       70  32.9   0.0560  0.1890       0
## 3471          Removed     XNYS      100 107.6   0.0800  0.2700       0
## 3472          Removed     XNYS       16 107.6   0.0128  0.0432       0
##      SEC.Fee NASD.Fee Rounded.Order.Commission
## 3467       0        0                    0.35
## 3468       0        0                    0.00
## 3469       0        0                    0.26
## 3470       0        0                    0.00
## 3471       0        0                    0.35
## 3472       0        0                    0.06
```

Problem 2:

First, I will rename the columns using easier names.

```
file.df <- data.frame(file)
colnames(file.df) <- c("Account", "Security", "ID", "TradeDate", "TradeTime", "Side", "Liquidity", "Rou
head(file.df)
```

```
##       Account Security             ID TradeDate TradeTime Side Liquidity
## 1 MRTTRADING1      EWZ '146357881511' 9/30/2014  15:58:35    3   Removed
## 2 MRTTRADING1      EWZ '163537750695' 9/30/2014  15:58:35    3   Removed
## 3 MRTTRADING1      EWZ '180717619879' 9/30/2014  15:58:35    3   Removed
## 4 MRTTRADING1      EWZ '945221798567' 9/30/2014  15:58:36    3   Removed
## 5 MRTTRADING1      EWZ '962401667751' 9/30/2014  15:58:36    3   Removed
## 6 MRTTRADING1      EWZ '979581536935' 9/30/2014  15:58:36    3   Removed
##   Route Quantity Price LimeFee ECNFee ACTFee SECFee NASDFee
## 1 BATSY      100 43.61    0.08  -0.16      0    0.1    0.02
## 2 EDGAA      100 43.61    0.08  -0.02      0    0.1    0.02
## 3   BSX      100 43.61    0.08  -0.04      0    0.1    0.02
## 4 BATSY      100 43.61    0.08  -0.16      0    0.1    0.02
## 5 EDGAA      100 43.61    0.08  -0.02      0    0.1    0.02
## 6   BSX      100 43.61    0.08  -0.04      0    0.1    0.02
##   RoundedOrderComm
## 1             0.04
## 2             0.18
## 3             0.16
## 4             0.04
## 5             0.18
## 6             0.16
```

Each trade is an observation. As such, we can summarize how many trades were made in each symbol.

```
trades.per.sym <- count(file.df$Security)
head(trades.per.sym)
```

```
##      x freq
## 1  ADP    2
## 2  AIZ    3
```

```
## 3 ALTR   16
## 4 AMGN    1
## 5  AMP    2
## 6  AMT   13
```

```
tail(trades.per.sym)
```

```
##        x freq
## 113   XL    3
## 114  XLE  141
## 115  XOP   49
## 116  XRT   20
## 117  XRX    8
## 118 YHOO    3
```

```
colnames(trades.per.sym) <- c("Symbol", "NumberOfTrades")
trades.per.symdf <- data.frame(trades.per.sym)
```

The symbol in which the most trades were made today was the stock GDX which was traded 925 times.

```
trades.per.symdf[which(trades.per.symdf$NumberOfTrades == max(trades.per.symdf$NumberOfTrades)),]
```

```
##    Symbol NumberOfTrades
## 37    GDX            925
```

The symbols in which the least trades were made today were the stocks AMGN, APD, AZO, CMI, GD, IBM, LNC, ROP, SIAL, WHR each with just 1 trade.

```
trades.per.symdf[which(trades.per.symdf$NumberOfTrades == min(trades.per.symdf$NumberOfTrades)),]
```

```
##     Symbol NumberOfTrades
## 4     AMGN              1
## 7      APD              1
## 9      AZO              1
## 15     CMI              1
## 36      GD              1
## 44     IBM              1
## 58     LNC              1
## 80     ROP              1
## 85    SIAL              1
## 109    WHR              1
```

The average price of each of the 118 unique stock symbols traded is summarized below.

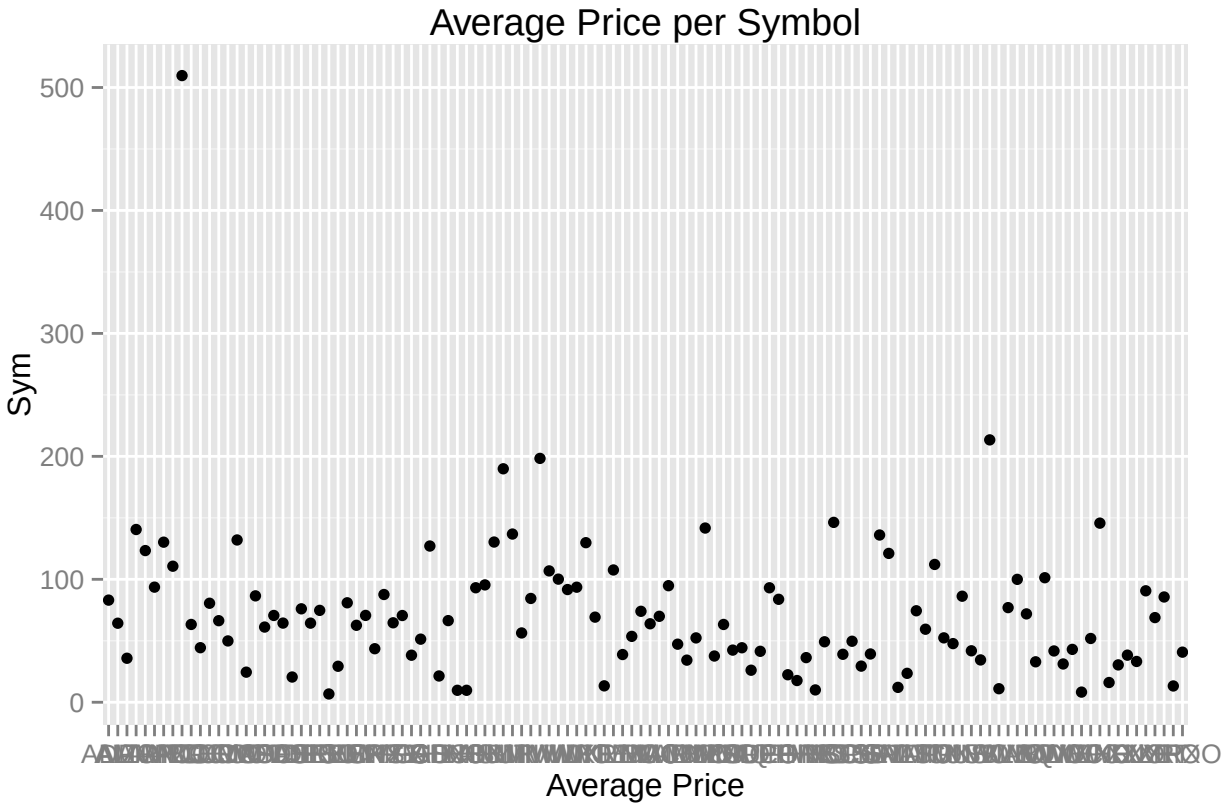
```
library(ggplot2)
sym.price <- data.frame(file.df$Security,file.df$Price)
colnames(sym.price) <- c("Sym", "Price")
summary.symprice <- ddply(sym.price, .(Sym), summarize, avg=mean(Price))
head(summary.symprice)
```

```
##    Sym    avg
## 1  ADP  83.07
## 2  AIZ  64.28
## 3 ALTR  35.79
## 4 AMGN 140.51
## 5  AMP 123.37
## 6  AMT  93.63
```

```
tail(summary.symprice)
```

```
##      Sym   avg
## 113   XL 33.17
## 114  XLE 90.65
## 115  XOP 68.86
## 116  XRT 85.62
## 117  XRX 13.25
## 118 YHOO 40.74
```

```
summary.symprice.df <- data.frame(summary.symprice)
colnames(summary.symprice.df) <- c("Sym", "Price")
avg.price.persym <- ggplot(summary.symprice.df,aes(Sym, Price))+geom_point() + labs(title = "Average Pri
avg.price.persym
```



The most expensive stock traded was AZO. The average price traded of AZO was 509.68 dollars and can easily be identified as the outlier in the scatter plot above.

```
summary.symprice.df[which(summary.symprice.df$Price == max(summary.symprice.df$Price)),]
```

```
##   Sym Price
## 9 AZO 509.7
```

The average price per share of stock traded was $72.00

```
avg.px.alldf <- data.frame(mean(summary.symprice.df$Price))
colnames(avg.px.alldf) <- c("AvgPricePerShare")
avg.px.alldf
```

```
##   AvgPricePerShare
## 1               72
```

Interestingly, all trades traded were on the same date between 15:58:35 and 15:59:58. Furthermore, the scatter plot below shows that before 15:58:48 every trade is 100 shares in size.

Over the next 10 seconds between 15:58:48 and 15:58:58 the algorithm starts to trade in more varied quantities. For instance at 15:58:52 there are 7 trades below 25 shares a clip. In the last 3 seconds of trading, most of the day's volume is created.

At 15:59:57 in particular the algorithm trades a wide variety of quantity sizes ranging from a just one share per trade to 100 shares per trade. In fact within that 1 second of time there are 13 trades of less than 25 shares each and 12 trades of between 25 and 50 shares each.

At 15:59:57 there are a total 25 trades of 50 shares or less as compared with 0 trades of 50 shares or less at 15:58:48.

This behavioral pattern could be attributed to rapidly changing liquidity dynamics and should be an area of further research. Some important questions include,

"Is there a greater impact cost of larger trade sizes after 15:59:57 or is there a greater impact cost of larger trade sizes prior to 15:58:48?"

"How does market liquidity over each interval of time affect the transaction cost in terms of average bid-ask spreads of stocks?"

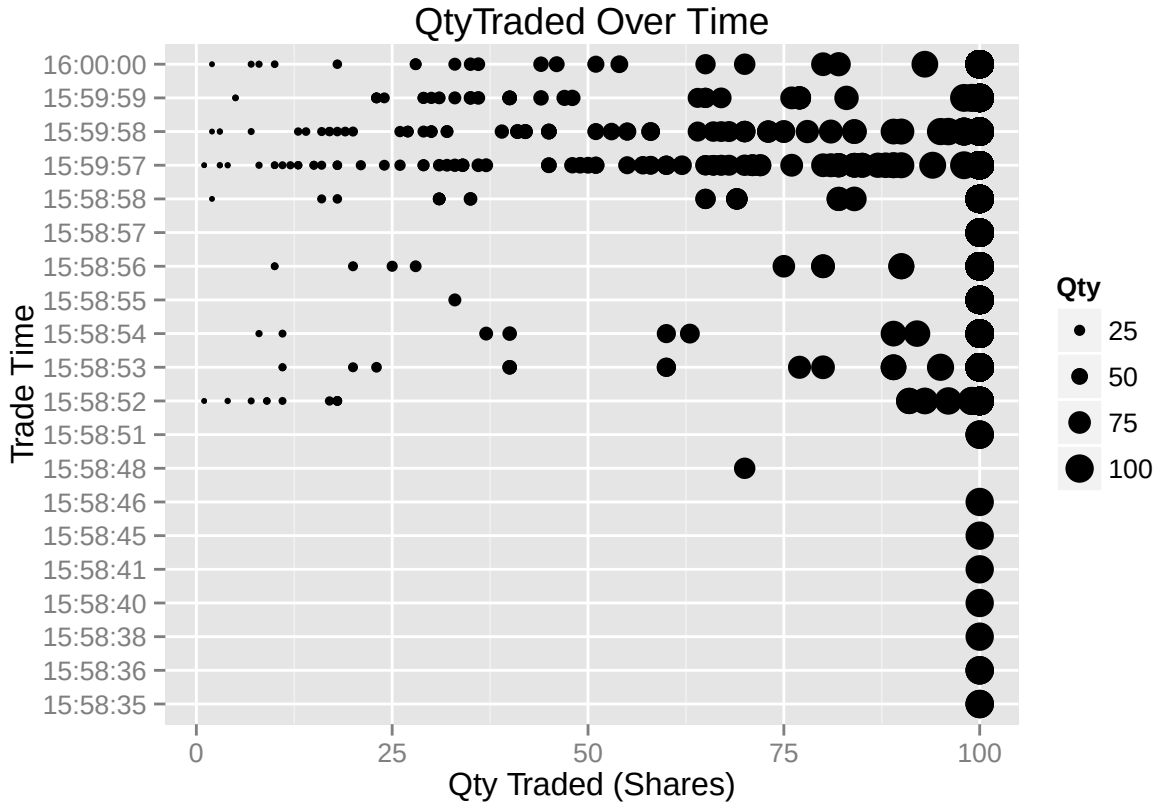
```
all.qty <- file.df$Quantity
all.time <- file.df$TradeTime
all.qty.timedf <- data.frame(file.df$Quantity,file.df$TradeTime)
colnames(all.qty.timedf) <- c("Qty", "TradeTime")
head(all.qty.timedf)
```

```
##   Qty TradeTime
## 1 100  15:58:35
## 2 100  15:58:35
## 3 100  15:58:35
## 4 100  15:58:36
## 5 100  15:58:36
## 6 100  15:58:36
```

```
tail(all.qty.timedf)
```

```
##      Qty TradeTime
## 3467  30  15:59:58
## 3468  70  15:59:58
## 3469   3  15:59:58
## 3470  70  15:59:58
## 3471 100  15:59:58
## 3472  16  15:59:58
```

```
qty.vs.time <- ggplot(all.qty.timedf, aes(x = Qty, y = TradeTime))+geom_point(aes(size = Qty))+scale_si
qty.vs.time
```



There are 5 fees that are paid for each trade made. Those fees include a brokerage fee, an ECN fee, an ACT fee, an SEC fee and an NASD fee.

The total fees paid for the trades in this data set was $720.4062 for a total of 333,535 shares traded.

The total fees per share paid was $0.002159912 per share traded.

```
comm.df <- data.frame(file.df$LimeFee, file.df$ECNFee, file.df$ACTFee, file.df$SECFee, file.df$NASDFee)
colnames(comm.df) <- c("BrokerFee", "ECNFee", "ACTFee", "SECFee", "NASDFee")
all.qty.time.fee.df <- data.frame(all.qty.timedf, comm.df)
total.fees.paid <- data.frame(sum(comm.df))
colnames(total.fees.paid) <- c("TotalFeesPaid")
sum.qty <- data.frame(sum(all.qty))
colnames(sum.qty) <- c("TotalSharesTraded")
sum.qty
```

```
##   TotalSharesTraded
## 1            333535
```

```
total.fees.paid
```

```
##   TotalFeesPaid
## 1         720.4
```

```
fees.per.share <- data.frame((total.fees.paid/sum.qty))
colnames(fees.per.share) <- c("Fees Per Share Paid")
fees.per.share
```

```
##   Fees Per Share Paid
## 1             0.00216
```

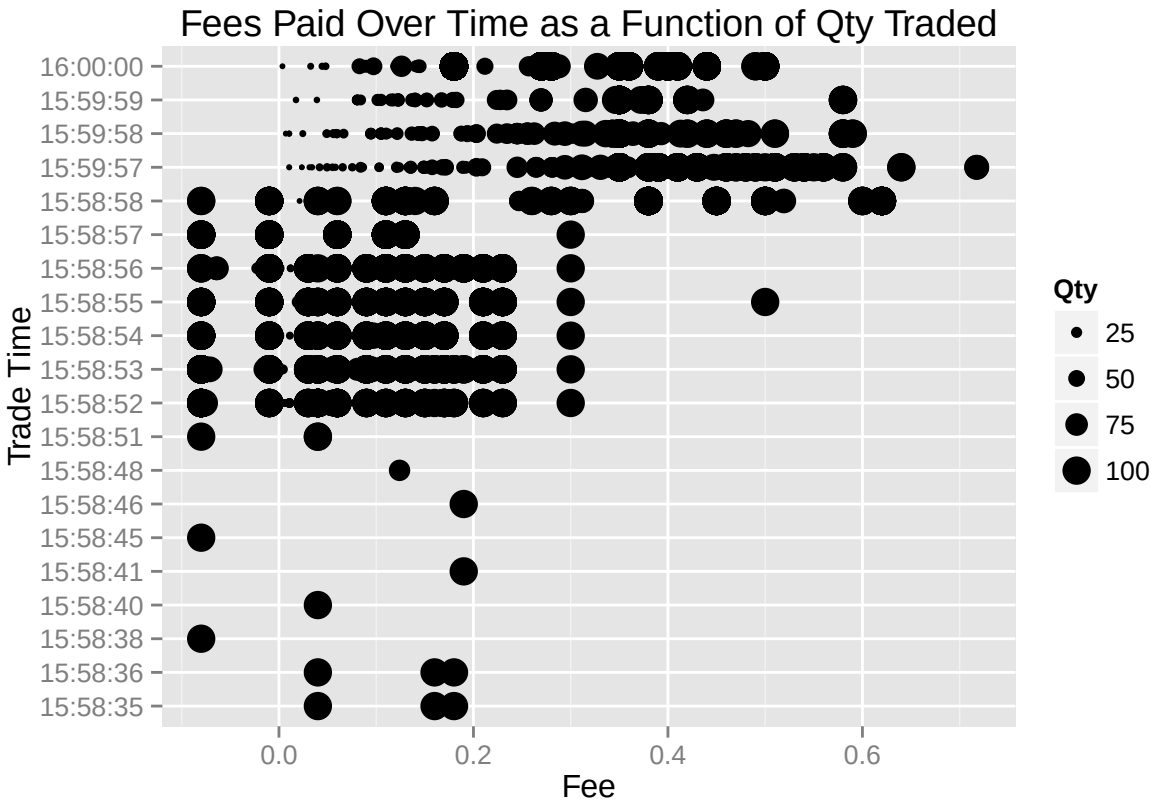
```
fee.time.qtydf <- data.frame(all.qty.time.fee.df$Qty, all.qty.time.fee.df$TradeTime,(all.qty.time.fee.d
colnames(fee.time.qtydf) <- c("Qty", "TradeTime", "Fee")
head(fee.time.qtydf)
```

```
##   Qty TradeTime  Fee
## 1 100  15:58:35 0.04
## 2 100  15:58:35 0.18
## 3 100  15:58:35 0.16
## 4 100  15:58:36 0.04
## 5 100  15:58:36 0.18
## 6 100  15:58:36 0.16
```

```
tail(fee.time.qtydf)
```

```
##      Qty TradeTime    Fee
## 3467  30  15:59:58 0.1050
## 3468  70  15:59:58 0.2450
## 3469   3  15:59:58 0.0105
## 3470  70  15:59:58 0.2450
## 3471 100  15:59:58 0.3500
## 3472  16  15:59:58 0.0560
```

```
time.qty.fee.plot <- ggplot(fee.time.qtydf, aes(x = Fee, y = TradeTime))+geom_point(aes(size = Qty)) + s
time.qty.fee.plot
```



From the scatter plot above we can clearly see that the majority of fees paid were paid within the last one minute of trading after 15:58:58.

The average trade size is 96.06423 shares. The min trade size is 1 share and the max trade size is 100 shares.

Interestingly, the average qty traded drops to 76.70769 shares per trade at 15:59:58 from 100 shares average per trade at 15:58:57 and increases back to 99.21886 shares per trade on average at 16:00:00. At 15:59:57 the average fees paid peaks at $0.37608 per share traded.

A correlation analysis shows that a weak inverse relationship between average quantity per trade over each time interval. This can be interpreted as the smaller the average share size per trade the higher the average fees paid. The reason for this may be due to the fact that some of the components that constitute the fee are based on the number of trades made rather than the number of shares traded. That is, the SEC Fee for example will be the same whether the trade size is 1 share or 100 shares. Therefore executing many small trades will drive the part of the fee component higher.

Other parts of the fee component, such as brokerage commissions, are charged on a per share basis.

```
avg.tradesize <- data.frame(mean(fee.time.qtydf$Qty))
colnames(avg.tradesize) <- c("Average Trade Size (Shares)")
avg.tradesize
```

```
##   Average Trade Size (Shares)
## 1                       96.06
```

```
max.tradesize <- data.frame(max(fee.time.qtydf$Qty))
colnames(max.tradesize) <- c("Max Trade Size")
max.tradesize
```

```
##   Max Trade Size
## 1            100
```

```
min.tradesize <- data.frame(min(fee.time.qtydf$Qty))
colnames(min.tradesize) <- c("Min Trade Size")
min.tradesize
```

```
##   Min Trade Size
## 1              1
```

```
summary.tradesize <- ddply(fee.time.qtydf, .(TradeTime), summarize, AvgQtyByTimePeriod=mean(Qty))
summary.tradesize
```

```
##    TradeTime AvgQtyByTimePeriod
## 1   15:58:35             100.00
## 2   15:58:36             100.00
## 3   15:58:38             100.00
## 4   15:58:40             100.00
## 5   15:58:41             100.00
## 6   15:58:45             100.00
## 7   15:58:46             100.00
## 8   15:58:48              70.00
## 9   15:58:51             100.00
## 10  15:58:52              93.60
## 11  15:58:53              97.44
## 12  15:58:54              97.71
## 13  15:58:55              99.71
## 14  15:58:56              97.66
## 15  15:58:57             100.00
## 16  15:58:58              96.24
## 17  15:59:57              88.57
## 18  15:59:58              76.71
## 19  15:59:59              85.46
## 20  16:00:00              99.22
```

```
summary.fee <- ddply(fee.time.qtydf, .(TradeTime), summarize, AvgFeeByTimePeriod = mean(Fee))
summary.fee
```

```
##    TradeTime AvgFeeByTimePeriod
## 1   15:58:35            0.12667
## 2   15:58:36            0.12667
## 3   15:58:38           -0.08000
## 4   15:58:40            0.04000
## 5   15:58:41            0.19000
## 6   15:58:45           -0.08000
## 7   15:58:46            0.19000
## 8   15:58:48            0.12400
## 9   15:58:51            0.00000
## 10  15:58:52            0.05506
## 11  15:58:53            0.07876
## 12  15:58:54            0.09566
## 13  15:58:55            0.10539
## 14  15:58:56            0.10726
## 15  15:58:57            0.05986
## 16  15:58:58            0.34916
```

```
## 17  15:59:57            0.37608
## 18  15:59:58            0.30389
## 19  15:59:59            0.31639
## 20  16:00:00            0.22173
```

```
avg.fee.tradesize.byintervaldf <- data.frame(summary.tradesize$AvgQtyByTimePeriod, summary.fee$AvgFeeBy
colnames(avg.fee.tradesize.byintervaldf) <- c("AvgQty", "AvgFee")
avg.fee.tradesize.byintervaldf
```

```
##    AvgQty   AvgFee
## 1  100.00  0.12667
## 2  100.00  0.12667
## 3  100.00 -0.08000
## 4  100.00  0.04000
## 5  100.00  0.19000
## 6  100.00 -0.08000
## 7  100.00  0.19000
## 8   70.00  0.12400
## 9  100.00  0.00000
## 10  93.60  0.05506
## 11  97.44  0.07876
## 12  97.71  0.09566
## 13  99.71  0.10539
## 14  97.66  0.10726
## 15 100.00  0.05986
## 16  96.24  0.34916
## 17  88.57  0.37608
## 18  76.71  0.30389
## 19  85.46  0.31639
## 20  99.22  0.22173
```

```
cor((avg.fee.tradesize.byintervaldf$AvgQty), (avg.fee.tradesize.byintervaldf$AvgFee))
```

```
## [1] -0.4309
```

As such, one must weigh the costs and benefits of trading larger to minimize fees paid versus trading smaller to minimize the bid-ask spread paid as a function of illiquidity over various time intervals throughout the day. Liqidity analysis to measure impact cost is beyond the scope of this project but is a logical next step.