# is607 Project 4

*Prashant B. Bhuyan*

*November 9, 2014*

My use case is to build a data base that will allow me query various trading analytics based a transaction data set. The data used are daily transaction files. For the purpose of this project I'll be using two different tables where each table represents one calendar day. The first table (file1) represents transactions from 11.7.2014 and the second table (file2) represents transactions from 11.6.2014. Our goal is to put each set of transactions into a database so that we can make queries to breakdown our commissions paid or compute our average buy or sell price or query the distribution of trades by symbol name. Ultimately we'd like to compute our daily and cumulative realized pnl for each day.

file1 consists of 1193 observations of 16 variables while file2 contains 1033 observati# ons of 16 variables.

Part 1 -Obtaining the Data

here are the heads for file1 and file2 of the data that I'm using for the project.

```
setwd("~/Desktop/si607Project4")

# 11.7.2014 Transactinos
file1 <- read.csv("~/Downloads/account-summary-MRTTRADING1-20141107.csv")


head(file1)
```

```
##       Account Security        Order.ID Trade.Date Trade.Time Side
## 1 MRTTRADING1      IWF   '43281156775'  11/7/2014   15:58:35    1
## 2 MRTTRADING1      XOP  '309569129127'  11/7/2014   15:58:36    1
## 3 MRTTRADING1      IWF  '730475924135'  11/7/2014   15:58:37    1
## 4 MRTTRADING1      IWF '1409080756903'  11/7/2014   15:58:38    1
## 5 MRTTRADING1      IWF '2422693038759'  11/7/2014   15:58:40    1
## 6 MRTTRADING1      IWF '3436305320615'  11/7/2014   15:58:42    1
##   Liquidity Route Quantity Price Lime.Fee ECN.Fee ACT.Fee SEC.Fee NASD.Fee
## 1   Removed BATSY      100 94.44     0.08   -0.16       0       0        0
## 2   Removed BATSY      100 61.62     0.08   -0.16       0       0        0
## 3   Removed BATSY      100 94.44     0.08   -0.16       0       0        0
## 4   Removed BATSY      100 94.44     0.08   -0.16       0       0        0
## 5   Removed BATSY      100 94.44     0.08   -0.16       0       0        0
## 6   Removed BATSY      100 94.44     0.08   -0.16       0       0        0
##   Rounded.Order.Commission
## 1                    -0.08
## 2                    -0.08
## 3                    -0.08
## 4                    -0.08
## 5                    -0.08
## 6                    -0.08
```

```
# 11.6.2014 Transactions
file2 <- read.csv("~/Downloads/account-summary-MRTTRADING1-20141106.csv")


head(file2)
```

```
##       Account Security        Order.ID Trade.Date Trade.Time Side
## 1 MRTTRADING1      GDX '1383310887591'  11/6/2014   15:58:38    3
## 2 MRTTRADING1      GDX '1400490756775'  11/6/2014   15:58:38    3
## 3 MRTTRADING1      GDX '1417670625959'  11/6/2014   15:58:38    3
## 4 MRTTRADING1      EWY '2757700422311'  11/6/2014   15:58:41    1
## 5 MRTTRADING1      EWY '3049758198439'  11/6/2014   15:58:41    1
## 6 MRTTRADING1      GDX '3986061068967'  11/6/2014   15:58:43    3
##   Liquidity Route Quantity Price Lime.Fee ECN.Fee ACT.Fee SEC.Fee NASD.Fee
## 1   Removed BATSY      100 17.37   0.0800 -0.1600       0    0.04     0.02
## 2   Removed EDGAA      100 17.37   0.0800 -0.0200       0    0.04     0.02
## 3   Removed   BSX      100 17.37   0.0800 -0.0400       0    0.04     0.02
## 4   Removed EDGAA       37 56.68   0.0296 -0.0074       0    0.00     0.00
## 5   Removed EDGAA      100 56.68   0.0800 -0.0200       0    0.00     0.00
## 6   Removed EDGAA      100 17.37   0.0800 -0.0200       0    0.04     0.02
##   Rounded.Order.Commission
## 1                    -0.02
## 2                     0.12
## 3                     0.10
## 4                     0.03
## 5                     0.06
## 6                     0.12
```

Part2 - Bring the Data into R

Here I will append the transactions from 11.6.2014 to the transactions from 11.7.2014 (imported above) and save all transactions as one data set. This will help me compute trade analytic queries on all transactions over the two days including distribution of trades over symbols, commissions paid and realized pnl from avg buy and sell values.

```
# combine transaction data from 11.6.2014 and 11.7.2014 into one data set
allTransactions <- rbind(file1,file2)

# save the data set as an .RData file.
save(allTransactions, file = "allTx.RData")
```

The structure of the allTx.RData data set consists of 16 column variables including Account, Security, OrderID, TradeDate, TradeTime, Side, Liquidity, Route, Quantity, Price, LimeFee, ECNFee, ACTFee, SECFee, NASDFee, RoundedOrderCommission. The data set consists of 2,226 observations of these 16 variables. Over those two days 11.6.2014and 11.7.2014 trades were made over 10 accounts in 147 different securities routed to 7 different exchanges/ecns. This is broken down as follows:

```
str(allTransactions)
```

```
## 'data.frame':    2226 obs. of  16 variables:
##  $ Account                : Factor w/ 10 levels "MRTTRADING1",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Security               : Factor w/ 147 levels "ACE","ADBE","AES",..: 40 104 40 40 40 40 40 42 103
##  $ Order.ID               : Factor w/ 2102 levels "'1005353829474'",..: 875 822 968 356 790 835 866
##  $ Trade.Date             : Factor w/ 2 levels "11/7/2014","11/6/2014": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Trade.Time             : Factor w/ 40 levels "10:26:19","10:26:27",..: 16 17 18 19 20 21 22 23 24
##  $ Side                   : int  1 1 1 1 1 1 1 1 3 3 ...
##  $ Liquidity              : Factor w/ 6 levels "Closing-Auction",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Route                  : Factor w/ 7 levels "ARCA","BATSY",..: 2 2 2 2 2 2 2 5 2 4 ...
##  $ Quantity               : int  100 100 100 100 100 100 100 100 100 100 ...
##  $ Price                  : num  94.4 61.6 94.4 94.4 94.4 ...
```

```
##  $ Lime.Fee                : num  0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 ...
##  $ ECN.Fee                 : num  -0.16 -0.16 -0.16 -0.16 -0.16 -0.16 -0.16 -0.02 -0.16 -0.04 ...
##  $ ACT.Fee                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SEC.Fee                 : num  0 0 0 0 0 0 0 0 0.08 0.08 ...
##  $ NASD.Fee                : num  0 0 0 0 0 0 0 0 0.02 0.02 ...
##  $ Rounded.Order.Commission: num  -0.08 -0.08 -0.08 -0.08 -0.08 -0.08 -0.08 0.06 0.02 0.14 ...
```

Part 3- Bring Data into PostgreSQL

First, I create a new pgSQL database called tradingTxs that will hold each day's transactions where each day is represented by data tables each with 16 fixed column variables where the number of rows per table (or trades per day) varies depending on that day's market activity.

– Database: "tradingTxs"

– DROP DATABASE "tradingTxs";

CREATE DATABASE "tradingTxs" WITH OWNER = postgres ENCODING = 'UTF8' TABLESPACE = pg_default LC_COLLATE = 'C' LC_CTYPE = 'C' CONNECTION LIMIT = -1;

Next I create a table for the transactions made on the first of the two days that I'm using for this project. After creating the table 'firstday' I instantiate column variables and declare data types.

– Table: firstday

– DROP TABLE firstday;

CREATE TABLE firstday ( "Account" text, "Security" text, "Order.ID" text, "Trade.Date" text, "Trade.Time" text, "Side" text, "Liquidity" text, "Route" text, "Quantity" text, "Price" text, "Lime.Fee" text, "ECN.Fee" text, "ACT.Fee" text, "SEC.Fee" text, "NASD.Fee" text, "Rounded.Order.Commission" text ) WITH ( OIDS=FALSE ); ALTER TABLE firstday OWNER TO postgres;

Next I create another table called 'secondday' in similar fashion to the 'firstday' table created above.

Next I want to append 'secondday' to 'firstday' into one file and output that query as a csv file called 'alltrades.csv' so that I can bring that data into R later (for example).

COPY (SELECT * FROM firstday UNION ALL SELECT * FROM secondday) to '/Users/Shared/alltrades.csv' With CSV;

Finally I create a new table called 'alltrades' from which I can work to perform additional queries as per my use case.

CREATE TABLE alltrades AS SELECT * FROM firstday UNION ALL SELECT * FROM secondday

Some Output:

SELECT * FROM firstday

"Account";"Security";"Order ID";"Trade Date";"Trade Time";"Side";"Liquidity";"Route";"Quantity";"Price";"Lime Fee";"ECN Fee";"ACT Fee";"SEC Fee";"NASD Fee";"Rounded Order Commission" "MRTTRADING1";"GDX";" '1383310887591' ";"6-Nov-14";"15:58:38";"3";"Removed";"BATSY";"100";"17.37";"0.08";"-0.16";"0";"0.04";"0.02";"-0.02" "MRTTRADING1";"GDX";" '1400490756775' ";"6-Nov-14";"15:58:38";"3";"Removed";"EDGAA
0.02";"0";"0.04";"0.02";"0.12" "MRTTRADING1";"GDX";" '1417670625959' ";"6-Nov-14";"15:58:38";"3";"Removed";"BSX";"1
0.04";"0";"0.04";"0.02";"0.1" "MRTTRADING1";"EWY";" '2757700422311' ";"6-Nov-14";"15:58:41";"1";"Removed";"EDGAA"
0.0074";"0";"0";"0";"0.03"

SELECT * FROM secondday

"Account";"Security";"Order ID";"Trade Date";"Trade Time";"Side";"Liquidity";"Route";"Quantity";"Price";"Lime Fee";"ECN Fee";"ACT Fee";"SEC Fee";"NASD Fee";"Rounded Order Commission" "MRTTRADING1";"IWF";" '43281156775' ";"11-7-14";"15:58:35";"1";"Removed";"BATSY";"100";"94.44";"0.08";"-0.16";"0";"0";"0";"-0.08" "MRTTRADING1";"XOP";" '309569129127' ";"11-7-14";"15:58:36";"1";"Removed";"BATSY";"100";"

0.16";"0";"0";"0";"-0.08" "MRTTRADING1";"IWF";" '730475924135' ";"11-7-14";"15:58:37";"1";"Removed";"BATSY";"100";"9
0.16";"0";"0";"0";"-0.08" "MRTTRADING1";"IWF";" '1409080756903' ";"11-7-14";"15:58:38";"1";"Removed";"BATSY";"100";"
0.16";"0";"0";"0";"-0.08"

SELECT * FROM alltrades

"Account";"Security";"Order ID";"Trade Date";"Trade Time";"Side";"Liquidity";"Route";"Quantity";"Price";"Lime Fee";"ECN Fee";"ACT Fee";"SEC Fee";"NASD Fee";"Rounded Order Commission" "MRTTRADING1";"GDX";" '1383310887591' ";"6-Nov-14";"15:58:38";"3";"Removed";"BATSY";"100";"17.37";"0.08";"-0.16";"0";"0.04";"0.02";"-0.02" "MRTTRADING1";"GDX";" '1400490756775' ";"6-Nov-14";"15:58:38";"3";"Removed";"EDGAA
0.02";"0";"0.04";"0.02";"0.12" "MRTTRADING1";"GDX";" '1417670625959' ";"6-Nov-14";"15:58:38";"3";"Removed";"BSX";"1
0.04";"0";"0.04";"0.02";"0.1" "MRTTRADING1";"EWY";" '2757700422311' ";"6-Nov-14";"15:58:41";"1";"Removed";"EDGAA"
0.0074";"0";"0";"0";"0.03"

Part 4- Bring Data into MongoDB

First I connect to mongo.

microstearchmbp:Shared MicrostrRes$ cd /usr/local/Cellar/mongodb/2.6.5/bin/db microstearchmbp:db MicrostrRes$ mkdir /usr/local/Cellar/mongodb/2.6.5/bin/db mkdir: /usr/local/Cellar/mongodb/2.6.5/bin/db: File exists microstearchmbp:db MicrostrRes$ cd . . microstearchmbp:db MicrostrRes$ mongod -dbpath /usr/local/Cellar/mongodb/2.6.5/bin/db 2014-11-09T20:15:51.507-0500 [initandlisten] MongoDB starting : pid=2941 port=27017 dbpath=/usr/local/Cellar/mongodb/2.6.5/bin/db 64-bit host=Microstructure-Research-MacBook-Pro.local

Next I import trade data from 11.6.2014 and 11.7.2014 into a database called tradedata.

Microstructure-Research-MacBook-Pro:db MicrostrRes$ mongoimport -db tradedata -collection firstdd -type csv -headerline -file /Users/MicrostrRes/Downloads/account-summary-MRTTRADING1-20141106.csv connected to: 127.0.0.1 2014-11-09T20:27:57.646-0500 check 9 1034 2014-11-09T20:27:57.676-0500 imported 1033 objects

Microstructure-Research-MacBook-Pro:db MicrostrRes$ mongoimport -db tradedata -collection seconddd -type csv -headerline -file /Users/MicrostrRes/Downloads/account-summary-MRTTRADING1-20141107.csv connected to: 127.0.0.1 2014-11-09T20:28:30.125-0500 check 9 1194 2014-11-09T20:28:30.148-0500 imported 1193 objects

Now I start Mongo.

Microstructure-Research-MacBook-Pro:db MicrostrRes$ mongo MongoDB shell version: 2.6.5 connecting to: test Server has startup warnings: 2014-11-09T20:15:51.507-0500 [initandlisten] 2014-11-09T20:15:51.507-0500 [initandlisten] ** WARNING: soft rlimits too low. Number of files is 256, should be at least 1000 >

Now I want to check to see if the collections 'firstdd' and 'seconddd' are there in the 'tradedata' database.

> show dbs admin (empty) employment 0.078GB itunes 0.078GB local 0.078GB paste_employees 0.078GB pastemployees 0.078GB test 0.078GB tradedata 0.078GB tradesmrt (empty) txss (empty) unitedstates 0.078GB unitedstatesdata 0.078GB unitedstatestest 0.078GB use tradedata switched to db tradedata

> show collections firstdd seconddd system.indexes

Indeed the collections 'firstdd' and 'seconddd' are in the 'tradedata' data base.

Now I'll append the 'firstdd' collection to the 'seconddd' collection to get one collection of documents. I accomplish this using the function below to loop through each document in the source file 'firstdd' and appending each document to the target file 'seconddd'.

db.firstdd.find().forEach(function(obj){ db.seconddd.insert(obj) });

Now we see that 'seconddd' has 2,226 documents while 'firstdd' still has 1033. 'second ddd' had 1,193 documents before calling the function above.

db.firstdd.count() 1033 db.seconddd.count() 2226

Here is some output from the combined collection 'secondddd' which includes data from 11.6.2014 and 11.7.2014.

db.seconddd.find().pretty()

{ "_id" : ObjectId("546014be601b52b992ebaddf"), "Account" : "MRTTRADING1", "Security" : "IWF", "Order ID" : " '43281156775' ", "Trade Date" : "11/7/2014", "Trade Time" : "15:58:35", "Side" : 1, "Liquidity" : "Removed", "Route" : "BATSY", "Quantity" : 100, "Price" : 94.44, "Lime Fee" : 0.08, "ECN Fee" : -0.16, "ACT Fee" : 0, "SEC Fee" : 0, "NASD Fee" : 0, "Rounded Order Commission" : -0.08 } { "_id" : ObjectId("546014be601b52b992ebade0"), "Account" : "MRTTRADING1", "Security" : "XOP", "Order ID" : " '309569129127' ", "Trade Date" : "11/7/2014", "Trade Time" : "15:58:36", "Side" : 1, "Liquidity" : "Removed", "Route" : "BATSY", "Quantity" : 100, "Price" : 61.62, "Lime Fee" : 0.08, "ECN Fee" : -0.16, "ACT Fee" : 0, "SEC Fee" : 0, "NASD Fee" : 0, "Rounded Order Commission" : -0.08 } { "_id" : ObjectId("546014be601b52b992ebade1"), "Account" : "MRTTRADING1", "Security" : "IWF", "Order ID" : " '730475924135' ", "Trade Date" : "11/7/2014", "Trade Time" : "15:58:37", "Side" : 1, "Liquidity" : "Removed", "Route" : "BATSY", "Quantity" : 100, "Price" : 94.44, "Lime Fee" : 0.08, "ECN Fee" : -0.16, "ACT Fee" : 0, "SEC Fee" : 0, "NASD Fee" : 0, "Rounded Order Commission" : -0.08 } { "_id" : ObjectId("546014be601b52b992ebade2"), "Account" : "MRTTRADING1", "Security" : "IWF", "Order ID" : " '1409080756903' ", "Trade Date" : "11/7/2014", "Trade Time" : "15:58:38", "Side" : 1, "Liquidity" : "Removed", "Route" : "BATSY", "Quantity" : 100, "Price" : 94.44, "Lime Fee" : 0.08, "ECN Fee" : -0.16, "ACT Fee" : 0, "SEC Fee" : 0, "NASD Fee" : 0, "Rounded Order Commission" : -0.08 } { "_id" : ObjectId("546014be601b52b992ebade3"), "Account" : "MRTTRADING1", "Security" : "IWF", "Order ID" : " '2422693038759' ", "Trade Date" : "11/7/2014", "Trade Time" : "15:58:40", "Side" : 1, "Liquidity" : "Removed", "Route" : "BATSY", "Quantity" : 100, "Price" : 94.44, "Lime Fee" : 0.08, "ECN Fee" : -0.16, "ACT Fee" : 0, "SEC Fee" : 0, "NASD Fee" : 0, "Rounded Order Commission" : -0.08 } { "_id" : ObjectId("546014be601b52b992ebade4"), "Account" : "MRTTRADING1", "Security" : "IWF", "Order ID" : " '3436305320615' ", "Trade Date" : "11/7/2014", "Trade Time" : "15:58:42", "Side" : 1, "Liquidity" : "Removed", "Route" : "BATSY", "Quantity" : 100, "Price" : 94.44, "Lime Fee" : 0.08, "ECN Fee" : -0.16, "ACT Fee" : 0, "SEC Fee" : 0, "NASD Fee" : 0, "Rounded Order Commission" : -0.08 } { "_id" : ObjectId("546014be601b52b992ebade5"), "Account" : "MRTTRADING1", "Security" : "IWF", "Order ID" : " '4114910153383' ", "Trade Date" : "11/7/2014", "Trade Time" : "15:58:44", "Side" : 1, "Liquidity" : "Removed", "Route" : "BATSY", "Quantity" : 100, "Price" : 94.44, "Lime Fee" : 0.08, "ECN Fee" : -0.16, "ACT Fee" : 0, "SEC Fee" : 0, "NASD Fee" : 0, "Rounded Order Commission" : -0.08 } { "_id" : ObjectId("546014be601b52b992ebade6"), "Account" : "MRTTRADING1", "Security" : "IYR", "Order ID" : " '8074870000295' ", "Trade Date" : "11/7/2014", "Trade Time" : "15:58:51", "Side" : 1, "Liquidity" : "Removed", "Route" : "EDGAA", "Quantity" : 100, "Price" : 75.18, "Lime Fee" : 0.08, "ECN Fee" : -0.02, "ACT Fee" : 0, "SEC Fee" : 0, "NASD Fee" : 0, "Rounded Order Commission" : 0.06 } { "_id" : ObjectId("546014be601b52b992ebade7"), "Account" : "MRTTRADING1", "Security" : "XME", "Order ID" : " '8521546599079' ", "Trade Date" : "11/7/2014", "Trade Time" : "15:58:52", "Side" : 3, "Liquidity" : "Removed", "Route" : "BATSY", "Quantity" : 100, "Price" : 36.2, "Lime Fee" : 0.08, "ECN Fee" : -0.16, "ACT Fee" : 0, "SEC Fee" : 0.08, "NASD Fee" : 0.02, "Rounded Order Commission" : 0.02 } { "_id" : ObjectId("546014be601b52b992ebade8"), "Account" : "MRTTRADING1", "Security" : "XME", "Order ID" : " '8555906337447' ", "Trade Date" : "11/7/2014", "Trade Time" : "15:58:52", "Side" : 3, "Liquidity" : "Removed", "Route" : "BSX", "Quantity" : 100, "Price" : 36.2, "Lime Fee" : 0.08, "ECN Fee" : -0.04, "ACT Fee" : 0, "SEC Fee" : 0.08, "NASD Fee" : 0.02, "Rounded Order Commission" : 0.14 } { "_id" : ObjectId("546014be601b52b992ebade9"), "Account" : "MRTTRADING1", "Security" : "IWM", "Order ID" : " '8616035879591' ", "Trade Date" : "11/7/2014", "Trade Time" : "15:58:52", "Side" : 1, "Liquidity" : "Removed", "Route" : "BATSY", "Quantity" : 100, "Price" : 116.58, "Lime Fee" : 0.08, "ECN Fee" : -0.16, "ACT Fee" : 0, "SEC Fee" : 0, "NASD Fee" : 0, "Rounded Order Commission" : -0.08 } { "_id" : ObjectId("546014be601b52b992ebadea"), "Account" : "MRTTRADING1", "Security" : "IWM", "Order ID" : " '8641805683367' ", "Trade Date" : "11/7/2014", "Trade Time" : "15:58:52", "Side" : 1, "Liquidity"

: “Removed”, “Route” : “EDGAA”, “Quantity” : 100, “Price” : 116.58, “Lime Fee” : 0.08, “ECN Fee” : -0.02, “ACT Fee” : 0, “SEC Fee” : 0, “NASD Fee” : 0, “Rounded Order Commission” : 0.06 } { “_id" : ObjectId(“546014be601b52b992ebadeb”), “Account” : “MRTTRADING1”, “Security” : “IWM”, “Order ID” : “ ‘8658985552551’ ”, “Trade Date” : “11/7/2014”, “Trade Time” : “15:58:52”, “Side” : 1, “Liquidity” : “Removed”, “Route” : “BSX”, “Quantity” : 100, “Price” : 116.58, “Lime Fee” : 0.08, “ECN Fee” : -0.04, “ACT Fee” : 0, “SEC Fee” : 0, “NASD Fee” : 0, “Rounded Order Commission” : 0.04 } { “_id" : ObjectId(“546014be601b52b992ebadec”), “Account” : “MRTTRADING1”, “Security” : “TLT”, “Order ID” : “ ‘8684755356327’ ”, “Trade Date” : “11/7/2014”, “Trade Time” : “15:58:52”, “Side” : 1, “Liquidity” : “Removed”, “Route” : “BATSY”, “Quantity” : 100, “Price” : 119.72, “Lime Fee” : 0.08, “ECN Fee” : -0.16, “ACT Fee” : 0, “SEC Fee” : 0, “NASD Fee” : 0, “Rounded Order Commission” : -0.08 } { “_id" : ObjectId(“546014be601b52b992ebaded”), “Account” : “MRTTRADING1”, “Security” : “TLT”, “Order ID” : “ ‘8701935225511’ ”, “Trade Date” : “11/7/2014”, “Trade Time” : “15:58:52”, “Side” : 1, “Liquidity” : “Removed”, “Route” : “EDGAA”, “Quantity” : 100, “Price” : 119.72, “Lime Fee” : 0.08, “ECN Fee” : -0.02, “ACT Fee” : 0, “SEC Fee” : 0, “NASD Fee” : 0, “Rounded Order Commission” : 0.06 } { “_id" : ObjectId(“546014be601b52b992ebadee”), “Account” : “MRTTRADING1”, “Security” : “TLT”, “Order ID” : “ ‘8727705029287’ ”, “Trade Date” : “11/7/2014”, “Trade Time” : “15:58:52”, “Side” : 1, “Liquidity” : “Removed”, “Route” : “BSX”, “Quantity” : 100, “Price” : 119.72, “Lime Fee” : 0.08, “ECN Fee” : -0.04, “ACT Fee” : 0, “SEC Fee” : 0, “NASD Fee” : 0, “Rounded Order Commission” : 0.04 } { “_id" : ObjectId(“546014be601b52b992ebadef”), “Account” : “MRTTRADING1”, “Security” : “GDX”, “Order ID” : “ ‘8787834571431’ ”, “Trade Date” : “11/7/2014”, “Trade Time” : “15:58:52”, “Side” : 3, “Liquidity” : “Removed”, “Route” : “BATSY”, “Quantity” : 100, “Price” : 18.61, “Lime Fee” : 0.08, “ECN Fee” : -0.16, “ACT Fee” : 0, “SEC Fee” : 0.04, “NASD Fee” : 0.02, “Rounded Order Commission” : -0.02 } { “_id" : ObjectId(“546014be601b52b992ebadf0”), “Account” : “MRTTRADING1”, “Security” : “GDX”, “Order ID” : “ ‘8805014440615’ ”, “Trade Date” : “11/7/2014”, “Trade Time” : “15:58:52”, “Side” : 3, “Liquidity” : “Removed”, “Route” : “EDGAA”, “Quantity” : 100, “Price” : 18.61, “Lime Fee” : 0.08, “ECN Fee” : -0.02, “ACT Fee” : 0, “SEC Fee” : 0.04, “NASD Fee” : 0.02, “Rounded Order Commission” : 0.12 } { “_id" : ObjectId(“546014be601b52b992ebadf1”), “Account” : “MRTTRADING1”, “Security” : “GDX”, “Order ID” : “ ‘8822194309799’ ”, “Trade Date” : “11/7/2014”, “Trade Time” : “15:58:52”, “Side” : 3, “Liquidity” : “Removed”, “Route” : “BSX”, “Quantity” : 100, “Price” : 18.61, “Lime Fee” : 0.08, “ECN Fee” : -0.04, “ACT Fee” : 0, “SEC Fee” : 0.04, “NASD Fee” : 0.02, “Rounded Order Commission” : 0.1 } { “_id" : ObjectId(“546014be601b52b992ebadf2”), “Account” : “MRTTRADING1”, “Security” : “LQD”, “Order ID” : “ ‘8882323851943’ ”, “Trade Date” : “11/7/2014”, “Trade Time” : “15:58:52”, “Side” : 1, “Liquidity” : “Removed”, “Route” : “BATSY”, “Quantity” : 100, “Price” : 119.06, “Lime Fee” : 0.08, “ECN Fee” : -0.16, “ACT Fee” : 0, “SEC Fee” : 0, “NASD Fee” : 0, “Rounded Order Commission” : -0.08 } Type “it” for more >

. . .

{ “_id" : ObjectId(“5460149d601b52b992eba9fa”), “Account” : “MRTTRADING1”, “Security” : “EWW”, “Order ID” : “ ‘9122841954983’ ”, “Trade Date” : “11/6/2014”, “Trade Time” : “15:58:52”, “Side” : 1, “Liquidity” : “Removed”, “Route” : “BATSY”, “Quantity” : 100, “Price” : 67.2, “Lime Fee” : 0.08, “ECN Fee” : -0.16, “ACT Fee” : 0, “SEC Fee” : 0, “NASD Fee” : 0, “Rounded Order Commission” : -0.08 } { “_id" : ObjectId(“5460149d601b52b992eba9fb”), “Account” : “MRTTRADING1”, “Security” : “EWW”, “Order ID” : “ ‘9148611758759’ ”, “Trade Date” : “11/6/2014”, “Trade Time” : “15:58:52”, “Side” : 1, “Liquidity” : “Removed”, “Route” : “EDGAA”, “Quantity” : 100, “Price” : 67.2, “Lime Fee” : 0.08, “ECN Fee” : -0.02, “ACT Fee” : 0, “SEC Fee” : 0, “NASD Fee” : 0, “Rounded Order Commission” : 0.06 } { “_id" : ObjectId(“5460149d601b52b992eba9fc”), “Account” : “MRTTRADING1”, “Security” : “EWW”, “Order ID” : “ ‘9165791627943’ ”, “Trade Date” : “11/6/2014”, “Trade Time” : “15:58:52”, “Side” : 1, “Liquidity” : “Removed”, “Route” : “BSX”, “Quantity” : 100, “Price” : 67.2, “Lime Fee” : 0.08, “ECN Fee” : -0.04, “ACT Fee” : 0, “SEC Fee” : 0, “NASD Fee” : 0, “Rounded Order Commission” : 0.04 } { “_id" : ObjectId(“5460149d601b52b992eba9fd”), “Account” : “MRTTRADING1”, “Security” : “GDX”, “Order ID” : “ ‘9234511104679’ ”, “Trade Date” : “11/6/2014”, “Trade Time” : “15:58:53”, “Side” : 3, “Liquidity” : “Removed”, “Route” : “BATSY”, “Quantity” : 100, “Price” : 17.35, “Lime Fee” : 0.08, “ECN Fee” : -0.16, “ACT Fee” : 0, “SEC Fee” : 0.04, “NASD Fee” : 0.02, “Rounded Order Commission” : -0.02 } { “_id" : ObjectId(“5460149d601b52b992eba9fe”), “Account” : “MRTTRADING1”, “Security” : “GDX”, “Order ID” : “ ‘9251690973863’ ”, “Trade Date” : “11/6/2014”, “Trade Time” : “15:58:53”, “Side” : 3, “Liquidity”

: “Removed”, “Route” : “EDGAA”, “Quantity” : 100, “Price” : 17.355, “Lime Fee” : 0.08, “ECN Fee” : -0.02, “ACT Fee” : 0, “SEC Fee” : 0.04, “NASD Fee” : 0.02, “Rounded Order Commission” : 0.12 } { “_id" : ObjectId(“5460149d601b52b992eba9ff”), “Account” : “MRTTRADING1”, “Security” : “GDX”, “Order ID” : “ ‘9260280908455’ ”, “Trade Date” : “11/6/2014”, “Trade Time” : “15:58:53”, “Side” : 3, “Liquidity” : “Removed”, “Route” : “BSX”, “Quantity” : 100, “Price” : 17.35, “Lime Fee” : 0.08, “ECN Fee” : -0.04, “ACT Fee” : 0, “SEC Fee” : 0.04, “NASD Fee” : 0.02, “Rounded Order Commission” : 0.1 } { “_id" : ObjectId(“5460149d601b52b992ebaa00”), “Account” : “MRTTRADING1”, “Security” : “LQD”, “Order ID” : “ ‘9311820516007’ ”, “Trade Date” : “11/6/2014”, “Trade Time” : “15:58:53”, “Side” : 1, “Liquidity” : “Removed”, “Route” : “BATSY”, “Quantity” : 100, “Price” : 118.41, “Lime Fee” : 0.08, “ECN Fee” : -0.16, “ACT Fee” : 0, “SEC Fee” : 0, “NASD Fee” : 0, “Rounded Order Commission” : -0.08 } { “_id" : ObjectId(“5460149d601b52b992ebaa01”), “Account” : “MRTTRADING1”, “Security” : “RSX”, “Order ID” : “ ‘9346180254375’ ”, “Trade Date” : “11/6/2014”, “Trade Time” : “15:58:53”, “Side” : 3, “Liquidity” : “Removed”, “Route” : “BATSY”, “Quantity” : 100, “Price” : 20.58, “Lime Fee” : 0.08, “ECN Fee” : -0.16, “ACT Fee” : 0, “SEC Fee” : 0.05, “NASD Fee” : 0.02, “Rounded Order Commission” : -0.01 } { “_id" : ObjectId(“5460149d601b52b992ebaa02”), “Account” : “MRTTRADING1”, “Security” : “RSX”, “Order ID” : “ ‘9363360123559’ ”, “Trade Date” : “11/6/2014”, “Trade Time” : “15:58:53”, “Side” : 3, “Liquidity” : “Removed”, “Route” : “EDGAA”, “Quantity” : 100, “Price” : 20.58, “Lime Fee” : 0.08, “ECN Fee” : -0.02, “ACT Fee” : 0, “SEC Fee” : 0.05, “NASD Fee” : 0.02, “Rounded Order Commission” : 0.13 } { “_id" : ObjectId(“5460149d601b52b992ebaa03”), “Account” : “MRTTRADING1”, “Security” : “IYR”, “Order ID” : “ ‘9423489665703’ ”, “Trade Date” : “11/6/2014”, “Trade Time” : “15:58:53”, “Side” : 1, “Liquidity” : “Removed”, “Route” : “EDGAA”, “Quantity” : 100, “Price” : 75.24, “Lime Fee” : 0.08, “ECN Fee” : -0.02, “ACT Fee” : 0, “SEC Fee” : 0, “NASD Fee” : 0, “Rounded Order Commission” : 0.06 } { “_id" : ObjectId(“5460149d601b52b992ebaa04”), “Account” : “MRTTRADING1”, “Security” : “IYR”, “Order ID” : “ ‘9440669534887’ ”, “Trade Date” : “11/6/2014”, “Trade Time” : “15:58:53”, “Side” : 1, “Liquidity” : “Removed”, “Route” : “BSX”, “Quantity” : 23, “Price” : 75.24, “Lime Fee” : 0.0184, “ECN Fee” : -0.0092, “ACT Fee” : 0, “SEC Fee” : 0, “NASD Fee” : 0, “Rounded Order Commission” : 0.04 }

Part 5: Compare Methods Compare the three approaches for purposes of managing and using your data. You should address any practical concerns you might have for each technology as well as any theoretical advantages or disadvantages for that technology. You can include in your comparison some commentary on ease of use as well.

---

Since the data set that I’m using for this project is extremely well structured all three approaches work fine. However, I believe PostgreSQL and R to be limiting in many ways if my use case were to expand beyond making simple computations on the data (for computing commission costs, distribution of trades across symbols, computing trading volumes and realized pnls from average buy and sell values).

PostgreSQL would become cumbersome and complicated if I were to add unstructured data to the database in the case that I wanted to see how news items historically affected the realized pnl and other factors like trading volume or symbol distribution. The lack of Map-Reduce support would make using PostgreSQL fairly complicated if I were scraping websites across multiple nodes. That said, for processing very structured data like I’m using in this project, PostgreSQL is fast and works well. In particular PostgreSQL supports user defined functions which I don’t believe Mongo does- so that can be a plus. Nevertheless I find the ease of use of PostgreSQL to be less intuitive than Mongo.

I think the best combination would be using Mongo with R because R provides you with the visualization and modelling tools that you need to work on the data- and Mongo provides you with an intuitive, document driven, horizontally scalable platform. For example, my use case for basic trade analytics data set could evolve into a complex intraday backtester and benchmarker. I may be reconciling the transactions from each day with high resolution (100 microsecond) tick and quote data to essentially replay historical performance in markets. If that were the case the sheer size of the market data and the basic data manipulations involved would likely invalidate PostgreSQL and R as acquisition technologies. Add to that the idea of bringing in historical news and social media data. Mongo’s ability to normalize files of different structures into a

document based format is extremely powerful. Further, Map-Reduce and the javascript functions would allow me to perform large scale queries and pipe the results to R for modelling and visualization much faster and more efficiently that I could with PostgreSQL or with R alone.

R is the best for summarizing the structure of data and for visualization/modelling on samples. However, I'm still not sure of how to use R to model massive data that would returned by Mongo queries or how to use R in a distributed computing sense. I do know that using R even for basic computations on smaller data sets - for example merging 200 files with 16 variables and approximately 1k ovbersvations each- takes a lot of time. I don't know how I'd perform computations using functions like dapply on files with tens of millions of rows and hundreds or thousands of columns (for example some kind of optimization).