Name: Sai Lahari Jalaparthi UIN: 658795799

Homework 2

Loading Data into Database:

1. Creation of Database in MySQL Workbench:

```
CREATE DATABASE `TwitterResearch_Fall2014`;
```

2. Loading and cleaning of data in ‘earnings_processed_Sept30_2014_1110pm’ table:

✓ Creation of table ‘earnreldate’

```
Create Table EarnRelDate (

            ticker VARCHAR(8),
            symbol VARCHAR(8),
            cname VARCHAR(40),
            pends DATE,
            pdicity VARCHAR(8),
            anndats DATE,
            anntimes decimal,
            actualdate DATE,
            earnrelease_date DATE,
            earnrelease_time TIME);
```

✓ Loading data into ‘earnreldate’ table

```
LOAD DATA INFILE 'C:/earnings_processed_Sept30_2014_1110pm.csv'  INTO TABLE
EarnRelDate
CHARACTER SET utf8
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\r'
IGNORE 1 LINES
(ticker, symbol, cname, pends, pdicity, anndats, anntimes, actualdate,  earnrelease_date,
earnrelease_time);
```

✓ 41581 rows fetched into ‘earnreldate’ table

```
Select count(*) from EarnRelDate;
```

✓ On total we have 39799 rows where symbol is not null

```
Select * from EarnRelDate where symbol is not null and symbol != "" order by earnrelease_date,
earnrelease_time;
```

✓ To clean data, create table ‘earnreldate2’ with primary key as ticker,earnrelease_date, earnrelease_time and pdicity. Load the distinct data present in ‘earnreldate’ table into ‘earnreldata2’ table

```
Create Table EarnRelDate2 (
            ticker VARCHAR(8),
            symbol VARCHAR(8),
            cname VARCHAR(40),
            pends DATE,
            pdicity VARCHAR(8),
            anndats DATE,
            anntimes decimal,
            actualdate DATE,
            earnrelease_date DATE,
            earnrelease_time TIME,
            PRIMARY KEY (ticker, earnrelease_date,earnrelease_time, pdicity)
) ENGINE=InnoDB;

Insert into EarnRelDate2 (ticker, earnrelease_date,earnrelease_time, pdicity )
Select distinct ticker, earnrelease_date,earnrelease_time, pdicity from EarnRelDate;
```

✓ 40504 rows are inserted into ‘earnreldate2’ table. This implies that there are 1077 duplicates with our defined criteria in the ‘earnreldate’ table

```
Select count(*) from EarnRelDate2;
```

✓ As we doesn’t every column and necessary attributes are ticker, earnrelease_date, earnrelease_time and symbol, I am creating a new table with primary key as ticker,earnrelease_date and earnrelease_time. Loading distinct data into that table

```
Create Table EarnRelDate3 (

            ticker VARCHAR(8),
            symbol VARCHAR(8),
            earnrelease_date DATE,
            earnrelease_time TIME,
            PRIMARY KEY (ticker, earnrelease_date,earnrelease_time)
) ENGINE=InnoDB;

Insert into EarnRelDate3 (ticker, earnrelease_date,earnrelease_time )
Select distinct ticker, earnrelease_date,earnrelease_time from EarnRelDate;
```

✓ 34174 data is loaded into ‘earnreldate3’ table

```
Select count(*) from EarnRelDate3;
```

3. Loading and cleaning of data in ‘TwitterYahoo_INTERM1_Oct4_2013.csv’:

✓ Creation of table ‘tweets’

```
Create Table Tweets (

            smblid VARCHAR(10),
            symbol VARCHAR(8),
            periodnum INT,
            periodnum_inday INT,
            volumestart INT,
            volumeend INT,
            twittermentions INT,
            twitterpermin DECIMAL(10,2),
            averagefollowers DECIMAL(10,2),
            datestart DATE,
            timestart TIME,
            dateend DATE,
            timeend TIME    );
```

- ✓ Loading data into ‘tweets’ table

```
LOAD DATA INFILE 'C:/TwitterYahoo_INTERM1_Oct4_2013.csv'  INTO TABLE  Tweets
CHARACTER SET utf8
FIELDS TERMINATED BY ','
 OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(smblid, symbol, periodnum, periodnum_inday, @volumestart, @volumeend,
@twittermentions, @twitterpersec, @averagefollowers, datestart, timestart,dateend, timeend)
SET volumestart = IF(@volumestart='',0,@volumestart),
volumeend = IF(@volumeend='',0,@volumeend),
twittermentions = IF(@twittermentions='',null,@twittermentions),
twitterpermin = IF(@twitterpersec='',null,60*@twitterpersec),
averagefollowers = IF(@averagefollowers='',null,@averagefollowers);
```

- ✓ 1223906 rows are loaded into ‘tweets’ table

```
select count(*) from Tweets;
```

- ✓ As there are many rows, deleting the duplicate rows and inserting the data into new table tweets2. ‘Tweets2’ table is created with the primary key and using index.

```
Create Table Tweets2 (
        smblid VARCHAR(10),
        periodnum INT,
        periodnum_inday INT,
        datestart DATE,
        timestart TIME,
        dateend DATE,
        timeend TIME,
        PRIMARY KEY (smblid, periodnum),
        INDEX tstart (smblid, datestart, timestart),
        INDEX tend (smblid, dateend, timeend));
```

```
Insert into Tweets2 (smblid,periodnum, periodnum_inday, datestart, timestart, dateend, timeend)
Select distinct smblid,periodnum, periodnum_inday, datestart, timestart, dateend, timeend from Tweets;
```

✓ 1223906 rows are loaded into ‘tweets2’ table. Even though there are no duplicates between tweets and tweets2, performance of queries can be enhanced using tweets2 table by using index

```
select count(*) from Tweets2;
```

4. Creation of new table ‘earnrematched’ and loading the data into table by joining the two tables ‘earnreldate3’ and ‘tweets2’

✓ Creation of new table ‘earnrelmatched’ with primary key as smblid and periodnum

```
Create Table EarnRelMatched (
        ticker VARCHAR(10),
        earnrelease_date DATE,
        earnrelease_time TIME,
        smblid VARCHAR(10),
        periodnum INT,
        periodnum_inday INT,
        datestart DATE,
        timestart TIME,
        dateend DATE,
        timeend TIME,
        PRIMARY KEY (smblid, periodnum)
);
```

✓ Loading data into ‘earnrelmatched’ table by full join of ‘earnreldate3’ table and ‘tweets2’ table using the below query

```
INSERT into EarnRelMatched
Select e.ticker, e.earnrelease_date, e.earnrelease_time,
t.smblid, t.periodnum, t.periodnum_inday, t.datestart, t.timestart, t.dateend, t.timeend
 from Tweets2 t, EarnRelDate3 e
where e.earnrelease_date = t.datestart
and e.earnrelease_time BETWEEN t.timestart AND t.timeend
and e.ticker = t.smblid;
```

✓ 213 rows are loaded into ‘earnrelmatched’ table

```
Select count(*) from EarnRelMatched;
```

5. For calculation of baseline values, we need to know the day name of the ‘earnrelease_date’ attribute in the ‘earnrelmatched’ table and ‘datestart’ attribute in the ‘tweets’ table.

So, I am creating a new table which includes date and day name from May 1, 2012 to September 18, 2013 with the primary key as Date

- ✓
  ```
  create table datetable (date1 DATE, weekname varchar(10), PRIMARY KEY(date1));
  ```
- ✓
  ```
  LOAD DATA INFILE 'C:/datetable.csv' INTO TABLE datetable FIELDS TERMINATED BY ',' ENCLOSED
  BY '"'LINES TERMINATED BY '\n'
  IGNORE 1 ROWS;
  ```
- ✓ 506 rows are loaded into datetable
  ```
  select count(date1) from datetable;
  ```

6. As the ‘earnrelmatched’ table doesn’t contain the column which includes day name for the ‘earnrelease_date’ attribute, I am creating a new table earnrelmatched1 with attributes same as ‘earnrelmatched’ table but with extra attribute ‘earnrelease_weekname’.

- ✓ Creation of ‘earnrelmatched’ table
  ```
  Create Table EarnRelMatched1 (
          ticker VARCHAR(10),
          earnrelease_date DATE,
          earnrelease_time TIME,
          smblid VARCHAR(10),
          periodnum INT,
          periodnum_inday INT,
          datestart DATE,
          timestart TIME,
          dateend DATE,
          timeend TIME,
          earnrelease_weekname varchar(10),
          PRIMARY KEY (smblid, periodnum)
  );
  ```

- ✓ Insert values into earnrelmatched1 by joining the ‘datetable’ table and ‘earnrelmatched’ table on ‘earnrelease_date’ attribute from ‘earnrelmatched’ table and ‘date1’ attribute from ‘datetable’ table
  ```
  insert into earnrelmatched1 select e.ticker,e.earnrelease_date,e.earnrelease_time,
  e.smblid,e.periodnum,e.periodnum_inday,
  e.datestart,e.timestart,e.dateend,e.timeend,d.weekname from earnrelmatched e,datetable d
  where e.earnrelease_date=d.date1;
  ```

- ✓ For all 213 rows, day name is added
  ```
  select * from earnrelmatched1;
  ```

7. As the ‘tweets’ table doesn’t contain the column which includes day name for the datestart, primary keys and index, I am creating a new table tweets3 with attributes that are necessary and by using primary keys and index.

- ✓ Creation of ‘tweets3’ table

```
Create Table Tweets3 (
        smblid VARCHAR(10),
        periodnum INT,
        periodnum_inday INT,
        volumestart INT,
        volumeend INT,
        twitterpermin DECIMAL(10,2),
        datestart DATE,
        timestart TIME,
        dateend DATE,
        timeend TIME,
        Datestartname varchar(10),
        PRIMARY KEY (smblid, periodnum),
        INDEX tstart (smblid, datestart, timestart),
        INDEX tend (smblid, dateend, timeend)
);
```

- ✓ Insert values into tweets3 by joining the ‘datetable’ table and ‘tweets’ table on ‘datestart’ attribute from ‘tweets’ table and ‘date1’ attribute from ‘datetable’

```
insert into tweets3
(smblid,periodnum,periodnum_inday,volumestart,volumeend,twitterpermin,datestart,timestart
,dateend,timeend,Datestartname) select t.smblid, t.periodnum,
t.periodnum_inday,t.volumestart,t.volumeend,t.twitterpermin, t.datestart,t.timestart,
t.dateend, t.timeend,d.weekname from tweets t,datetable d where t.datestart=d.date1;
```

- ✓ 1223906 rows are loaded into ‘tweets3’ table.

```
select count(*) from tweets3;
```

QUESTION 1:

Different firms represented in at least 30 days:

As the tweets table contains data for the firms with twitterpermin as zero also, I had considered only the firms that are represented in at least 30 days with twitterpermin not equal to zero

```
select count(*) from (select smblid,count(distinct datestart) as c from tweets where twitterpermin!=0
group by smblid having c>=30)t;
```

Result: 95

QUESTION 2:

- ➢ By observing the data in the ‘earnrelmatched’ table, I identified that one firm have the same earnrelease_date, earnrelease_time but the timestart for that is differed by 01 seconds. This is because for that firm on that there are two different types of tweets differed by 01 seconds on that day. So for calculation of average twitter levels for the firms, I used ‘group by’ with the attributes ‘ticker’, ‘earnrelease_date’, ’earnrelease_time’ and ‘timestart’.

- Since some of the firms may not have data related to tweets at the period interval we are calculating, I am using left outer join to include those firms also.

(a) Average tweet levels in the 40minutes following earnings release:

For calculation of average tweet levels in the 40minutes following earning release time is calculated by getting the periodnum_inday for the firm at that earn release date and time. The 40minutes interval that I have considered is the periodnum_inday+1 to periodnum_day+4 as the tweets are recorded with 10minutes increments.

```
select e.ticker,e.earnrelease_date,e.earnrelease_time,e.timestart,avg(t.twitterpermin) from
EarnRelMatched e left outer join tweets t on t.smblid=e.ticker and
t.datestart=e.earnrelease_date and t.periodnum_inday BETWEEN e.periodnum_inday+1 and
e.periodnum_inday+4 group by e.ticker,e.earnrelease_date,earnrelease_time,e.timestart;
```

Result: Refer to file ‘2a.csv’

Analysis:

1. To analyze which firms doesn’t have data regarding tweets in the ‘tweets’ table after the periodnum_inday for the firm on that earn release date, I had used the below query

   ```
   select ticker,earnrelease_date,earnrelease_time from  EarnRelMatched where
   (ticker,earnrelease_date,earnrelease_time) not in
   (select e.ticker,e.earnrelease_date,e.earnrelease_time from EarnRelMatched e, tweets2
   t where t.smblid=e.ticker
   and t.datestart=e.earnrelease_date and t.periodnum_inday BETWEEN
   e.periodnum_inday+1 and e.periodnum_inday+4
   group by e.ticker,e.earnrelease_date,earnrelease_time,e.timestart)
   ```

   Result set contains 9 firms that doesn’t have any data regarding the tweets after the earn release period.
2. If we observe the results on total there are 30 firms(21 firms with tweets per min as 0 and 9 firms mentioned in above point) that doesn’t have any tweets

(b) Average tweet levels in the 2hours following earnings release:

For calculation of average tweet levels in the 2hours following earning release time is calculated by getting the periodnum_inday for the firm at that earn release date and time. The 120minutes interval that I have considered is the periodnum_inday+1 to periodnum_day+12 as the tweets are recorded with 10minutes increments.

```
select e.ticker,e.earnrelease_date,e.earnrelease_time,e.timestart,avg(t.twitterpermin) from
EarnRelMatched e left outer join tweets t on t.smblid=e.ticker and
t.datestart=e.earnrelease_date and t.periodnum_inday BETWEEN e.periodnum_inday+1 and
e.periodnum_inday+12 group by e.ticker,e.earnrelease_date,earnrelease_time,e.timestart;
```

Result: Refer to file '2b.csv'

Analysis:

1. To analyze which firms doesn't have data regarding tweets in the 'tweets' table after the periodnum_inday for the firm on that earn release date, I had used the below query

   ```
   select ticker,earnrelease_date,earnrelease_time from  EarnRelMatched where
   (ticker,earnrelease_date,earnrelease_time) not in
   (select e.ticker,e.earnrelease_date,e.earnrelease_time from EarnRelMatched e, tweets2
   t where t.smblid=e.ticker
   and t.datestart=e.earnrelease_date and t.periodnum_inday BETWEEN
   e.periodnum_inday+1 and e.periodnum_inday+12
   group by e.ticker,e.earnrelease_date,earnrelease_time,e.timestart)
   ```

   Result set contains 9 firms that doesn't have any data regarding the tweets after the earn release period.
2. On total 28 firms doesn't have any tweets per minute

(c) Average tweet levels in the 24 hours following earnings release:

As we had to calculate average tweet levels for the 24hours following earnings release, periodnum_inday doesn't work. Therefore I am using TIMESTAMP to concatenate date and time. By using the addtime() function I am adding 24 hours to the earn release time which is considered as next day with the same earn release time

```
select e.ticker,e.earnrelease_date,e.earnrelease_time,e.timestart,avg(t.twitterpermin) from
EarnRelMatched e left outer join tweets t on t.smblid=e.ticker and
TIMESTAMP(t.datestart,t.timestart) between
TIMESTAMP(e.earnrelease_date,e.earnrelease_time) and
addtime((TIMESTAMP(e.earnrelease_date,e.earnrelease_time)),'24:00:00') group by
e.ticker,e.earnrelease_date,earnrelease_time,e.timestart;
```

Result: Refer to file '2c.csv'

Analysis:

1. On total there are 9 firms that doesn't have tweets per minute.

(d) Average tweet levels in the 1 week following earnings release:

As we had to calculate average tweet levels for the 1week following earnings release, periodnum_inday doesn’t work. Therefore I am using TIMESTAMP to concatenate date and time. By using the addtime() function I am adding 168 hours to the earn release time which is considered as next week with the same earn release time

```
Select e.ticker, e.earnrelease_date, e.earnrelease_time, e.timestart, avg(t.twitterpermin),
addtime((TIMESTAMP(e.earnrelease_date,e.timestart)),'168:00:00') as timeperiod_endtime
from EarnRelMatched e, tweets t where t.smblid=e.ticker and
TIMESTAMP(t.datestart,t.timestart) between
TIMESTAMP(e.earnrelease_date,e.earnrelease_time) and
addtime((TIMESTAMP(e.earnrelease_date,e.earnrelease_time)),'168:00:00') group by
e.ticker,e.earnrelease_date,earnrelease_time,e.timestart;
```

Result: Refer to file ‘2d.csv’

Analysis:

1. On total there are 9 firms that doesn’t have tweets per minute.

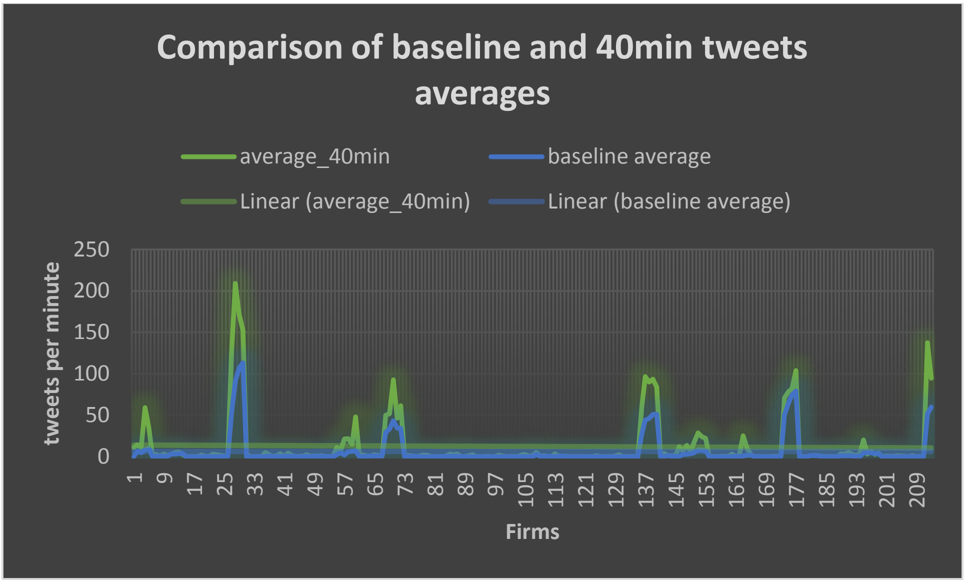
(e and i) Baseline average:

Baseline average is calculated for the firm on the same weekday as the earn release date from May 1, 2012 to earnrelease_date-7 and half hour time period is considered as endtime-10 to endtime+20.

```
Select e.ticker, e.earnrelease_date, e.earnrelease_time, e.timestart, avg(t.twitterpermin),
t.Datestartname from EarnRelMatched1 e left outer join tweets3 t on t.smblid=e.ticker and
t.datestart>=2012-05-01 and t.datestart<=e.earnrelease_date-7 and
t.Datestartname=e.earnrelease_weekname and t.timestart between
subtime(e.timeend,'00:10:00') and addtime(e.timeend,'00:20:00') group by
e.ticker,e.earnrelease_date,earnrelease_time,e.timestart;
```

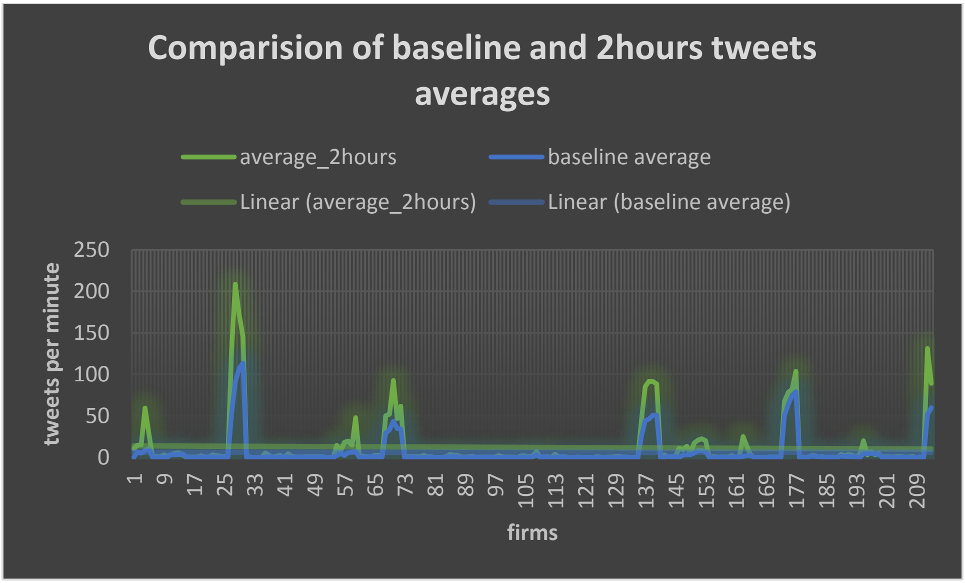
Result: Refer to file ‘baseline tweets per minute.csv’

(f) Comparison of average of tweet levels of 40min with baseline:



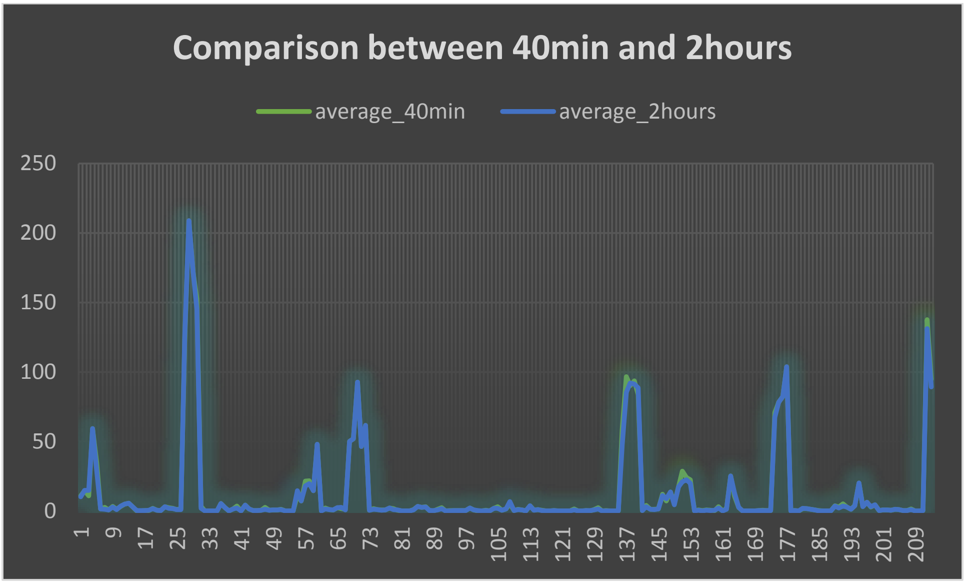
The above graph shows that tweets per minute is higher for period with 40minutes following the earnings release than baseline. Both the 40min and baseline almost follows the same pattern. The trading line for the average of 40min is above the trading line for the baseline. Even though few firms have tweets per minute less than trading line, the number of tweets per minute is higher when we compare it with its baseline

Comparison of average of tweet levels of 2hours with baseline:

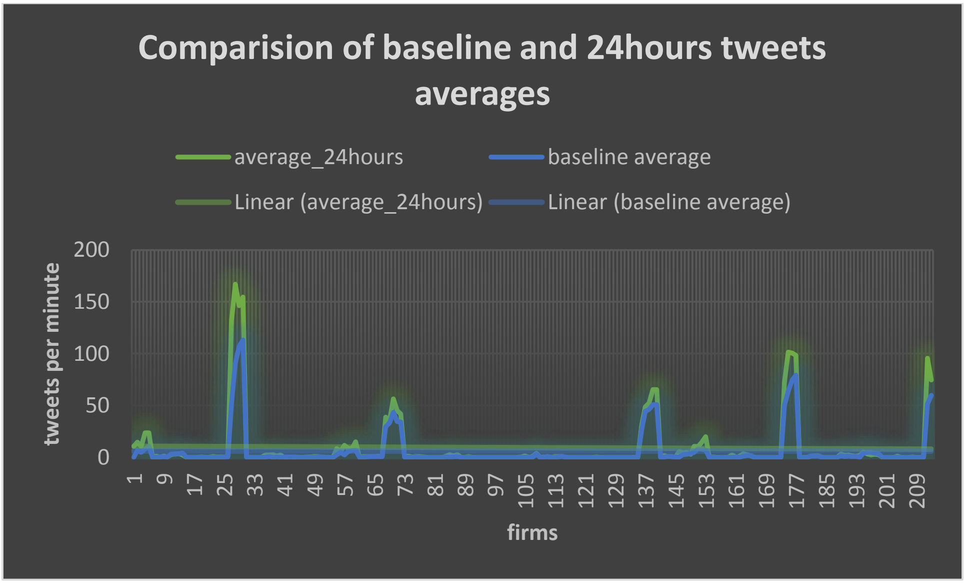


The above graph shows that tweets per minute is higher for period with 2hours following the earnings release than baseline. The trading line for the average of 2hours is above the trading line for the baseline.

If we compare the tweets per minute for 40min and 2hours, it is approximately same.

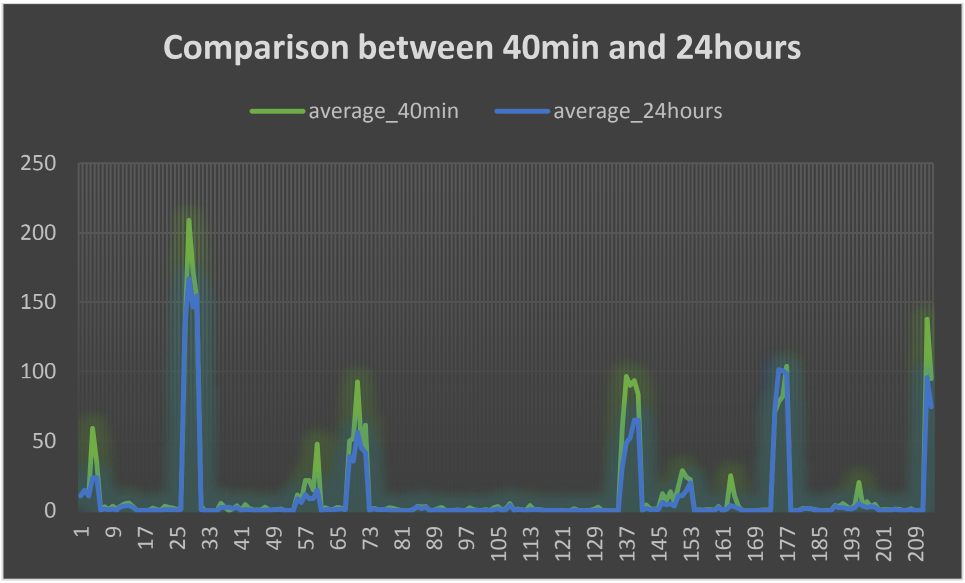


Comparison of average of tweet levels of 24hours with baseline:

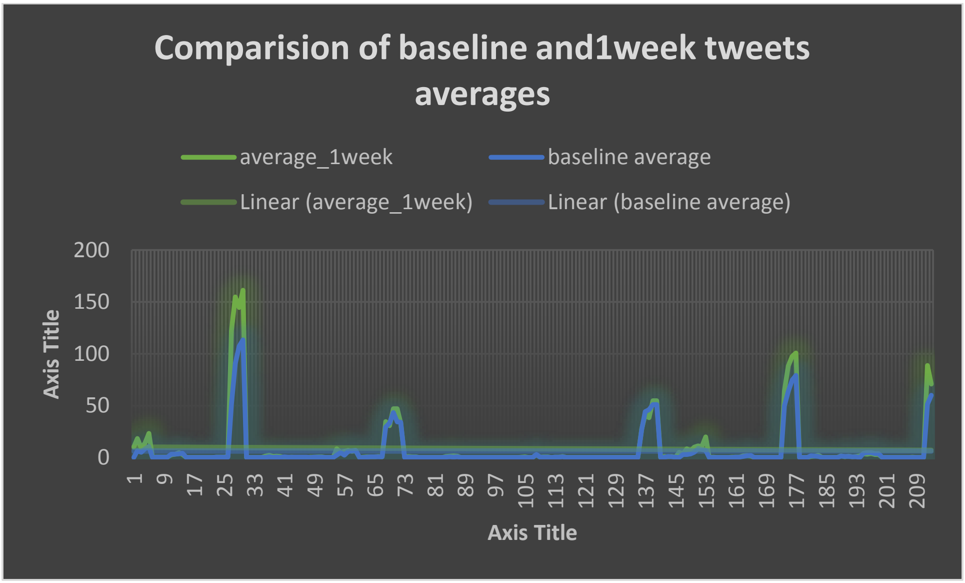


The above graph shows that tweets per minute is higher for period with 24hours following the earnings release than baseline. The trading line for the average of 24hours is above the trading line for the baseline.

If we compare the 24hours with 40min, for some firms the tweets level is higher in period 40min following the earnings release which can be observed from the below graph



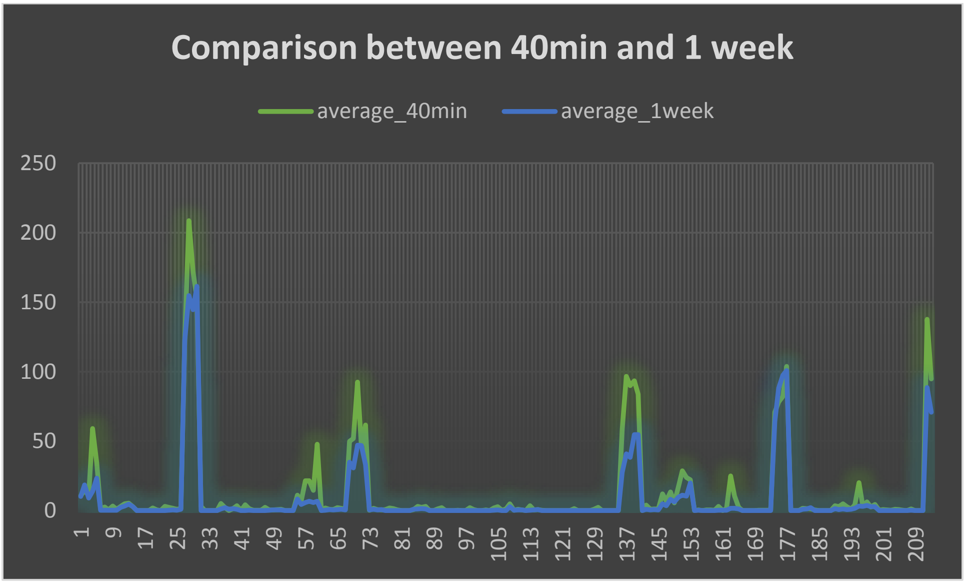
Comparison of average of tweet levels of 1week with baseline:



The above graph shows that for few firms, tweets per minute is higher for period with 1week following the earnings release than baseline. For some firms, the difference between both 1week and baseline is too less. The trading line for the average of 1week is approximately equal the trading line for the baseline.

Overall analysis for tweets per minute:

The tweet level is much higher for the period 40minutes following the earnings release when we compare it with others. If we compare one week with 40minutes, the number of tweets per minute is decreasing.



With the increase of time following the earn release, the number of tweets per minute is decreasing.

QUESTION 3:

(a) Average trading volume levels in the 40minutes following earnings release:

For calculation of average volume levels in the 40minutes following earning release time is calculated by getting the periodnum_inday for the firm at that earn release date and time. The 40minutes interval that I have considered is the periodnum_inday+1 to periodnum_day+4 as the tweets are recorded with 10minutes increments.

```
select e.ticker,e.earnrelease_date,e.earnrelease_time,e.timestart,avg(t.volumeend) from
EarnRelMatched e left outer join tweets t on t.smblid=e.ticker and
t.datestart=e.earnrelease_date and t.periodnum_inday BETWEEN e.periodnum_inday+1 and
e.periodnum_inday+4 group by e.ticker,e.earnrelease_date,earnrelease_time,e.timestart;
```

Result: Refer to file ‘3a.csv’

Analysis:

1. To analyze which firms doesn’t have data regarding tweets in the ‘tweets’ table after the periodnum_inday for the firm on that earn release date, I had used the below query

```
select ticker,earnrelease_date,earnrelease_time from  EarnRelMatched where
(ticker,earnrelease_date,earnrelease_time) not in
(select e.ticker,e.earnrelease_date,e.earnrelease_time from EarnRelMatched e, tweets2
t where t.smblid=e.ticker
and t.datestart=e.earnrelease_date and t.periodnum_inday BETWEEN
e.periodnum_inday+1 and e.periodnum_inday+4
group by e.ticker,e.earnrelease_date,earnrelease_time,e.timestart)
```

Result set contains 9 firms that doesn't have any data regarding the tweets after the earn release period.

2. On total there are 14 firms that has 0 volume end level

(b) Average trading volume levels in the 2hours following earnings release:

For calculation of average volume levels in the 2hours following earning release time is calculated by getting the periodnum_inday for the firm at that earn release date and time. The 120minutes interval that I have considered is the periodnum_inday+1 to periodnum_day+12 as the tweets are recorded with 10minutes increments.

```
select e.ticker,e.earnrelease_date,e.earnrelease_time,e.timestart,avg(t. volumeend) from
EarnRelMatched e left outer join tweets t on t.smblid=e.ticker and
t.datestart=e.earnrelease_date and t.periodnum_inday BETWEEN e.periodnum_inday+1 and
e.periodnum_inday+12 group by e.ticker,e.earnrelease_date,earnrelease_time,e.timestart;
```

Result: Refer to file '3b.csv'

Analysis:

1. To analyze which firms doesn't have data regarding tweets in the 'tweets' table after the periodnum_inday for the firm on that earn release date, I had used the below query

   ```
   select ticker,earnrelease_date,earnrelease_time from  EarnRelMatched where
   (ticker,earnrelease_date,earnrelease_time) not in
   (select e.ticker,e.earnrelease_date,e.earnrelease_time from EarnRelMatched e, tweets2
   t where t.smblid=e.ticker
   and t.datestart=e.earnrelease_date and t.periodnum_inday BETWEEN
   e.periodnum_inday+1 and e.periodnum_inday+12
   group by e.ticker,e.earnrelease_date,earnrelease_time,e.timestart)
   ```

   Result set contains 9 firms that doesn't have any data regarding the tweets after the earn release period.

(c) Average trading volume levels in the 24 hours following earnings release:

As we had to calculate average volume end levels for the 24hours following earnings release, periodnum_inday doesn't work. Therefore I am using TIMESTAMP to concatenate date and time. By using the addtime() function I am adding 24 hours to the earn release time which is considered as next day with the same earn release time

```
select e.ticker,e.earnrelease_date,e.earnrelease_time,e.timestart,avg(t. volumeend) from
EarnRelMatched e left outer join tweets t on t.smblid=e.ticker and
TIMESTAMP(t.datestart,t.timestart) between
TIMESTAMP(e.earnrelease_date,e.earnrelease_time) and
addtime((TIMESTAMP(e.earnrelease_date,e.earnrelease_time)),'24:00:00') group by
e.ticker,e.earnrelease_date,earnrelease_time,e.timestart;
```

Result: Refer to file '3c.csv'

Analysis:

1. To analyze which firms doesn't have data regarding tweets in the 'tweets' table after the periodnum_inday for the firm on that earn release date, I had used the below query

   ```
   select ticker,earnrelease_date,earnrelease_time from  EarnRelMatched where
   (ticker,earnrelease_date,earnrelease_time) not in
   (select e.ticker,e.earnrelease_date,e.earnrelease_time from EarnRelMatched e, tweets2
   t where t.smblid=e.ticker
   and t.datestart=e.earnrelease_date and t.periodnum_inday BETWEEN
   e.periodnum_inday+1 and e.periodnum_inday+12
   group by e.ticker,e.earnrelease_date,earnrelease_time,e.timestart)
   ```

   Result set contains 1 firm that doesn't have any data regarding the tweets after the earn release period.

(d) Average trading volume levels in the 1 week following earnings release:

As we had to calculate average volume end levels for the 1week following earnings release, periodnum_inday doesn't work. Therefore I am using TIMESTAMP to concatenate date and time. By using the addtime() function I am adding 168 hours to the earn release time which is considered as next week with the same earn release time

```
Select e.ticker, e.earnrelease_date, e.earnrelease_time, e.timestart, avg(t. volumeend),
addtime((TIMESTAMP(e.earnrelease_date,e.timestart)),'168:00:00') as timeperiod_endtime
from EarnRelMatched e, tweets t where t.smblid=e.ticker and
TIMESTAMP(t.datestart,t.timestart) between
TIMESTAMP(e.earnrelease_date,e.earnrelease_time) and
addtime((TIMESTAMP(e.earnrelease_date,e.earnrelease_time)),'168:00:00') group by
e.ticker,e.earnrelease_date,earnrelease_time,e.timestart;
```

Result: Refer to file '3d.csv'

Analysis:

1. All firms have the volume end level

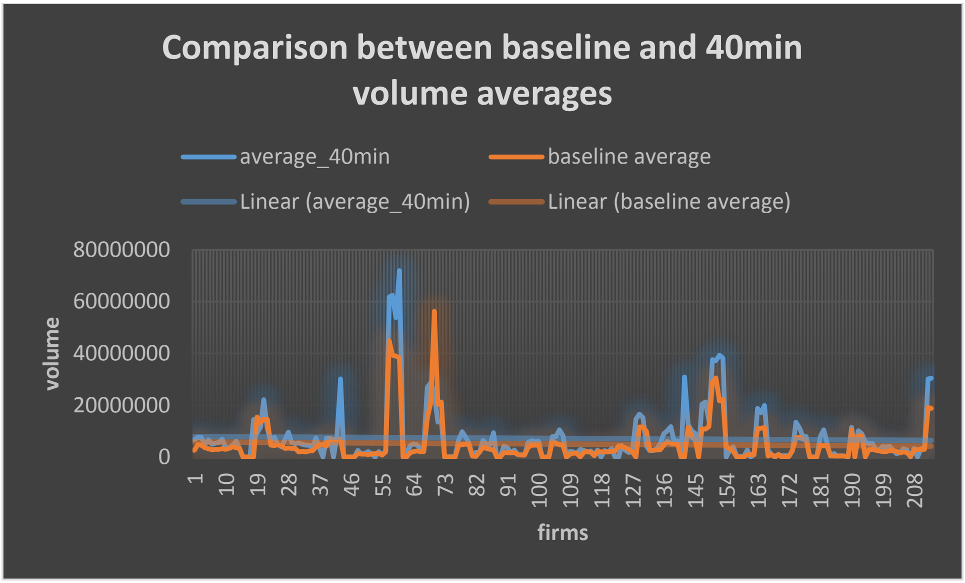
(e and i) Baseline average:

Baseline average is calculated for the firm on the same weekday as the earn release date from May 1, 2012 to earnrelease_date-7 and half hour time period is considered as endtime-10 to endtime+20.

```
Select e.ticker, e.earnrelease_date, e.earnrelease_time, e.timestart, avg(t. volumeend),
t.Datestartname from EarnRelMatched1 e left outer join tweets3 t on t.smblid=e.ticker and
t.datestart>=2012-05-01 and t.datestart<=e.earnrelease_date-7 and
t.Datestartname=e.earnrelease_weekname and t.timestart between
subtime(e.timeend,'00:10:00') and addtime(e.timeend,'00:20:00') group by
e.ticker,e.earnrelease_date,earnrelease_time,e.timestart;
```

Result: Refer to file ‘baseline volume.csv’

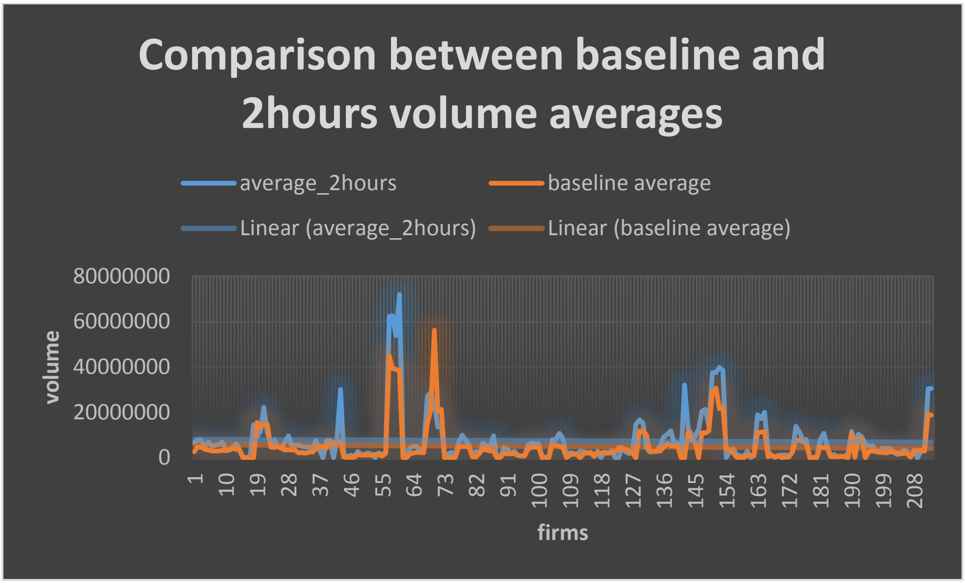
(f) Comparison between baseline and 40min average volume levels



The above graph shows that most of the firms have the volume end level greater for period 40minutes following the earning release when compared with the baseline average. Only few firms (2 to 10) have baseline average greater but that is negligible as the firm count is less and other factor may effect for that at that time.

To compare the overall, we are considering trading line. The trading line for average volume level for period 40minutes following the earning release is above the trading line for baseline average.

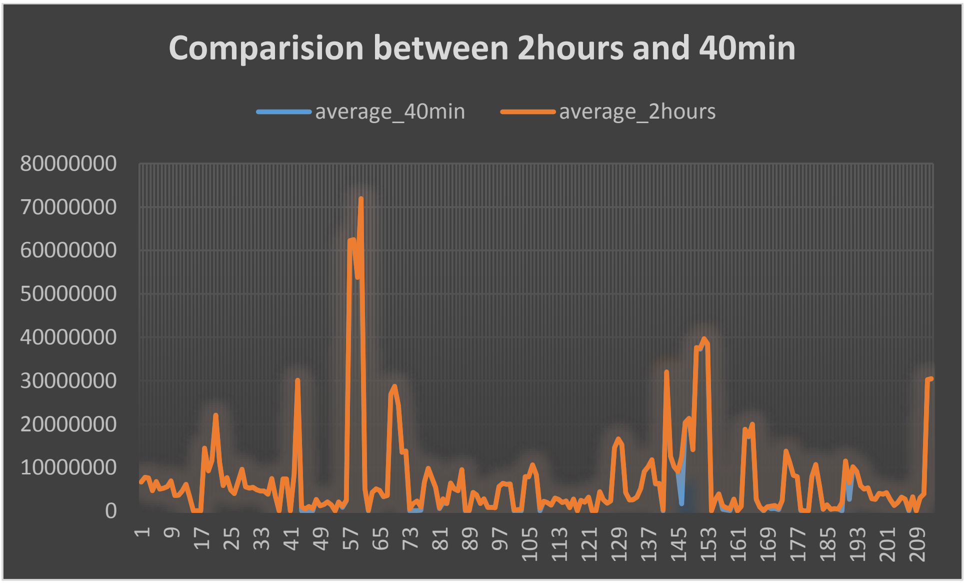
Comparison between baseline and 2hours average volume levels



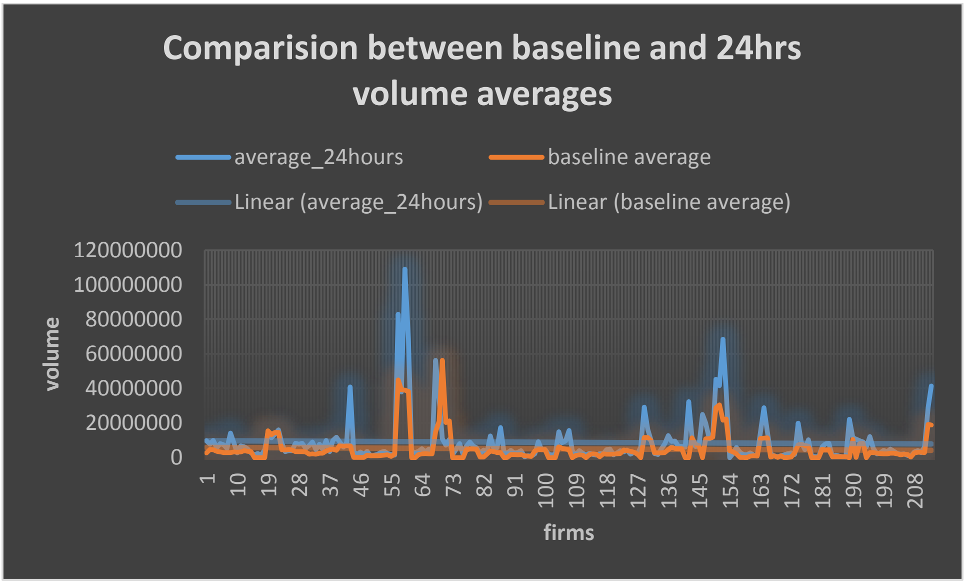
The above graph shows that most of the firms have the volume end level greater for period 2hours following the earning release when compared with the baseline average. Only few firms (2 to 10) have baseline average greater but that is negligible as the firm count is less and other factor may effect for that at that time.

To compare the overall, we are considering trading line. The trading line for average volume level for period 40minutes following the earning release is above the trading line for baseline average.

If we compare the trading volume levels for 40min period and 2hours period following the earning release, the average trading volume is approximately same

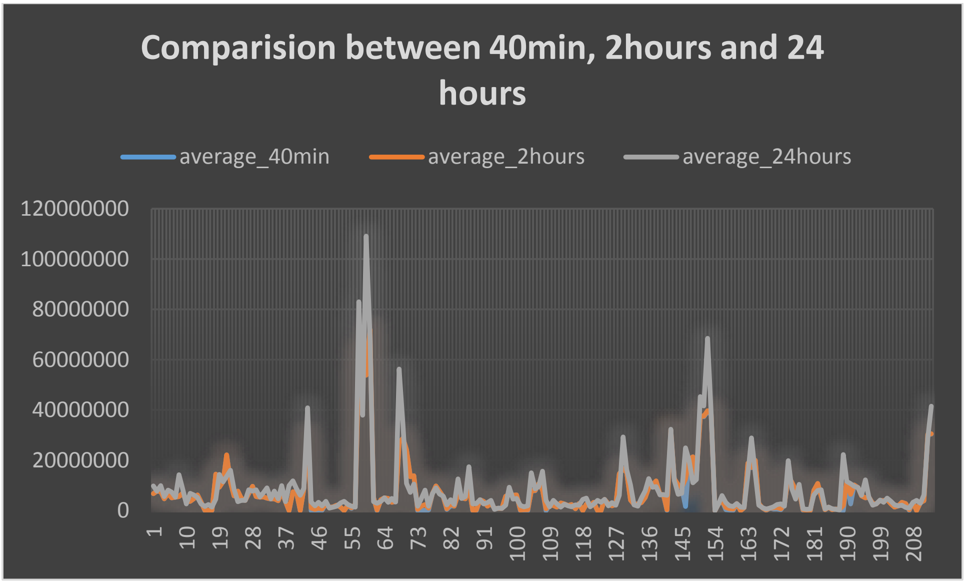


Comparison between baseline and 24hours average volume levels:

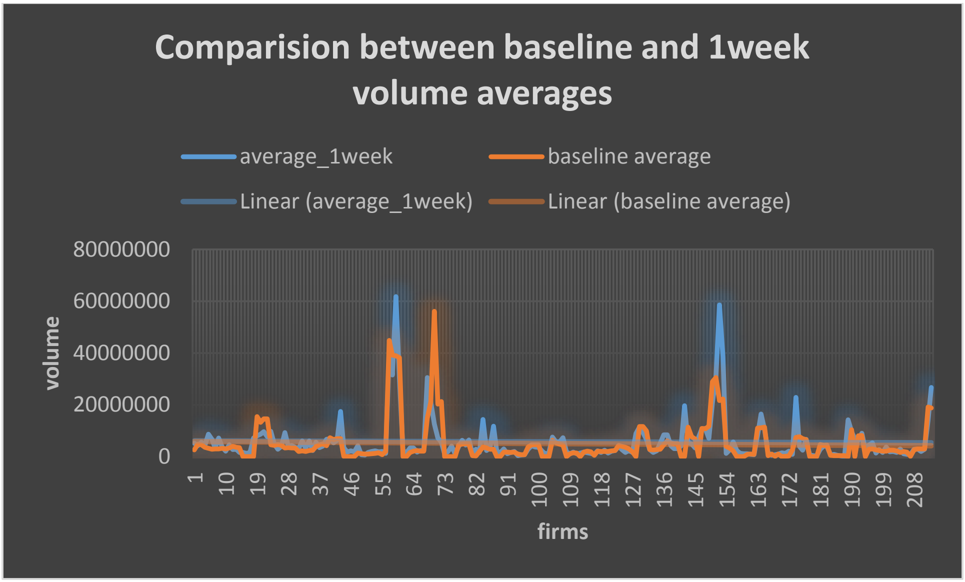


The above graph shows that most of the firms have the volume end level greater for period 24hours following the earning release when compared with the baseline average.

For the firms that have higher difference in the volume levels for 40min and 2hrs, the difference is much higher when we compare it with 24hours which can be observed from the below graph.



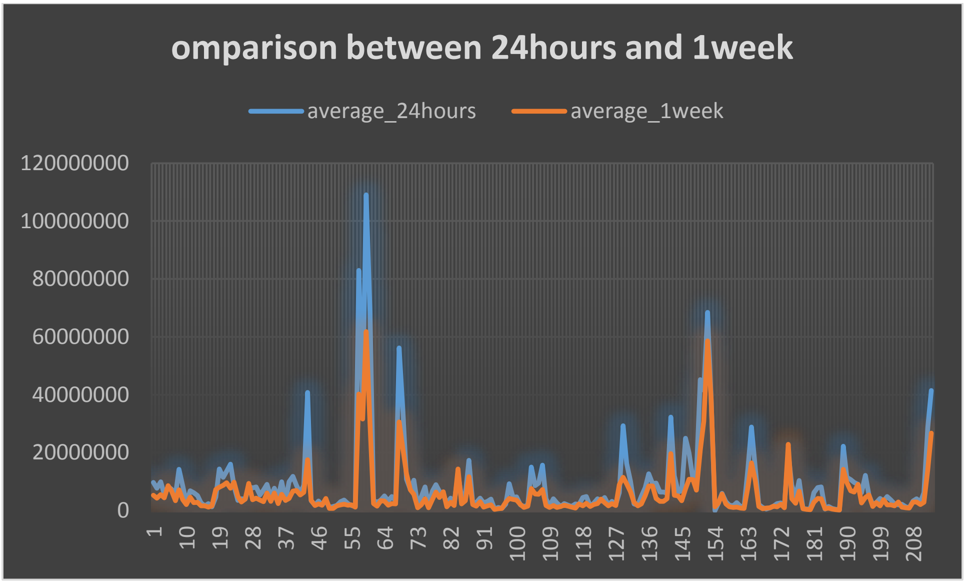
Comparison between baseline and 1week average volume levels:



The above graph shows that most of the firms have the volume end level greater for period 1week following the earning release when compared with the baseline average. Graph is similar with the 40min.

Overall analysis for volume end level:

If we compare the volume end level of 1week with 24hrs, the volume end levels is decreasing for 1week.



After 24hrs of earning release, volume end levels are much higher when compared with 40min, 2hours and 24 hours for most of the firms

For 40min with baseline, 2 hours with baseline, 24hours with baseline and 1week with baseline, trading line is above the baseline.