DATA EXPLORERS

A Project Report

Submitted by

Prateek Joshi 77813106

Siddhant Deshmukh 36568046

Under the Guidance of

Prof. Alin Dobra

In partial fulfilment for the award of the degree of

Master of Science in Computer Science

At

UNIVERSITY OF FLORIDA

DEC 2015

Annexure-II
DECLARATION

I, Prateek Joshi, Siddhant Deshmukh, understand that plagiarism is defined as anyone or combination of the following:

1. Un-credited verbatim copying of individual sentences, paragraphs or illustration (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.

2. Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)

3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did wrote what. (Source: IEEE, The institute, Dec. 2004)

4. I have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.

5. I affirm that no portion of my work can be considered as plagiarism and I take full responsibility if such a complaint occurs. I understand fully well that the guide of the seminar/ project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature of the Student:
Name: Prateek Joshi , Siddhant Deshmukh
Roll No. 77813106 , 36568046
Date: 12-8-2015

Annexure-III

CERTIFICATE

This is to certify that the project entitled “Data Explorers” is the bonafide work carried out by students of MS (Computer Science), University of Florida, during the academic year 2015, in partial fulfilment of the requirements for the award of the Degree of Master of Science in Computer Science. The project work has been assessed and found to be satisfactory.

Prof Ain Dobra

ACKNOWLEDGEMENT

On the occasion of completion of this project we would like to express our gratitude to Prof. Alin Dobra who guided and encouraged us to take this project up and helped us overcome the difficulties that we faced. We would also like to thank Jon Claus who helped us when we encountered any sort of problems.

At last, we would like to thank our parents and friends who have helped us directly or indirectly to accomplish our goals.

Annexure IV

Table of contents

INTRODUCTION

In this project the main objective was to do data analysis on a large data set which was provided by the financial analytics company Nanex. Using the Fgrokit application which unlike other big data platforms uses the Generalized Linear Aggregate Model coupled with other peripheral data processing models like generalized transforms and generalized input to process large amounts of data.

Figure 1: Fgrokit

PROJECT OVERVIEW

We used two data sets namely the sec filings data set and the Nanex financial dataset.

Project Phase 1:

Problem/challenge: load the SEC filings (annual reports, quarterly reports) as well as the Compustat data into tiGrokit, and use textual analysis to match the data in Compustat with the underlying filings (based on similarity in assets, sales, number of shares, etc). In order words, identify the original filing for each record in Compustat.

Project Phase 2:

The first goal:

To create a dataset that contains the intraday market moves at a one minute granularity. The second goal is to construct a trading strategy using the abnormal return (i.e. firm return minus market return) following the release of 8K (material event) filings.

For the second goal:

The purpose is to set up a trading strategy based on the release of new information (the dataset with 8K filings). For each 10-minute window following a filing, the firm's abnormal return is constructed as the 10-minute firm return minus the 10-minute market return.

Alternative second goal:

A more elaborate trading strategy would be first to identify stocks that are highly correlated, and then to go 'long' or 'short' in the paired stock. Example: firm A and firm B have highly correlated returns. Firm A releases a bad-news 8K filing (its share drops relative to the market index): go 'short' in firm B (going short in firm B means to borrow shares of firm B from a broker, sell these shares for cash on the market, in order to gain from price declines of share B that materialize when you repurchase shares B and return them to the broker; this is basically the opposite of investing). This strategy would yield a positive return if the market is slow to incorporate intra-industry news into stock prices.

REVIEW OF LITERATURE

Project Phase 1:

We had to match the records from the compustat data to the corresponding sec filing documents which were provided to us in the form of raw text data. So to study the structure of sec filings we referred a paper titled "Using XBRL to conduct a large-scale study of discrepancies between the accounting numbers in Compustat and SEC 10-K filings" by Roman Chychyla and Alexander Kogan from Rutgers Business School – Newark and New Brunswick, Rutgers, The State University of New Jersey

Project Phase 2:

For this phase we referred the queries previously used by Prof. Alin Dobra and Jonathan M. Claus available on the following links:

https://gist.github.com/alinVD
https://gist.github.com/jonathanmclaus

We also referred "GLADE: A Scalable Framework for Efficient Analytics" by Prof. Alin Dobra and Florin Rusu

REQUIREMENT AND ANALYSIS

Project Phase 1:

Publicly listed firms (and some private firms with more than 500 shareholders) are required to file with the SEC, where these filings are made publicly available for investors/researchers. The SEC provides archives going back to 1993 with filing details such company name, form type, filing date and the url where the filings can be retrieved. Common filings are the annual report (10-K) and quarterly report (10-Q).

Standards & Poor (S&P) is the main data vendor for annual and quarterly financial data of US firms. Their datasets (Compustat Fundamental Annual, and Compustat Fundamental Quarterly) contain about 1,000 variables for each filing. Variables include assets, accounts receivables, sales, equity, net income, etc. S&P also creates new variables that are not in the original filing, and sometimes they adjust some numbers. So, not all variables may be traced back to the original filing (but that should be systematic across observations), and some numbers may differ (for example sales in their data may not be the exact same sales in the 10-K). To populate their datasets, S&P uses the publicly available filings that everybody can download, but in the datasets that they sell they do not provide the url to the filing they used to populate each record. I.e., the do not help researchers to find the original filings.

Problem/challenge: load the SEC filings (annual reports, quarterly reports) as well as the Compustat data into tiGrokit, and use textual analysis to match the data in Compustat with the underlying filings (based on similarity in assets, sales, number of shares, etc). In order words, identify the original filing for each record in Compustat.

The firm identifier in Compustat is gvkey; the identifier for the SEC is CIK (central index key). Compustat does provide the current CIK, but not the historic CIK. If the CIK did not change over time, researchers can search the SEC archives for the filings and then finding the matching original filing is easy. However, if the CIK did change, there is no way to find the SEC filing, as the SEC uses the historic CIK, and Compustat the current CIK. This is the case for roughly 20-30% of firm-years in Compustat. Matching on company name does not work well either, as names can be spelled differently (e.g. IBM versus International Business Machines).

Relevance: many researchers (including myself) need to use the underlying original SEC filings for data in Compustat. It would be valuable if no (or few) observations would be lost when matching is improved.

Data that was available:

- Compustat Fundamental Annual (as well as Quarterly, if needed)
- SEC Archives (summary data, company name, filing date, historical CIK)
- Annual reports filed since 1993 (roughly 220,000 files) (Quarterly filings could be downloaded as well, if needed)

Project Phase 2:

Intraday market return index

In finance and accounting research intraday market return studies become more common. With these studies, short event windows are used to examine the impact of new information on stock prices. For example, to measure the effect of FED speeches on bank-stocks in the 10 minute following the release of some item (i.e. interest rate). In the studies that examine the effect of information over several days it is common to control for market wide movements. For example, if a stock has gained 2% after the release of some information, but the market increased 3%, the stock relatively lost 1%. The goal of this project is to construct a dataset that contains market-wide moves within each day. This way, intraday event studies can control for market wide moves.

The first goal:

To create a dataset that contains the intraday market moves at a one minute granularity. The second goal is to construct a trading strategy using the abnormal return (i.e. firm return minus market return) following the release of 8K (material event) filings.

The second goal:

To set up a trading strategy based on the release of new information (the dataset with 8K filings). For each 10-minute window following a filing, the firm's abnormal return is constructed as the 10-minute firm return minus the 10-minute market return.

Alternative second goal:

A more elaborate trading strategy would be first to identify stocks that are highly correlated, and then to go 'long' or 'short' in the paired stock. Example: firm A and firm B have highly correlated returns. Firm A releases a bad-news 8K filing (its share drops relative to the market index): go 'short' in firm B (going short in firm B means to borrow shares of firm B from a broker, sell these shares for cash on the market, in order to gain from price declines of share B that materialize when you repurchase shares B and return them to the broker; this is basically the opposite of investing). This strategy would yield a positive return if the market is slow to incorporate intra-industry news into stock prices.

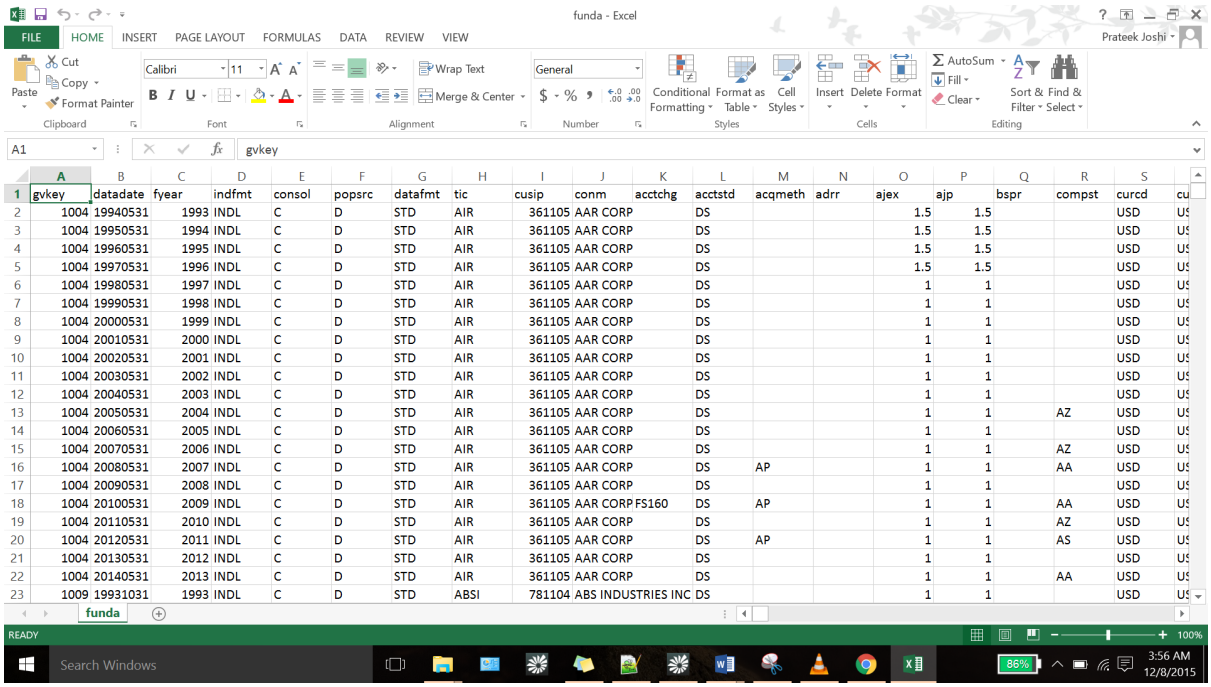
DATASETS

Project Phase 1:

The Compustat data:

Commonly used variables:

- Assets
- Equity
- Accounts Receivable
- Accounts Payable
- Sales -- Depending On Industry
- Cost of Goods Sold



funda - Excel

gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	cusip	conm	acctchg	acctstd	acqmeth	adrr	ajex	ajp	bspr	compst	curcd
1004	19940531	1993	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1.5	1.5			USD
1004	19950531	1994	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1.5	1.5			USD
1004	19960531	1995	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1.5	1.5			USD
1004	19970531	1996	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1.5	1.5			USD
1004	19980531	1997	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1			USD
1004	19990531	1998	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1			USD
1004	20000531	1999	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1			USD
1004	20010531	2000	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1			USD
1004	20020531	2001	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1			USD
1004	20030531	2002	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1			USD
1004	20040531	2003	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1			USD
1004	20050531	2004	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1		AZ	USD
1004	20060531	2005	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1			USD
1004	20070531	2006	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1		AZ	USD
1004	20080531	2007	INDL	C	D	STD	AIR	361105	AAR CORP		DS	AP		1	1		AA	USD
1004	20090531	2008	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1			USD
1004	20100531	2009	INDL	C	D	STD	AIR	361105	AAR CORP	FS160	DS	AP		1	1		AA	USD
1004	20110531	2010	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1		AZ	USD
1004	20120531	2011	INDL	C	D	STD	AIR	361105	AAR CORP		DS	AP		1	1		AS	USD
1004	20130531	2012	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1			USD
1004	20140531	2013	INDL	C	D	STD	AIR	361105	AAR CORP		DS			1	1		AA	USD
1009	19931031	1993	INDL	C	D	STD	ABSI	781104	ABS INDUSTRIES INC		DS			1	1			USD

Figure 2: COMPUSTAT DATASET

Sec Filing Data:

FILER:

COMPANY DATA:
COMPANY CONFORMED NAME: ADC TELECOMMUNICATIONS INC
CENTRAL INDEX KEY: 0000061478
STANDARD INDUSTRIAL CLASSIFICATION: TELEPHONE & TELEGRAPH APPARATUS [3661]
IRS NUMBER: 410743912
STATE OF INCORPORATION: MN
FISCAL YEAR END: 1031

FILING VALUES:
FORM TYPE: 10-K405/A
SEC ACT:
SEC FILE NUMBER: 000-01424
FILM NUMBER: 1681275

Project Phase 2:

Nanex Trades Dataset:

Type	Symbol	MsOfDay	Date	ListedExg	ReportingExg	Price	TradeConditi	ConditionFla	BATECode	Size
Equity	COCO	40976325	6/15/07	NQNM	PACF	15.01	Regular	0	NULL	100
Equity	AA	57525725	2/3/14	NYSE	EDGX	11.22	IntermarketS	0	NULL	100
Equity	HMY	41328900	1/4/13	NYSE	NYSE	8.28	Regular	0	NULL	100
Equity	TAP	49295725	10/19/12	NYSE	BATS	43.925	Regular	0	NULL	100
Equity	FOE	38549225	6/16/09	NYSE	NYSE	3.45	Regular	0	NULL	100
Equity	HAR	57531375	9/21/06	NYSE	NYSE	81.58	Regular	0	NULL	100
Equity	BEE	55350875	4/16/14	NYSE	BTRF	10.395	Regular	0	NULL	100
Equity	LCC	57084625	12/13/12	NYSE	CBOE	12.95	Regular	0	NULL	100
Equity	EP	52450425	7/15/11	NYSE	BOST	20.02	Regular	0	NULL	100
Equity	DELL	36236900	10/20/10	NQNM	RUSL	14.58	IntermarketS	0	NULL	100
Equity	ALV	46274400	6/9/08	NYSE	NYSE	53.8	IntermarketS	0	NULL	100
Equity	JAS	49138600	12/2/05	NYSE	NYSE	11.94	Regular	0	NULL	500
Equity	MITI	43463150	10/1/09	NQNM	NQEX	6.33	IntermarketS	0	NULL	200
Equity	EEM	38566200	9/25/08	PACF	BATS	36.31	Regular	0	NULL	100
Equity	ORCL	36236425	5/27/09	NQNM	NQEX	19.14	Regular	0	NULL	100
Equity	MDVN	53513150	3/19/10	NQNM	NQEX	11.99	IntermarketS	0	NULL	300
Equity	ATML	54334550	5/24/11	NQNM	RUSL	14.08	Regular	0	NULL	100

FIGURE 3: NANEX TRADES DATASET

8k Sec filings DataSet:

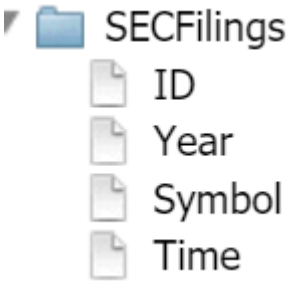


FIGURE 4: 8K SEC FILING DATASETS

PRE-PROCESSING

Preprocessing Phase 1:

For the phase 1 of the project we decided to use regular expressions to extract the important fields from the 10k filing text files using python text extraction queries and then create a table which could be loaded directly into grokit where we can later on use this table to do collision resolution queries and find out the text document associated with a row in the compustat data set.

Our initial approach was to extract values for certain important fields in the 10k text files like firm name. here is a part of our python code to extract field names from the text files.

```
import re
#file = open('F:/sec filings/1.txt', 'r')
for i in range(1,101,1):
    with open('F:/sec filings/'+str(i)+'.txt') as infile, open('C:/Users/prate_000/output.txt', 'w') as outfile:
        copy = False
        for line in infile:
            #if line.strip() == "CONSOLIDATED STATEMENTS OF CASH FLOWS":
                CompanyName = re.search(r"^\s*Net cash provided from operating activities\s*\.*\s*\$?(\d*\,*\d*)\s*(\d*\,*\d*)\s*(\d*\,*\d*)\s*\n"
                # print str(i)+".txt  -> "+
                if (CompanyName!=None):
                    print (str(i)+".txt  -> "+CompanyName.group(1))
                else:
                    CompanyName = re.search(r"^<.*>\s*<.*>\s*(Net cash provided from operating activities)\s*<.*>\s*<.*>\n", line)#next(infile)
                    if (CompanyName!=None):
                        print (str(i)+".txt  -> "+CompanyName.group(1))
                    else:
                        CompanyName = re.search(r"^<.*>\s*<.*>\s*(Net cash provided from \(used in\) operating activities)\s*<.*>\s*<.*>\n", line)
                        if (CompanyName!=None):
                            print (str(i)+".txt  -> "+CompanyName.group(1))
                        else:
                            CompanyName = re.search(r"^\s*<.*>\s*<.*>\s*(Net cash provided from \(used in\) operating)\s*\n", line)#next(infile)
                            if (CompanyName!=None):
                                print (str(i)+".txt  -> "+next(infile))
                            else:
                                CompanyName = re.search(r"^\s*<.*>\s*<.*>\s*(Net cash provided from)\s*\n", line)#next(infile)
                                if (CompanyName!=None):
                                    print (str(i)+".txt  -> "+next(infile))
                                else:
                                    CompanyName = re.search(r"^\s*(Total cash from operating activities)\s*\$*(\d*\,*\d*)\s*(\d*\,*\d*)\s*(\d*\,*\
                                    if (CompanyName!=None):
                                        print (str(i)+".txt  -> "+CompanyName.group(2))
                                    else:
                                        CompanyName = re.search(r"^\s*<.*>\s*<.*>\s*(Total cash from operating activities)<.*>\s*<.*>\n", line)#ne
                                        if (CompanyName!=None):
                                            print (str(i)+".txt  -> "+CompanyName.group(1))
```

Figure 5:SOURCE CODE TO EXTRACT COMPANY NAMES

```
45.txt  ->              COMPANY CONFORMED NAME:                 AFP IMAGING CORP
46.txt  ->              COMPANY CONFORMED NAME:                 AFP IMAGING CORP
47.txt  ->              COMPANY CONFORMED NAME:                 AFP IMAGING CORP
48.txt  ->              COMPANY CONFORMED NAME:                 AFP IMAGING CORP
49.txt  ->              COMPANY CONFORMED NAME:                 AFP IMAGING CORP
50.txt  ->              COMPANY CONFORMED NAME:                 AFP IMAGING CORP
51.txt  ->              COMPANY CONFORMED NAME:                 AFP IMAGING CORP
52.txt  ->              COMPANY CONFORMED NAME:                 AFP IMAGING CORP
53.txt  ->              COMPANY CONFORMED NAME:                 AFP IMAGING CORP
54.txt  ->              COMPANY CONFORMED NAME:                 AFP IMAGING CORP
55.txt  ->              COMPANY CONFORMED NAME:                 AFP IMAGING CORP
56.txt  ->              COMPANY CONFORMED NAME:                 AFP IMAGING CORP
58.txt  ->              COMPANY CONFORMED NAME:                 A L PHARMA INC
59.txt  ->              COMPANY CONFORMED NAME:                 ALPHARMA INC
60.txt  ->              COMPANY CONFORMED NAME:                 ALPHARMA INC
61.txt  ->              COMPANY CONFORMED NAME:                 ALPHARMA INC
62.txt  ->              COMPANY CONFORMED NAME:                 ALPHARMA INC
```

Figure 6: THE GENERATED OUTPUT

For extracting other values from table inidices from the text files we tried extracting the tables as a whole from the text file.Here is the python code for table extraction.

```
import re
pattern1 = re.compile(r"CONSOLIDATED STATEMENTS? OF CASH FLOWS[\s\n]+<TABLE>.+?<\/TABLE>" ,re.S )
pattern2 = re.compile(r"CONSOLIDATED STATEMENTS OF CASH FLOWS.+?<\/(TABLE|table)>" ,re.S )
pattern3 = re.compile(r"Consolidated Statements of Cash Flows<BR>.+?<\/(TABLE|table)>" ,re.S | re.IGNORECASE )
pattern4 = re.compile(r"Consolidated Statements of Cash Flows</B>.+?<\/(TABLE|table)>" ,re.S | re.IGNORECASE )
#Consolidated Statements of Cash Flows</B>
ctr=0
for i in range(1,101,1):
    ifile1 = open('files/'+ str(i) +'.txt', 'r')
    ifile2 = open('files/'+ str(i) +'.txt', 'r')
    ifile3 = open('files/'+ str(i) +'.txt', 'r')
    ifile4 = open('files/'+ str(i) +'.txt', 'r')
    ofile = open('output/table'+ str(i) +'.txt', 'w')
    table = re.search(pattern1,ifile1.read())
    if table == None :
        table1 = re.search(pattern2,ifile2.read())
        if table1 != None :
            ofile.write(table1.group(0))
        else :
            table2 = re.search(pattern3,ifile3.read())
            if table2 != None :
                ofile.write(table2.group(0))
            else :
                table3 = re.search(pattern4,ifile4.read())
                if table3 != None :
                    ofile.write(table3.group(0))
                else :

                    ofile.write('no table found')
                    ctr=ctr+1
                    print i
    else :
        ofile.write(table.group(0))

    ifile1.close()
    ifile2.close()
    ofile.close()
```

Figure 7:PYTHON CODE FOR TABLE EXTRACTION

```
CONSOLIDATED STATEMENTS OF CASH FLOWS    <BR>  </B></FONT></P>

<!-- User-specified TAGGED TABLE -->
<TABLE WIDTH="100%" BORDER=0 CELLSPACING=0 CELLPADDING=0>
<TR VALIGN="BOTTOM">
<TH COLSPAN=4 ALIGN="LEFT"><FONT SIZE=2> </FONT><BR></TH>
<TH WIDTH="2%"><FONT SIZE=1> </FONT></TH>
<TH COLSPAN=8 ALIGN="CENTER"><FONT SIZE=1><B>For the Year Ended May 31,</B></FONT><HR NOSHADE></TH>
<TH WIDTH="1%"><FONT SIZE=1> </FONT></TH>
</TR>
<TR VALIGN="BOTTOM">
<TH COLSPAN=4 ALIGN="LEFT"><FONT SIZE=1> </FONT><BR></TH>
<TH WIDTH="2%"><FONT SIZE=1> </FONT></TH>
<TH COLSPAN=2 ALIGN="CENTER"><FONT SIZE=1><B>2001</B></FONT><HR NOSHADE></TH>
<TH WIDTH="2%"><FONT SIZE=1> </FONT></TH>
<TH COLSPAN=2 ALIGN="CENTER"><FONT SIZE=1><B>2000</B></FONT><HR NOSHADE></TH>
<TH WIDTH="2%"><FONT SIZE=1> </FONT></TH>
<TH COLSPAN=2 ALIGN="CENTER"><FONT SIZE=1><B>1999</B></FONT><HR NOSHADE></TH>
<TH WIDTH="1%"><FONT SIZE=1> </FONT></TH>
</TR>
<TR VALIGN="BOTTOM">
<TH COLSPAN=4 ALIGN="LEFT"><FONT SIZE=1> </FONT><BR></TH>
<TH WIDTH="2%"><FONT SIZE=1> </FONT></TH>
<TH COLSPAN=8 ALIGN="CENTER"><FONT SIZE=1><B>(In thousands)<BR> </B></FONT><BR></TH>
<TH WIDTH="1%"><FONT SIZE=1> </FONT></TH>
</TR>
<TR BGCOLOR="#CCEEFF" VALIGN="BOTTOM">
<TD COLSPAN=4><FONT SIZE=2>Cash flows from operating activities:</FONT></TD>
```

Figure 8:THE OUTPUT PRDUCED

As it can be seen from the output the structure of the table indices across the dataSet varied from a simple text table to an html embedded table.

Preprocessing Phase 2:

The nanex trades data set was loaded into grokit.Since to achieve one second granuarity we had to reduce the data and filter on symbols which effect the market the most.We decided on the most important stock symbols only and we loaded this list of names on grokit.The symbol names and the validity period of this symbol names were filtered beforehand to give us only the current existing names of the symbol.

PERMNO	NAMEDT	NAMEENDT	SHRCD	EXCHCD	SICCD	NCUSIP	COMNAM	SHRCLS	TSYMBOL	PRIMEXCH	TRDSTAT	SECSTAT	PERMCO	COMPNO	ISSUNO	FF_IND	IND_48
10026	6/10/2004	12/31/2013	11	3	2052	46603210	J & J SNACK FOODS C		JJSF	Q	A	R	7976	60007929	10433	FOOD	2
10032	6/10/2004	12/31/2013	11	3	3670	72913210	PLEXUS CORP		PLXS	Q	A	R	7980	60007933	10437	CHIPS	36
10044	6/10/2004	12/31/2013	11	3	2060	77467840	ROCKY MOUNTAIN C		RMCF	Q	A	R	7992	60007945	10454	FOOD	2
10065	6/10/2004	12/31/2013	14	1	6726	621210	ADAMS EXPRESS CO		ADX	N	A	R	20023	0	0	FIN	47
10107	6/10/2004	12/31/2013	11	3	7370	59491810	MICROSOFT CORP		MSFT	Q	A	R	8048	60008001	10539	BUSSV	34
10145	6/10/2004	12/31/2013	11	1	3724	43851610	HONEYWELL INTERN		HON	N	A	R	22168	0	0	AERO	24
10147	6/10/2004	12/31/2013	11	1	3572	26864810	E M C CORP MA		EMC	N	A	R	8093	60008046	10615	COMPS	35
10200	6/10/2004	12/31/2013	11	3	2830	75991610	REPLIGEN CORP		RGEN	Q	A	R	8135	60008088	10673	DRUGS	13
10207	6/10/2004	12/31/2013	14	3	6720	78080N10	ROYCE FOCUS TRUST		FUND	Q	A	R	9806	60009741	13072	FIN	47
10239	6/10/2004	12/31/2013	11	3	6330	5775520	BALDWIN B		BWINB	Q	A	R	714	60000713	924	INSUR	45
10252	6/10/2004	12/31/2013	11	3	6020	45383610	INDEPENDENT BANK		INDB	Q	A	R	8179	60008131	10733	BANKS	44
10297	6/10/2004	12/31/2013	11	3	6030	64472210	NEW HAMPSHIRE TH		NHTB	Q	A	R	8217	60008169	10785	BANKS	44
10299	6/10/2004	12/31/2013	11	3	3670	53567810	LINEAR TECHNOLOG		LLTC	Q	A	R	8220	60008172	10788	CHIPS	36
10355	6/10/2004	12/31/2013	11	3	2710	23391210	DAILY JOURNAL COR		DJCO	Q	A	R	8278	60008230	10879	BOOKS	8
10375	6/10/2004	12/31/2013	11	1	6021	87227510	T C F FINANCIAL COR		TCB	N	A	R	8292	60008244	10895	BANKS	44
10395	6/10/2004	12/31/2013	11	3	6330	63890410	NAVIGATORS GROUP		NAVG	Q	A	R	8314	60008266	10934	INSUR	45
10397	6/10/2004	12/31/2013	11	3	4210	95075510	WERNER ENTERPRISE		WERN	Q	A	R	8317	60008269	10939	TRANS	40
10421	6/10/2004	12/31/2013	11	3	4510	83087910	SKYWEST INC		SKYW	Q	A	R	8340	60008291	10975	TRANS	40
10443	6/10/2004	12/31/2013	11	3	4210	70337B10	PATRIOT TRANSPORT		PATR	Q	A	R	8360	60008310	11003	TRANS	40
10507	6/10/2004	12/31/2013	11	3	3840	58449L10	MEDICAL ACTION IN		MDCI	Q	A	R	7340	60007306	9511	MEDEQ	12
10530	6/10/2004	12/31/2013	11	3	2835	58958410	MERIDIAN BIOSCIEN		VIVO	Q	A	R	8441	60008391	11118	DRUGS	13
10606	6/10/2004	12/31/2013	11	1	3491	94274910	WATTS W.A		WTS	N	A	R	8508	60008458	11211	BLDMT	17

Figure 9: IMPORTANT SYMBOLS FOR CREATING THE INDEX

AFTER HOURS TRADING

Since the Nanex data had very fine granularity and contained all the stocks traded in the 24 hour period of a day we managed to do some after hour trading analysis. The market hours of the typical market the following :

PRE-MARKET HOURS : 6 to 9:28 a.m.

NORMAL MARKET HOURS : 9:30 a.m. to 4 p.m.

AFTER HOURS TRADING : 4:02 to 8 p.m.

The following figure showed us the number of trades made over the entire data vs the time of the day when the trade was made. We can estimate the amount of after hour trading done by the chart below.

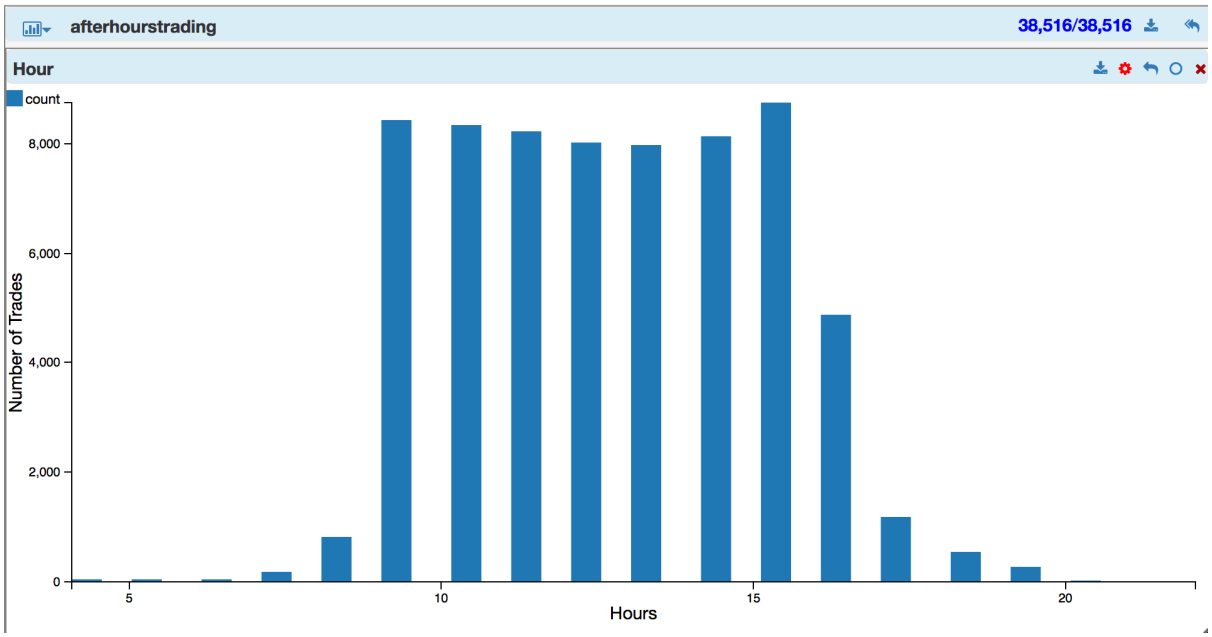


Figure 10: NUMBER OF TRADES AT DIFFERENT TIMES OF THE DAY

We tabulated the result and tried to analyze it to find some peculiar results. In our data 8pm had a value of 72,000,000 and we sorted our data based on the last time of a trade made in the trading window. We found a few peculiar results with trading time well after 8 pm. We researched more about traders and after hour trading and found out that only privileged members could trade during these hours. The following table gives a snapshot of the data and the query is provided below.

market

#	Hour	FirstPrice	LastPrice	FirstTime	LastTime	marketchan
1	17,644,580	1,000.52	999.88	72,000,000	74,132,000	0.00064
2	17,644,820	10,775.86	10,762.32	72,000,150	74,108,550	0.00126
3	17,658,860	2,080.626	2,082.336	72,006,525	73,746,875	-0.00082
4	17,676,236	244,894.35585	245,219.0857	72,000,000	73,200,875	-0.00133
5	17,675,708	6,857.7744	6,865.5374	72,000,000	73,200,350	-0.00113
6	17,644,844	397.05	365.00	72,008,275	72,584,650	0.08072
7	17,653,940	3,493.12	3,493.12	72,000,050	72,578,000	0.00
8	17,623,700	9.50	9.50	72,540,850	72,547,000	0.00
9	17,637,620	66.62	66.85	72,489,125	72,489,725	-0.00345
10	17,643,500	2,038.6803	2,039.0753	72,000,000	72,482,275	-0.00019
11	17,634,908	141.18	141.02	72,000,200	72,478,325	0.00113
12	17,623,292	44.44	44.44	72,462,100	72,462,100	0.00
13	17,637,428	1,083.98	1,083.98	72,070,275	72,431,850	0.00

Figure 11: PECULIAR RESULTS AFTER HOURS TRADING

```
library(nanex)

data <- Read(nanex_trades)

agg <- Segmenter(GroupBy(
  data,
  group = c(HourID = MsOfDay %/% 3600000 + 24 * (Date$GetYear() * 366 + Date$GetDayOfYear() - 1),Symbol),
  OrderBy(LastTime = dsc(MsOfDay), inputs = Price, outputs = LastPrice, limit = 1),
  OrderBy(FirstTime = asc(MsOfDay), inputs = Price, outputs = FirstPrice, limit = 1),
  Count = Count(),
  Total = Sum(Size)
))



market <- Generate(agg, marketchange = (FirstPrice - LastPrice) / FirstPrice, Hour = HourID %% 24)

market <- market[.(5) <= Hour && Hour <= .(8)]
## No need for a limit here. There are less than 40,000 hours in the data.
market <- OrderBy(market, dsc(Total),limit=200000)

View(market)
```

Figure 12: AFTER HOURS TRADING QUERY

We also wanted to see the names of some companies that traded after hours and sorted them by the highest number of trades before hours and got the following data.

Symbol	LastTime	LastPrice	FirstTime	FirstPrice	Count	marketchan	Hour
BAC	32,399,675	11.52	28,800,525	10.28	74,201	-0.12062	8
BAC	32,399,900	15.61	28,800,200	15.30	70,601	-0.02026	8
C	32,399,925	1.52	28,800,300	1.20	70,555	-0.26667	8
FB	32,399,975	24.53	28,800,650	22.81	61,043	-0.07541	8
C	32,399,750	3.13	28,800,075	3.24	60,562	0.03395	8
WB	32,399,800	0.98	28,800,325	3.88	59,975	0.74742	8
BAC	32,398,350	14.58	28,800,550	13.8248	58,718	-0.05463	8
SPY	32,399,975	121.41	28,800,400	120.53	52,366	-0.0073	8
FB	32,399,950	34.45	28,800,100	32.83	50,718	-0.04935	8
BAC	32,399,750	11.78	28,800,050	11.76	49,936	-0.0017	8
SPY	32,398,925	108.63	28,801,025	109.76	49,660	0.0103	8
SPY	32,398,875	102.67	28,800,450	103.00	48,658	0.0032	8
SPY	32,398,900	113.37	28,800,300	113.10	48,579	-0.00239	8
BAC	32,399,250	8.41	28,800,775	7.27	47,916	-0.15681	8
CSCO	32,399,825	20.40	28,800,050	20.31	46,467	-0.00443	

Figure 13: HIGHEST NUMBER OF TRADES BEFORE HOURS

The names of some familiar companies that we found with the most number of after hours trading were:

1. Bank of America
2. Citi Group
3. Facebook
4. Weibo (Chinese version of Facebook or Twitter)

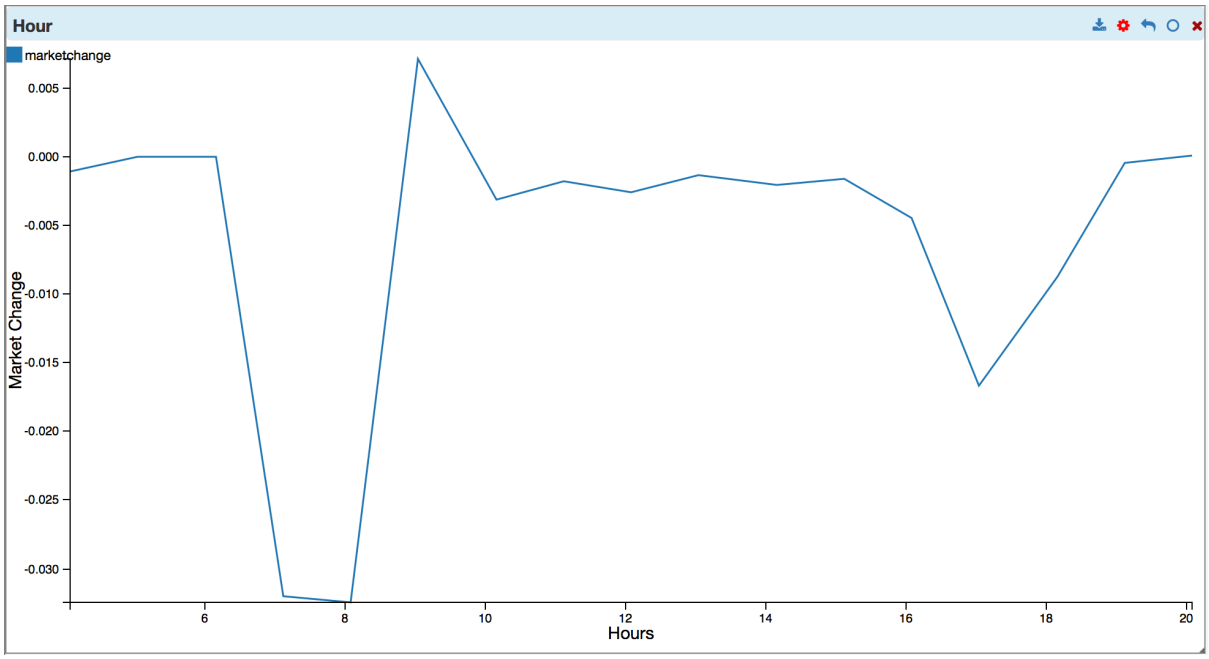


Figure 14: MARKET CHANGE AFTER HOURS

HOUR ID TO IMPLEMENT WINDOWS

We implemented hour id as a unique id for each window in the Nanex data. To verify that our hour id was producing accurate results we compared it our charts to the charts generated by Google.

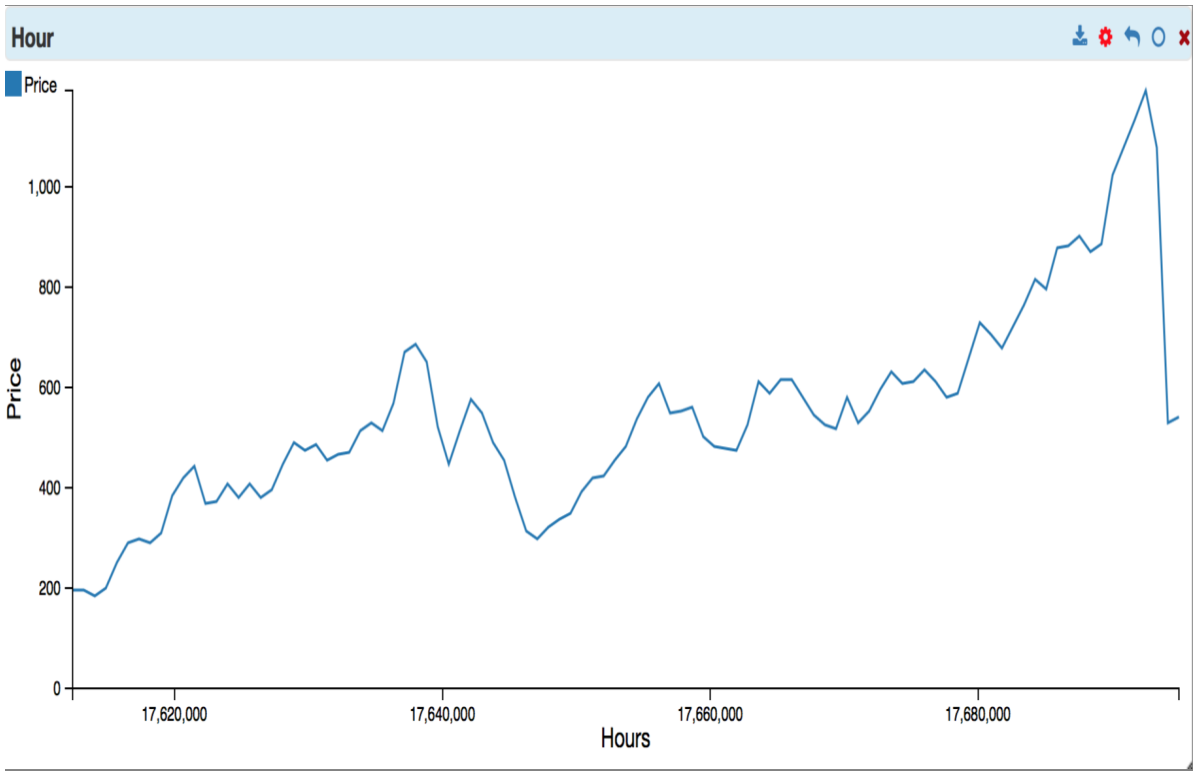


Figure 15: VERIFICATION OF OUR METHODS

PECULIAR RESULT IN APPLE STOCK PRICE

We find a very unusual event in the Apple stock price as you can see below. We took this topic up with our project mentor and he said there was nothing peculiar and it is a common occurrence.

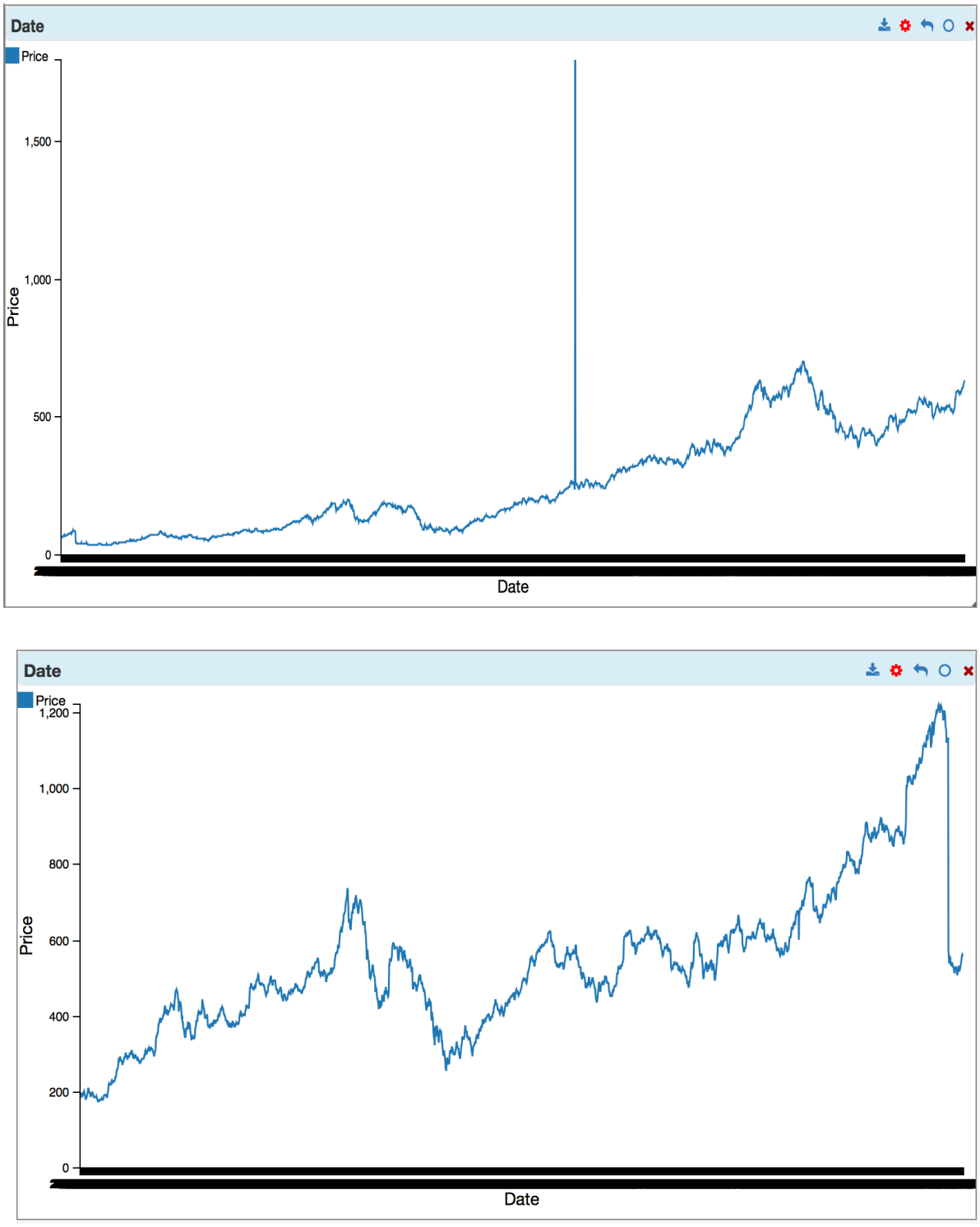


Figure 16: PECULIAR OCCURANCE IN APPLE DATA

MARKET CHANGE AND ABNORMAL RETURNS QUERIES

By using the hourly windows that we created using ids we wanted to calculate the hourly market change as that was what our mentor needed for his research. We used this hourly market change to calculate the abnormal returns of the stock. We saved the hourly market index on Grokit and read the data with our second query.

```
library(gtBase)
library(nanex)
library(methods)

data <- Read(nanex_trades)

agg <- Segmenter(
GroupBy(data,
group=c(Hour = MsOfDay %/% 3600000 + 24 * (Date$GetYear() * 366 + Date$GetDayOfYear()-1),Symbol),
OrderBy(LastTime = dsc(MsOfDay), inputs = Price, outputs = LastPrice, limit = 1),
OrderBy(FirstTime = asc(MsOfDay), inputs = Price, outputs = FirstPrice, limit = 1),
Count = Count(),
Total = Sum(Size)
));

##View(CountDistinct(agg, Hour))

market <- Segmenter(GroupBy(agg,group=Hour, Sum(FirstPrice), Sum(LastPrice)))

market <- Generate(market, marketchange = (FirstPrice - LastPrice) / FirstPrice)



market <- OrderBy(market, asc(Hour), limit=200000);

View(market)

WriteCSV(market, "/tmp/hourly_market_index.csv", Hour, marketchange);
```

Figure 17: MARKET CHANGE QUERY

Below we have a snapshot of the intermediate data and the second query that reads the data and computes the abnormal return.

In finance, an abnormal return is the difference between the actual return of a security and the expected return. Abnormal returns are sometimes triggered by "events."

A	B	C	D
Hour	FirstPrice	LastPrice	marketchange
17611976	19250.758	19258.359	-0.000394842
17611977	264246.19	263633.835	0.002317379
17611978	271433.78	270539.282	0.003295458
17611979	257673.76	257732.779	-0.000229028
17611980	261803.3	261767.378	0.000137228
17611981	257683.91	257392.255	0.001131822
17611982	259157.71	259155.093	1.00966E-05
17611983	266660.58	266151.753	0.001908161
17611984	126167.76	126350.996	-0.001452344
17611985	26300.56	26292.7215	0.00029802
17611986	31344.427	31332.4382	0.000382476

Figure 18: INTERMEDIATE SAVED DATA SNAPSHOT

```
library(gtBase)
library(nanex)
library(methods)

data <- Read(nanex_trades)

market <- ReadCSV("/tmp/hourly_market_index.csv", c(,), header = TRUE, sep = "|")

agg <- Segmenter(
GroupBy(data,
group=c(SymbolHour = MsOfDay %/% 3600000 + 24 * (Date$GetYear() * 366 + Date$GetDayOfYear()-1),Symbol),
OrderBy(LastTime = dsc(MsOfDay), inputs = Price, outputs = LastPrice, limit = 1),
OrderBy(FirstTime = asc(MsOfDay), inputs = Price, outputs = FirstPrice, limit = 1),
Count = Count(),
Total = Sum(Size)
));

agg <- Generate(agg, symbolchange = (FirstPrice - LastPrice) / FirstPrice)


datajoin <- Join( agg, SymbolHour,market, Hour)

#datajoin <- OrderBy(datajoin, dsc(Count), limit = 200000)


final <- Generate(datajoin, totalchange= marketchange - symbolchange)


final <- OrderBy(final, asc(Hour), limit=200000);

View(final)
```

Figure 19: ABNORMAL RETURN QUERY

query2

#	Hour	SymbolHour	Symbol	LastTime	LastPrice	FirstTime	FirstPrice	Count	Total	syml
1	17,611,976	17,611,976	PMCS	31,210,975	11.36	28,862,750	11.37	17	6,700	
2	17,611,976	17,611,976	PDLI	31,464,075	20.73	31,464,075	20.73	1	500	
3	17,611,976	17,611,976	SGMS	30,997,225	23.84	28,921,025	23.84	3	50,250	
4	17,611,976	17,611,976	AVAN	29,604,000	2.13	29,604,000	2.13	2	1,000	
5	17,611,976	17,611,976	PNCL	32,223,100	13.94	32,223,100	13.94	1	17,300	
6	17,611,976	17,611,976	INCY	31,320,950	9.99	31,320,950	9.99	1	800	
7	17,611,976	17,611,976	VSEA	32,062,100	36.32	28,846,025	36.69	15	7,550	
8	17,611,976	17,611,976	ICOR	28,862,875	0.65	28,862,875	0.65	1	2,000	
9	17,611,976	17,611,976	TGAL	30,312,175	1.67	29,215,200	1.67	5	3,500	
10	17,611,976	17,611,976	CPN	31,948,100	3.95	30,068,350	3.93	4	2,700	

Figure 20: ABNORMAL RETURN RESULT FULL TABLE

symbolchan	marketchan	totalchange
0.99873	0.00191	0.99682
0.99	0.00261	0.98739
0.98667	0.00001	0.98666
0.98667	0.00101	0.98565
0.98	0.00101	0.97899
0.96	0.00232	0.95768
0.94444	-0.0016	0.94604
0.90909	0.00014	0.90895
0.90	-0.00239	0.90239
0.90	-0.00085	0.90085
0.90103	0.00084	0.90018
0.8968	-0.00245	0.89925
0.90	0.00084	0.89916
0.90	0.00191	0.89809
0.90	0.00261	0.89739
0.8961	0.00232	0.89379
0.88889	-0.00239	0.89128
0.86957	-0.00239	0.87195

Figure 21: ABNORMAL RETURN TABLE (MAIN PART)

MIDWAY PROPOSED FUTURE WORK

What Next?

Join Sec filing data and achieve higher granularity

Correlation between stocks and grouping similar stocks

Developing a intra-day trading strategy

Doing further Analysis on after hours trading

Analyzing Nanex data before and after certain important events (like 8-k filings)

JOIN SEC FILING DATA TO ACHIEVE HIGHER GRANULARITY

Joost, our project guide, was not interested in all the companies in the Nanex data set and gave us another data set with company names that he was interested. We cleaned this data as we had to join this dataset to the Nanex data and columns liked date had to be standardized.

```
PERMNO|NAMEDT|NAMEENDT|SHRCD|EXCHCD|SICCD|NCUSIP|COMNAM|SHRCLS|TSYMBOL|PRIMEXCH|
TRDSTAT|SECSTAT|PERMCO|COMPNO|ISSUNO|FF_IND|IND_48
10000|19860107|19861203|10|3|3990|68391610|OPTIMUM MANUFACTURING INC|A|OMFGA|Q|A|R|
7952|60007905|10396||
10000|19861204|19870309|10|3|3990|68391610|OPTIMUM MANUFACTURING INC|A|OMFAC|Q|A|R|
7952|60007905|10396||
10000|19870310|19870611|10|3|3990|68391610|OPTIMUM MANUFACTURING INC|A|OMFGA|Q|A|R|
7952|60007905|10396||
10001|19860109|19931121|11|3|4920|39040610|GREAT FALLS GAS CO||GFGC|Q|A|R|7953|
60007906|10398|UTIL|31
10001|19931122|20040609|11|3|4920|29274A10|ENERGY WEST INC||EWST|Q|A|R|7953|60007906|
10398|UTIL|31
10001|20040610|20041018|11|3|4920|29274A10|ENERGY WEST INC||EWST|Q|A|R|7953|60007906|
10398|UTIL|31
10001|20041019|20041226|11|3|4920|29274A10|ENERGY WEST INC||EWSTE|Q|A|R|7953|
60007906|10398|UTIL|31
10001|20041227|20080204|11|3|4920|29274A10|ENERGY WEST INC||EWST|Q|A|R|7953|60007906|
10398|UTIL|31
10001|20080205|20080304|11|3|4920|29274A20|ENERGY WEST INC||EWSTD|Q|A|R|7953|
60007906|10398|UTIL|31
10001|20080305|20090803|11|3|4920|29274A20|ENERGY WEST INC||EWST|Q|A|R|7953|60007906|
10398|UTIL|31
10001|20090804|20091217|11|3|4920|29269V10|ENERGY INC||EGAS|Q|A|R|7953|60007906|
10398|UTIL|31
10001|20091218|20100708|11|2|4925|29269V10|ENERGY INC||EGAS|A|A|R|7953|60007906|
10398|UTIL|31
10001|20100709|20131231|11|2|4925|36720410|GAS NATURAL INC||EGAS|A|A|R|7953|60007906|
10398|UTIL|31
10002|19860110|19930929|10|3|6710|60740110|MOBILE NATIONAL CORP||MBNC|Q|A|R|7954|
60007907|10399|FIN|47
```

Figure 22: RAW DATA CONTAINING RELEVANT SYMBOLS AND OTHER INFORMATION

We had to date and time was stored in a single tuple and we had to clean the data by extracting the time and date in order to load the data to Grokit. After a lot of cleaning and standardization we loaded the data with the help of Jon.

```
data <- Read(nanex_trades)
sym <- Read(nanex_symbols)
data <- Join(data, c(Symbol), sym, c(Symbol))[Start <= Date && Date <= Stop]
```

Figure 23: DECREASING THE GRANULARITY OF DATA WITH THE HELP OF A JOIN

query11

```
library(gtBase)
library(nanex)
library(methods)


#data <- Bernoulli(Read(nanex_trades), 0.5)
data <- Read(nanex_trades)
sym <- Read(nanex_symbols)
data <- Join(data, c(Symbol), sym, c(Symbol))[Start <= Date && Date <= Stop]


agg <- Segmenter(
GroupBy(data,
group=c(Minute = MsOfDay %/% 60000 + base::DATETIME(Date)$AsMinutes(),Symbol),
OrderBy(LastTime = dsc(MsOfDay), inputs = Price, outputs = LastPrice, limit = 1),
OrderBy(FirstTime = asc(MsOfDay), inputs = Price, outputs = FirstPrice, limit = 1),
Count = Count(),
Total = Sum(Size)
));

##View(CountDistinct(agg, Hour))

market <- Segmenter(GroupBy(agg,group=Minute, Sum(FirstPrice), Sum(LastPrice),Min(FirstTime),Max(LastTime)))

market <- Generate(market, marketchange = (FirstPrice - LastPrice) / FirstPrice)



market <- OrderBy(market, asc(Minute), limit=200000);

View(market)

##WriteCSV(market, "/tmp/hourly_market_index.csv", Hour, marketchange);
```

Figure 24: QUERY WITH MINUTE GRANULARITY

CORRELATION BETWEEN STOCKS AND DEVELOPING A TRADING STRATEGY

In statistics, the Pearson product-moment correlation coefficient (/ˈpɪərsɪn/) (sometimes referred to as the PPMCC or PCCor Pearson's r) is a measure of the linear correlation between two variables X and Y, giving a value between +1 and −1 inclusive, where 1 is total positive correlation, 0 is no correlation, and −1 is total negative correlation. It is widely used in the sciences as a measure of the degree of linear dependence between two variables.

One of our goals was to find out correlation between various symbols so as to develop a trading strategy based on how two stocks behave on some special event in the 8-k filings.
We used the following query for this:

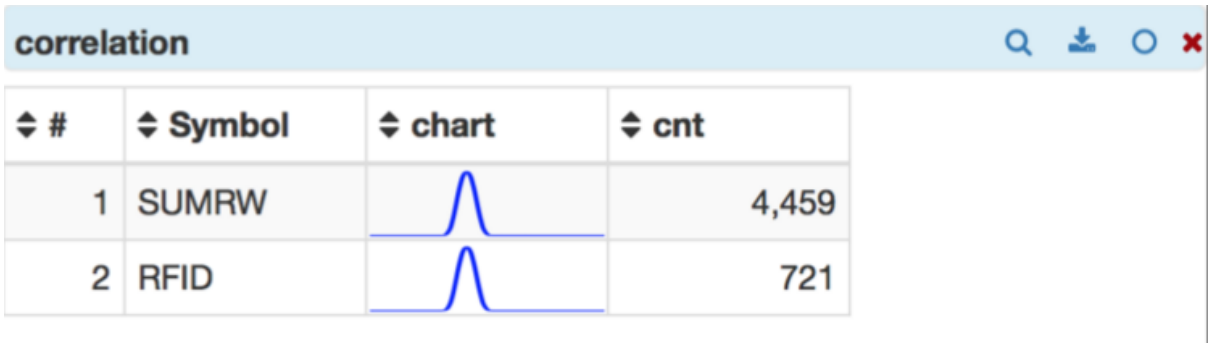
```
library(gtStats)
library(methods)
library(nanex)
data <- Read(nanex_trades)
data <- Generate(data, Seconds = MsOfDay %/% 1000 + DateTime(Date)$AsSeconds())

charts <- GroupBy(data, group = Symbol,
                  chart = LineChart(inputs = c(MsOfDay / 3600000, Price), length = 24),
                  cnt = Count())[cnt > 100]

covariance <- BigMatrix(charts, inputs = c(Symbol, chart), outputs = c(x, y, covariance))[x != y]

ordering <- OrderBy(covariance, dsc(covariance), limit = 200000)

View(ordering)
```

Figure 25: QUERY TO GENERATE COVARIANCE

covariance	x	y
0.99969	CHH	FTD
0.99975	ENZ	FTD
0.99966	STR	FTD
0.99966	ATU	FTD
0.99972	HZO	FTD
0.99962	UTI	ARG
0.99975	ENTG	ARG
0.99972	K	ARG
0.99985	JEC	ARG
0.9999	CVC	ARG
0.99988	KBR	ARG
0.99989	NYB	ARG
0.99976	RAH	ARG

Figure 26: THE OUTPUT GENERATED WAS

We were also able to plot the symbol graphs to show there approximate behaviour during the period the data was recorded.



correlation

#	Symbol	chart	cnt
1	SUMRW		4,459
2	RFID		721

Figure 27: A VISUAL REPRESENTATION OF COVARIANCE BETWEEN 2 SYMBOLS

```
library(nanex)
library(gtStats)
library(methods)

data <- Read(nanex_trades)
data <- Generate(data, Seconds = MsOfDay %/% 1000 + DateTime(Date)$AsSeconds())
##data <- Generate(data, Index=Seconds-
View(Multiplexer(data, min = Min(Seconds), max = Max(Seconds)))


charts <- GroupBy(data, group = Symbol,
                  chart = LineChart(inputs = c(MsOfDay / 3600000, Price), length = 24),
                  cnt = Count())[cnt > 100]

View(charts[Symbol == "SUMRW" || Symbol == "RFID"])

covariance <- BigMatrix(charts, inputs = c(Symbol, chart), outputs = c(x, y, covariance))[x != y]

ordering <- OrderBy(covariance, dsc(covariance), limit = 200000)

View(ordering)
```

Figure 28: Query for correlation

EVENT ANALYSIS ON NANEX DATA

We got a list of events from Joost which consisted of the 8-k sec filings. We had done a lot of extracting with this data hence we were very familiar with it. We analyzed the behavior of the stocks 300 seconds before and after each of these events (sec filings) to try to find instances of insider trading or any other illegal acts. We loaded the sec filings and Nanex trades and created a join between them based on Symbol and Date. We tried to find a specific pattern in the output. If the graph had an upward trend before the filing and downward trend after we filtered such stocks as they could be instances of insider trading.

sec filing events

```
## For each SEC filings, trades of the corresponding stock in the interval [time - before, time + after]
## are analyzed, where "time" is the time of the filing and "before", "after" are parameters.
## To reduce system load, only trades occurring on the same day as the filing are accounted for.
before <- 300
after <- 300

library(gtNanex)

## Trades can be filtered because all stocks filed with the SEC are equity stocks, not bonds.
events <- Load(SECFilings)
trades <- Load(nanex_trades)[Type == "Equity"]

events <- Generate(events, Date = Date(Time), Seconds = base::Time(Time)$as_seconds(), .overwrite = TRUE)

trades <- Join(trades, c(Symbol, Date), events, c(Symbol, Date))

trades <- Generate(trades, difference = Seconds - MsOfDay %/% 1000)
trades <- trades[.(-before) <= difference && difference <= .(after)]

info <- GroupBy(trades, c(ID, Symbol, Time),
                SharesTraded = Sum(if (difference < 0) Size else 0),
                  TotalVolume = Sum(Size),

                  OrderBy(MidTime = asc(abs(difference)), inputs = c(MidPrice = Price, MidSize = Size), limit = 1),
                OrderBy(LastTime = dsc(MsOfDay), inputs = c(LastPrice = Price, LastSize = Size), limit = 1),
                OrderBy(FirstTime = asc(MsOfDay), inputs = c(FirstPrice = Price, FirstSize = Size), limit = 1))

info <- info [ FirstPrice < MidPrice && MidPrice > LastPrice]

info <- OrderBy(info, dsc((LastPrice), limit = 200000)

##info <- OrderBy(info, PortionSold = dsc(SharesTraded / TotalVolume))

## info <- Generate(info, LastTime = base::Time(LastTime), FirstTime = base::Time(FirstTime), .overwrite = TRUE)

##WriteCSV(info, "/tmp/hourly_market_index.csv");


View(info)
```

Figure 29: SEC FILINGS EVENTS ANALYSIS ON NANEX DATA

We realized that we are getting many values that are going up by a very small amount and again decreasing by a negligible amount. To solve this problem, we change the last part of the query as shown below.

```
## For each SEC filings, trades of the corresponding stock in the interval [time - before, time + after]
## are analyzed, where "time" is the time of the filing and "before", "after" are parameters.
## To reduce system load, only trades occurring on the same day as the filing are accounted for.
before <- 300
after <- 300

library(gtNanex)

## Trades can be filtered because all stocks filed with the SEC are equity stocks, not bonds.
events <- Load(SECFilings)
trades <- Load(nanex_trades)[Type == "Equity"]

events <- Generate(events, Date = Date(Time), Seconds = base::Time(Time)$as_seconds(), .overwrite = TRUE)

trades <- Join(trades, c(Symbol, Date), events, c(Symbol, Date))

trades <- Generate(trades, difference = Seconds - MsOfDay %/% 1000)
trades <- trades[.(-before) <= difference && difference <= .(after)]

info <- GroupBy(trades, c(ID, Symbol, Time),
                SharesTraded = Sum(if (difference < 0) Size else 0),
                  TotalVolume = Sum(Size),

                  OrderBy(MidTime = asc(abs(difference)), inputs = c(MidPrice = Price, MidSize = Size), limit = 1),
                OrderBy(LastTime = dsc(MsOfDay), inputs = c(LastPrice = Price, LastSize = Size), limit = 1),
                OrderBy(FirstTime = asc(MsOfDay), inputs = c(FirstPrice = Price, FirstSize = Size), limit = 1))

info <- info [ FirstPrice < MidPrice && MidPrice > LastPrice]

info <- OrderBy(info, dsc((MidPrice - LastPrice) / MidPrice), limit = 200000)

##info <- OrderBy(info, PortionSold = dsc(SharesTraded / TotalVolume))

## info <- Generate(info, LastTime = base::Time(LastTime), FirstTime = base::Time(FirstTime), .overwrite = TRUE)

##WriteCSV(info, "/tmp/hourly_market_index.csv");


View(info)
```

sec filing events

Figure 30: MODIFIED SEC FILINGS QUERY

sec filing events

#	_orderAtt1	ID	Symbol	Time	SharesTrade	TotalVolume	MidTime	MidPrice	MidSize	Last
1	0.57268	60,187	YRCW	Sep 16, 2011 1	3,539,245	3,668,170	2	0.2635	100	40,4
2	0.39289	11,981	JMP	Mar 12, 2008 1	2,200	4,700	8	9.85	1,800	36,8
3	0.3007	31,718	JAZZ	Apr 03, 2009 0	10,775	10,975	2	1.43	10,000	34,2
4	0.2971	77,531	HTM	Nov 17, 2011 0	3,335	9,135	0	0.69	5,000	34,2
5	0.29268	57,925	STXS	Aug 09, 2011 0	208,674	243,636	0	1.64	30,512	34,5
6	0.28141	90,457	CABG	Nov 16, 2005 0	54,663	59,246	2	1.99	9,697	34,4
7	0.27964	81,646	ABI	Oct 25, 2006 0	1,400	1,900	25	50.53	100	34,5
8	0.268	56,109	PRST	Aug 13, 2009 0	74,272	76,872	0	2.50	100	34,4
9	0.25349	41,768	RAMR	May 20, 2008 0	56,923	59,015	3	1.46	7,400	34,5
10	0.2527	14,340	PACR	Feb 12, 2009 0	207,645	212,518	2	7.40	24,043	34,4
11	0.25	6,620	EEE	Mar 27, 2009 0	602,270	611,000	0	1.16	100	34,4
12	0.24313	80,981	SPTN	Jul 21, 2005 09	152,468	154,503	2	14.56	3,617	34,4
13	0.23529	20,110	WRES	Feb 26, 2009 0	167,830	187,831	3	1.19	52,716	34,5

Figure 31: SEC FILING EVENTS FINAL TABLE

Some companies with peculiar patterns in the final table:

1. Anheuser-Busch InBev
2. Metabolix
3. Alaska Air Group
4. Empire State Reality Trust
5. Hartford Financial Services Group
6. National Interstate Corporation
7. YRC Worldwide

sec filing events

#	ıde	TotalVolume	MidTime	MidPrice	MidSize	LastTime	LastPrice	LastSize	FirstTime	FirstPrice
1	ł5	3,668,170	2	0.2635	100	40,473,375	0.1126	200	40,082,575	0.2578
2	)0	4,700	8	9.85	1,800	36,815,450	5.98	200	36,336,400	6.02
3	'5	10,975	2	1.43	10,000	34,297,150	1.00	575	34,081,125	1.22
4	ł5	9,135	0	0.69	5,000	34,265,050	0.485	335	34,090,125	0.54
5	'4	243,636	0	1.64	30,512	34,500,500	1.16	100	34,195,225	1.60
6	ł3	59,246	2	1.99	9,697	34,492,750	1.43	3,500	33,932,450	1.95
7	)0	1,900	25	50.53	100	34,527,475	36.40	100	34,187,975	35.00
8	'2	76,872	0	2.50	100	34,466,350	1.83	800	34,081,525	1.99
9	ł3	59,015	3	1.46	7,400	34,500,000	1.0899	2,500	33,935,700	1.30
10	ł5	212,518	2	7.40	24,043	34,494,200	5.53	100	33,917,175	6.88
11	'0	611,000	0	1.16	100	34,498,750	0.87	800	34,026,675	1.11
12	ł8	154,503	2	14.56	3,617	34,499,500	11.02	100	33,942,225	14.49
13	)0	187,831	3	1.19	52,716	34,500,925	0.91	500	34,082,150	1.01

Figure 32: SHOWING THE PATTERN IN THE FINAL TABLE

CONCLUSION

We have learnt a lot in the course and got first hand experience in data science. The challenge of using regular expressions and handling raw, dirty data made us appreciate the effort needed to clean data. We realized that the process of ETL (Extraction, Transformation and Loading) was one of the most important and tedious step in data analysis. We also tried to look at the Nanex data from various different angles and realized that data analysis is more of an art and one needs to be very innovative to find new, useful information in the data. Only ones love for patterns and exploring data can invoke an interest in this field. Our data exploration journey does not end here and we plan to continue to use Grokit to dive deeper into the Nanex data set and analyze other interesting datasets like N-gram and network data.

REFERENCES

1. https://gist.github.com/alinVD

2. https://gist.github.com/jonathanmclaus

3. We also referred “GLADE: A Scalable Framework for Efficient Analytics” by Prof. Alin Dobra and Florin Rusu

4. “Using XBRL to conduct a large-scale study of discrepancies between the accounting numbers in Compustat and SEC 10-K filings” by Roman Chychyla and Alexander Kogan from Rutgers Business School – Newark and New Brunswick, Rutgers, The State University of New Jersey