NYU TANDON SCHOOL OF ENGINEERING

Earning releases impact story told by Surprise teller robot.

FRE6883 – Group 3
Supavitch Nakburee / Thumthiti Pinto / Hanlu Xia / Wenjun Jia
05/15/2020

Outline:

- Executive summary
- Task allocation
- How do we select stocks from Zack's
- Diagram design
- Functionality and Implementation
- Conclusion and Enrichment

NYU TANDON SCHOOL OF ENGINEERING

Executive summary

BMM's Impact

Reported Earnings that "Beat", "Meet", "Miss" expectation of investors impacted stock's return significantly. The miss-match in expectation and reality caused a fluctuation in prices according to good/bad news.

5 Functions

Leveraged computational power of C++ programming to study an effect of %Surprise of stocks in Russell 1000 to stocks' return with 5 main functionalities:

- Historical prices of a stock, one stock information from one group, average abnormal return, cumulative abnormal return, standard deviation, and a plot, to analyzed the effect of reported earning.

Zack's

Scraped Zack's EPS analysis of stocks in Russell 1000 and categorized them into 3 groups (BMM)– Beat, Meet, and Miss – ascendingly based on %Surprise.

Programming structures

Built 2 classes with 35 member functions and 18 independent functions showing statistically figure of how BMM's cumulative returns responded to announcement date of reported earning in three weeks.

Plot result

Our plot of CAARs showed significant impacts of Earning release by BMM groups with impact ranging in number of days.

*

NYU TANDON SCHOOL OF ENGINEERING

Task allocation

	Week 1	Week 2	Week 3
Supavitch	1. Planning the project and respective timeline. 2. Design Bootstrapping algorithm for CAAR, AAR, calculation. Develop the calculation of metric of return. 3. Revise the logic from zack extraction data.	1. Develop the bootstrapping algorithm to randomizing and return aggregation and integrated with the return calculation metric. 2. Develop the bootstrapping algorithm for testing with few price data 10 stocks 3. Debug and revise the stock class.	1. Debugging the print information. And yahoo finance bug. 2. Develop the presentation file to communicate with clients. 3. Debug and revise the bootstrapping algorithm 4. Final revisit of the data flow and application.
Thumthiti	1. Design and develop stock class, stock header file and stock cpp file with function inside them such as get_eps, get_surprise etc. 2. Design the information flow of the data and function.	1. Integrate the yahoo finance adj price with the stock metric calculation. 2. Design and develop the menus operation and integrate all the file together 3. co-development of bootstrapping algorithm of randomization and revise calculation of return,	1. Debug and revise the customization of N number. 2. Revise and debug the display menu and their operation. 3. Develop the presentation file to communicate with clients.
Hanlu	1. Create Matlab Algorithm to extract Earning from zack website. 2. Calculate EPS metric and divided in to 3 group according to earning surprise 3. Design the information flow of the data and function.	1. Add the function necessity for the stock class (Debug and revise stock class). 2. Design and develop. The Yahoo finance retrieving stock price and store in stocklist. 3. Design and develop. The Yahoo finance retrieving stock IWB benchmark data. to get their data with respective date. 4. Integrate with Yahoo finance retrieving data with the matlab data.	1. Revise and debug the bootstrap calculation to yield realistic result. 2. Test the application with data of150 stocks from test set. 3. Debugging the yahoo finance code to retrieve all data of 868 stocks. 4. Debugging the bootstrap algorithm for the static vector. 5. Debug and revise the menus and customization.
Wenjun	1. Design and develop stock class, stock header file and stock cpp file with function inside them such as get_eps, get_surprise etc. 2. Revise and debug the bootstrap design and information flow design.	1. Develop the code to Retrieve the stock data from csv file contain EPS. 2. Design and Develop gnuplot algorithm to plot the CAAR from each group of earning surprise. 3. Debug and revise the bootstrapping algorithm	1. Integrate the return metric with the gnuplot code to have respectively realistic plot. 2. Revise & debug the visualization and showing result according the requirement, such as change N to 2N for CAAR , AAR calculation 3. Develop the presentation file for clients.

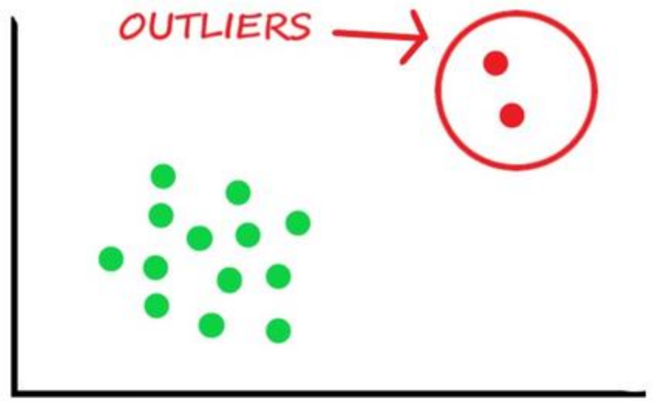
How your group selected stocks from Zack's ?

No outlier

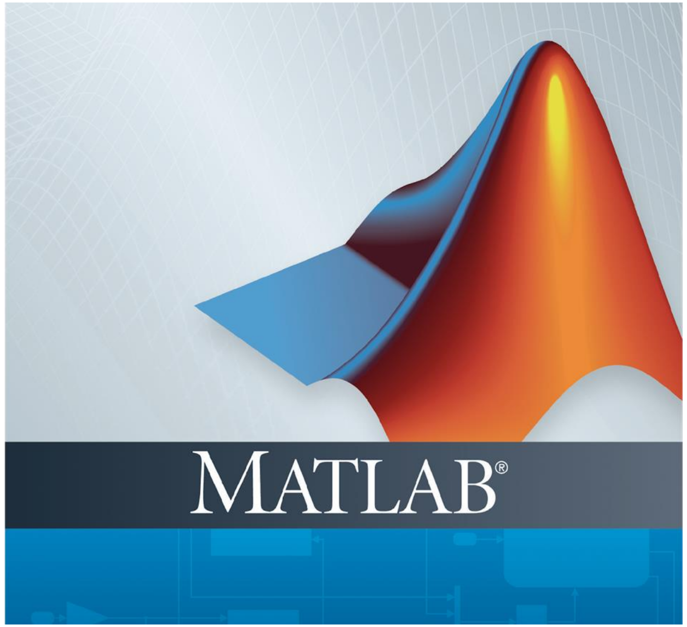
Even Though the application queries Russell 1000 stocks, we won't be retrieving all stock information. Since some of the stocks might have very large outliner in earning surprise : for instance if the Expected Earning for stock 'AAA' is 0.01 but actual earning is 10 -> so we have (10-0.01)/0.01 = 99900% these stock will be eliminated, since they are marked as outlier.

Extra conditions:

15 stocks does have 2n+1 size – eliminate Announcement date not matching



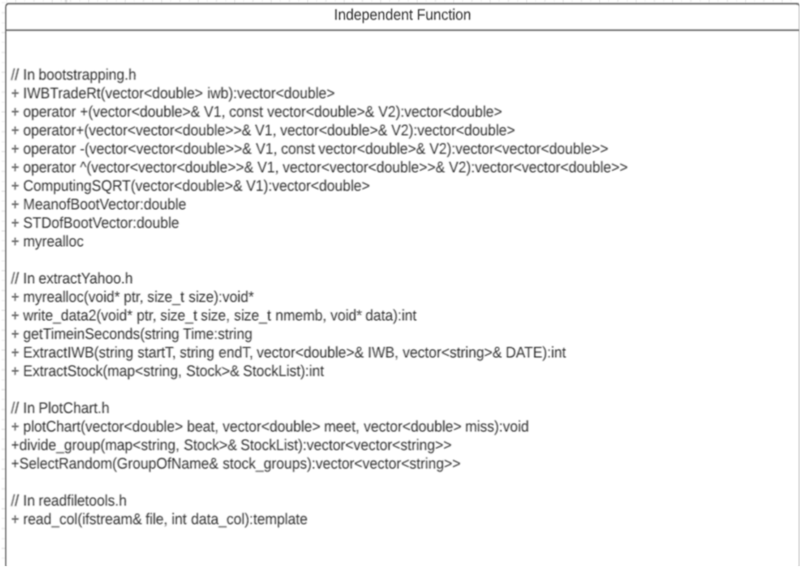
Csv file from Matlab



TickerName	Date	Estimate	Reported	SurpriseP	Group
ORCL	9/10/2020	0.86	0.93	8.1	Meet
ADBE	9/15/2020	2.4	2.57	7	Meet
AZO	9/22/2020	24.74	30.93	25	Beat
CTAS	9/23/2020	2.16	2.78	28.7	Beat
COST	9/24/2020	2.85	3.13	9.8	Meet
MU	9/29/2020	1	1.08	8	Meet
CAG	10/1/2020	0.57	0.7	22.8	Beat
STZ	10/1/2020	2.51	2.76	9.9	Meet
DPZ	10/8/2020	2.75	2.49	-9.4	Miss
JPM	10/13/2020	2.35	2.92	24.2	Beat
JNJ	10/13/2020	1.99	2.2	10.5	Meet
DAL	10/13/2020	-3.14	-3.3	-5.1	Miss
USB	10/14/2020	0.93	0.99	6.4	Meet
PGR	10/14/2020	1.69	1.87	10.6	Meet
PACW	10/14/2020	0.63	0.38	-39.6	Miss
UAL	10/14/2020	-7.63	-8.16	-6.9	Miss
MS	10/15/2020	1.26	1.59	26.1	Beat

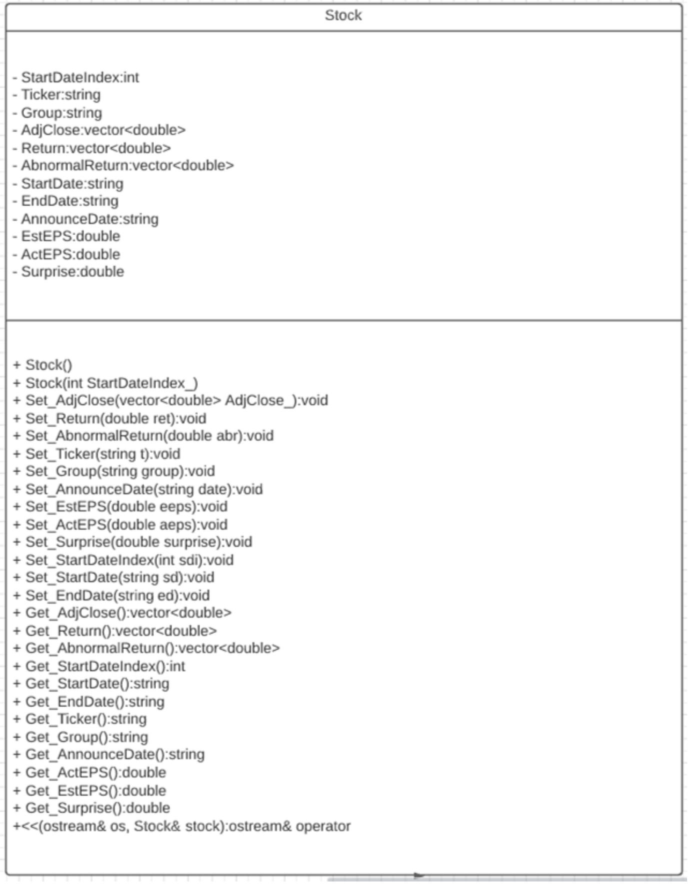
NYU TANDON SCHOOL OF ENGINEERING

Diagram design



Independent Function

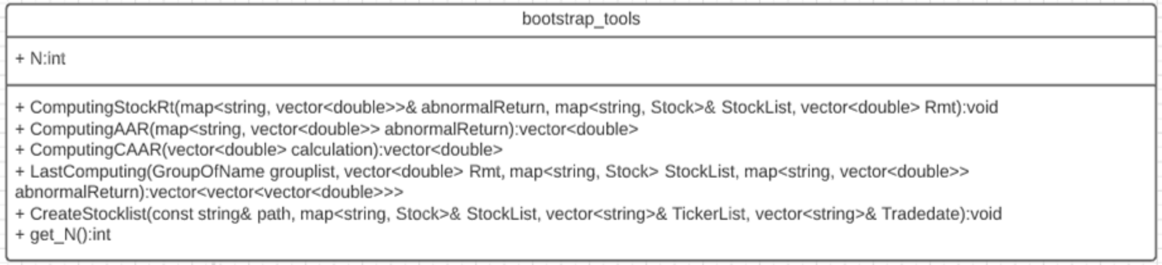
```
// In bootstrapping.h
+ IWBTradeRt(vector<double> iwb):vector<double>
+ operator +(vector<double>& V1, const vector<double>& V2):vector<double>
+ operator+(vector<vector<double>>& V1, vector<double>& V2):vector<double>
+ operator -(vector<vector<double>>& V1, const vector<double>& V2):vector<vector<double>>
+ operator ^(vector<vector<double>>& V1, vector<vector<double>>& V2):vector<vector<double>>
+ ComputingSQRT(vector<double>& V1):vector<double>
+ MeanofBootVector:double
+ STDofBootVector:double
+ myrealloc

// In extractYahoo.h
+ myrealloc(void* ptr, size_t size):void*
+ write_data2(void* ptr, size_t size, size_t nmemb, void* data):int
+ getTimeinSeconds(string Time:string
+ ExtractIWB(string startT, string endT, vector<double>& IWB, vector<string>& DATE):int
+ ExtractStock(map<string, Stock>& StockList):int

// In PlotChart.h
+ plotChart(vector<double> beat, vector<double> meet, vector<double> miss):void
+divide_group(map<string, Stock>& StockList):vector<vector<string>>
+SelectRandom(GroupOfName& stock_groups):vector<vector<string>>

// In readfiletools.h
+ read_col(ifstream& file, int data_col):template
```



Stock

```
- StartDateIndex:int
- Ticker:string
- Group:string
- AdjClose:vector<double>
- Return:vector<double>
- AbnormalReturn:vector<double>
- StartDate:string
- EndDate:string
- AnnounceDate:string
- EstEPS:double
- ActEPS:double
- Surprise:double
```

```
+ Stock()
+ Stock(int StartDateIndex_)
+ Set_AdjClose(vector<double> AdjClose_):void
+ Set_Return(double ret):void
+ Set_AbnormalReturn(double abr):void
+ Set_Ticker(string t):void
+ Set_Group(string group):void
+ Set_AnnounceDate(string date):void
+ Set_EstEPS(double eeps):void
+ Set_ActEPS(double aeps):void
+ Set_Surprise(double surprise):void
+ Set_StartDateIndex(int sdi):void
+ Set_StartDate(string sd):void
+ Set_EndDate(string ed):void
+ Get_AdjClose():vector<double>
+ Get_Return():vector<double>
+ Get_AbnormalReturn():vector<double>
+ Get_StartDateIndex():int
+ Get_StartDate():string
+ Get_EndDate():string
+ Get_Ticker():string
+ Get_Group():string
+ Get_AnnounceDate():string
+ Get_ActEPS():double
+ Get_EstEPS():double
+ Get_Surprise():double
+<<(ostream& os, Stock& stock):ostream& operator
```



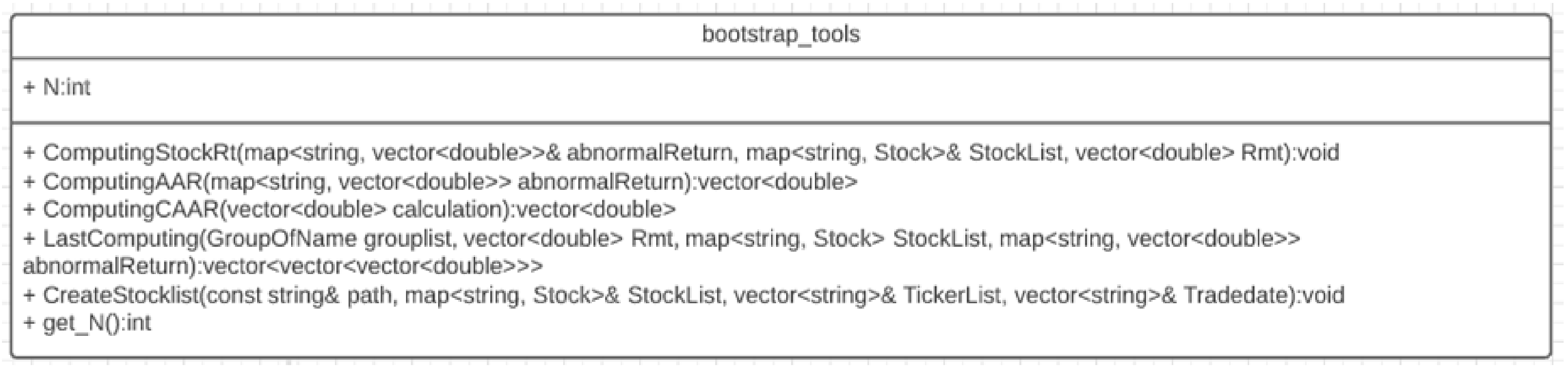
bootstrap_tools

```
+ N:int
```

```
+ ComputingStockRt(map<string, vector<double>>& abnormalReturn, map<string, Stock>& StockList, vector<double> Rmt):void
+ ComputingAAR(map<string, vector<double>> abnormalReturn):vector<double>
+ ComputingCAAR(vector<double> calculation):vector<double>
+ LastComputing(GroupOfName grouplist, vector<double> Rmt, map<string, Stock> StockList, map<string, vector<double>>
abnormalReturn):vector<vector<vector<double>>>
+ CreateStocklist(const string& path, map<string, Stock>& StockList, vector<string>& TickerList, vector<string>& Tradedate):void
+ get_N():int
```

Diagram design

ComputingStockRt -> computing return of each stock and computing abnormal return of each stock: stored in stocklist
ComputingAAR -> computing AAR
ComputingCAAR -> computing cumulative AAR
Lastcomputing -> will combine (select randomstock, AAR , CAAR and find their average / sd)
CreateStocklist -> Used to read CSV file from Matlab and create various vector to stored stock information such as return



bootstrap_tools

+ N:int

+ ComputingStockRt(map<string, vector<double>>& abnormalReturn, map<string, Stock>& StockList, vector<double> Rmt):void
+ ComputingAAR(map<string, vector<double>> abnormalReturn):vector<double>
+ ComputingCAAR(vector<double> calculation):vector<double>
+ LastComputing(GroupOfName grouplist, vector<double> Rmt, map<string, Stock> StockList, map<string, vector<double>> abnormalReturn):vector<vector<vector<double>>>
+ CreateStocklist(const string& path, map<string, Stock>& StockList, vector<string>& TickerList, vector<string>& Tradedate):void
+ get_N():int

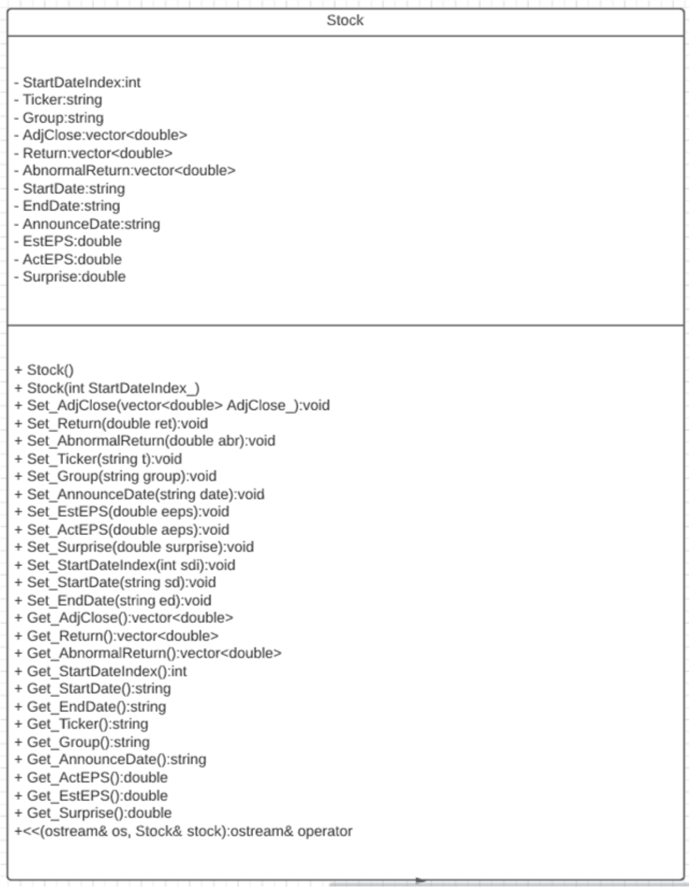
Diagram design

Stock class: contains the individual stock information such as their EPS, Surprise%, Group , Return , Abnormal return.

Their intention is to store the stocks information and will be put into the map call stocklist

stocklist map <ticker , stock class>
so that the application can queries the stock information though the tickername.

*



```
Stock

- StartDateIndex:int
- Ticker:string
- Group:string
- AdjClose:vector<double>
- Return:vector<double>
- AbnormalReturn:vector<double>
- StartDate:string
- EndDate:string
- AnnounceDate:string
- EstEPS:double
- ActEPS:double
- Surprise:double

+ Stock()
+ Stock(int StartDateIndex_)
+ Set_AdjClose(vector<double> AdjClose_):void
+ Set_Return(double ret):void
+ Set_AbnormalReturn(double abr):void
+ Set_Ticker(string t):void
+ Set_Group(string group):void
+ Set_AnnounceDate(string date):void
+ Set_EstEPS(double eeps):void
+ Set_ActEPS(double aeps):void
+ Set_Surprise(double surprise):void
+ Set_StartDateIndex(int sdi):void
+ Set_StartDate(string sd):void
+ Set_EndDate(string ed):void
+ Get_AdjClose():vector<double>
+ Get_Return():vector<double>
+ Get_AbnormalReturn():vector<double>
+ Get_StartDateIndex():int
+ Get_StartDate():string
+ Get_EndDate():string
+ Get_Ticker():string
+ Get_Group():string
+ Get_AnnounceDate():string
+ Get_ActEPS():double
+ Get_EstEPS():double
+ Get_Surprise():double
+<<(ostream& os, Stock& stock):ostream& operator
```

NYU TANDON SCHOOL OF ENGINEERING

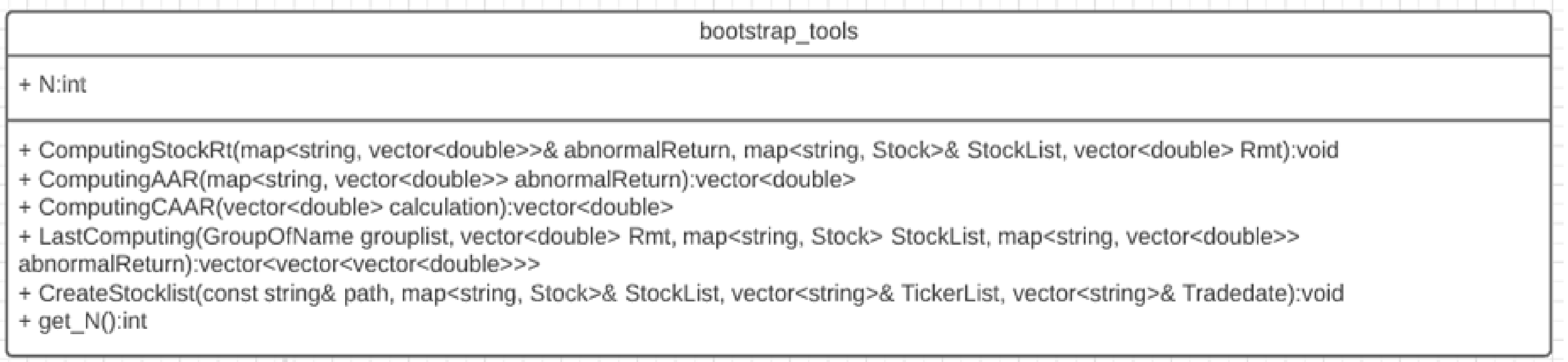
Diagram design

bootstrap_tools: contains the metric calculation which related to user input N

N: is the number of range that will be input by application user. Range between (30-60)

For instance, N = 32 mean that user would like to queries information
[from announcement-32 trading day to announcement+32 trading day]

The function in the bootstrap_tools class will use the N as one of the parameters and calculate the metric such as AAR, CAAR.



bootstrap_tools

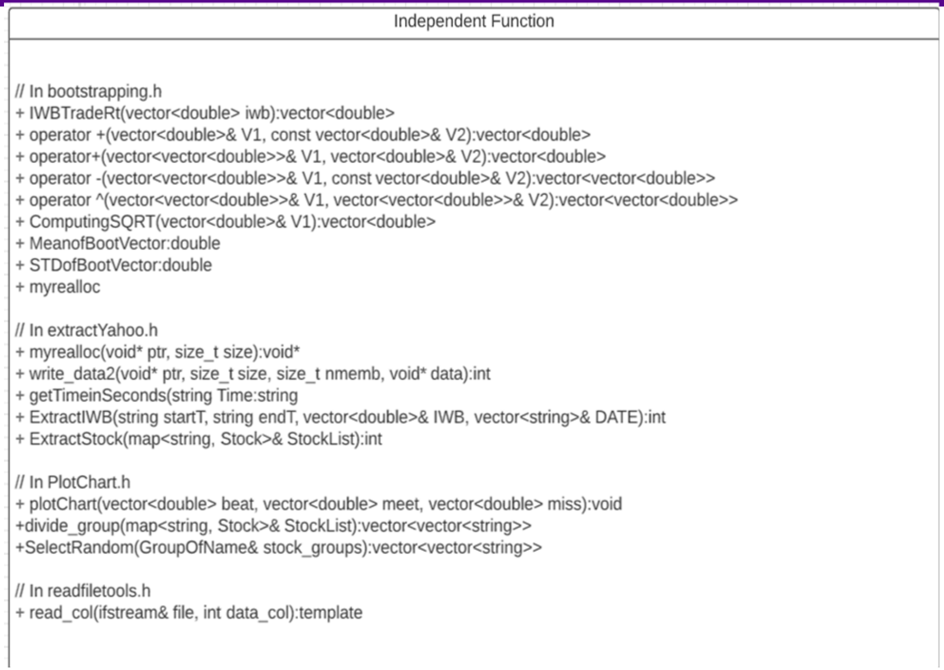
+ N:int

+ ComputingStockRt(map<string, vector<double>>& abnormalReturn, map<string, Stock>& StockList, vector<double> Rmt):void
+ ComputingAAR(map<string, vector<double>> abnormalReturn):vector<double>
+ ComputingCAAR(vector<double> calculation):vector<double>
+ LastComputing(GroupOfName grouplist, vector<double> Rmt, map<string, Stock> StockList, map<string, vector<double>> abnormalReturn):vector<vector<vector<double>>>
+ CreateStocklist(const string& path, map<string, Stock>& StockList, vector<string>& TickerList, vector<string>& Tradedate):void
+ get_N():int

*

NYU TANDON SCHOOL OF ENGINEERING

Diagram design



Independent Function

```
// In bootstrapping.h
+ IWBTradeRt(vector<double> iwb):vector<double>
+ operator +(vector<double>& V1, const vector<double>& V2):vector<double>
+ operator+(vector<vector<double>>& V1, vector<double>& V2):vector<double>
+ operator -(vector<vector<double>>& V1, const vector<double>& V2):vector<vector<double>>
+ operator ^(vector<vector<double>>& V1, vector<vector<double>>& V2):vector<vector<double>>
+ ComputingSQRT(vector<double>& V1):vector<double>
+ MeanofBootVector:double
+ STDofBootVector:double
+ myrealloc

// In extractYahoo.h
+ myrealloc(void* ptr, size_t size):void*
+ write_data2(void* ptr, size_t size, size_t nmemb, void* data):int
+ getTimeinSeconds(string Time:string
+ ExtractIWB(string startT, string endT, vector<double>& IWB, vector<string>& DATE):int
+ ExtractStock(map<string, Stock>& StockList):int

// In PlotChart.h
+ plotChart(vector<double> beat, vector<double> meet, vector<double> miss):void
+divide_group(map<string, Stock>& StockList):vector<vector<string>>
+SelectRandom(GroupOfName& stock_groups):vector<vector<string>>

// In readfiletools.h
+ read_col(ifstream& file, int data_col):template
```

Function in
bootstrap.h will primary concentrate on bootstrapping algorithm and return calculation

extractYahoo.h will primary concentrate in data extraction from yahoo finance.

Plotchart.h will primary concentrate in chart plotting

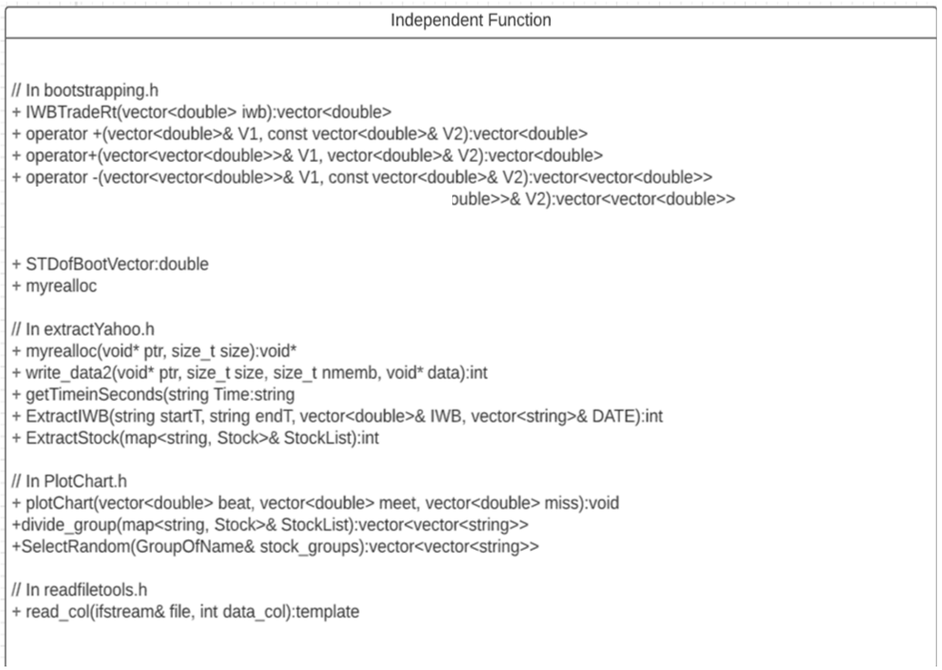
readfiletools.h will primary concentrate in reading the EPS file from csv.

Independent function: This is not the class, however contain the important function that will be used throughout the application

Each of them can come from different headers such as bootstrap.h , extractYahoo , plotchat.h , readfiletools.h.
In which each of them and their own unique objective.

NYU TANDON SCHOOL OF ENGINEERING

Diagram design



Independent Function

```
// In bootstrapping.h
+ IWBTradeRt(vector<double> iwb):vector<double>
+ operator +(vector<double>& V1, const vector<double>& V2):vector<double>
+ operator+(vector<vector<double>>& V1, vector<double>& V2):vector<double>
+ operator -(vector<vector<double>>& V1, const vector<double>& V2):vector<vector<double>>
                                                              ouble>>& V2):vector<vector<double>>

+ STDofBootVector:double
+ myrealloc

// In extractYahoo.h
+ myrealloc(void* ptr, size_t size):void*
+ write_data2(void* ptr, size_t size, size_t nmemb, void* data):int
+ getTimeinSeconds(string Time:string
+ ExtractIWB(string startT, string endT, vector<double>& IWB, vector<string>& DATE):int
+ ExtractStock(map<string, Stock>& StockList):int

// In PlotChart.h
+ plotChart(vector<double> beat, vector<double> meet, vector<double> miss):void
+divide_group(map<string, Stock>& StockList):vector<vector<string>>
+SelectRandom(GroupOfName& stock_groups):vector<vector<string>>

// In readfiletools.h
+ read_col(ifstream& file, int data_col):template
```

In bootstrapping.h
IWBTradeRt : calculate return of benchmark IWB
Operator+ -> overload operator for +
Operator- -> overload operator for –
Operator^ -> operator for matrix square
ComputingSQRT -> find sqrt of vector

//**inPlotchart**
plotChart -> plot the result using Gnuplot

//**in Stocklist.h**
Divide_group -> stored the stockgroup reading from csv in the vector<vector<string>> format
Selectrandom -> select random stck

//**in readfiletools.h**
Read_col ->read column from csv.

Functionality & Implementation

MATLAB

Retrieve Russell_1000 Stock Tickers, Announcement date, reported EPS, expected EPS, and %Surprise from Zacks

↓

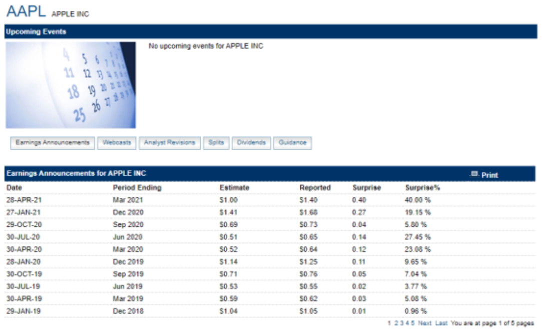
Eliminating %Surprise outliners

↓

sorting all stocks by %Surprise and separate into three groups with same size

↓

Save to “Allgroup.csv”



AAPL APPLE INC

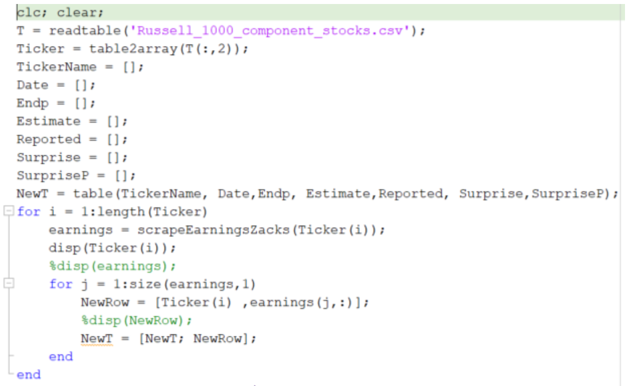
Upcoming Events

No upcoming events for APPLE INC

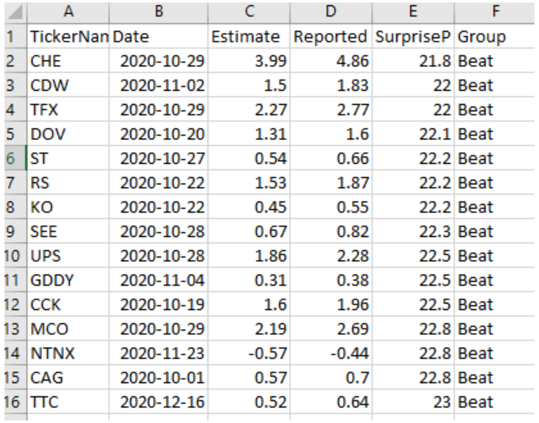
Earnings Announcements for APPLE INC

Date	Period Ending	Estimate	Reported	Surprise	Surprise%
28-APR-21	Mar 2021	$1.00	$1.40	0.40	40.00 %
27-JAN-21	Dec 2020	$1.41	$1.68	0.27	19.15 %
29-OCT-20	Sep 2020	$0.69	$0.73	0.04	5.80 %
30-JUL-20	Jun 2020	$0.51	$0.65	0.14	27.45 %
30-APR-20	Mar 2020	$0.52	$0.64	0.12	23.08 %
28-JAN-20	Dec 2019	$1.14	$1.25	0.11	9.65 %
30-OCT-19	Sep 2019	$0.71	$0.76	0.05	7.04 %
30-JUL-19	Jun 2019	$0.53	$0.55	0.02	3.77 %
30-APR-19	Mar 2019	$0.59	$0.62	0.03	5.08 %
29-JAN-19	Dec 2018	$1.04	$1.05	0.01	0.96 %

1 2 3 4 5 Next Last You are at page 1 of 5 pages



```
clc; clear;
T = readtable('Russell_1000_component_stocks.csv');
Ticker = table2array(T(:,2));
TickerName = [];
Date = [];
Endp = [];
Estimate = [];
Reported = [];
Surprise = [];
SurpriseP = [];
NewT = table(TickerName, Date,Endp, Estimate,Reported, Surprise,SurpriseP);
for i = 1:length(Ticker)
    earnings = scrapeEarningsZacks(Ticker(i));
    disp(Ticker(i));
    %disp(earnings);
    for j = 1:size(earnings,1)
        NewRow = [Ticker(i) ,earnings(j,:)];
        %disp(NewRow);
        NewT = [NewT; NewRow];
    end
end
```



	A	B	C	D	E	F
1	TickerNan	Date	Estimate	Reported	SurpriseP	Group
2	CHE	2020-10-29	3.99	4.86	21.8	Beat
3	CDW	2020-11-02	1.5	1.83	22	Beat
4	TFX	2020-10-29	2.27	2.77	22	Beat
5	DOV	2020-10-20	1.31	1.6	22.1	Beat
6	ST	2020-10-27	0.54	0.66	22.2	Beat
7	RS	2020-10-22	1.53	1.87	22.2	Beat
8	KO	2020-10-22	0.45	0.55	22.2	Beat
9	SEE	2020-10-28	0.67	0.82	22.3	Beat
10	UPS	2020-10-28	1.86	2.28	22.5	Beat
11	GDDY	2020-11-04	0.31	0.38	22.5	Beat
12	CCK	2020-10-19	1.6	1.96	22.5	Beat
13	MCO	2020-10-29	2.19	2.69	22.8	Beat
14	NTNX	2020-11-23	-0.57	-0.44	22.8	Beat
15	CAG	2020-10-01	0.57	0.7	22.8	Beat
16	TTC	2020-12-16	0.52	0.64	23	Beat

Functionality & Implementation

MATLAB

Retrieve Russell_1000 Stock Tickers, Announcement date, reported EPS, expected EPS, and %Surprise from Zacks

Eliminating %Surprise outliners

sorting all stocks by %Surprise and separate into three groups with same size

Save to **“Allgroup.csv”**

C++

Use ExtractIWB2() function to extract IWB info, stored price and date in vector IWB and DATE

“Allgroup.csv” □ Create a stock list with type StockMap

Use ExtractStock2() function to extract adj price for all stocks in stock list

C++:

A

Retrieving ALL data:

```
vector<double> IWB;
vector<string> DATE;

bootstrap_tools bt;
```

Initialize vectors and object

```
cout << "Downloading IWB data..." << endl;
ExtractIWB2("2020-04-01", "2021-04-20", IWB, DATE);
```

Store iwb data into IWB and the corresponding date into DATE

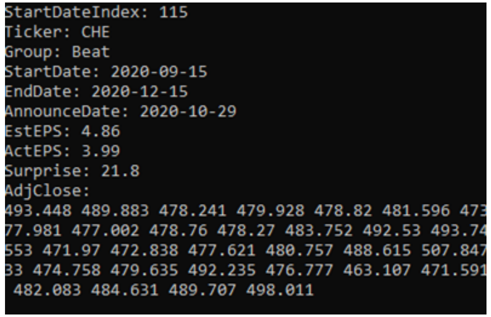
```
srand(unsigned(std::time(0)));
StockMap StockList;
vector<string> TickerList;
```

Initialize Stock List and a vector to store ticker

```
string path = "Allgroups.csv";
bt.CreateStocklist(path, StockList, TickerList, DATE);
ExtractStock2(StockList);
```

Download adj for all stocks and store into stock list

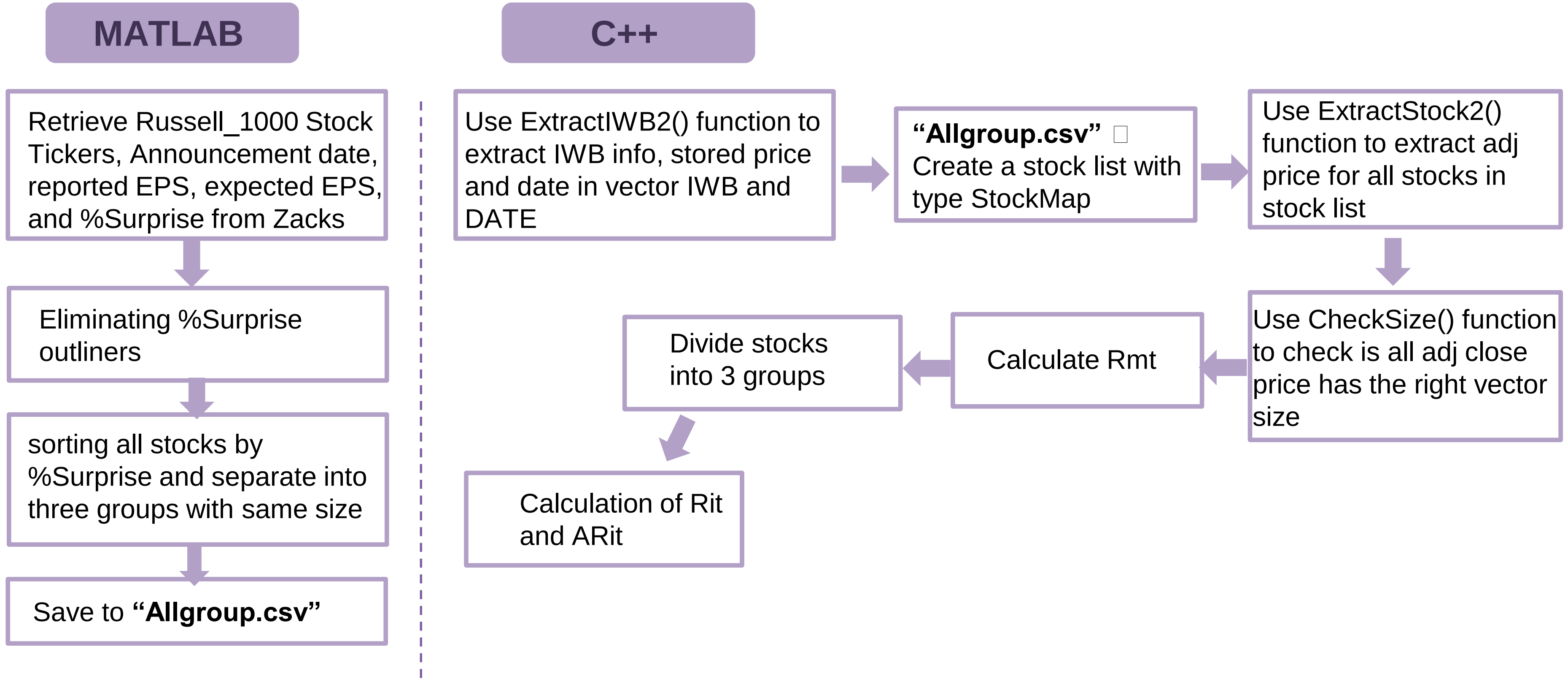
One Sample Stock from Stock List



```
StartDateIndex: 115
Ticker: CHE
Group: Beat
StartDate: 2020-09-15
EndDate: 2020-12-15
AnnounceDate: 2020-10-29
EstEPS: 4.86
ActEPS: 3.99
Surprise: 21.8
AdjClose:
493.448 489.883 478.241 479.928 478.82 481.596 473
77.981 477.002 478.76 478.27 483.752 492.53 493.74
553 471.97 472.838 477.621 480.757 488.615 507.847
33 474.758 479.635 492.235 476.777 463.107 471.591
 482.083 484.631 489.707 498.011
```

NYU TANDON SCHOOL OF ENGINEERING

Functionality & Implementation



MATLAB

Retrieve Russell_1000 Stock Tickers, Announcement date, reported EPS, expected EPS, and %Surprise from Zacks

Eliminating %Surprise outliners

sorting all stocks by %Surprise and separate into three groups with same size

Save to **“Allgroup.csv”**

C++

Use ExtractIWB2() function to extract IWB info, stored price and date in vector IWB and DATE

“Allgroup.csv” □ Create a stock list with type StockMap

Use ExtractStock2() function to extract adj price for all stocks in stock list

Use CheckSize() function to check is all adj close price has the right vector size

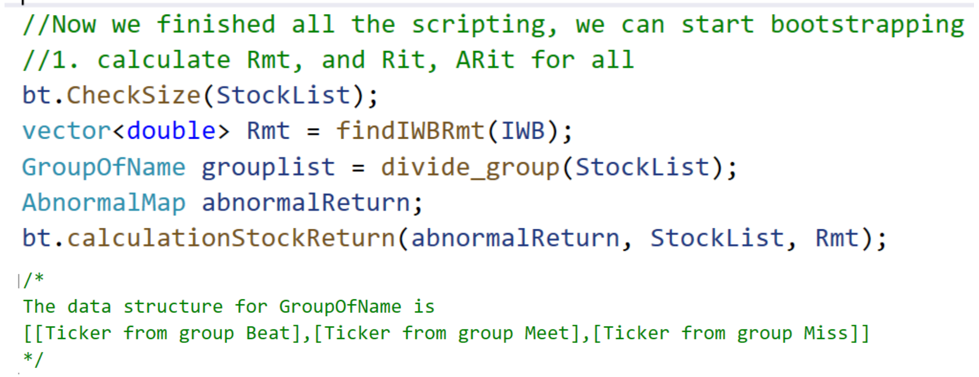
Calculate Rmt

Divide stocks into 3 groups

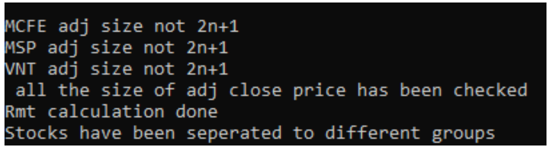
Calculation of Rit and ARit

B

C++: Prepare and Start Calculation

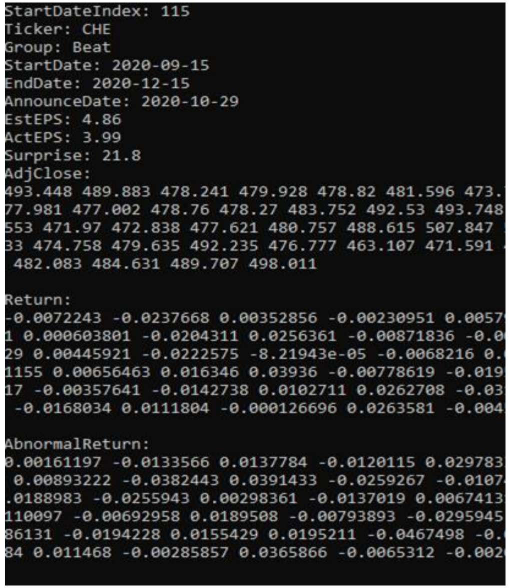


```cpp
//Now we finished all the scripting, we can start bootstrapping
//1. calculate Rmt, and Rit, ARit for all
bt.CheckSize(StockList);
vector<double> Rmt = findIWBRmt(IWB);
GroupOfName grouplist = divide_group(StockList);
AbnormalMap abnormalReturn;
bt.calculationStockReturn(abnormalReturn, StockList, Rmt);
/*
The data structure for GroupOfName is
[[Ticker from group Beat],[Ticker from group Meet],[Ticker from group Miss]]
*/
```



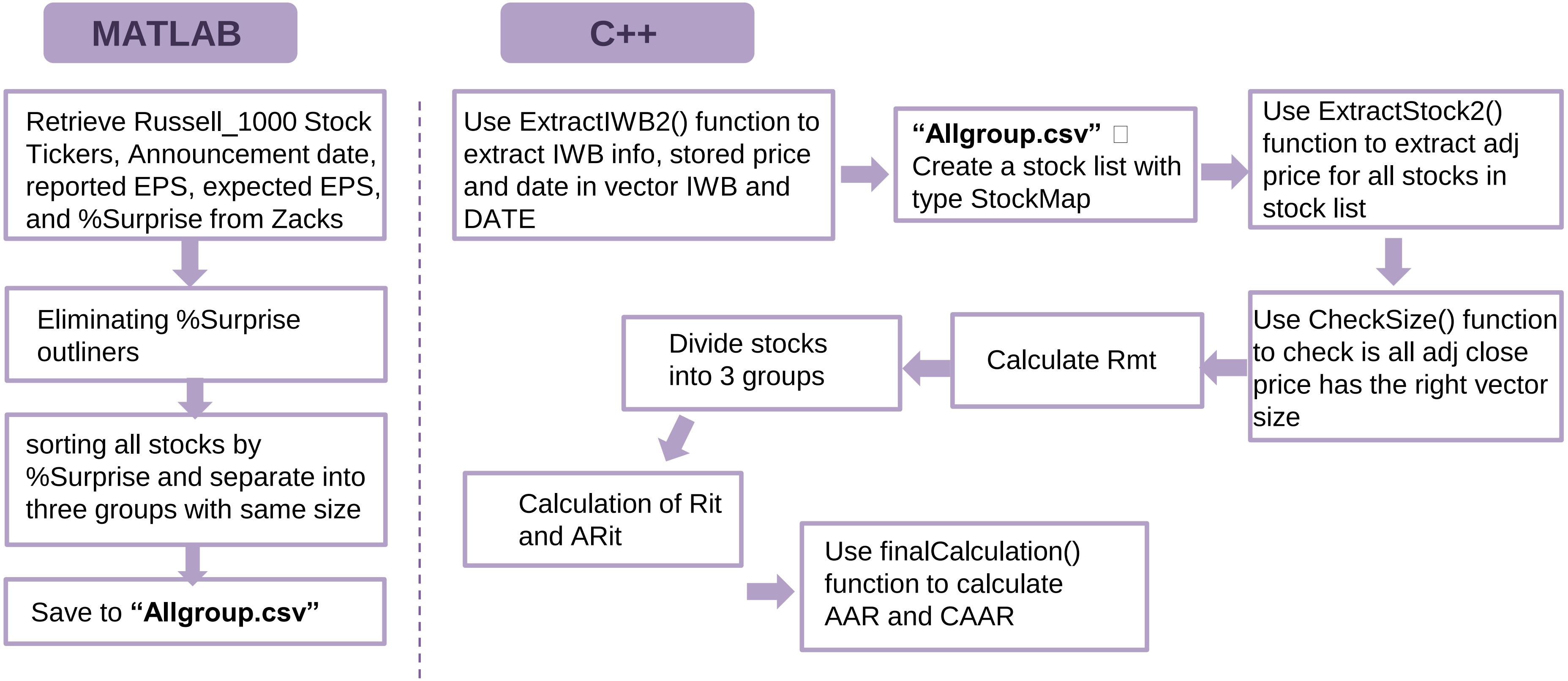
```
MCFE adj size not 2n+1
MSP adj size not 2n+1
VNT adj size not 2n+1
 all the size of adj close price has been checked
Rmt calculation done
Stocks have been seperated to different groups
```

One Sample Stock
from Stock List



```
StartDateIndex: 115
Ticker: CHE
Group: Beat
StartDate: 2020-09-15
EndDate: 2020-12-15
AnnounceDate: 2020-10-29
EstEPS: 4.86
ActEPS: 3.99
Surprise: 21.8
AdjClose:
493.448 489.883 478.241 479.928 478.82 481.596 473.
77.981 477.002 478.76 478.27 483.752 492.53 493.748
553 471.97 472.838 477.621 480.757 488.615 507.847
33 474.758 479.635 492.235 476.777 463.107 471.591
 482.083 484.631 489.707 498.011

Return:
-0.0072243 -0.0237668 0.00352856 -0.00230951 0.0057
1 0.000603801 -0.0204311 0.0256361 -0.00871836 -0.0
29 0.00445921 -0.0222575 -8.21943e-05 -0.0068216 0.
1155 0.00656463 0.016346 0.03936 -0.00778619 -0.019
17 -0.00357641 -0.0142738 0.0102711 0.0262708 -0.03
 -0.0168034 0.0111804 -0.000126696 0.0263581 -0.004

AbnormalReturn:
0.00161197 -0.0133566 0.0137784 -0.0120115 0.02978
 0.00893222 -0.0382443 0.0391433 -0.0259267 -0.0107
.0188983 -0.0255943 0.00298361 -0.0137019 0.0067413
110097 -0.00692958 0.0189508 -0.00793893 -0.0295945
86131 -0.0194228 0.0155429 0.0195211 -0.0467498 -0.
84 0.011468 -0.00285857 0.0365866 -0.0065312 -0.002
```

Functionality & Implementation



MATLAB

Retrieve Russell_1000 Stock Tickers, Announcement date, reported EPS, expected EPS, and %Surprise from Zacks

Eliminating %Surprise outliners

sorting all stocks by %Surprise and separate into three groups with same size

Save to **“Allgroup.csv”**

C++

Use ExtractIWB2() function to extract IWB info, stored price and date in vector IWB and DATE

“Allgroup.csv” □ Create a stock list with type StockMap

Use ExtractStock2() function to extract adj price for all stocks in stock list

Use CheckSize() function to check is all adj close price has the right vector size

Calculate Rmt

Divide stocks into 3 groups

Calculation of Rit and ARit

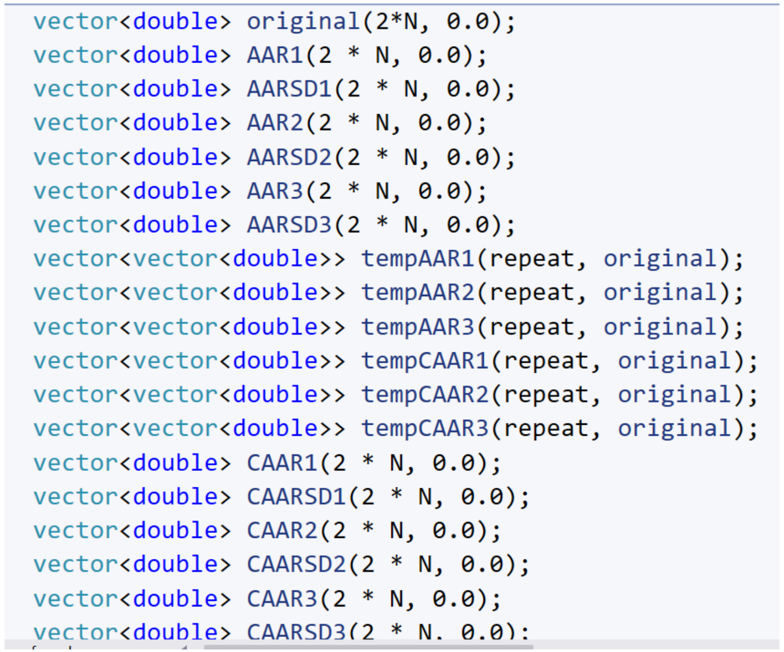
Use finalCalculation() function to calculate AAR and CAAR

C++:

Bootstrapping: Details of Finalcalculation function

1. Creating Matrix of [3 X (4X1)] to store average of AAR and CAAR, standard deviation of AAR and CAAR of three groups – Beat, Meet, and Miss.
2. AAR#, CAAR# [2*interval(N) x1]to store the average
3. tempAAR#, and tempCAAR# with [2 * interval(N) x 40] size to store result from all 40 computation.

```
vector<vector<vector<double>>> Result(3, vector<vector<double>>(4));
```



```
vector<double> original(2*N, 0.0);
vector<double> AAR1(2 * N, 0.0);
vector<double> AARSD1(2 * N, 0.0);
vector<double> AAR2(2 * N, 0.0);
vector<double> AARSD2(2 * N, 0.0);
vector<double> AAR3(2 * N, 0.0);
vector<double> AARSD3(2 * N, 0.0);
vector<vector<double>> tempAAR1(repeat, original);
vector<vector<double>> tempAAR2(repeat, original);
vector<vector<double>> tempAAR3(repeat, original);
vector<vector<double>> tempCAAR1(repeat, original);
vector<vector<double>> tempCAAR2(repeat, original);
vector<vector<double>> tempCAAR3(repeat, original);
vector<double> CAAR1(2 * N, 0.0);
vector<double> CAARSD1(2 * N, 0.0);
vector<double> CAAR2(2 * N, 0.0);
vector<double> CAARSD2(2 * N, 0.0);
vector<double> CAAR3(2 * N, 0.0);
vector<double> CAARSD3(2 * N, 0.0);
```

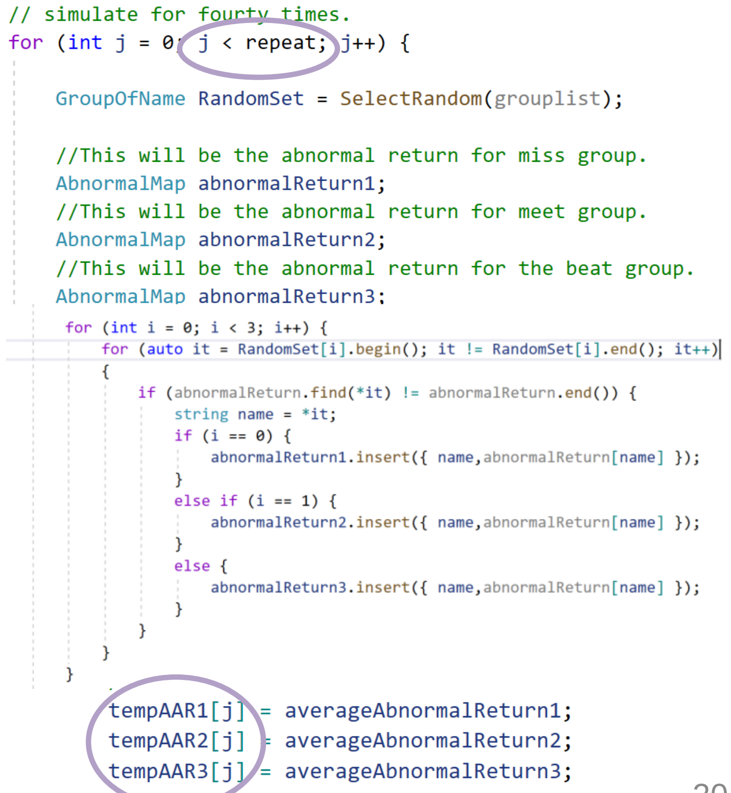
C

C++:

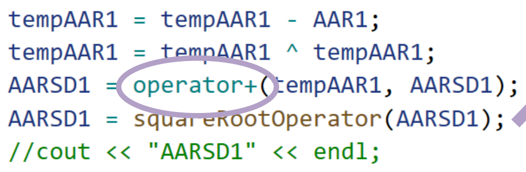
Bootstrapping: Details of Finalcalculation function

1. Randomly select 50 stocks from grouplist for calculation (40 times)

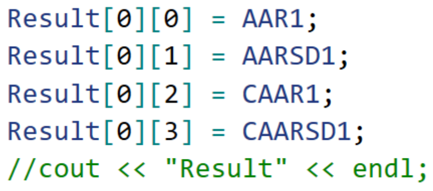
1. Overloading Operators for standard deviation computation



```cpp
// simulate for fourty times.
for (int j = 0; j < repeat; j++) {

    GroupOfName RandomSet = SelectRandom(grouplist);

    //This will be the abnormal return for miss group.
    AbnormalMap abnormalReturn1;
    //This will be the abnormal return for meet group.
    AbnormalMap abnormalReturn2;
    //This will be the abnormal return for the beat group.
    AbnormalMap abnormalReturn3;
     for (int i = 0; i < 3; i++) {
         for (auto it = RandomSet[i].begin(); it != RandomSet[i].end(); it++)
         {
             if (abnormalReturn.find(*it) != abnormalReturn.end()) {
                 string name = *it;
                 if (i == 0) {
                     abnormalReturn1.insert({ name,abnormalReturn[name] });
                 }
                 else if (i == 1) {
                     abnormalReturn2.insert({ name,abnormalReturn[name] });
                 }
                 else {
                     abnormalReturn3.insert({ name,abnormalReturn[name] });
                 }
             }
         }
     }
    tempAAR1[j] = averageAbnormalReturn1;
    tempAAR2[j] = averageAbnormalReturn2;
    tempAAR3[j] = averageAbnormalReturn3;
```

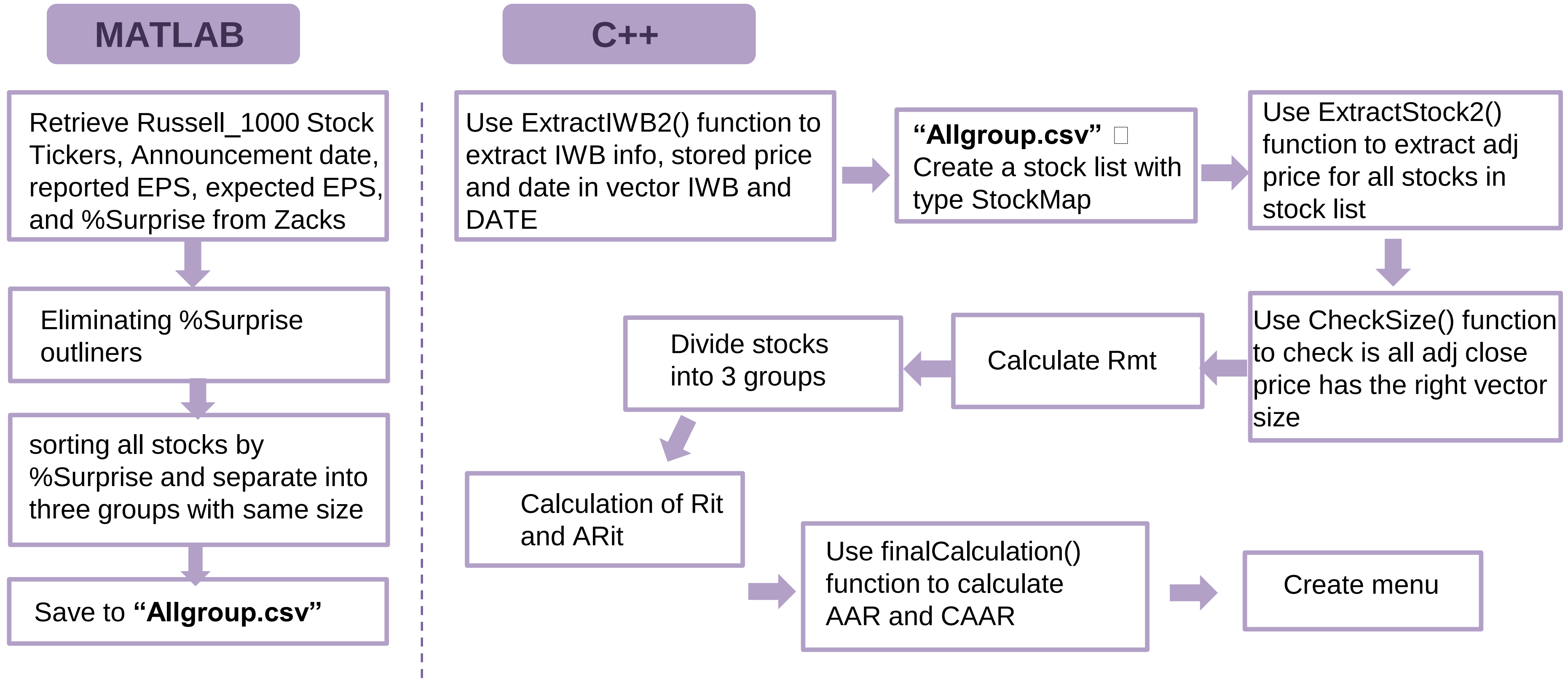


```cpp
tempAAR1 = tempAAR1 - AAR1;
tempAAR1 = tempAAR1 ^ tempAAR1;
AARSD1 = operator+(tempAAR1, AARSD1);
AARSD1 = squareRootOperator(AARSD1);
//cout << "AARSD1" << endl;
```



```cpp
Result[0][0] = AAR1;
Result[0][1] = AARSD1;
Result[0][2] = CAAR1;
Result[0][3] = CAARSD1;
//cout << "Result" << endl;
```

Functionality & Implementation



MATLAB

Retrieve Russell_1000 Stock Tickers, Announcement date, reported EPS, expected EPS, and %Surprise from Zacks

Eliminating %Surprise outliners

sorting all stocks by %Surprise and separate into three groups with same size

Save to **“Allgroup.csv”**

C++

Use ExtractIWB2() function to extract IWB info, stored price and date in vector IWB and DATE

“Allgroup.csv” □ Create a stock list with type StockMap

Use ExtractStock2() function to extract adj price for all stocks in stock list

Use CheckSize() function to check is all adj close price has the right vector size

Calculate Rmt

Divide stocks into 3 groups

Calculation of Rit and ARit

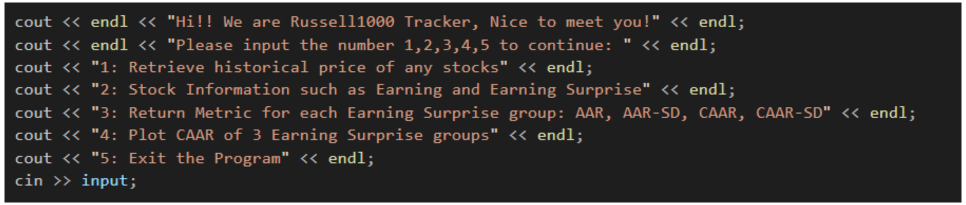
Use finalCalculation() function to calculate AAR and CAAR

Create menu

C++:

Creating menu: Switch function with 5 input options.

- The menu function is designed to ask for input as a number ranged from 1-5 with explanation of each menu option.
- The number input is plugged in to opt variable as an input to switch function.
- Every time the program finishes running one option, it will ask users whether users want to continue in using the same option or not by putting Y or y as yes and N or n as no. There are while loops checking the input or Y/N on every menu option.
- Results printed are set with precision of 6 digits.
- The menu option #5 is exit and it will reconfirm with users again with Y/N whether they really want to quit the program.



```
cout << endl << "Hi!! We are Russell1000 Tracker, Nice to meet you!" << endl;
cout << endl << "Please input the number 1,2,3,4,5 to continue: " << endl;
cout << "1: Retrieve historical price of any stocks" << endl;
cout << "2: Stock Information such as Earning and Earning Surprise" << endl;
cout << "3: Return Metric for each Earning Surprise group: AAR, AAR-SD, CAAR, CAAR-SD" << endl;
cout << "4: Plot CAAR of 3 Earning Surprise groups" << endl;
cout << "5: Exit the Program" << endl;
cin >> input;
```

C++:

Plot charts of CAAR1/ CAAR2/ CAAR3: Gnuplot.

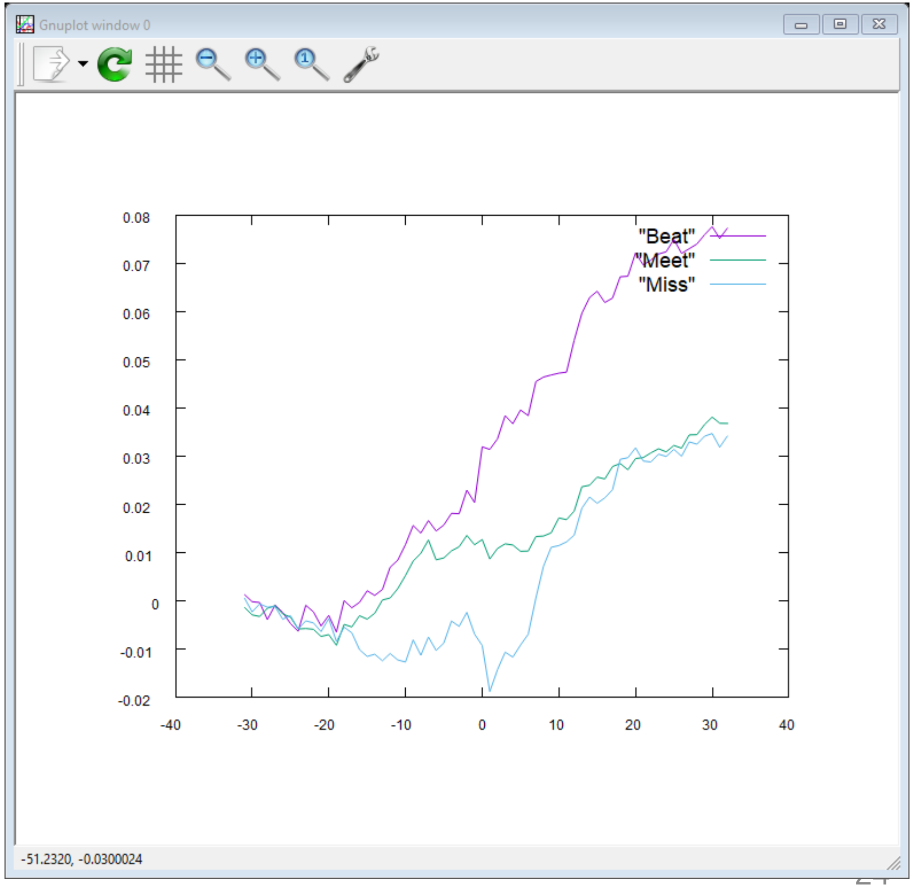
- Function plotChart is created as an independent function. The function is designed to receive input from three vectors of double from “Beat”, “Meet”, “Miss”
- x1, y1, x2, y2, x3, y3 are double variables setup to line plot.
- GnuplotPipe is setup to open gnuplot library from local environment.
- For each plot of CAAR by groups, x-axis shows as day number with 0 as EPS announcement date and y-axis shows value of CAAR for each group.
- The function plots line graph of three groups altogether.

Gnuplot & Conclusion

When we are trying to find the effect for earnings release on stock's returns against the benchmark, we use gnuplot to directly present the linear relationship and the vertical difference for the three different groups: beat, meet, and miss.
so the horizontal line represents the trade index: -N to N-1

x1 = x2 = x3 = [-N, N-1]
y1 = [CAAR of miss group]
y2 = [CAAR of meet group]
y3 = [CAAR of beat group]

Conclusion: the biggest jump of abnormal return usually occur during 1st 5 day after announcement date. From the trading strategy, people could leverage this in information for trend reversion strategy.



Enrichment

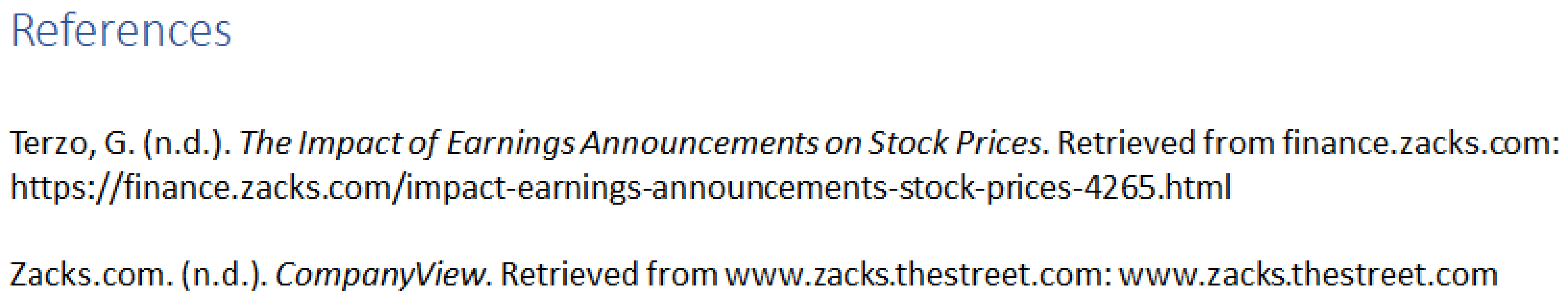
Enrichment

We create the application that will help investor track the stock return with their effect of earning surprise.

Since some of the investor, might blindly invest in the company that has high surprise and sell the company that has low surprise.

These investors will tend to lose money, however if the investor pay attention to our application, they can easily prevent this loss and generate new strategy for investing with earning announce surprise.

Reference



References

Terzo, G. (n.d.). *The Impact of Earnings Announcements on Stock Prices*. Retrieved from finance.zacks.com: https://finance.zacks.com/impact-earnings-announcements-stock-prices-4265.html

Zacks.com. (n.d.). *CompanyView*. Retrieved from www.zacks.thestreet.com: www.zacks.thestreet.com