
# FRE 6883
# Financial Computing

## Final Project – Trading Strategy on EPS

Prof. Song Tang

**12/16/2014**

**Authors:**
Manshen Lin ML3156@nyu.edu
Ruiyang Xu RX319@nyu.edu
Gaurav Sharma gaurav.sharma@nyu.edu
Bo Liu BL1615@nyu.edu

# Contents

## 1. Executive Summary

The project can be divided into 4 parts

(1) Construct the stock poll consisting of 93 stocks from S & P 500 through research and store their information in one file “StockInfo.txt”.

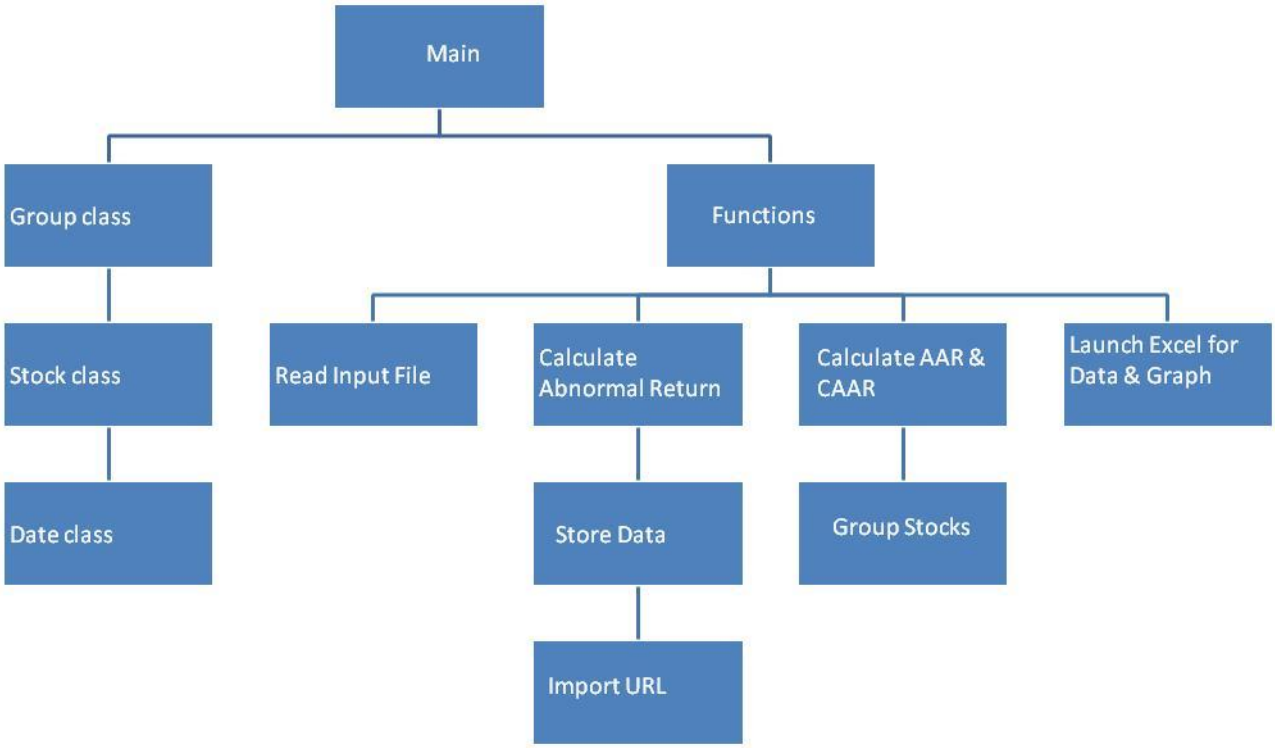
(2) Input the information from “StockInfo.txt” to C++ and store it in certain STL containers, use liburl to retrieve historical prices data of those stocks and SPY from Yahoo Finance and store them in certain STL containers or classes.

(3) Calculate the return, abnormal return ;Sort the stock into three groups according to their EPSs and calculate the AAR and CAAR for each group

(4) Create Excel charts to compare and present the AAR and CAAR of the groups and Evaluate the impact of earning report on stock price

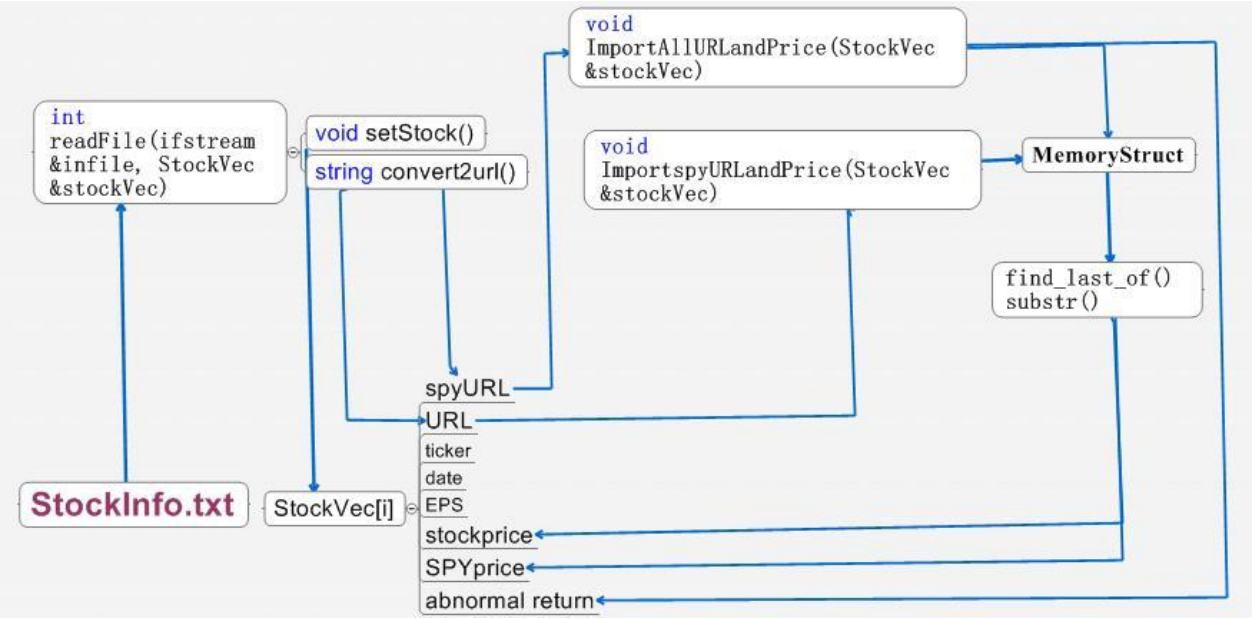
Note: Total Program Run Time approx. 37 Seconds. This time includes reading the input file, create URL and download data from URL, perform data calculation, and launch Excel to print all the original stock and spy prices and draw graphs.

## 2. Architectural Design



## 3. Implementation Detail

(1)Input data and store it




**int readFile(ifstream &infile, StockVec &stockVec);**

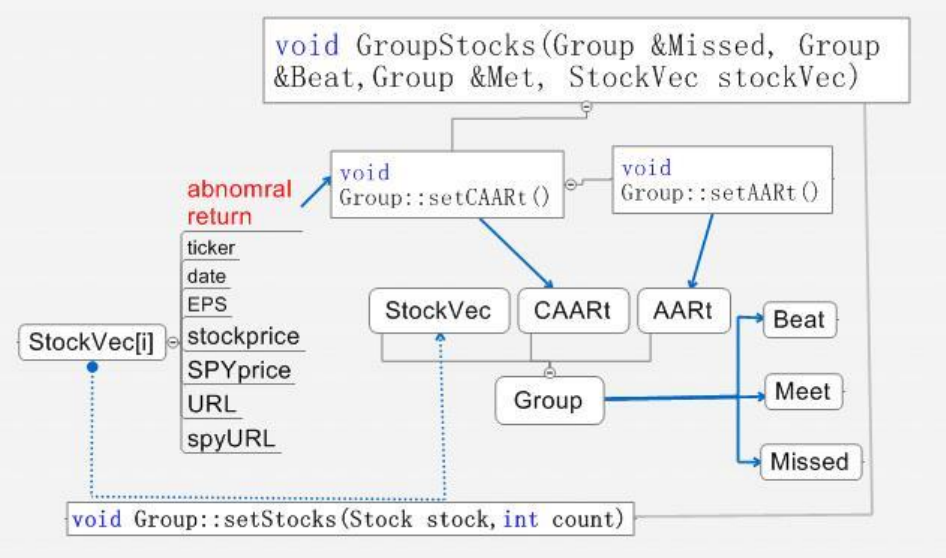
Read the stocks' information from "StockInfo.txt" and store all the data(including all the data of corresponding SPY) in StockVec[i] through **setStcok()** ,automatically create stock and spy URLs through through **convert2url().**

**void ImportAllURLandPrice(StockVec &stockVec)**

**void ImportspyURLandPrice(StockVec &stockVec)**

Use libcurl to retrieve the stocks' prices and spy prices and store them in StockVec[i] through **struct MemoryStruct**;

(2) Group and Calculation




Group Missed, Beat, Met;

**void GroupStocks(Group &Missed, Group &Beat,Group &Met, StockVec stockVec)**

Store the stock classes to different group classed through "if-else" loop on threshold judgment(Actual EPS-Estimated EPS) and **setStocks();**Calculate the AARt and CAARt through

**setAARt()** and **setCAARt()**;

(3)Print to excel

**void printtoexcel(StockVec &stockVec,Group &Beat,Group &Met,Group &Missed);**

The function called by main function to create a excel file with the data including the prices of stocks,the prices of Spys, the AARt and CAARt of all the groups and create line graphs of the AARt and CAARt for every group.

**Vector<double, long> v2V(Vector1 &V);**

Convert the Vector1 type to the Vector type defined by the exceldriver functions

**void Vectorlist(list<Vector<double, long> > &vectorList,Matrix1 &M);**

Convert the Matrix of Matrix1 type to list<Vector<double, long> > type

**Matrix<double,long> trans(Matrix1 &M);**

Convert the Matrix of self-defined type Matrix1 to type Matrix defined by the exceldriver functions

**void ExcelDriver::AddMatrix(const std::string& name, const NumericMatrix<double, long> & matrix, const list<std::string> & rowLabels, const list<std::string> & columnLabels)**

Print a matrix to one sheet of the excel file

**void ExcelDriver::CreateChart(const Vector<double, long> & x, const list<std::string> & labels, const list<Vector<double, long> > & vectorList, const std::string& chartTitle, const std::string& xTitle, const std::string& yTitle)**

Print the graph of **vectorList** to one sheet of the excel file

## 4. Delivery Stages

| Version | Description | Date | Status | Note |
|---|---|---|---|---|
| 1.0 | Read Infile & Import URL Data using libcurl | 11/24/2014 | Completed | |
| 2.0 | Perform calculation using independent function | 11/24/2014 | Completed | |
| 2.1 | Launch Excel and output data into Excel based on 2.0 | 11/27/2014 | Completed | |
| 2.0.1 | Data Structure(creating classes) based on version 2.0 | 12/4/2014 | Completed | |
| 3.0 | Launch Excel and draw its graphs based on version 2.1 | 12/4/2014 | Completed | Initial QA completed |
| 4.0 | Merge version 2.0.1 and version 3.0 | 12/7/2014 | Completed | 2nd QA completed |
| 4.0.1 | Remove the specific group assignment for a stock | 12/14/2014 | Completed | QA completed |
| 4.0.2 | Write libcurl into memory instead of txt file | 12/14/2014 | Completed | QA completed |

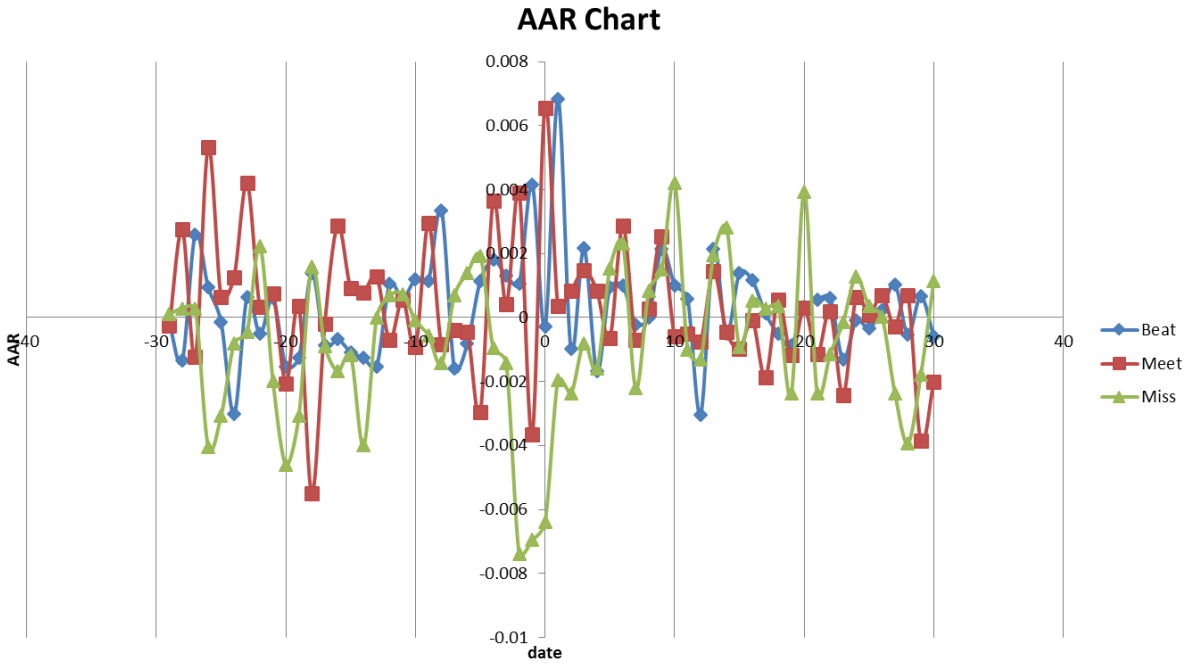
## 5. Task Allocation

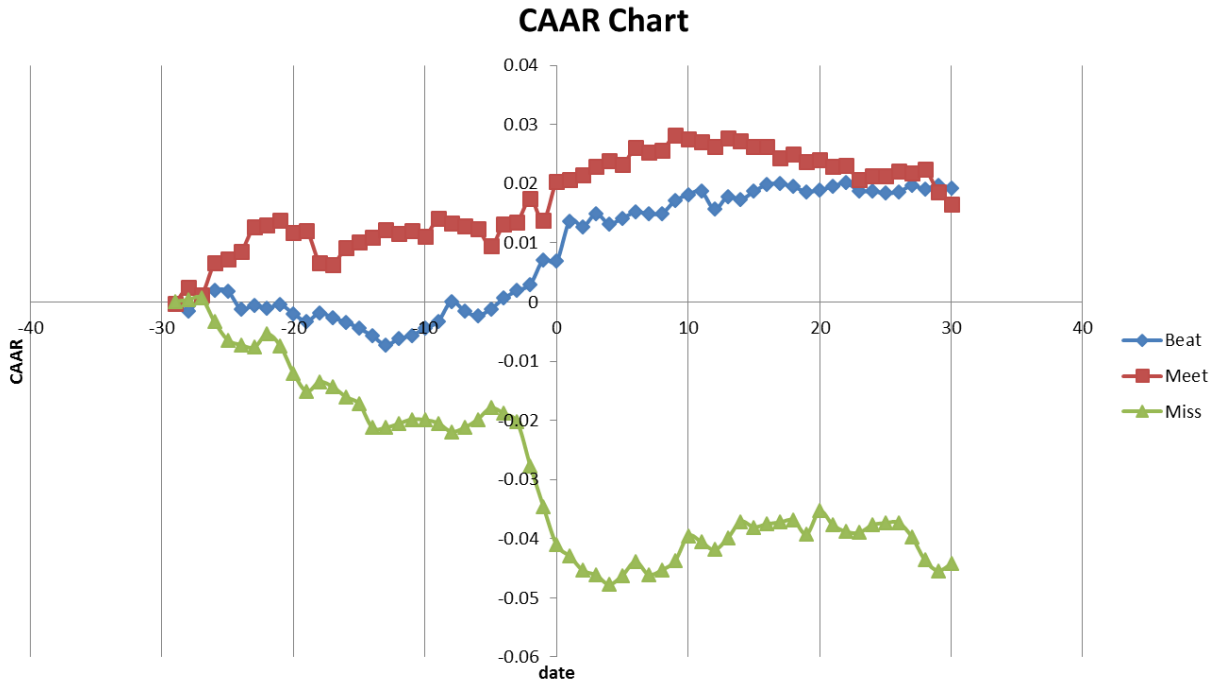
| Task | Allocation |
|---|---|
| **Stock Selection** | **Manshen, Gaurav, Bo** |
| **Algorithm Design** | **All** |
| **General Code Implementation** | **Ruiyang** |
| **Class Code Implementation** | **Manshen** |
| **Testing** | **Gaurav, Bo** |
| **Improvement (Libcurl to Memory)** | **All** |

## 6. Stock Selection

| Group | Threshold (H): H=5 cents | # of Stocks |
|---|---|---|
| Beat | Actual-Estimated>=H | 31 |
| Missed | 0<Actual-Estimated<H | 31 |
| Met | Actual-Estimated<0 | 31 |

## 7. Results


AAR Chart
0.008
0.006
0.004
0.002
0
-0.002
-0.004
-0.006
-0.008
-0.01
AAR
-40
-30
-20
-10
0
10
20
30
40
date
Beat
Meet
Miss


CAAR Chart
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
CAAR
-40
-30
-20
-10
0
10
20
30
40
date
Beat
Meet
Miss

# 8. Testing (Using MS Excel/MATLAB)

## 8.1 TESTING THE INPUT DATA

The input data of the stocks and SPY are tested by using an R Program to download the data into a csv file and using that to compare to the stock & SPY data downloaded into the C++ program.

The following files are needed to test the input data of the stocks and SPY.

- "InputData.xls" : This file contains the following columns of information about the selected stocks for each group : Group (Beat/Meet/Miss), Ticker, Estimated EPS, Actual EPS, Difference, Day 0 (Day of Earnings Announcement), 30 Day Prior (30 trading days prior to Earnings Announcement), 30 Day After (30 trading days after Earnings Announcement).

We will use the columns Ticker, 30 Day Prior & 30 Day After from the "InputData.xls" file. The following is procedure for downloading stock data:

- Create .txt files for the columns, 30 Day Prior & 30 Day After. (Please see "StartDates.txt" & "EndDates.txt")
- Create .csv file for the Ticker column. (Please see "ProjectStocks.csv")
- Run the R program. Please see "WorkingCode.txt" for the R code used. The program uses tseries library in R to run download stock data from Yahoo Finance as per three arrays of data i.e. stockstkr, startdates & enddates. Once this information is downloaded from Yahoo Finance the program then concatenates the data into a single data frame (matrix) and writes it into a .csv file in the working directory called "StockDataOutput.csv"

Testing the input and calculations of the program is conducted by creating a MS Excel file using the stock and SPY data downloaded with the R code. This file is called "TestingFileFinal.xls" and is essentially the C++ program in MS Excel. It has all the data and calculations and additionally we will use it for testing the inputs and output of the C++ program.
Now we output the C++ code to the following files: "SampleOutput by code4.0_v2".

Finally we can now test the data using the "TestingFileFinal.xls" (Please see "TestingFileFinal.xls"):

- Test Stock Data
    - In "TestingFileFinal.xls" the "Stocks(C++)" tab contains stock prices from "SampleOutput by code4.0_v2" and "Stocks(R)" tab contains stock prices from "StockDataOutput.csv". The data from "StockDataOutput.csv" has been pasted transpose into excel.

- In "TestingFileFinal.xls" the "SPY(C++)" tab contains stock prices from "SampleOutput by code4.0_v2" and "SPY(R)" tab contains stock prices from "SPYDataOutput.csv". The data from "SPYDataOutput.csv" has been pasted transpose into excel.
- In "TestingFileFinal.xls" the "StockTest" tab compares stock prices from "Stocks" & "Stocks2" tabs and shows number of errors
- In "TestingFileFinal.xls" the "SPYTest" tab compares SPY prices from "SPY" & "SPY2" tabs and shows number of errors

**Input Data Testing Summary: The input data extracted by using C++ are same as the input data extracted by using R code. No error.**

## 8.2 TESTING THE OUTPUT DATA

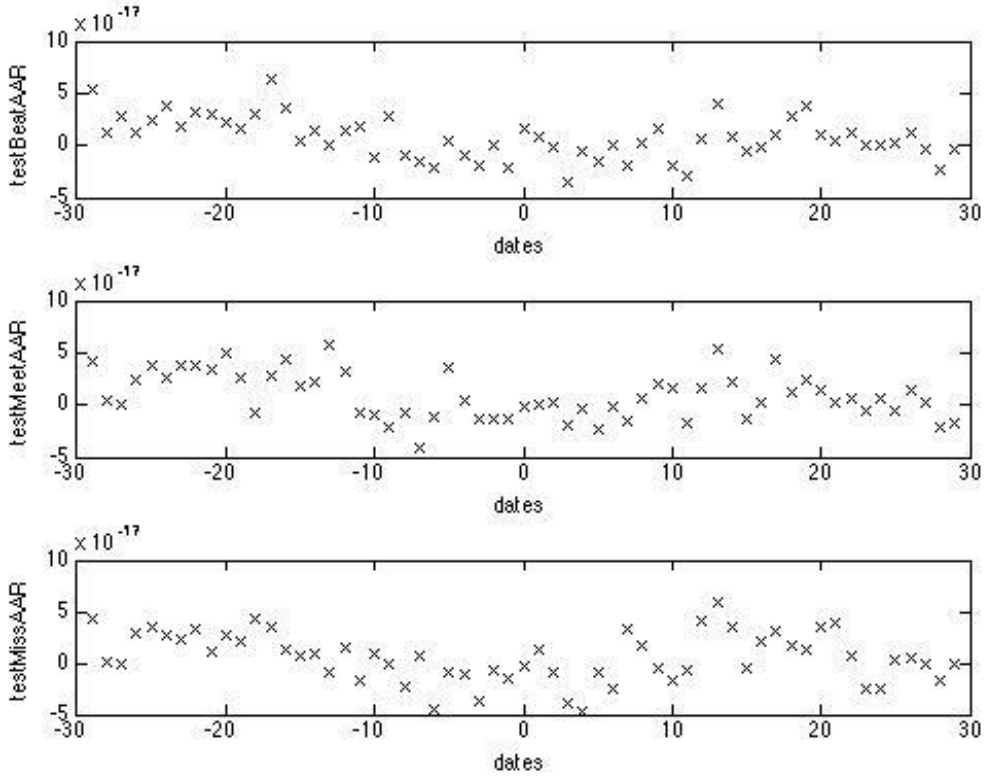
Once the input data is tested and validated, the Excel file and Matlab tests the CAAR Output & graphs.
Excel calculations are done in the file "TestingFileFinalFinal.xls", Matlab calculations are done in the file "testoutput 2".
Additional file needed: "SampleOutput by code4.0_v2"

In both the "TestingFileFinalFinal.xls":

- Calculate ARit
  - ARit is calculated for each stock for each day
  - Calculations in the "ARit" tab
- Calculate AARt
  - ARRt is calculated for each group of stocks for each day arranged by Beat, Meet, Miss
  - Calculations in the "ARRt" tab
- Calculate CAAR
  - Cumulative ARRt is calculated for each group of stocks for cumulative days arranged by Beat, Meet, Miss
  - Calculations in the "CARR" tab
  - Copy CAAR calculations from "SampleOutput by code4.0" and compare for errors
- Graph CAAR for each group of stocks
  - Graph is in the "Graph" tab & compare to graph in "SampleOutput by code4.0"
- Graph test errors for each group of stocks
  - Graph is in the "testerrors" tab in the "testoutput 2" file to show the difference between Matlab calculations and C++ calculations. For each groups of data, the graph shows errors between different calculations method are in 15 digits, almost nothing.

**Output Data Testing Summary: The calculation results from C++ are same as the results from Matlab and Excel.**

**Note: The following graph are the difference of AARt results between C++ and Matlab code. The error levels are 10^(-17) for the three groups. Since the error levels are so small, we consider the errors are due to computer truncation error, and the results from C++ are same as the results from Matlab.**




## 9. Conclusion

- AAR: Both Beat and Met have upward spikes after satisfied EPS. And the Missed group has a downward spike.
- CAAR: Both Beat and Met have stable upward movement after satisfied EPS. And the Missed group has a downward movement before of the disappointed EPS.

➔ Both AAR and CAAR behave the same way as we have expected.

## 10. Enrichment

1. Import data by using licurt memory, and store the data directly to the class without using a txt file as intermediary

2. Remove specific group assignment to a stock for future flexibility

3. QA Testing by using R, Excel & Matlab

## 11. Reference

[1] https://www.marketgrader.com
[2] http://investing.businessweek.com