

The background image shows an aerial view of the Chicago skyline, featuring the Loop area with its dense cluster of skyscrapers, and the surrounding city buildings extending towards the lakefront. The deep blue of Lake Michigan is visible to the right.

# The Impact of Earnings Announcements on Stock Price

FRE 6883: Financial Computing  
Team member: Fan Gao, Minhang Chen,  
LingLing Liu, Yimeng Ma, Zhuo Deng



# Executive Summary

- Aim: to analysis the impact of earnings releases on their stock prices
- Steps:
  - 1) Discussed the whole project design collectively and assigned tasks to each team member .
  - 2) Coded and debugged our assigned parts
  - 3) Module integration and testing
  - 4) Improved the codes to guarantee efficiency
- Result: successfully completed the project; and the program's running time is within 1.5 minutes.



# Task Allocation



1

Data Import and Cleaning :  
Lingling Liu, Minhang Chen,  
Yimeng Ma

2

Calculation Part: Fan Gao

3

Generate Graph: Zhuo Deng

4

Menu Part: Minhang Chen

5

PPT: Lingling Liu, Yimeng Ma

# Project Design



- Imported stock data from Bloomberg and categorize stocks into three groups
- Retrieved 61-day historical stock prices from Yahoo Finance
- Create classes and data structure
- Implement Bootstrap and calculate AAR and CAAR
- Use Excel Driver to draw the graph
- Design the menu
- Enhancement: bootstrap

# Import Data

- Download all the stocks' tickers, estimated and actual EPS, and EPS release date.
- In the excel file, decide the start-date and end-date of each stock's 61 days period and change all the date to format yyyy-mm-ddThh:mm:ss. Save as csv file.
- Read csv and store data in three maps (`map<string, string>` &`map1`, `map<string, vector<double>>`&`map2`, `map<string, vector<string>>` &`map3`)

# Libcurl



1

3 maps:

Call by value:

Map: get symbol, startdate and  
enddate

Call by reference:

Map: store symbol and stock/SPY  
adjusted close price

2

Generate vector of symbols  
Get cookies

3

For loop :get stock/SPY price

4

Store the value in maps

# Stock Selection

Beat:  
Actual EPS > 105% Estimated EPS

Meet:  
95% Estimated EPS <= Actual EPS <= 105% Estimated EPS

Miss:  
Actual EPS < 95% Estimated EPS

Use the iterator of  
`map<string,vector<double>> map2`  
to compare the actual EPS and  
estimated EPS of each stock

# Class & Containers

```
void setdata(stock,  
spy, releasedate,  
epsearning){....}
```

```
vector<vector<string> >  
categorize(double rate){....}  
1st row beat vector<string>  
2nd row meet  
3rd row miss
```

**Vectord AAR(){....}**  
Overload operation for  
vector and matrix:  
vector +/- vector  
vector / float  
matrix - matrix

```
class stockclass;  
map <string, Vectord > stock;  
map <string, Vectord > SPY;  
map <string, string> releasedate;  
map <string, Vectord > EPSEarning;
```

```
void setdata(stock, spy){....}
```

**Function Template**  
**T Select(vector<string> stockname, T obj)**  
for (vector<string>::const\_iterator iter = stockname.begin();  
iter != stockname.end(); iter++) {  
select[\*iter] = obj[\*iter];  
}

**Vectord CAAR(Vectord AAR) {**  
Vectord result;  
result.push\_back(AAR[0]);  
for (int i = 1; i < AAR.size(); i++) {  
result.push\_back(result.back() + AAR[i]);  
}



# UML

## Class stockclass

- + stock: map<string, vector<double> >
- + SPY: map<string, vector<double> >
- + releasedate: map<string, string> >
- + EPSEarning: map<string, vector<double> >

- + setdata(map<string, vector<double> > stock\_, map<string, vector<double> > spy\_, map<string, string> release, map<string, vector<double> > EPS):void
- + setdata(map<string, vector<double> > stock\_, map<string, vector<double> > spy\_):void
- + categorize(double rate):vector<vector<string> >
- + function template Select(vector<string> stockname, T obj):T
- + AAR():vector<double>
- + CAAR(vector<double> AAR):vector<double>

# BootStrapping(100 times)

Select 40 random stocks,

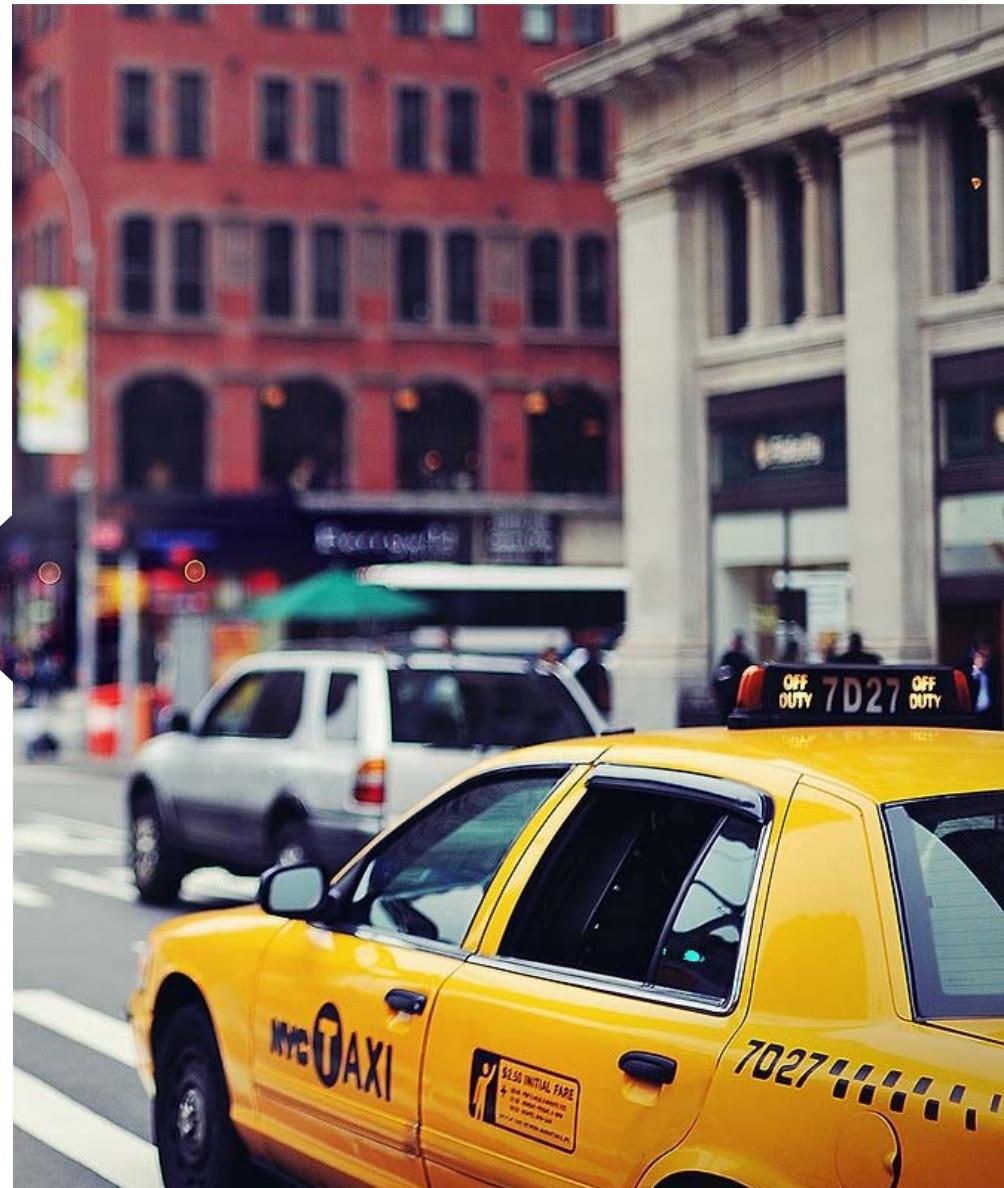
result (matrix of string, size = 100, do 100 times)

```
for (int i = 0; i < 40; i++) {  
    int randindex = rand() % N;  
    result[j].push_back(stockgroupname[k][randvector[randindex]]);  
    //erase the number I have selected each time to avoid stock overlap  
    randvector.erase(randvector.begin() + randindex);  
    N = randvector.size();
```

} calculate AAR, CAAR,

aar, caar(matrix of double, 100(times) X 60(pricedata) )

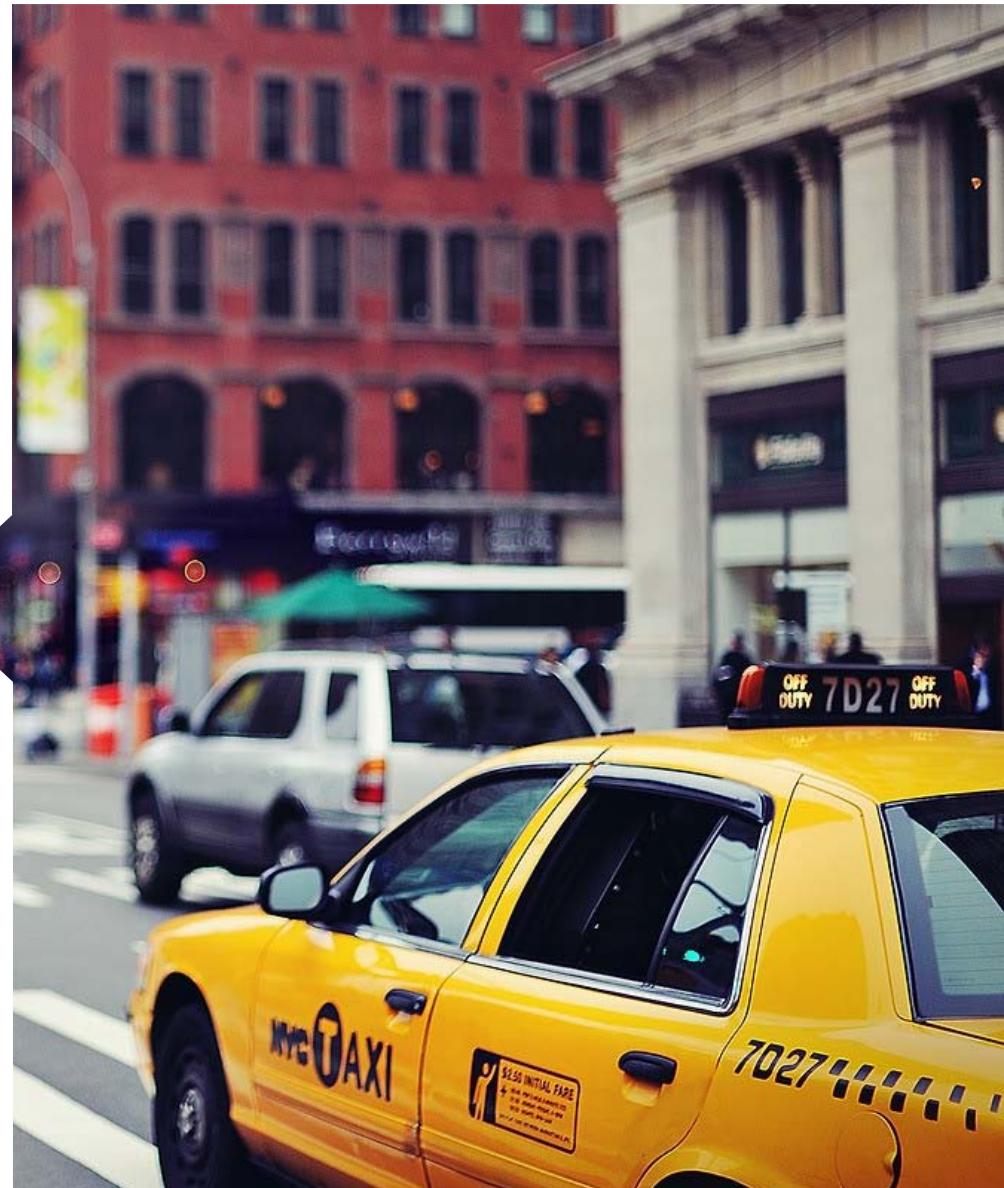
```
stockclass selectedGroup;  
vector<vector<string>> result;  
result.resize(100);  
for (int j = 0; j < 100; j++) {  
    int N = stockgroupname[k].size();  
    vector<int> randvector;  
    for (int i = 0; i < N; i++) {  
        randvector.push_back(i);  
    }  
    for (int i = 0; i < 40; i++) {  
        int randindex = rand() % N;  
        result[j].push_back(stockgroupname[k][randvector[randindex]]);  
        //erase the number I have selected each time to avoid stock overlap  
        randvector.erase(randvector.begin() + randindex);  
        N = randvector.size();  
    }  
    selectedGroup.setdata(threeGroup[k].Select<Map>(result[j], threeGroup[k].stock),  
                         threeGroup[k].Select<Map>(result[j], threeGroup[k].SPY));  
    aar.push_back(selectedGroup.AAR());  
    caar.push_back(selectedGroup.CAAR(aar.back()));  
}  
for (int i = 0; i < 100; i++) {  
    AARave = AARave + aar[i];  
    CAARave = CAARave + caar[i];  
}  
AARave = AARave / 100;  
CAARave = CAARave / 100;  
matrix_[k].push_back(AARave);  
matrix_[k].push_back(CAARave);
```



# BootStrapping(100 times)

```
return ResultMatrix;
```

	AAR	CAAR
Group beat	Vector<double> size = 60	Vector<double>
Group meet	Vector<double>	Vector<double>
Group miss	Vector<double>	Vector<double>



# Excel Chart

## 1. Function plot

PrintInExcel(xarr, list of yarr, text)

## 2. To be plotted, xarr, yarr are Daniel duffy vectors

Element & index type <double, long>

1) xarr[xarr.MinIndex()] = -29.0; for loop -29~30

2) yarr

Convert Class:

Initialize:daniel duffy

Member function get\_yarr()

3 object.get\_yarr - > 3 yarr



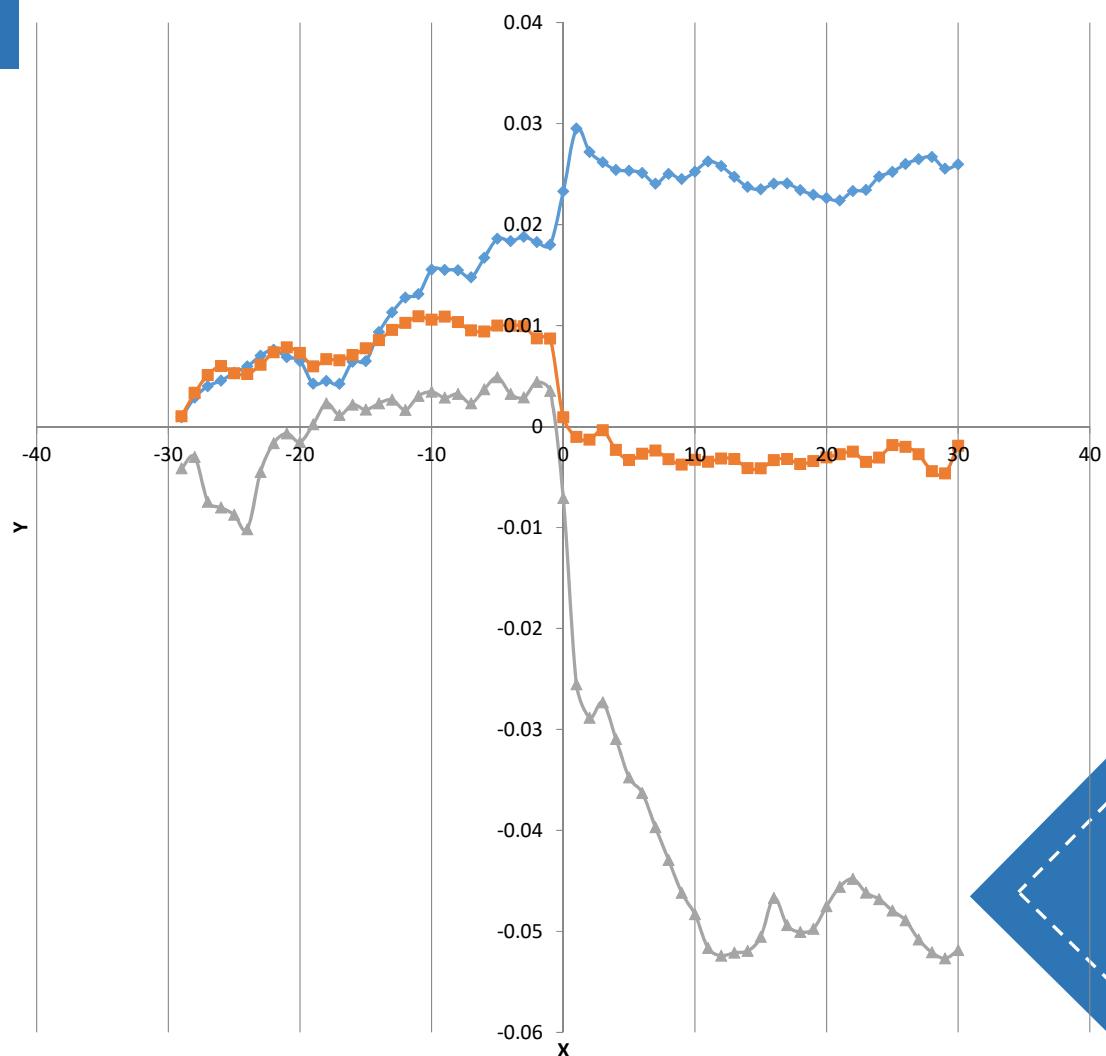
## Graphical Output

Plotting matrix of CAAR for 3 groups using ExcelDriver.

Group beat and Group miss have significant abnormal returns after announcement date.

## Excel Chart

CAAR



## Graphical Output

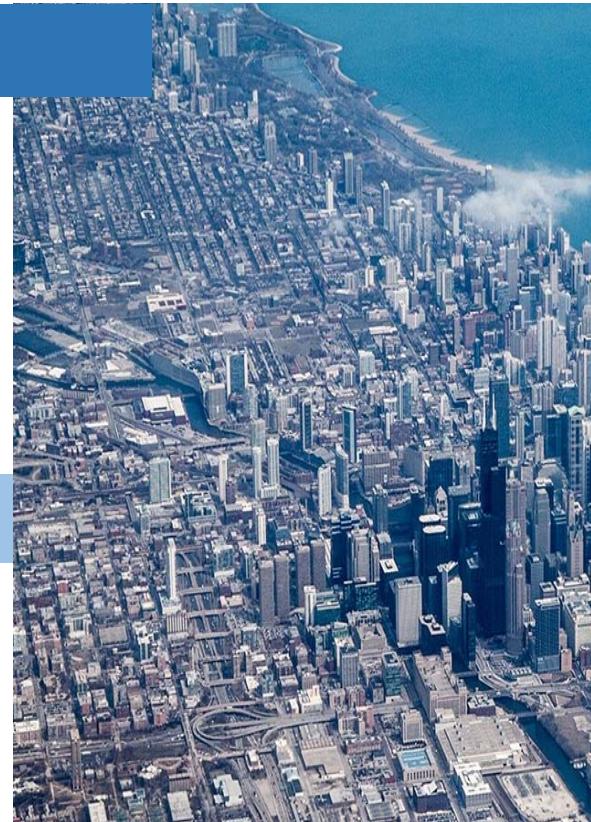
Plotting matrix of CAAR for 3 groups using ExcelDriver.

Group beat and Group miss have significant abnormal returns after announcement date.

# Results & Conclusion

## Group beat

Announcement of beating expected EPS has significant positive impact on stock return for the first 1 or 2 days after announcement date.



## Group miss

Announcement of missing expected EPS has significant negative impact on stock return for the first 10 days after announcement date.



# Reference

1

Yahoo Finance

2

Bloomberg

An aerial photograph of the Chicago skyline and surrounding urban landscape. The city is densely built with a mix of residential and commercial buildings. The Chicago River flows through the center, and Lake Michigan is visible to the east. A large, semi-transparent blue triangle is overlaid on the upper right portion of the image.

THANKS