Information Visualisation

Sebastião Queiroz e Mello
Instituto Superior Técnico
Lisbon, Portugal
sebastiao.q.mello@tecnico.ulisboa.pt

ABSTRACT

In this paper you will find a new way to visualise market data and interact with it.

The most precious time for a financial analyst is time, is the only thing that goes, never comes back, so get time back is the rule, one second can change the story during a trading day, this project can allow financial analyst seeing his favourite index in a different way.

This project was developed for the Information Visualisation course.

Keywords
D3.js; Python; Financial Markets, Quandl; Quantitative Analysis.

INTRODUCTION

Nowadays our world is surrounded by tons of information, texts, images, sounds. Managing data, in order to get useful and understandable information to someone is an enormous challenge.

When we develop a tool to be used by financial analysts, get time back is everything, time is the only thing we can't recover, but we can save it, by creating tools which allow this professionals to fully understand, in a very short time what does the information means. During a trading session, if we can save one second, per each trade, could the second which will make the difference, between winning or losing.

This project was done with the propose of create a visualisation of market data, to give time back to financial analysts.

With this assignment, I tried to answer to five main questions:

1. How the markets influence each others.
2. How the volatility index, VIX, influence the most important indexes.
3. What is the behaviour of volatility in an uptrend, downtrend or flat market,
4. What is the correlation between the index quote and the VIX index quote.
5. How historical moments influence the behaviour of a market.

By the end of this report, you should understand what is the behaviour of some of this markets, some questions could remain unanswered, to a non-financial person, due to the complexity of some of the technical methods and tools used to develop this visualisation, and the complexity of the work, the question 4 and 5 could not address the all of the problem.

The project was developed using D3.js and Python. All of the data was collected from quandl.com, a financial website, which has some free databases, with data updated every day, by Yahoo Finance.

RELATED WORK

During the last few years, I have been researching a lot about financial markets, this subject is my passion and my life, this work is part of my researching project.

The first time that someone explain me the concept of volatility, it was quite hard to properly understand what it real mean. Reading some definitions on the internet wasn't enough. Most of the main financial tools, such as Investing, Yahoo Finance or even Google Finance, don't allow you to have a view of some of the methods that I used in this assignment, so matching the need, that any of that websites couldn't answer, with my need of fully understanding the volatility concept, took me to choose this subject.

When I attended to Human-Computer Interaction course, of the most important concepts that I learn was the type of users: new user, intermediate user and advance user, so focus on new and intermediate users was what I wanted, because the learning curve to understand graphs is quite hard, so I tried to develop a very user-friendly interface.

In figure 1, as you can see Investing allow you a lot of interaction, but sometimes it becomes confusing, so you don't understand what can you do with this graph.

I tried to approach Google's user interface, is very clean simple to use, and doesn't have a lot of information, figure 2.

Some financial concepts that I applied in this assignment, I learnt from Investopedia, the biggest financial education website.

Figure 1 Investing graph interface

Figure 2 Google Finance graph interface

THE DATA

As presented in the introduction of this report, all of the data was collected from Quandl, which has an API, each allow us to get accurate and updated market data.

The data collected could be encoded into a JSON, CSV or XML file.

In this case, I used JSON files, because I have been working with JSON files and objects, from a long time, and also it is simpler to manage the information, compared with CSV or XML.

Since this is a standalone solution, I created a script to get all of the needed data, using Python, and I stored into a JSON file.

The datasets were manipulated using Javascript, this could be also manipulated by Python, but since we are working with JS, I decided to manipulate with this language.

The data used for this project is completely updated, and I decided get it since 2000 until today.

The datasets have the following structure:

[[“Date”,”Open”,”Low”,”High”,”Close”,”Volume”],…, [“Date”,”Open”,”Low”,”High”,”Close”,”Volume”]]

The dataset was transformed in the following structure:

```
{
        {0:
                { Date: ‘24-12-1992’
                 Open: ‘1992’
                 Close: ‘2010’
                 High: ‘2010’
                 Low: ‘1980’
                 Volatility: ‘0.003’
                 Moving_Average_25: ‘1980’
                 Moving_Average_200: ‘1970’
        }
….
        {20000:
                { Date: ‘24-12-2015’
                 Open: ‘1992’
                 Close: ‘2010’
                 High: ‘2010’
                 Low: ‘1980’
                 Volatility: ‘0.003’
                 Moving_Average_25: ‘1980’
                 Moving_Average_200: ‘1970’
                }
        }
}
```

- Date is the date of the trading day;
- Open is the open price of a trading day;
- Close is the close price of a trading day;
- High is the highest price of a trading day;
- Low is the lowest price of a trading day;
- Volatility is a financial indicator;

- Moving_Average_10 is a technical indicator, of the last 10 days.
- Moving_Average_200 is a technical indicator, of the last 200 days.

A JSON object/file, works as hash table, so we have a key and a value, the key is the position of the value in the table, so we can reference a value like this, dataset[0], and if we want a property of the the object we can do dataset[0].open.

To fully understand some of the decisions taken during the development process is important to understand some of the indicators.

To calculate this indicators I had the need to derive data, in order to get the desired results. A better explanation on how to get this measures will be presented later.

Volatility

Volatility is a statistical measure of the dispersion of returns for a given security or market index. Volatility can either be measured by using the standard deviation or variance between returns from that same security or market index. Commonly, the higher the volatility, the riskier the security.

To calculate the volatility we use the following method:

1. Calculate the average price for the number of periods or observations, in this case the number of periods is 10.
2. Determine each period's deviation.
3. Square each period's deviation
4. Sum the squared deviations.
5. Divide this sum by the number of observations.
6. The standard deviation is then equal to the square root of that number.

Moving Average

Moving Average is a widely used indicator in technical analysis that helps smooth out price action by filtering out the “noise” from random price fluctuations. A moving average (MA) is a trend-following or lagging indicator because it is based on past prices. There are two types of Moving Average, the simple (SMA) and the exponential (EMA). In this work I decide to use the SMA, because consume less resources than the EMA, and the method to get this value is quicker and easier to get.

The biggest difference between the SMA and EMA, is the EMA is more sensitive to the price changes than SMA.

This measure was used to identify if the index is in an uptrend, downtrend or in a flat market.

Figure 3- SMA(25) >SMA(200) - uptrend

As you will understand, this measure create some false-positives, because to identify the trend we need to compare SMA with different periods, for example, if the SMA(10) > SMA(200), so it is an uptrend, figure 1, if SMA(10) < SMA(200), so it is an downtrend, figure 4.

To calculate this indicator we need to calculate the average close price of the last n-periods .

Since I am using the last 200 and 10 periods, the first periods have a zero value associated to this measure. This problem create a scalability problem, because every time

we increase the number of periods or increase the timeframe, instead of using daily data we could use annual

Figure 4- SMA(25) < SMA(200) - downtrend

data, this means that if we want to create a SMA(200), with a annual timeframe, we need to compare the last 200 years, and of course we don’t have data available from the last 200 years.

VISUALISATION

Overall description

I choose to develop a very simple and clean user interface, figure 5, as presented before, some of interfaces are completely loaded of icons and buttons, and becomes very hard for a new user to analyse and understand the information.

In this solution, I implemented three different ways to the user interact with the information:

1. Index selection;
2. Timeframe selection, daily, weekly, monthly or annually data, figure 6;
3. Filtering by date, figure 7.

The index selection, was implemented using one or two selectors, depending on what question/task do you want to see, by default the selected index is the German index, DAX.

The date filter, was implemented by using to inputs, one for a start date and other for end date, by default, the data presented is since 2007 until today.

The timeframe is selector is a set of four buttons, and when the button is clicked, it becomes active. The default timeframe is daily.

The data is encoded into a line(s) or into a scatterplot.

Figure 5- Interface overview

Rationale

In my perspective, having a simple and easy to use solution it is always better than I hard an complex one, and we try to create a tool or a solution for one hardest and complex jobs in the world, we should develop easy to use solutions.

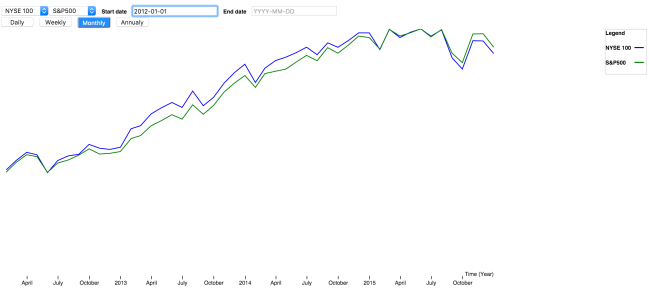

Figure 6- Monthly timeframe selected

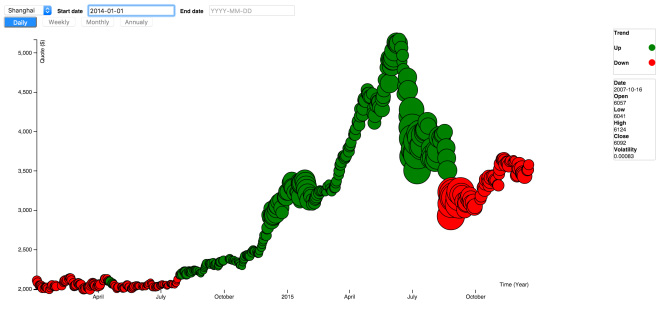

Figure 7- Scatterplot filtered by date, with daily data

Presenting the desired information to a user in a clean way is pure gold, getting clean information within seconds is giving time back to our users.

As always there are always a lot of things that we can improve, for example, on question 3 instead of having just red and green, I should implemented a gradient of red and green, in function of the volatility.

Since the first sketches until get the finished product, a lot of changes occur, mainly on the interaction with the data. In the beginning I thought just about static information, but with the development of the project, I had the need of knowing the data for each day, so I need to implement a legend to see the information.

The initial sketches was crucial to get the end product, because after designing the first sketches I looked for better solutions on the web, and I get to this product, figure 8 and 9.

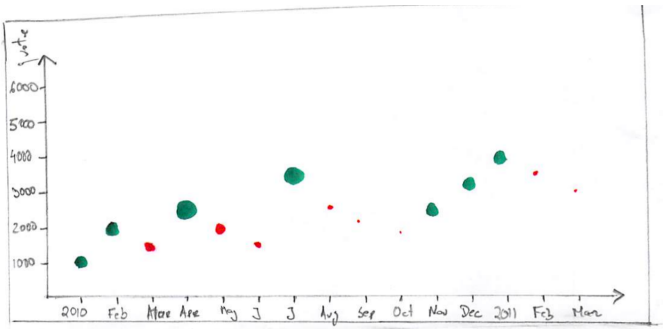

Figure 8- First sketch of third task

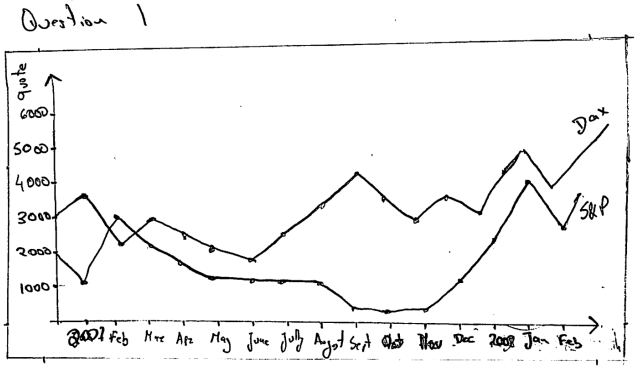

Figure 9- First sketch of first task

Potential

In this section I will present you some of use cases to applied in this project.

Before present the use cases, I need to give you a better insight about the visualisation. Many of the conclusions that you can make from this work could look trivial, but sometimes there are people that can't get the trivial things, that is way I focus my work on new and intermediate users.

1. Task 1 - Observe the correlation between DAX and S&P500, using weekly data between the 2008 and 2012.
    - Select the index DAX on one of the selectors;
    - Select the index S&P500 in the other selector
    - Introduce the start date 2008-01-01;
    - Introduce the end date 2012-12-31;
    - Click on the weekly button
    - Conclusion: we can observe that this two indexes are completely correlated, when one goes up the other goes up. (Figure 10)

Figure 10 -Result of task 1

2. Task 2 - Compare the behaviour of the index VIX and S&P500 between 2007 and 2011, using daily data.
    - Select the index S&P500
    - Introduce the start date 2007-01-01;
    - Introduce the end date 2011-12-31;
    - Conclusion: we can conclude that this two indexes move in opposite directions. (Figure 11)

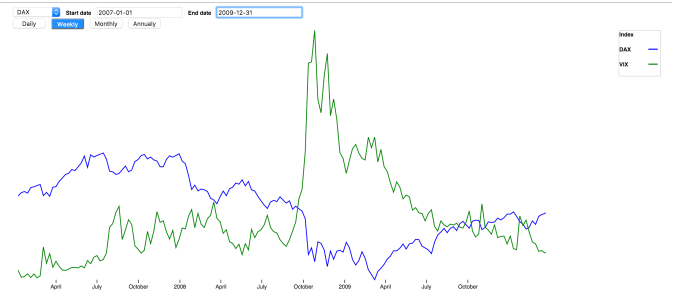

Figure 11 -Result of task 2

3. Task 2 - Compare the behaviour of the index VIX and the chinese index, using monthly data.
    - Select the index Shanghai
    - Select monthly.

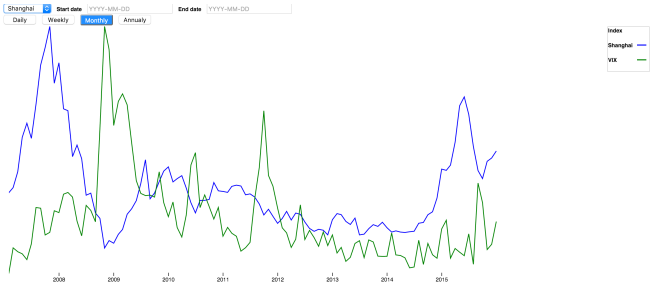

Figure 12 -Result of task 2

- Conclusion: we can conclude that the VIX and Shanghai index are more less correlated, so this index has a high volatility. (Figure 12)

4. Task 3 - Observe the volatility vs the trend of NYSE, between 2007 and 2010
    - Select the index NYSE
    - Introduce the start date 2007-01-01;
    - Introduce the end date 2010-01-01;
    - Conclusion: we can conclude the volatility is higher on the downtrends, due to the fear of the investors. (Figure 13)

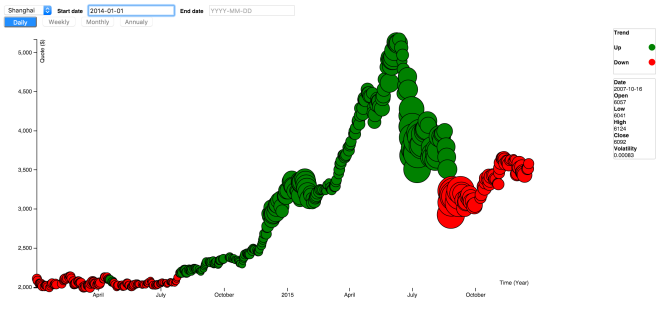

Figure 13 -Result of task 3

5. Task 5 - Observer the behaviour of the index VIX
    - Select the index VIX
    - Conclusion: we can conclude the VIX quote increases on historical moments, which means that the fear of something goes wrong increases.

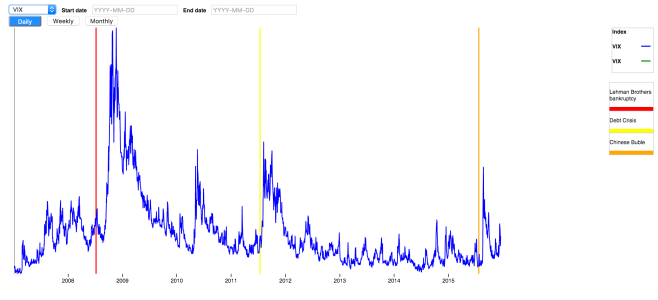

Figure 14 -Result of task 5

IMPLEMENTATION DETAILS

Using a powerful tool such as D3.js, sometimes could be dangerous, because the learning curve it is very hard.

In the beginning it was quite difficult to understand how to get things done, even by the end of the project for me it is

difficult to fully understand how to implement certain things.

Managing data wasn't a challenge for me, I am quite use to develop and create my own algorithms. As I presented in the data section I needed to implement two different algorithms to derive data, Volatility and Moving Average. I also implemented other algorithm to decide if the index is moving in an uptrend or in a downtrend.

The hardest part during this project was implementing the views, it took me too much time to understand how D3, encode data into a SVG element.

D3 also offered a way to create a legend for each visualisation, but in my understanding this is not as adaptable as I desired, so I implement a legend in HTML, to give to the user the information, about the visualisation, such as, what represents a color or what is the data in certain day.

Another problem there I faced during the implementation was creating a zoom in/out of the data, this was quite hard to implement mostly because I didn't have time to implement and investigate more about D3.

Implementing the HTML and managing the Javascript events, it was quite simple, so I choose to implement more things using this two technologies than using D3.js.

CONCLUSION

This project it was crucial to my education, because as I said before, I am constantly researching and studying about financial markets, and this project was the perfect match between what I have been doing and with this course.

If I had more one month and 3000€, I give up of my current job for one month just to work on this. Nevertheless I would like to improve the interaction between the user and the data, and I will improve the visual encoding.

Also I would improve the interaction between task and the layout of the page. I know that is quite poor, but hopefully I could improve it if I had time.

This was one of the hardest projects that I have done, due to the complexity of the chosen subject and for the fact that I am a student worker, for that reason I couldn't applied my self as I desired. Doing the same work as 3 people and working at the same time has hard.

I would like to thank Professor Sandra Gama, for all of the time spent and her precious help.