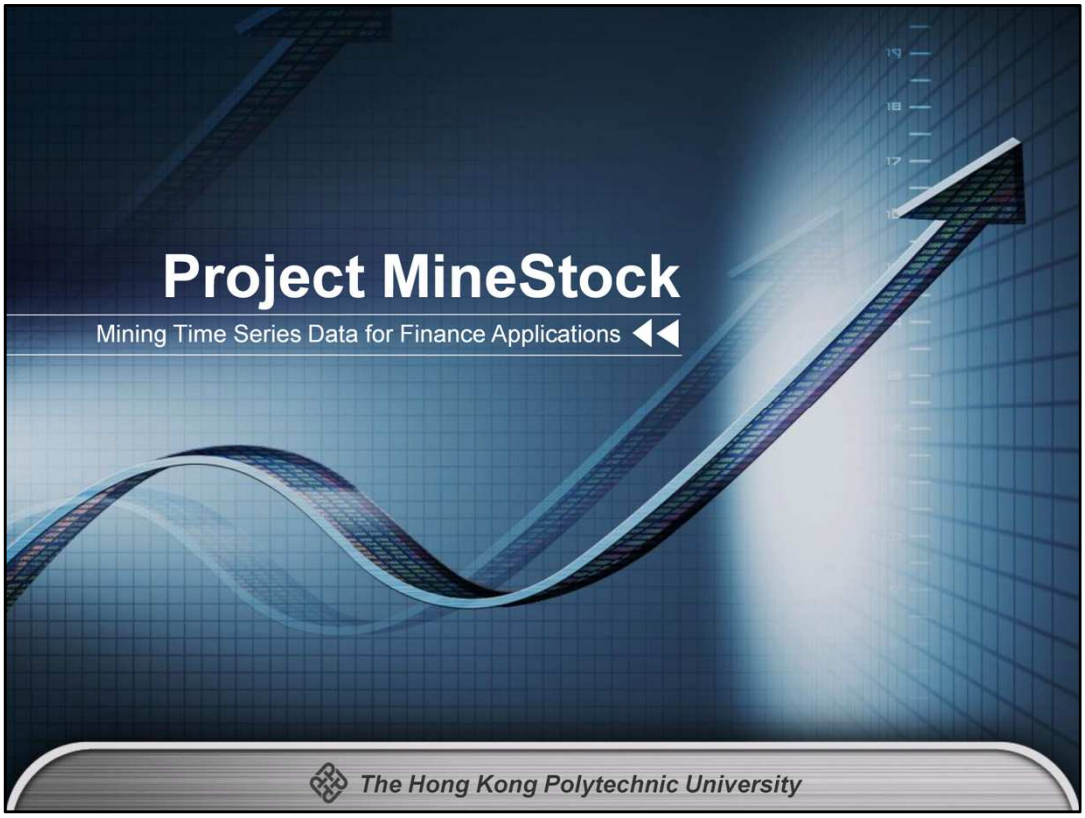


Project MineStock
Mining Time Series Data for Finance Applications

Presented by TONG Man Kin
Supervised by Dr. CHAN Chun Chung Keith,
Co-examined by Dr. ZHANG Lei and Dr. LO Chi Lik Eric

BSc (Hons.) Major in Computing (61031-ASC)
Department of Computing, The Hong Kong Polytechnic University

May 3, 2011

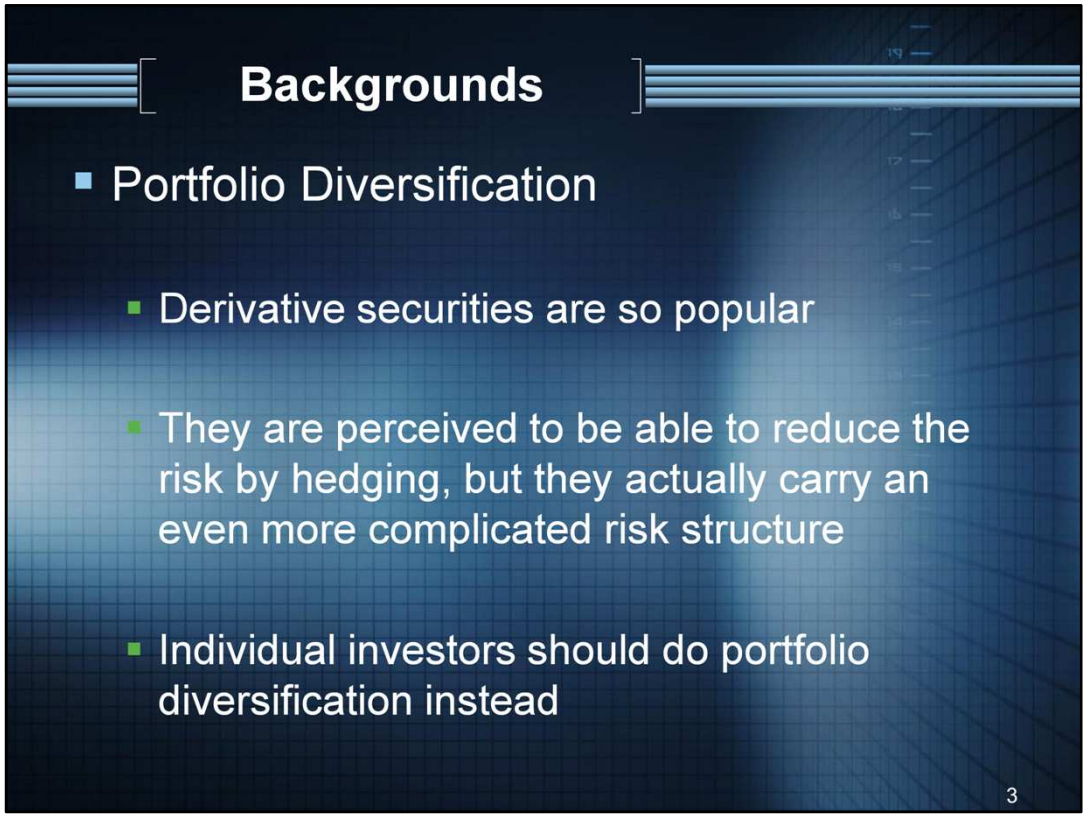
<presentation script>

In this half an hour, first of all, we will be having this PowerPoint presentation, after the presentation, we will see a demo of the system, the MineStock Workbench, and finally, we will have a small Q&A session.

The presentation's flow would be like this, we will talk about the backgrounds, objectives, methodologies and evaluation.

I will focus heavily on the methodologies sections because of the time constraint.

You can always refer to my report, or contact me, for more details on the whole project.



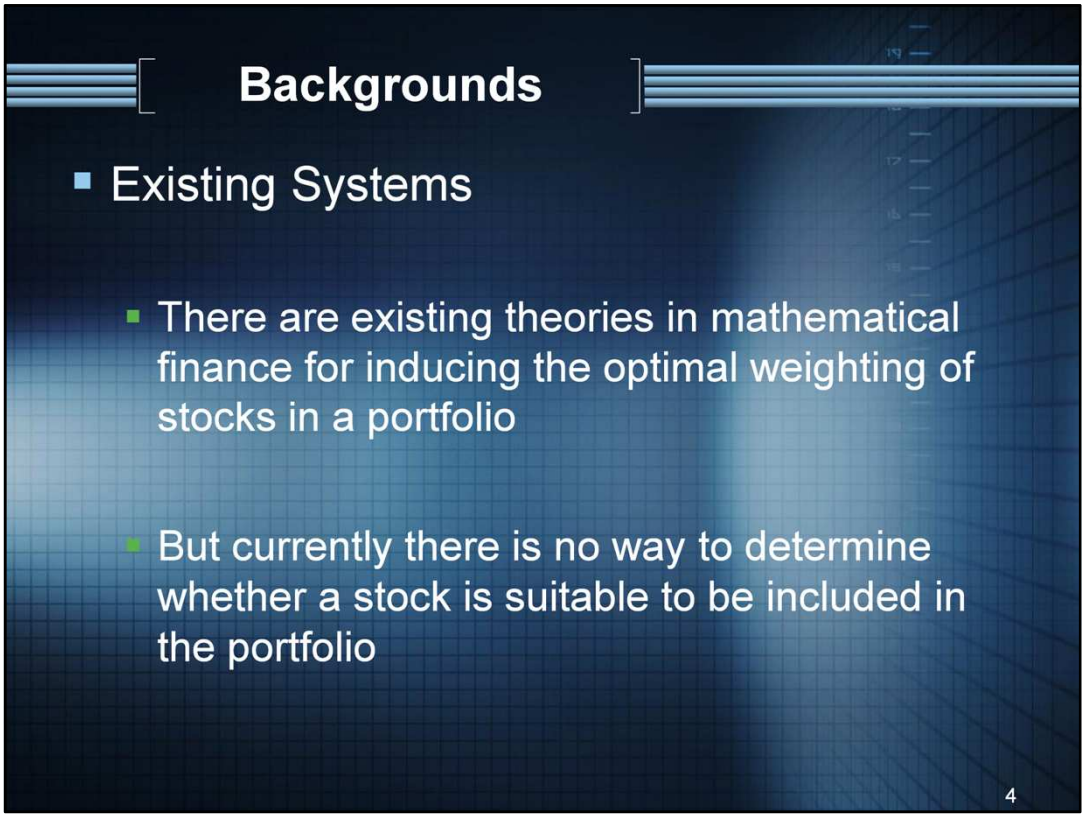
<presentation script>

Derivatives becomes so popular after they are invented.

And they are perceived to be able to reduced the investment risk by hedging.

However I don't think so, in the financial tsunami in 2008 we learnt that they actually carry an even more complicated risk structure.

I believe individual investors should do portfolio diversification instead, in order to reduce their investment risk.



<presentation script>

Here is a review of existing systems.

I found that there are existing theories in mathematical finance for inducing the optimal weighting of stocks in a portfolio. So after you pick the stocks, you can find out the weighting.

But currently there is no way to determine whether a stock is suitable, or not suitable to be included in the portfolio.

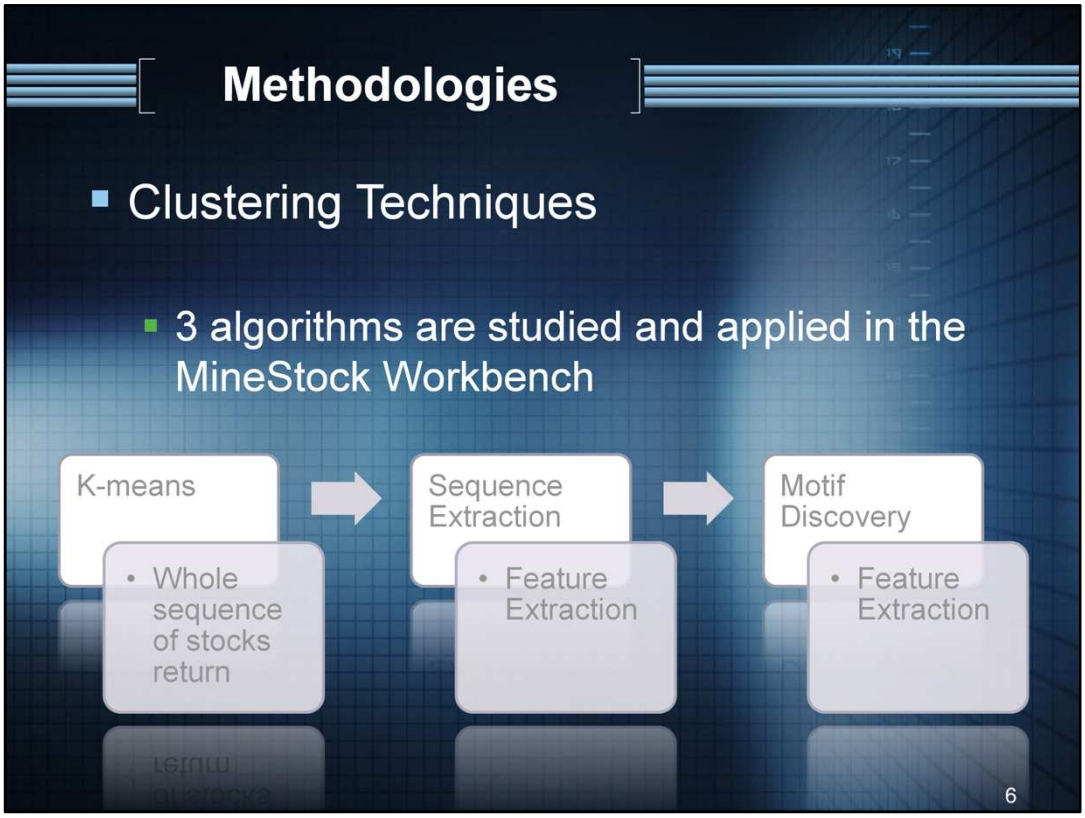
<presentation script>

Therefore this is what I want to do in this project.

I believe that data mining, or more specifically, the clustering techniques are able to solve this problem. For example, if I have divided all the available stocks into clusters, than you know stocks from same cluster are similar and should not be added into same portfolio. You will pick the stocks from different clusters as they not similar and thus, able to generate a greater diversification result.

The 1st objective of the project is to develop the clustering techniques on stocks. 2nd objective is to implement a software tool, MineStock Workbench.

This tool will allow investors to use the clustering techniques developed, to assist them to pick stocks more easily, and also provide the mathematical finance calculations that I just said, so after investors pick their stocks, they can find out the optimal asset allocation.



<presentation script>

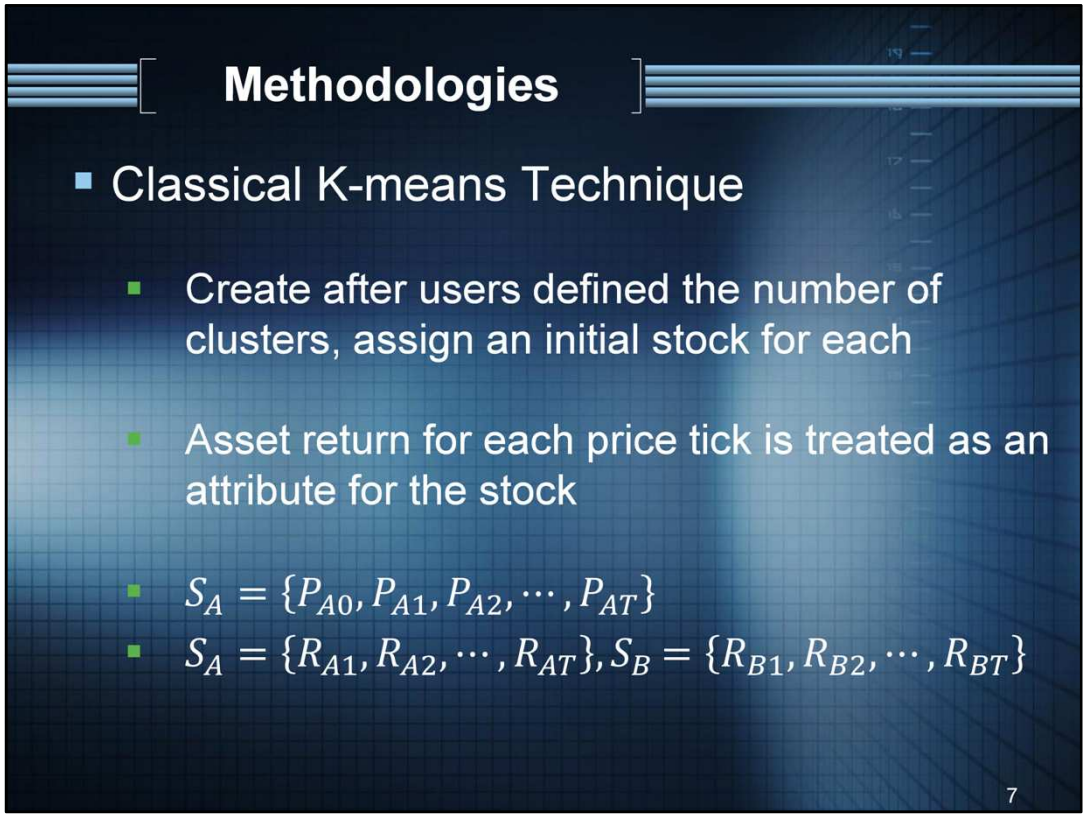
Then we are going to talk about the clustering techniques.

In this project and the MineStock Workbench, 3 algorithms are applied and studied. They are the classical k-means, identical sequence extraction, and similar motif discovery. The latter 2 are originally used in other domains. I have put them here with some modifications.

K-means is the simplest way to perform clustering, which compares whole sequence of stock prices at once. So we have 2 stocks fluctuating like this, and k-means will be used to compare the whole period.

However, in order to deal with the complexity of time series data, the latter 2 applied the concept of feature extraction.

I will discuss about these 3 algorithms, one by one in the following slides.



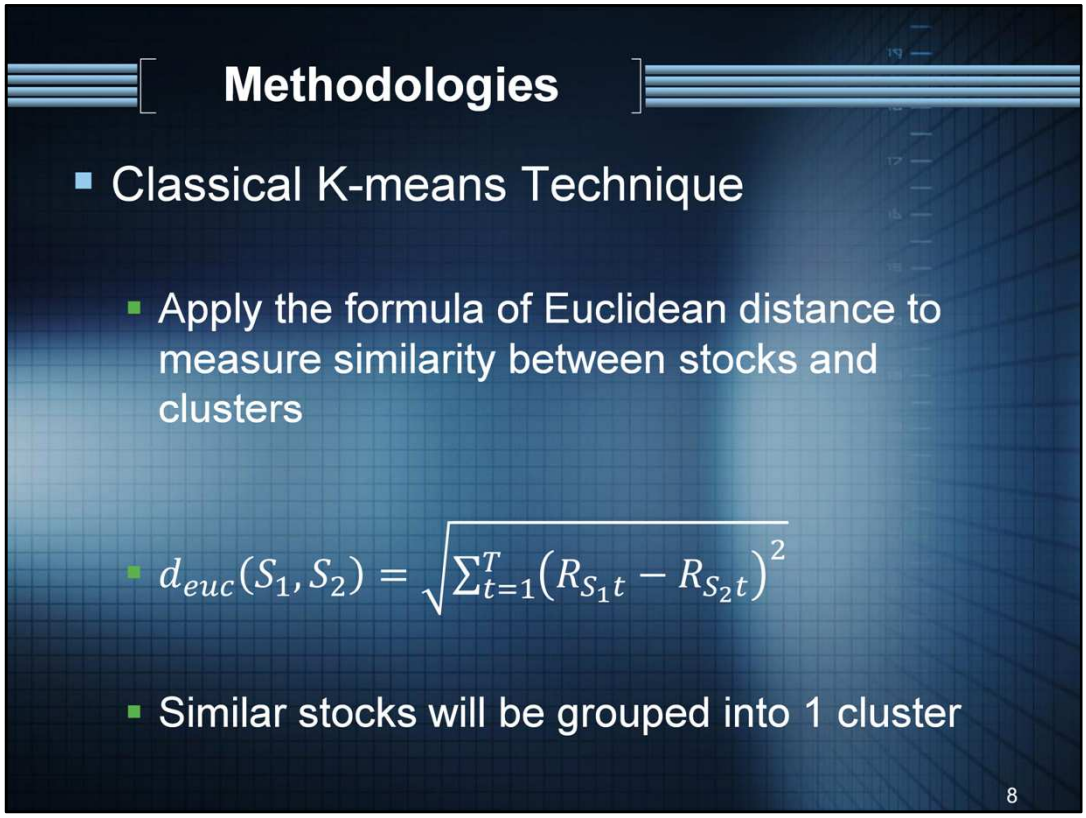
<presentation script>

First, the classical k-means, I am sure that you know better than I do, so this is just a recap.

First of all, after users defined the number of clusters, we then create them. And then, for each cluster, randomly assign a stock into it.

For each stock, we will have many days, or price ticks, downloaded. We compute the asset return for each tick, and each return is treated as an attribute for the stocks.

Like the sets shown below, we will have R(A1) compared with R(B1), R(A2) compared R(B2), and so on and so on.

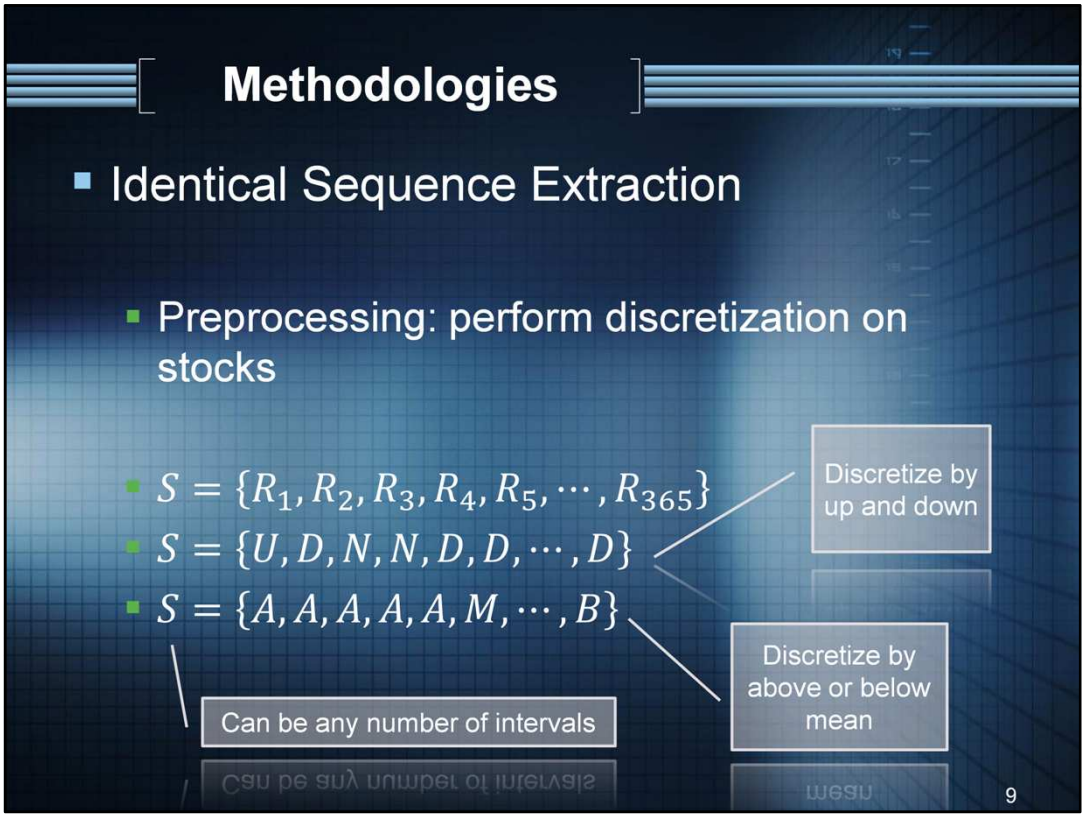


<presentation script>

So, how to compare? We will use the formula of Euclidean distance to compare the stocks and clusters. As you have see here.

Smaller distance implies a more similar pair in terms of their fluctuation patterns.

Stock will be assigned to the cluster which they are have the smaller distance. Therefore, similar stocks will be grouped into a single cluster.



<presentation script>

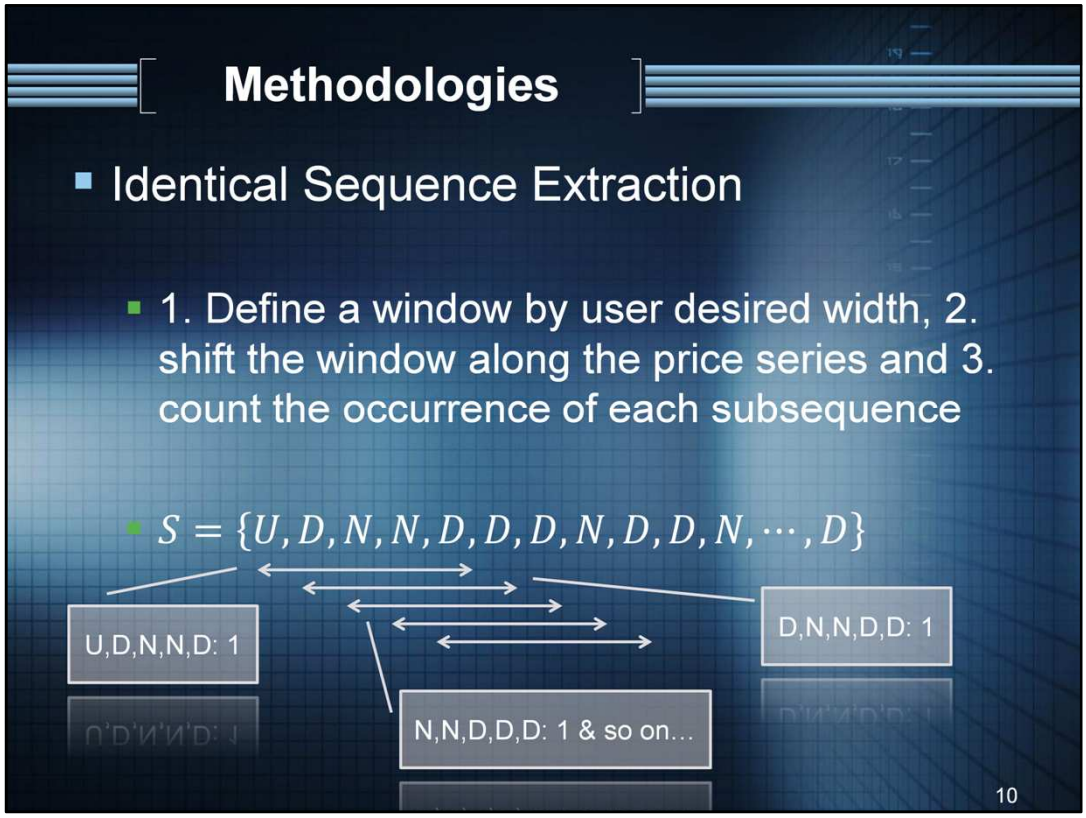
Identical Sequence Extraction.

First of all, unlike k-means, we have to perform a discretization process on stocks instead of comparing the numbers directly.

In MineStock Wenchbench, the discretization process can be done by two ways.

One of them is discretize by price changes, or ups and downs of price, for example, up-down-nochange-nochange-down-down. Another one the price level, say, in the example here, A is above mean, M is mean, B is below mean.

In these examples we performed discretization with 3 intervals, but actually it can be any numbers. If you choose 5 as interval, then you will have higher up, lower up, no change, lower down, higher down. Similar cases applied to both discretization methods.

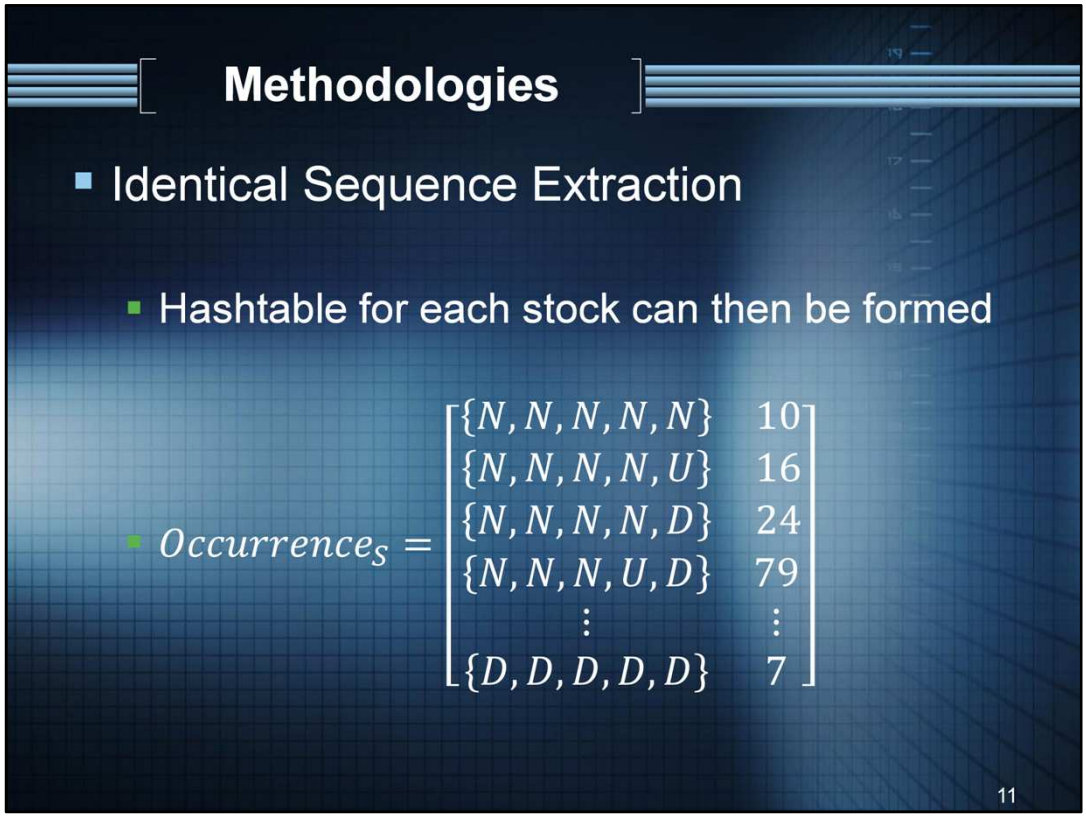


<presentation script>

After we have done the discretization preprocessing, we can extract the subsequence in the time series out.

First we define a window by user desired width, say 5, in this example, then we shift this window along the price series and count the occurrence of each subsequence, or pattern.

So you can see, in the first window we got U-D-N-N-D, the second we got D-N-N-D-D, the third is N-N-D-D-D and so on. This 1, is the count.

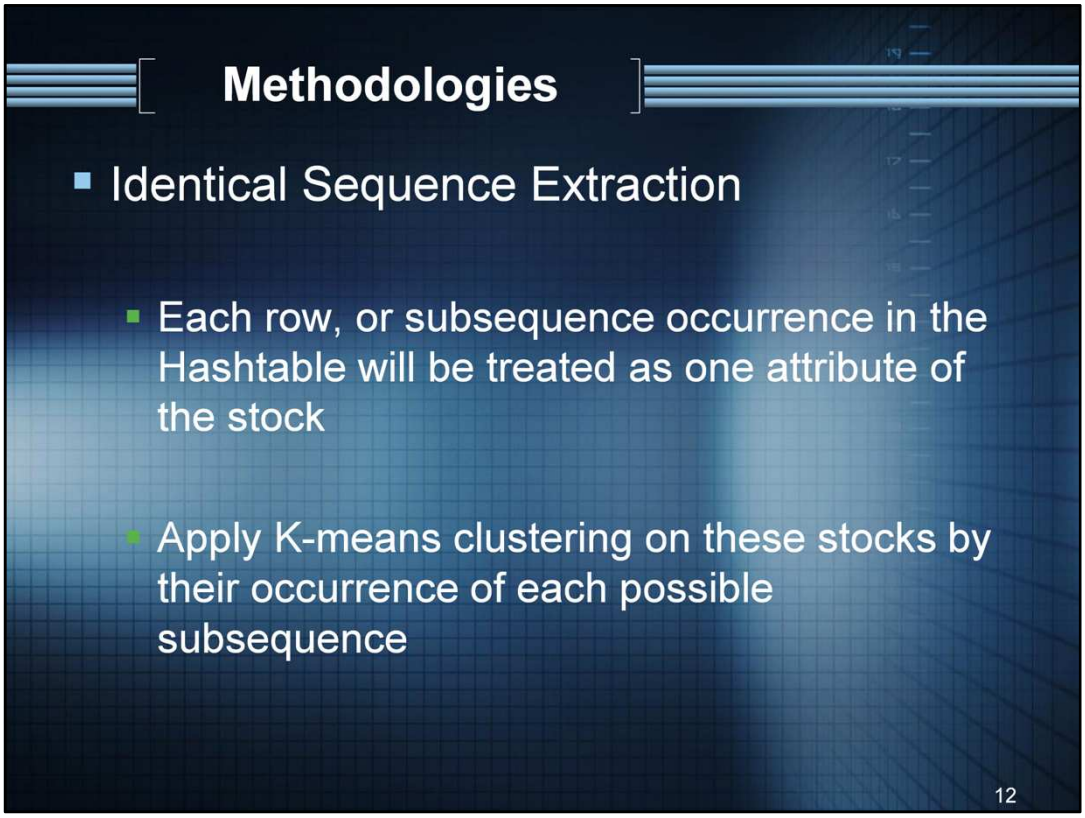


<presentation script>

After we finish shifting the window along the time series, we can form the following hashtable.

Left side is each unique subsequence, and the right side is their occurrence count.

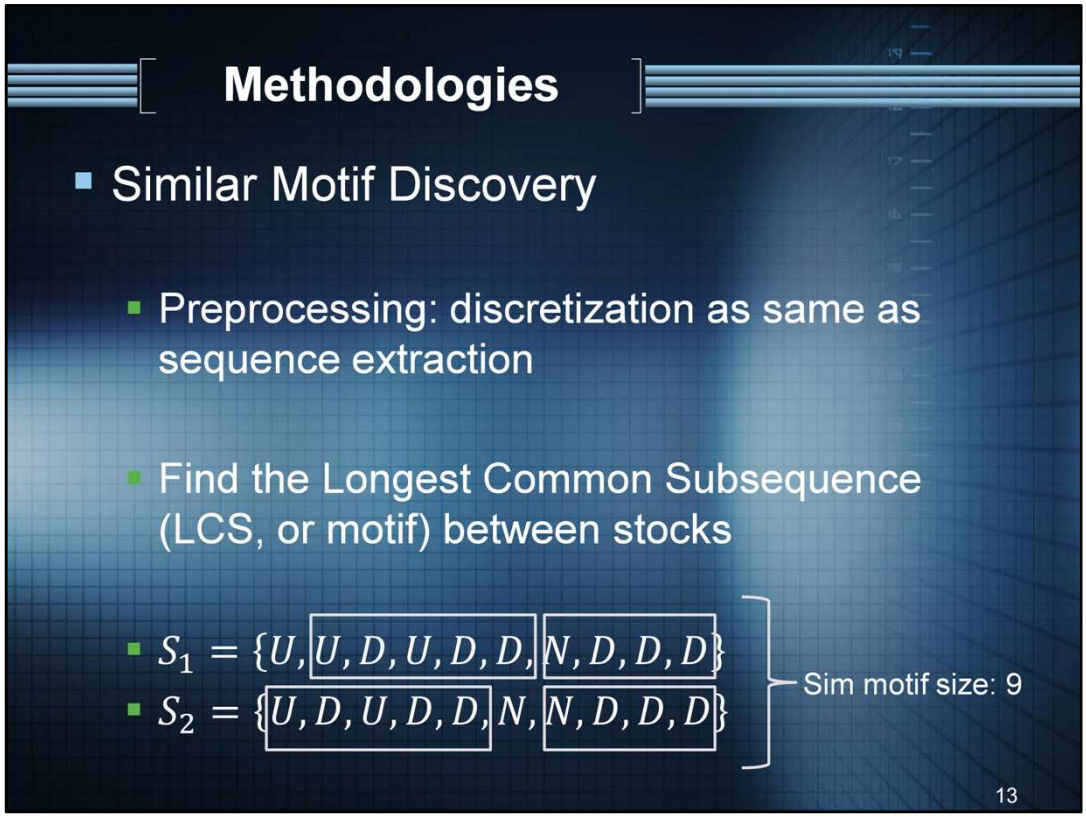
After we found out the occurrence hashtable for each stocks, we can compare them. Because similar stocks should have similar occurrence count for each pattern.



<presentation script>

Therefore, actually, each row, or subsequence occurrence in the table can be treated as one attribute of the stock.

We can apply K-means clustering on these stock. Different from the classical k-means that I just said, this one looks into the occurrence of patterns instead of price changes.

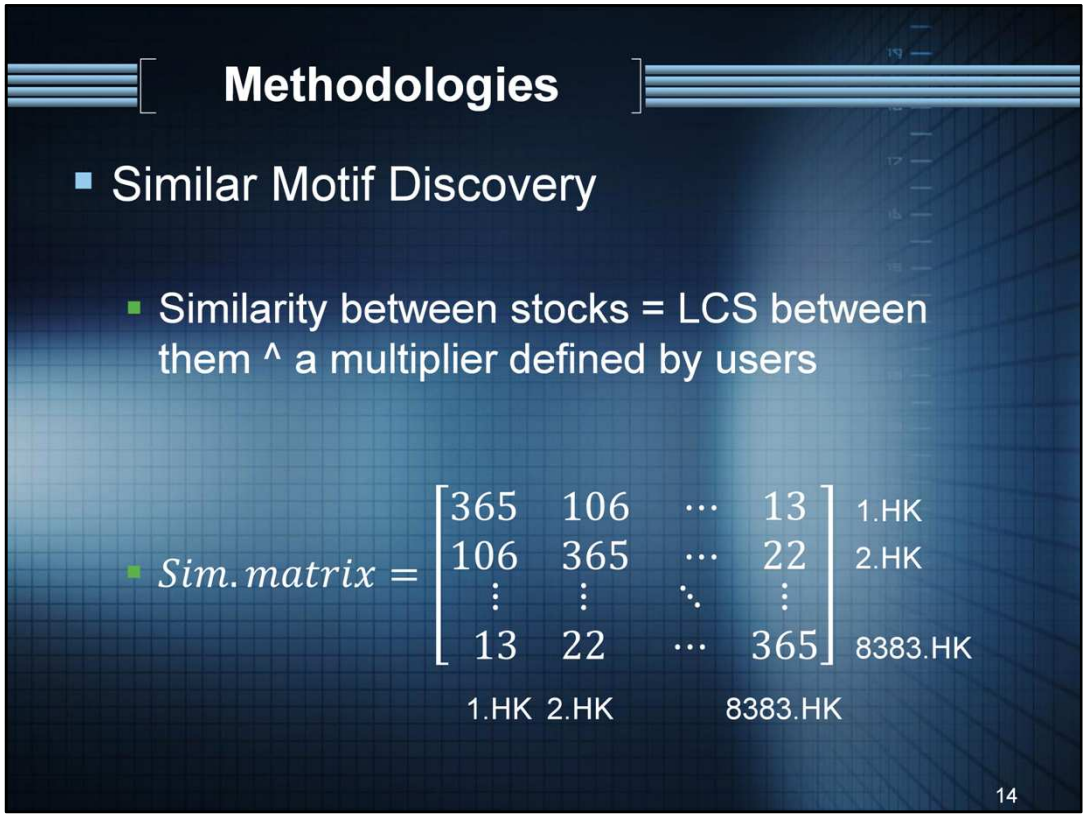


<presentation script>

After talking about the sequence extraction method, now we come to the similar motif method.

First of all, for this algorithm, we will need to perform the same discretization process as in identical sequence extraction method. Then we find the longest common subsequence among the stocks.

In this example, the longest common subsequence is U-D-U-D-D-N-D-D-D, which is having a size of 9.

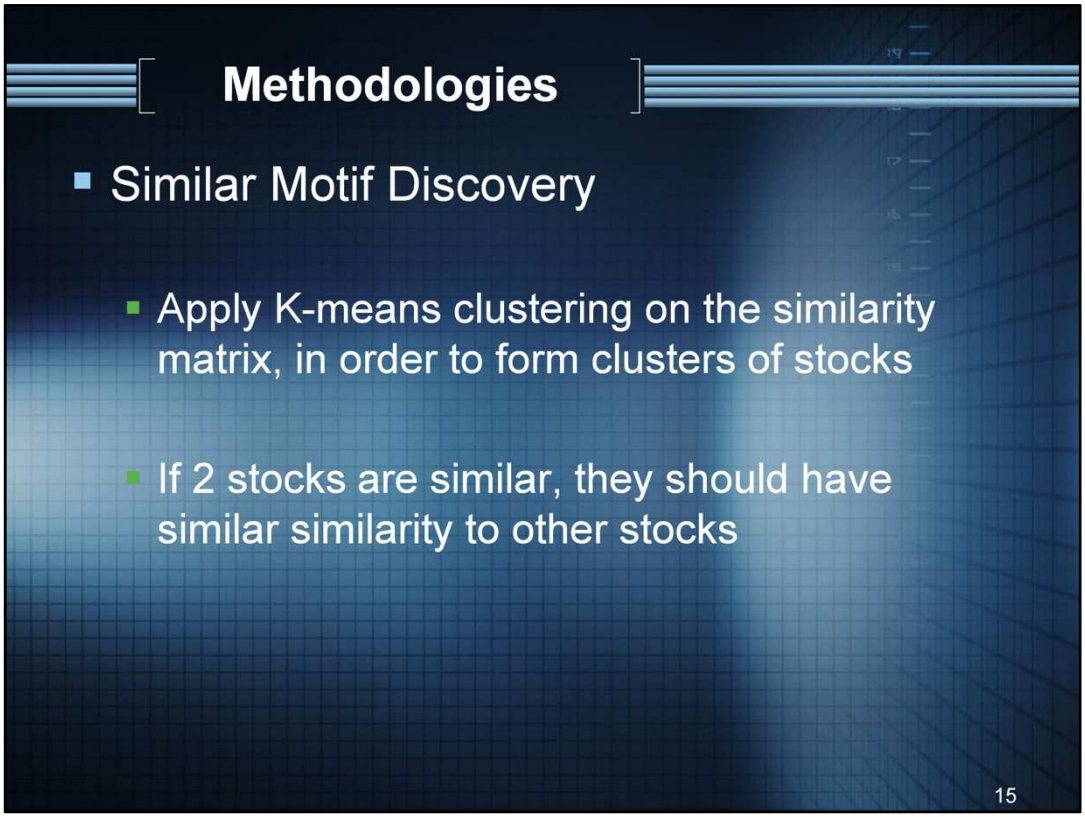


<presentation script>

Then my algorithm will compute the similarity between these 2 stocks by taking a power factor on the size of LCS. This power factor, or multiplier, is defined by users.

The power factor is intended to magnify the result of LCS, so higher LCS, high similarity, and lower LCS, low similarity.

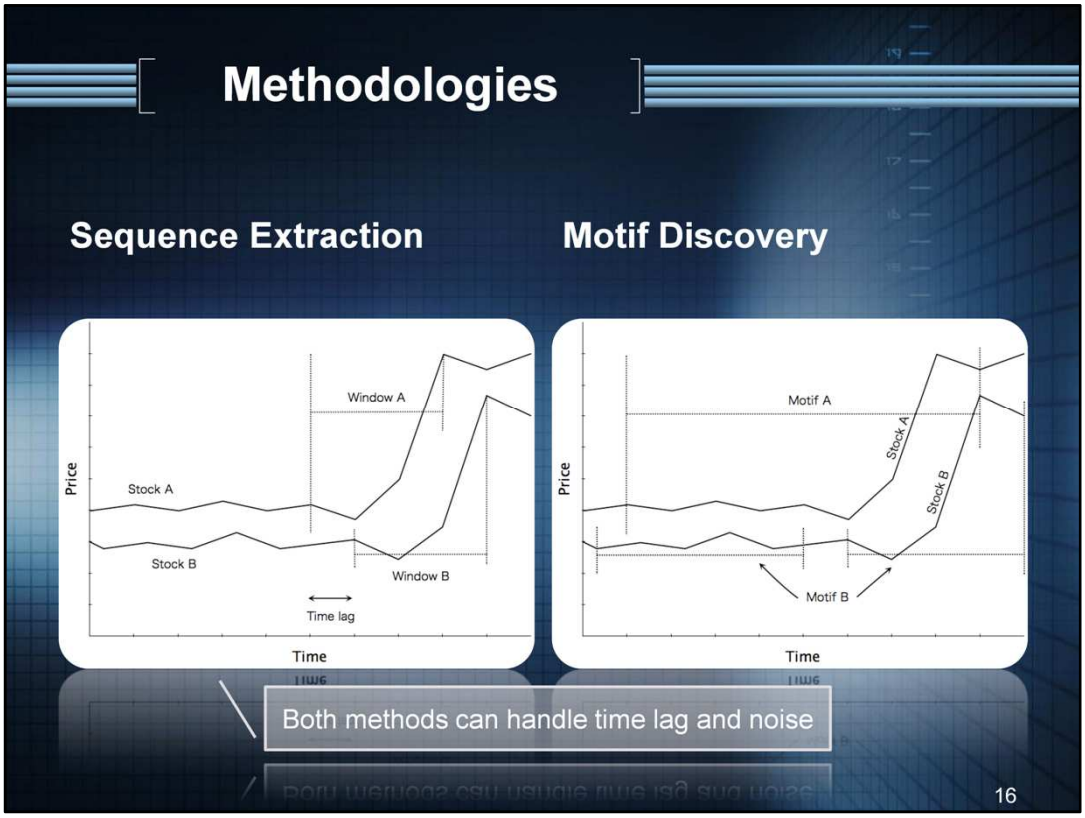
After extracting motifs of all stocks we can obtain the following similarity table.



<presentation script>

After we have got the matrix, we can apply K-means clustering on the similarity matrix, in order to form clusters of stocks.

If 2 stocks are similar, they should have similar similarity to other stocks.

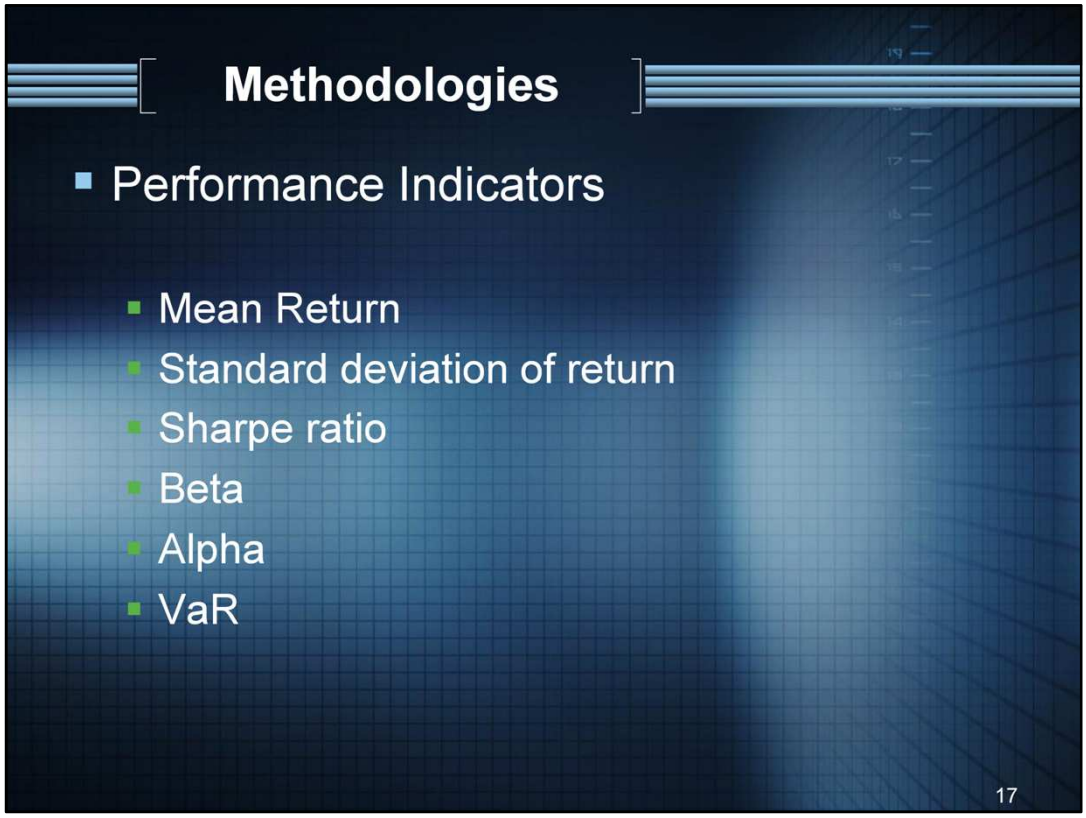


<presentation script>

We have discussed the 2 feature extraction algorithms which used in my system. I think thy are better than the classical k-means because they are able handle time lag and noise in the sequence.

For example, as we can see from both charts, there is a time lag here. And you can see, both ways can identify the similar patterns between 2 stocks without affected by the time lag.

In contrast, classicial k-means just comparing the change of this and this (note: t=0 vs t=0), and then this and this (note: t=1 vs t=1). So it will find out a very great difference at this point (note: the high-fly part), but not considering that they are actually similar.



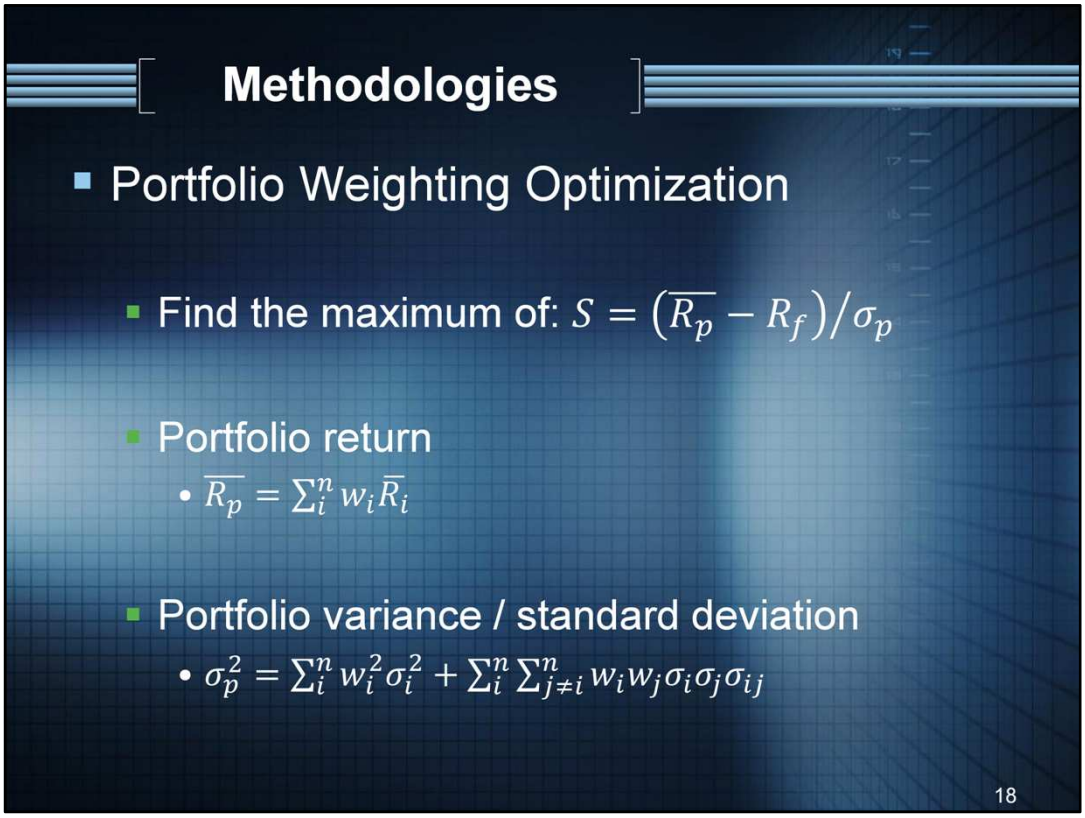
<presentation script>

After talking about the clustering techniques, here is some performance indicators which are provided in my system.

This slide is the basic one, we have average return and standard derivation of return. This one is also known as the total risk of a stock.

We also have the Sharpe ratio, this measures the risk adjusted return of a stock. It is calculated by this formula, expected return of asset minus risk free rate, divided by the total risk of asset.

We use US T-bill as risk free rate in our system.



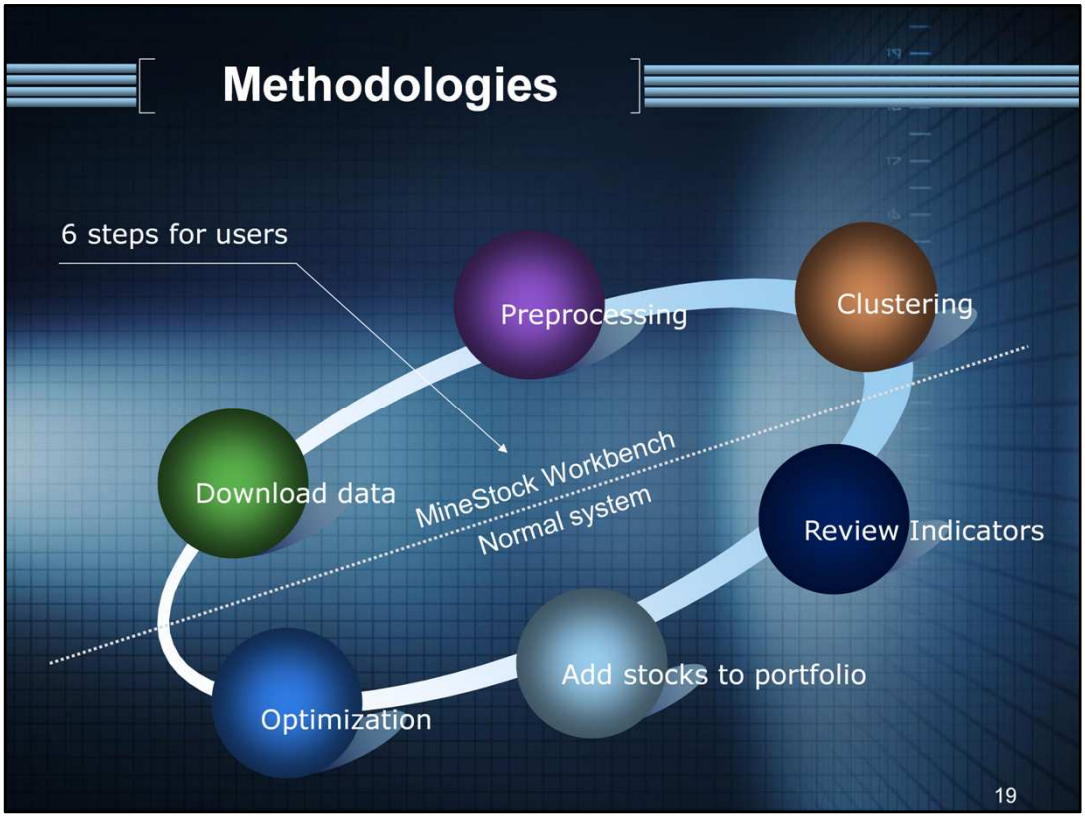
<presentation script>

OK, how to do portfolio optimization? We will look into the Sharpe ratio.

We can calculate the portfolio return and variance by these formula. 'W' here is the weighting of each individual asset.

We try out different 'W' and obtain the Sharpe ratio by this formula, and then we will know how the asset can be allocated better.

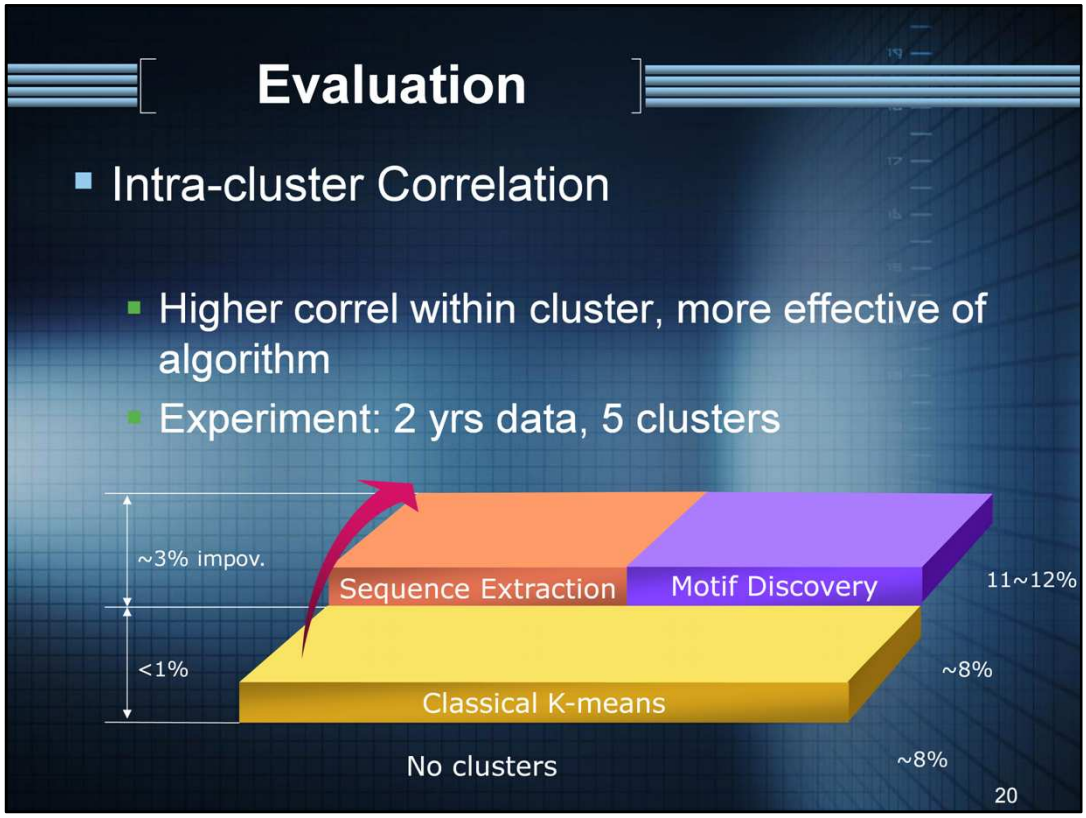
Notice that this sigma i-j is the correlation between 2 stocks. Smaller correlation smaller total risk, we will get back to this in the evaluation section.



<presentation script>

In summary this is the 6 steps for users to use the MineStock Workbench. First of all they will need to download data, then preprocessing, then clustering. Then for the stocks in each clusters, we have different performance indicators for them to review, and then they can pick stocks to form a portfolio, and finally do the weighting optimization.

While the current software tools and finance theories only helping us to do the lower part, MineStock Workbench providing the upper part as well, so it can assist investors in their whole decision making process, including the selection of stocks.



<presentation script>

We know that smaller correlation between stocks will generate a lower total risk, thus a higher risk adjusted return. Therefore, what I was trying to do in evaluating the effectiveness of the clustering algorithms is calculating their average correlation within a cluster.

I found that in my experiment, average correlation of stocks is about 8%. After I used the classical k-means algorithm, average correlation of stocks within cluster is also about 8%, only a very minor improvement.

For sequence extraction and motif discovery algorithm, the correlation goes to 11-12%. So they are better than the k-means.

Also, the correlation becomes much greater when we try to do further clustering on existing clusters. You may refer to my report for more details.

<presentation script>

Here we go to the demonstration session.