Turbulence & Absorption Ratio

* **Introduction:**

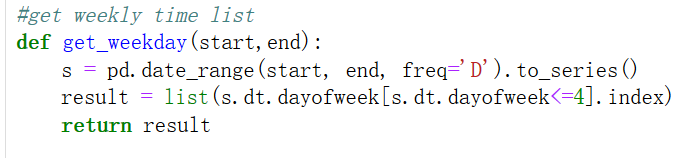
The greatest financial catastrophe in 2008 has been witnessed by all government and market practitioners, thus more and more importance is attached to developing effective tools to monitor and quantify all types of risk. Under this circumstance, many risk measures, simple or complex, micro or macro, have been proposed by scholars, trying to grasp the secret of how our financial market is operating.

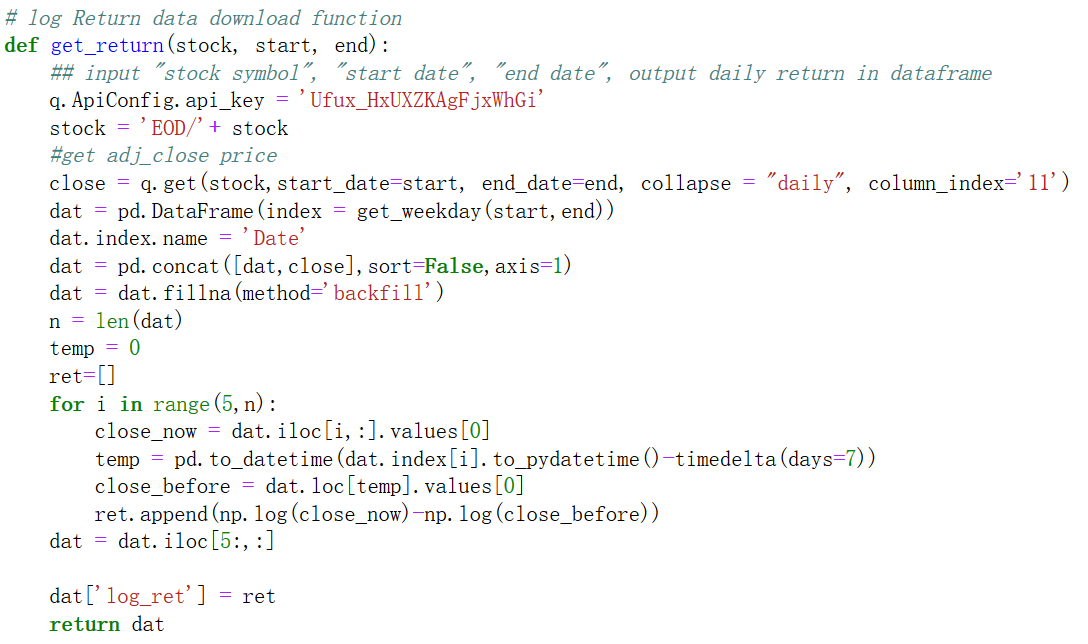
Two of such risk measures are Financial Turbulence and Absorption Ration, both of which do not require disgusting mathematical proof. I have conducted research on the application of these methods in different asset markets and validated the model in detecting extreme phenomenon during the last 20 years with intraday prices.

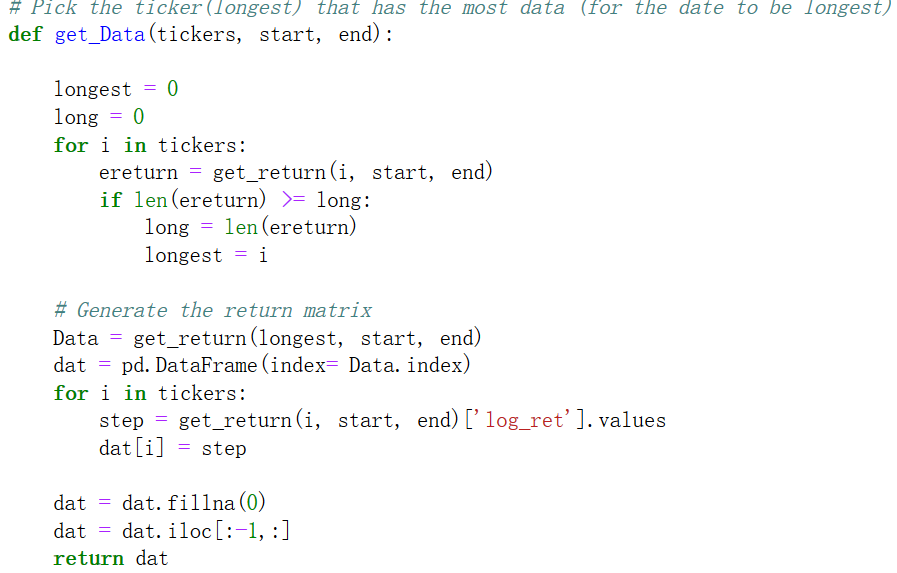
* **Data:**

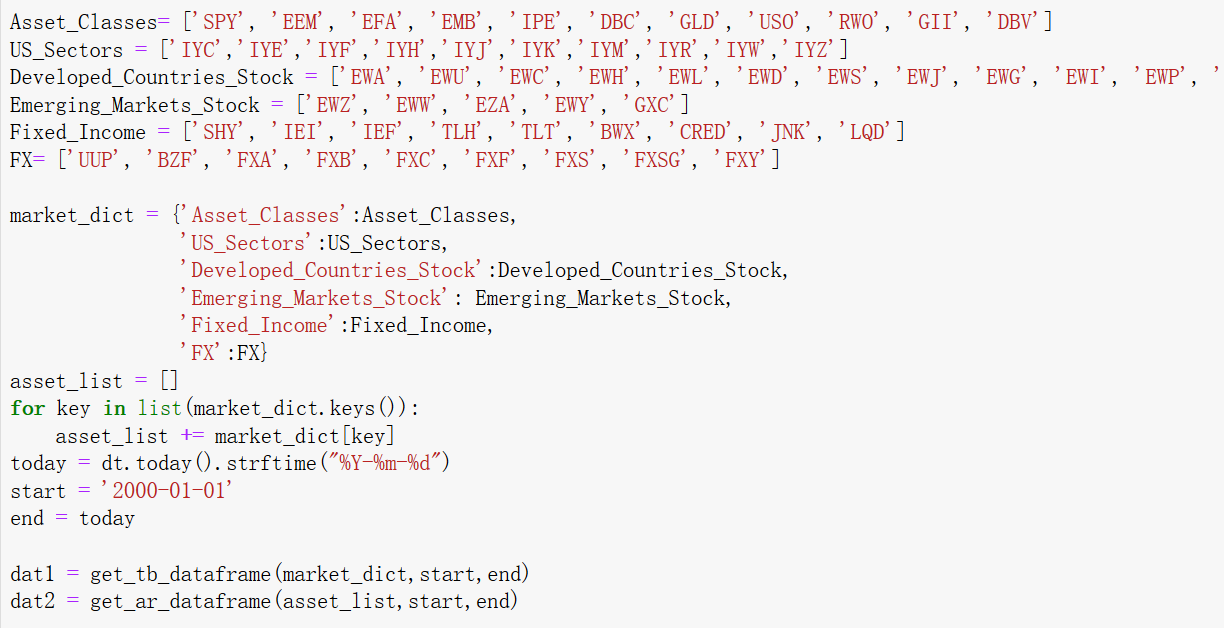
All the data is come from quandl (python package):

We firstly used ETF data just like previous intern(all the code is available in Data\_Require.py:









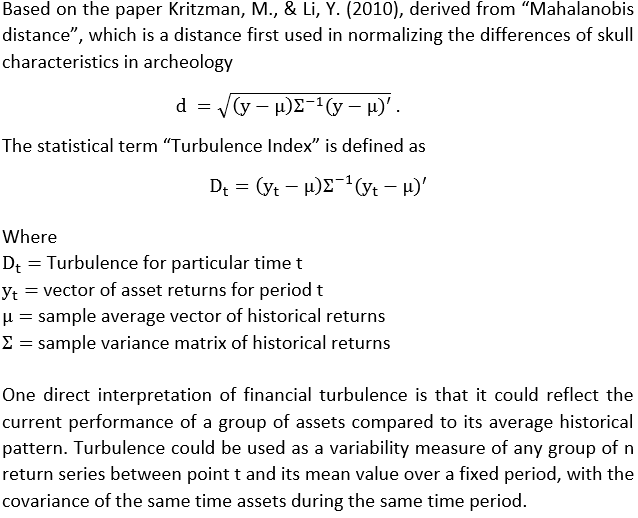
* **Methods:**

The method to calculate Turbulence & Absorption Ratio is totally based on two papers:

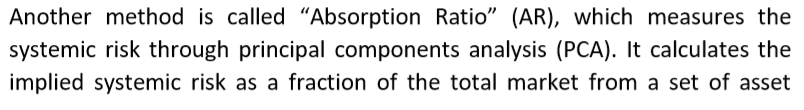
1. Kritzman, M., & Li, Y. (2010). Skulls, financial turbulence, and risk management. Financial Analysts Journal, 66(5), 30-41.
2. Kritzman, M., Li, Y., Page, S., & Rigobon, R. (2011). Principal components as a measure of systemic risk. The Journal of Portfolio Management, 37(4), 112-126. 3

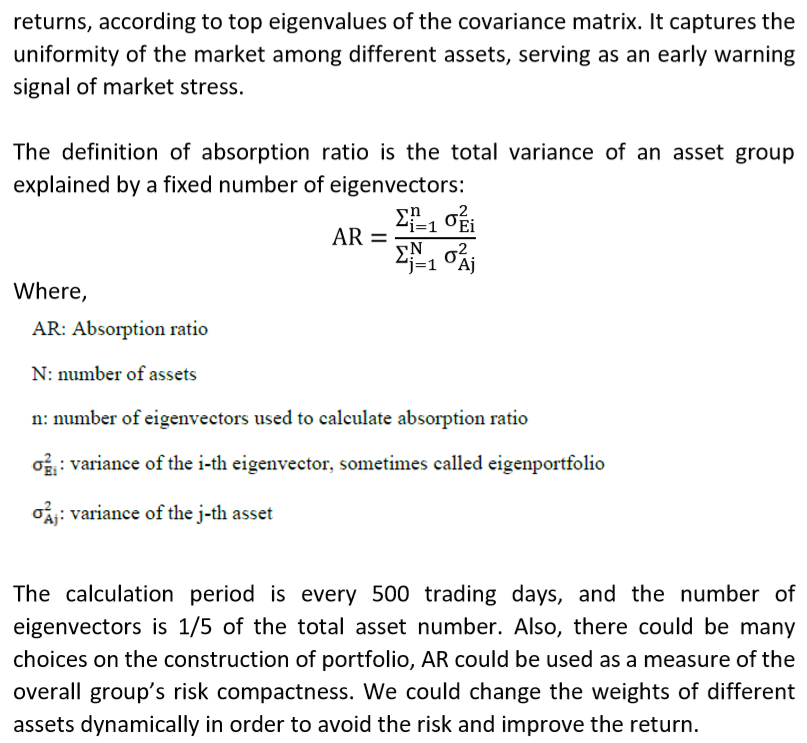
Specific methods:

1. **Turbulence:**



1. **Absorption Ratio**





* **Details:**

1. **Improve Previous Intern’s Work**

Previous intern used daily return and do not use exponential moving average/ exponential decay to calculate Turbulence & Absorption Ratio. To make our work more consistent with papers’ work, we use weekly log return and exponential moving average/ exponential decay.

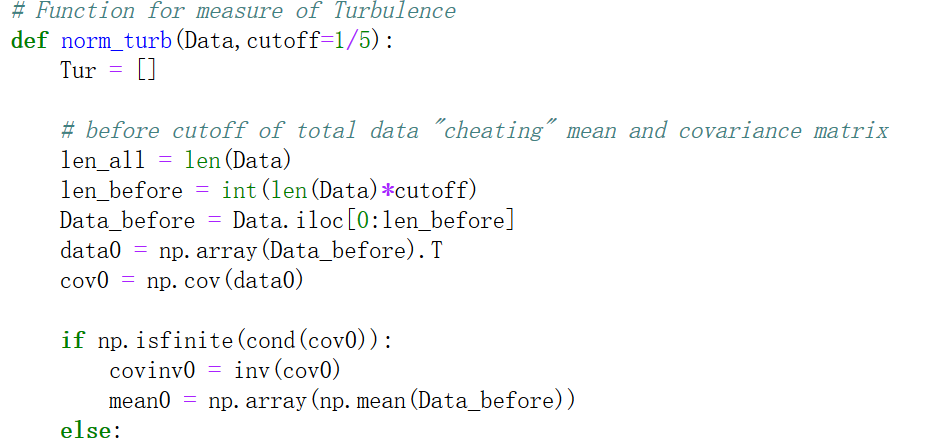
For the missing value of weekly data, we use backfill for these missing values.

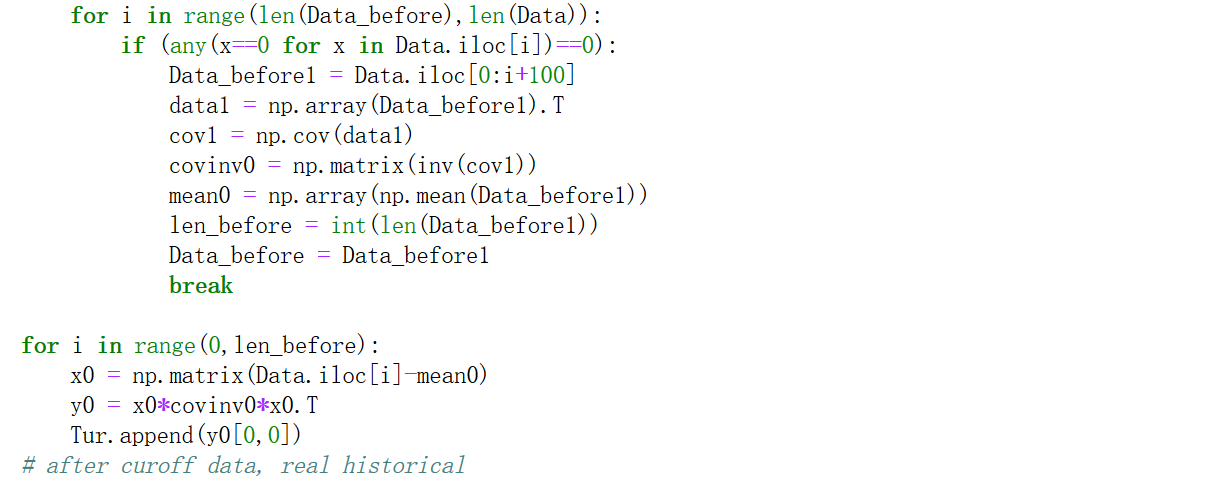
For turbulence Index, because use all the data to calculate previous index values is not make sense, we use cutoff method to handle it:

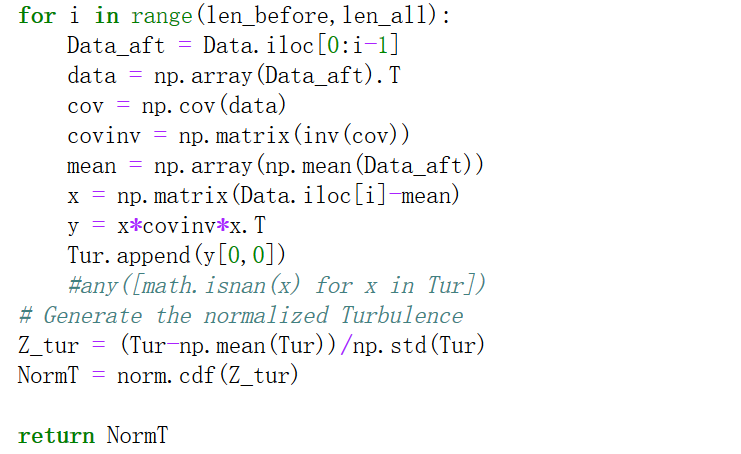
when we got all the data from 2000 to today, we cutoff 1/5(default, you can choose whatever you like) to generate fake index value. Then, we calculate Tb only base on previous data but not all data.

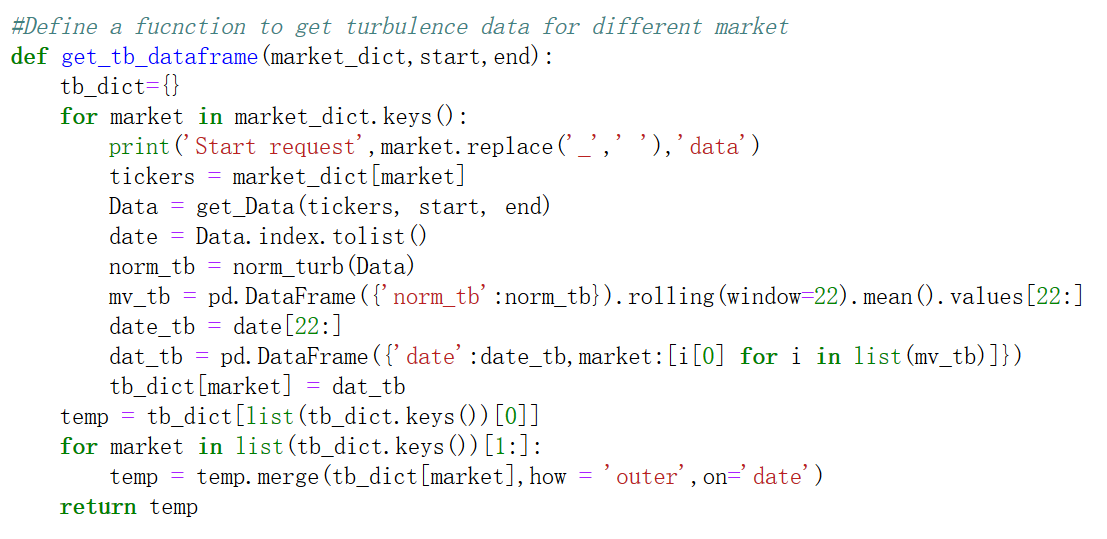
We also write full documents for all the ETF we use, they are in ETFs\_new.xlsx

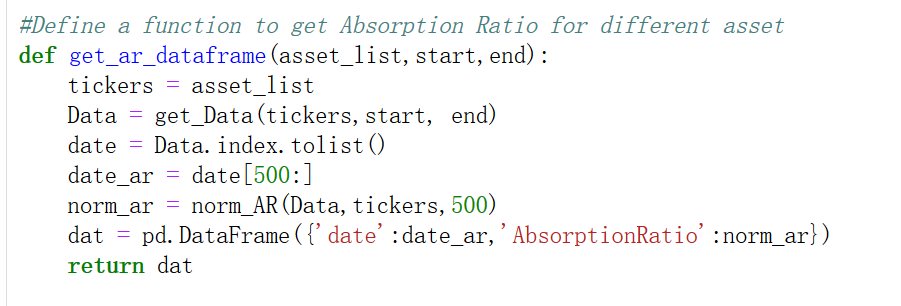
The code to calculate TB and AR:



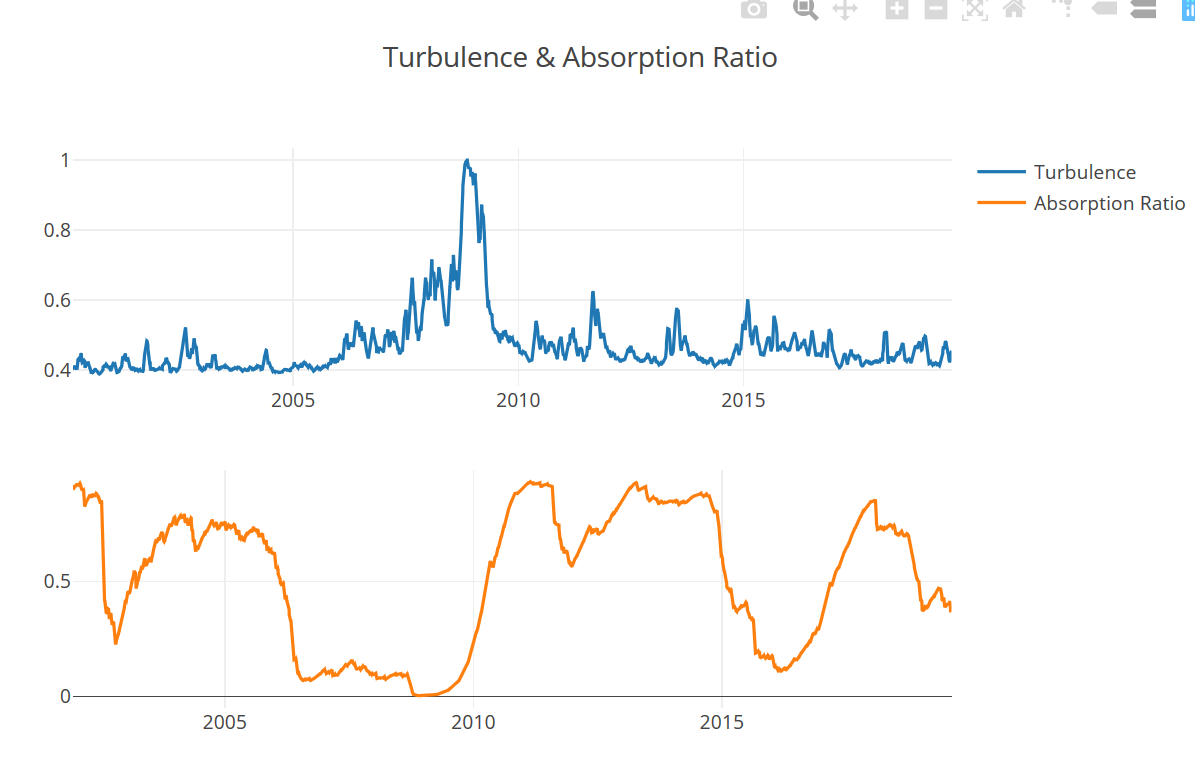






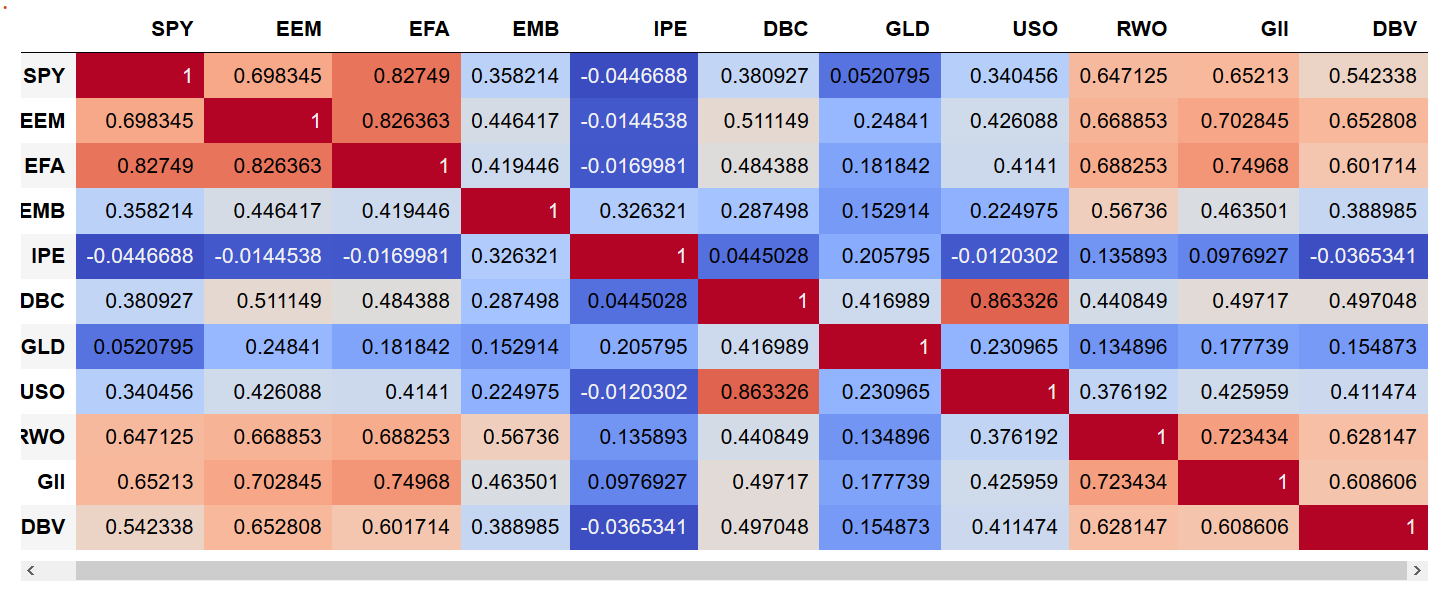


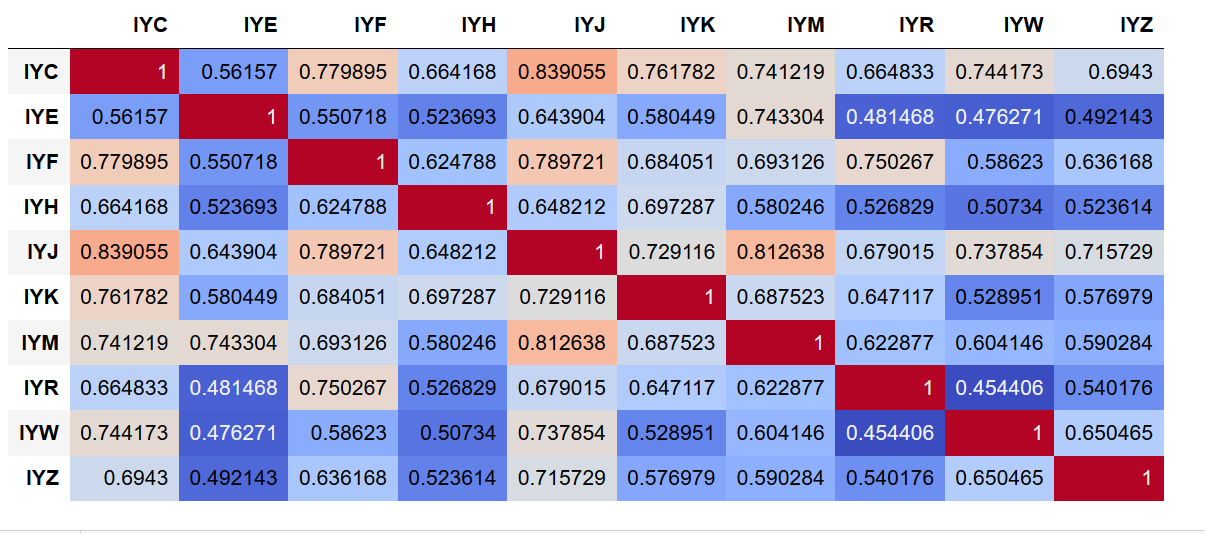
The sample result for Asset\_Classes: ['SPY', 'EEM', 'EFA', 'EMB', 'IPE', 'DBC', 'GLD', 'USO', 'RWO', 'GII', 'DBV']

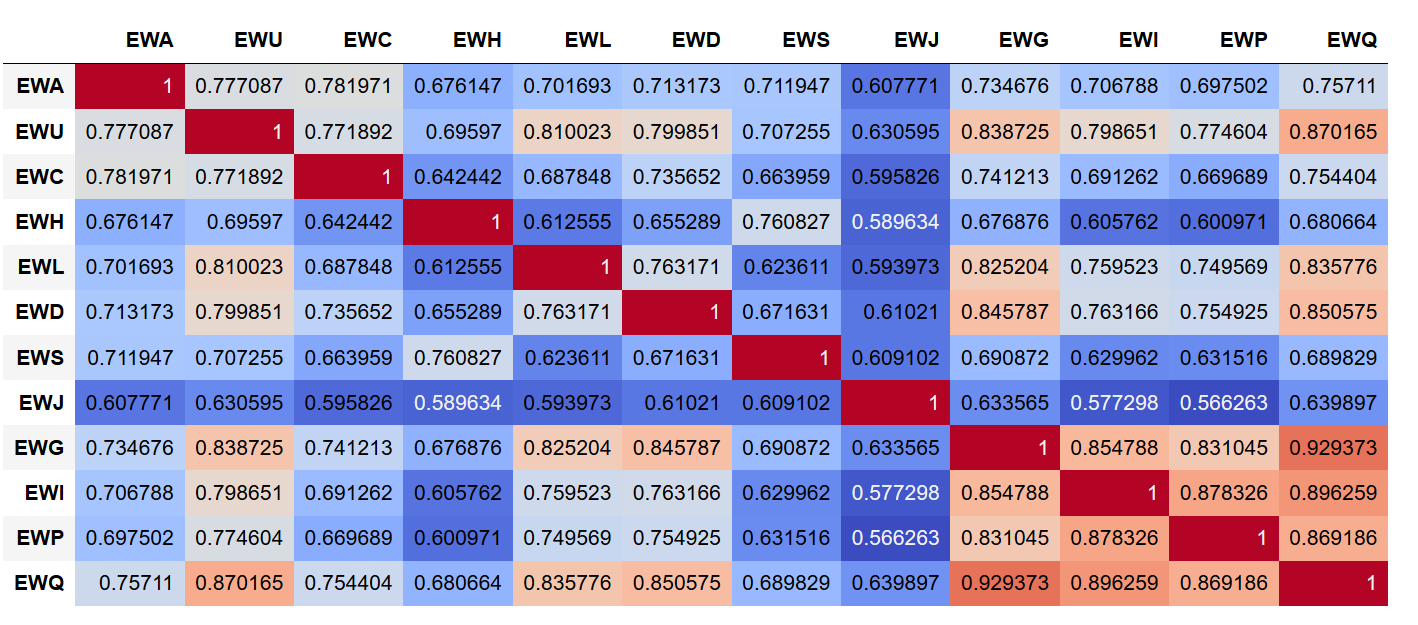


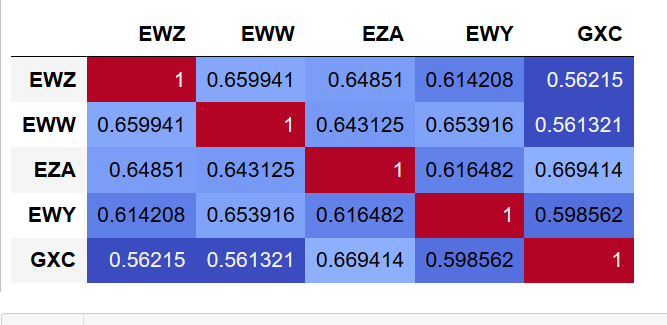
The full results was deployed on public by myself, there is link: <https://whispering-spire-22988.herokuapp.com/>

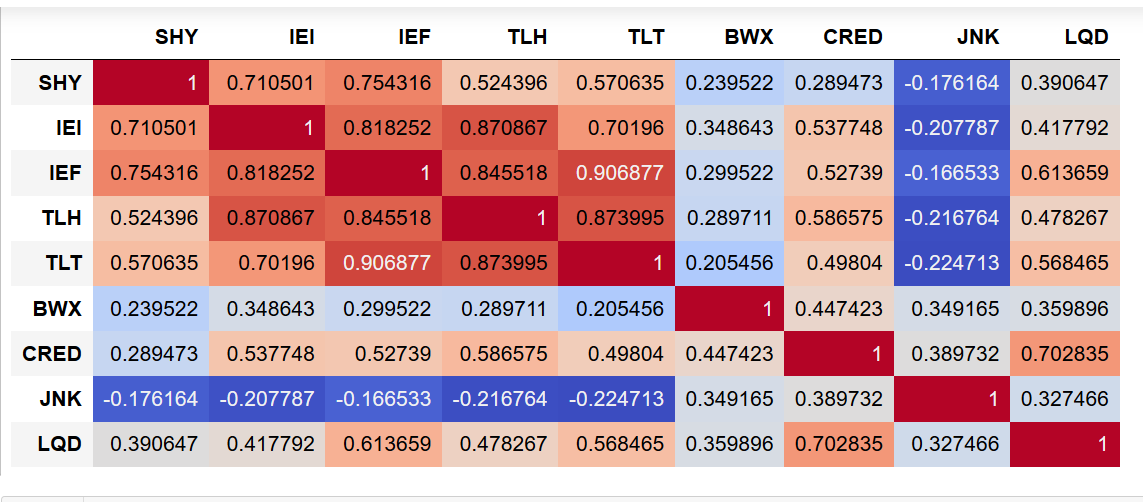
We also calculate correlation matrix for different ETFs in same asset:

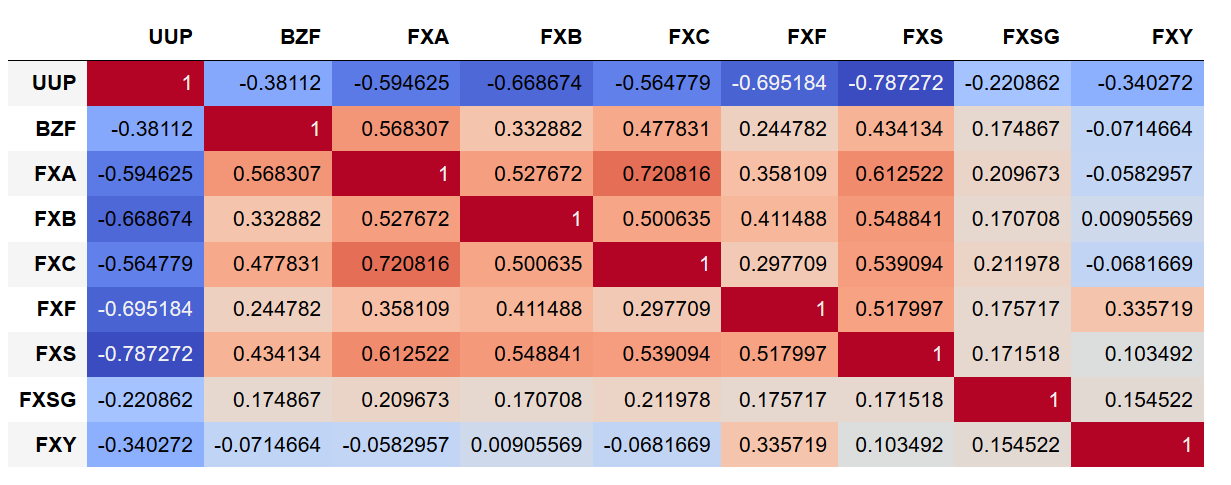












1. **Tb Score work (Ar score is same to Tb)**

In this part, because we want to map turbulence to S&P 500 stocks, we just use this stocks’ log return to calculate turbulence. Just list ETFs, we break stocks to 11 sectors and for each sector, we calculate turbulence index.

To calculate Tb and Ar Score, we use linear regression and add quadratic term:

X = Turbulence index for the sector (this stock is belonging to this sector)

X\_Square = Square of X

Y= Quantile of log return of this stock

We have the format for this linear regression for each stock:

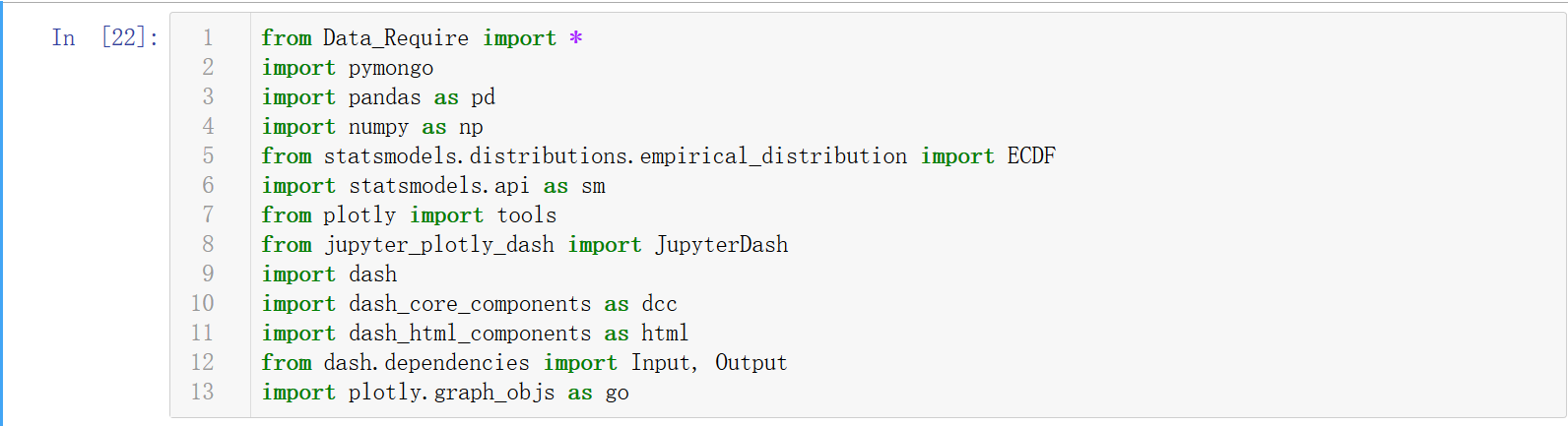
**Y = a\*X + b\*X\_Square + c**

We use lag 5 log return (became log return is weekly) to form the regression(just for prediction power)

We find the result for the model is not really good, the Adj R\_square is only about 0.03, but X and X\_Square are both statistically significant.

After form the regression model, we use this to get score for each stock (use it’s prediction power) and draw interactive plot for them. At this time, because the processing time is too long(exceed 30 seconds), we can not deploy the app to public, but we save the stock log return data in the Mongodb cloud and automatically update the value.

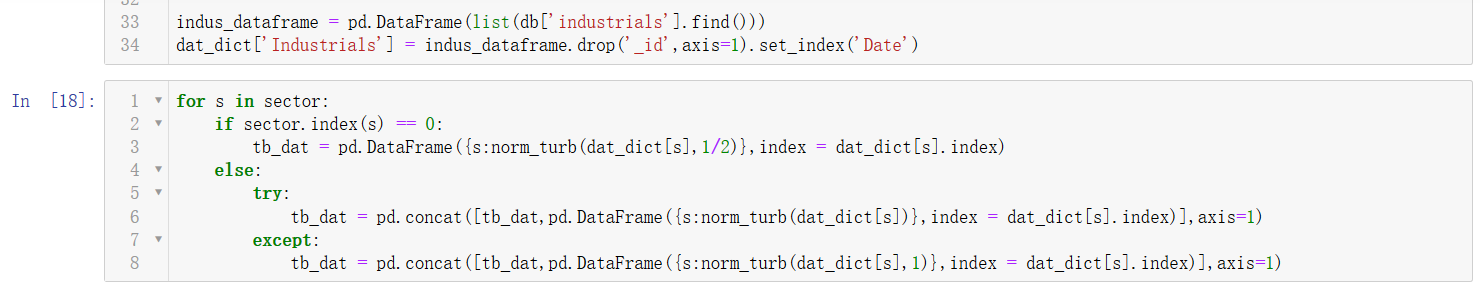
Code and sample result for the score(all the codes is in TB\_regression\_mongodb\_cloud.ipynb):

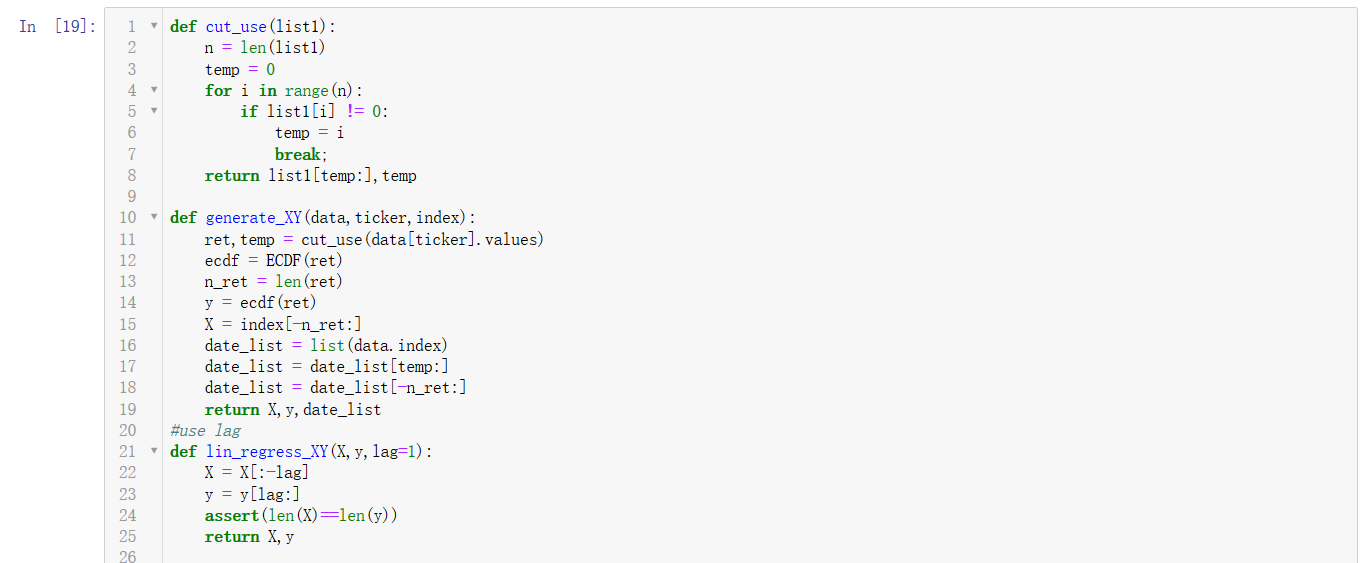


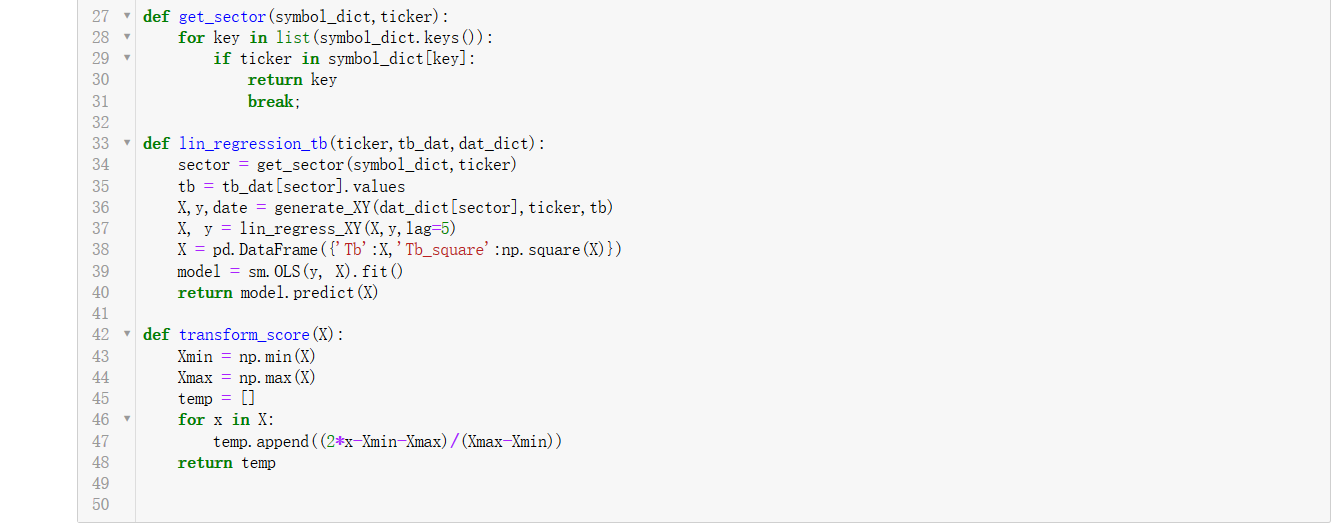




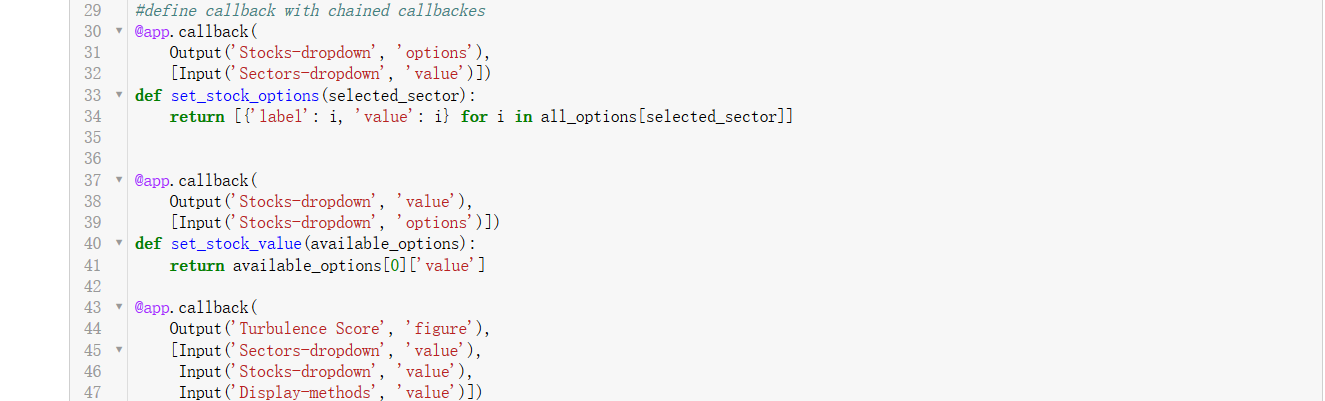


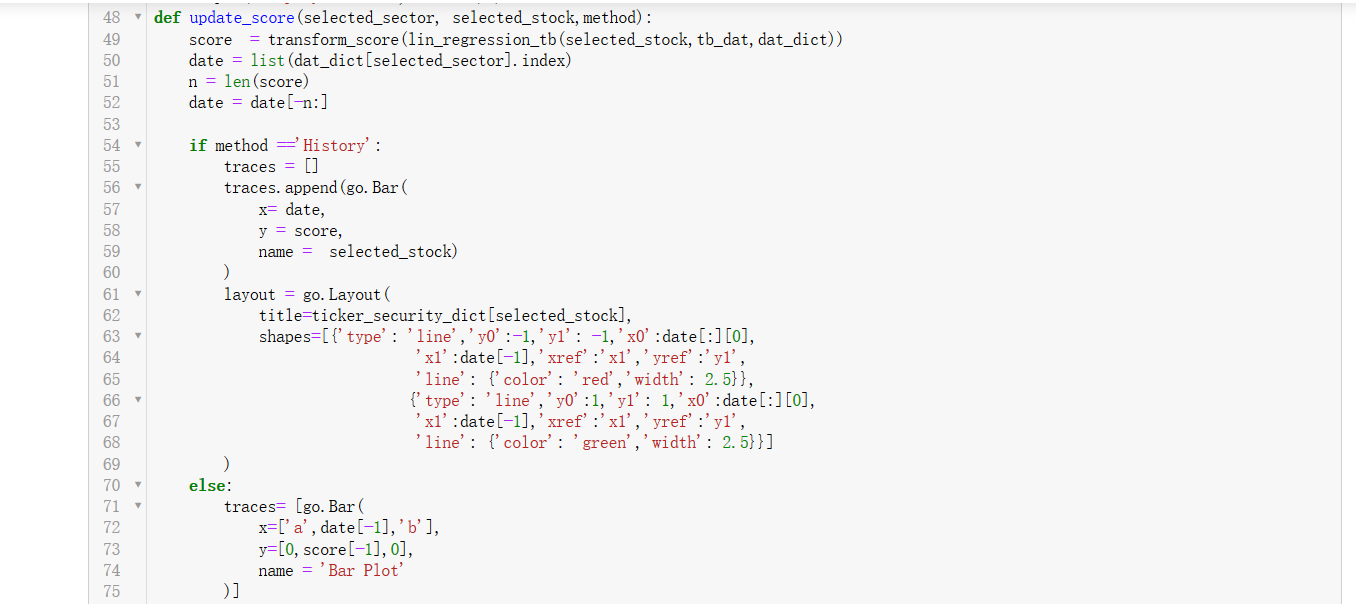






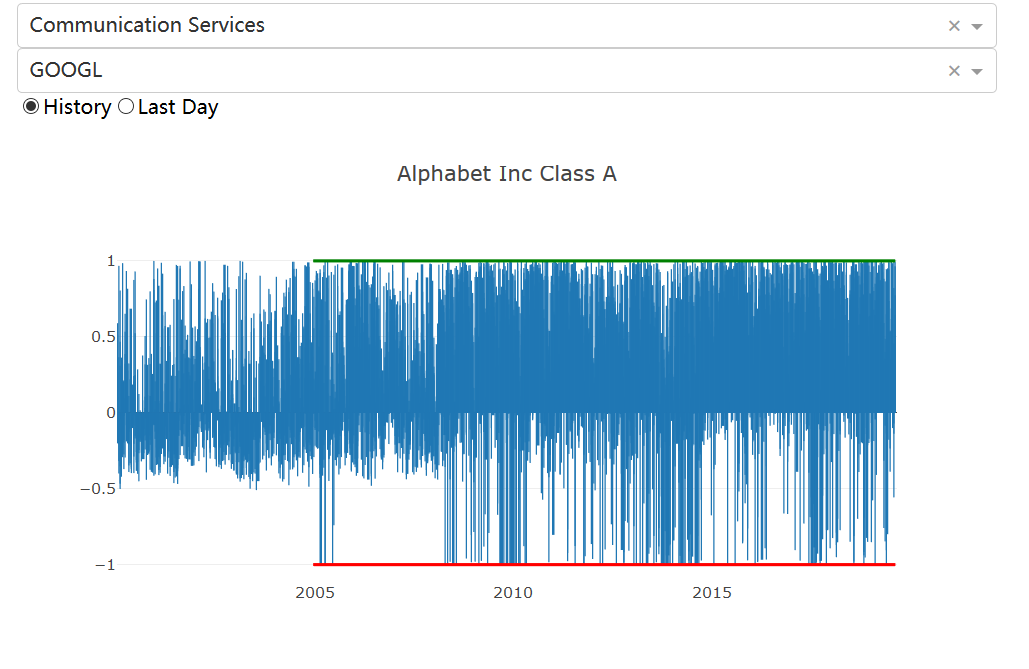


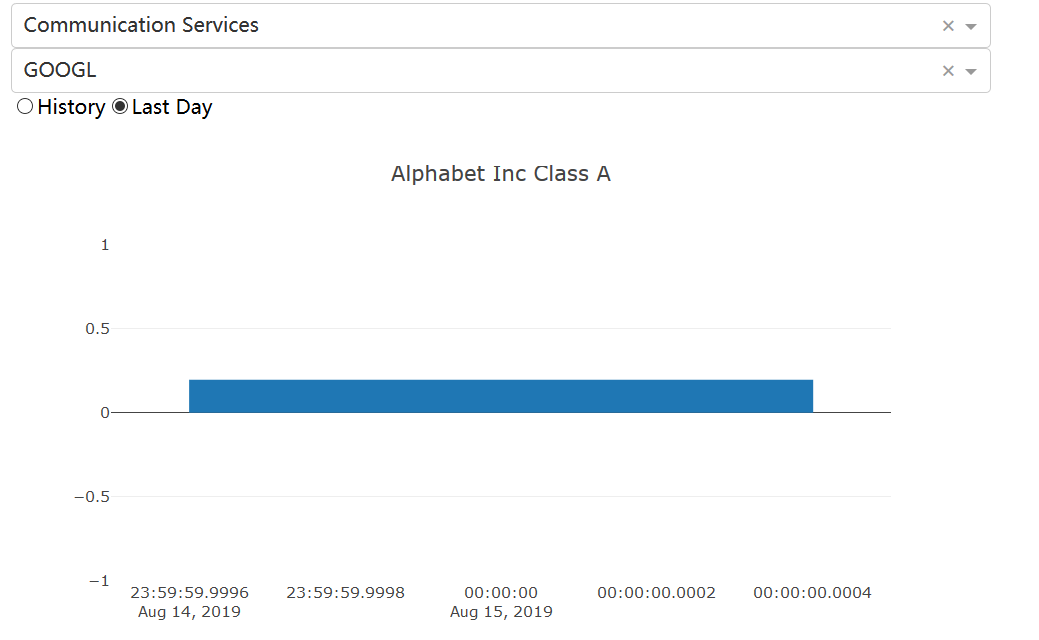






Sample result (for full result, please run program in your own computer):





* **Conclusion**

As we could view Turbulence as a volatility index, which incorporated both the individual volatility and their correlations, it’s explicit and simple. However, summarizing all information in a single index value may sacrifice if the data have high dimensions. It also means two groups of assets may vary greatly with each other even if they show the same turbulence.

For Absorption ratio, it is a good measure of market uniformity. But it also needs careful choice of the calculation period, and the number of eigenvectors chosen in numerator. The longer period for covariance matrix, the more information it includes but also at the risk of changed structure and noisy data.

* **Further work for TB and AR**

To calculate Tb regression, we could compare different log return (daily, weekly and monthly) and use moving average to see the result. But we still need a standard to test the result, maybe implied volatility or other indexes. For the score part, more reasonable method should be considered, calculate momentum of 1 year and 1 week and minus them maybe a good method to get the current value of tb index. Then we could use Markov regime and base on that result to do prediction. Use other time series or nonlinear method maybe better( LSTM maybe a good method)