

Executive Summary

Background

Low volatility can be a very effective predictor for future returns in China A-shares market. According to MSCI, shares with a low volatility has produced strong cumulative returns over the past five years and the MSCI China A Minimum Volatility Index returns outperformed the MSCI China A Index returns by 6.7% per year. As such, AXA Investment Managers wants to build a low volatility signal that is indicative of China A-shares market stock's current and forecasted volatility.

Problem Objective

Our objective is to build a company-level quantitative predictive model to identify stocks with low future volatility using fundamental data that consist of features of Chinese A-shares companies. Our model should strive to outperform a naive guesstimate of the future volatility.

Methodology and Results

Data Cleaning & Scoping: Columns that had many missing or little unique values were removed. Considering that we are forecasting stock volatility, we imperatively took note of the number of data points each company had. Companies that were tracked earlier had more data points. Upon inspection, we learnt that 632 of all 1,800 companies, had a full set of data - data points beginning from January 2001. We approached our model building with these *full-data companies*.

Defining Volatility: This project defines volatility, the main dependent variable, as a measure of the variance of a company's returns. The volatility of a company is calculated by taking the standard deviation of the last 12 values of the company's returns. This rolling window approach gives us an accurate indicator of a company's volatility by month.

Regression model attempt: Having established our dependent variable, we then attempted to find any correlation between the volatility and the other features. Iterating through each company, we found that the correlations calculated with each variable varies, making model building tricky.

ARIMA model attempt: Upon studying the autocorrelation of the volatility of the companies, we learnt that a company's volatility is highly dependent on its previous values. Hence, we leaned towards building a time series forecasting model using the ARIMA (auto regressive integrated moving average) model. Despite having visually assuring plots of our forecast, we needed a way to analyse and benchmark our model.

Model Analysis

Building a naive model that took the mean of the last 12 variance data of a company allowed us to analyse if our model was indeed better than a guesstimate. We found that the mean squared error of our model significantly outperformed the naive model by 52%.

Limitations

Firstly, training our model with full-data companies meant that companies with less data points may suffer data inaccuracy. The ARIMA model we built on also requires that data points are stationary. However, this is compromised due to the spikes in market volatility in 2008 and 2015.

Value Proposition

AXA IM's go-to approach has always been towards industry-level analysis. However, our company level approach was able to bring about fresh insights to the analysis of China A-shares. The company level approach was able to bring out tangible features that the industry-level approach may lack, giving us greater depth into our insights.

Company background

AXA Investment Managers (IM) operates as an investment management company. According to AXA's investment philosophy, low volatility can be an effective predictor for future returns in China A-shares market. Taking an active, long-term approach, it is vital that AXA IM constantly find new ways to analyse data to help clients secure a better investment experience.

Problem definition

In this project, we are required to establish our understanding of the volatility of a stock and use that definition to forecast future volatilities. We approach this by combining company fundamental data (from the income statement and balance sheet), establishing its current volatility to generate a quantitative signal that is indicative of a company's future volatility over the next 12 months. A description of the data provided can be found in [Appendix A](#).

Motivation Methodology

Data Cleaning

Before using the data for prediction modelling, we need to first define which columns (e.g. `asst_risk`, `total_ror`) have sufficient data that may become a possible variable for our model. We removed columns that had less than 10% of non-null data or unique values. Additionally, companies had varying number of data points; companies tracked earlier had more data, with the earliest data point at January 2001. We found that 632 out of 1,800 companies had a full dataset - a sizeable number we went forward to build our model with.

Defining Volatility

We have defined the main dependent variable, volatility, as the standard deviation of the rate of return; and we have explored two approaches in finding it.

Year-by-year volatility

Our first approach was to try to find a standard deviation value for one entire year. Our description of a year by year volatility can be found in [Appendix B](#). However, this approach has several limitations:

- We may miss periods within a year (e.g. 3-month or 6-month) where the volatility spiked or plummeted or both, which would not give a good way of prediction.
- Aggregating the volatility by year transforms 12 monthly values into 1 yearly value, making feature engineering for regression techniques difficult as we are comparing 12 values to 1.

Therefore, we decided to find a better way to calculate volatility as this approach data points, leading to an inaccurate model.

Rolling window volatility

The second approach finds the rolling standard deviation, enabling us to understand how volatility has changed over time or behaved in different market conditions. We computed the standard deviation from a rolling window of 12.

A representation on how the past rate of return data is used to calculate the present volatility can be found in [Appendix C](#). In addition, in the case of missing values, we set the window to accept a minimum of 4 values.

Exploration of data

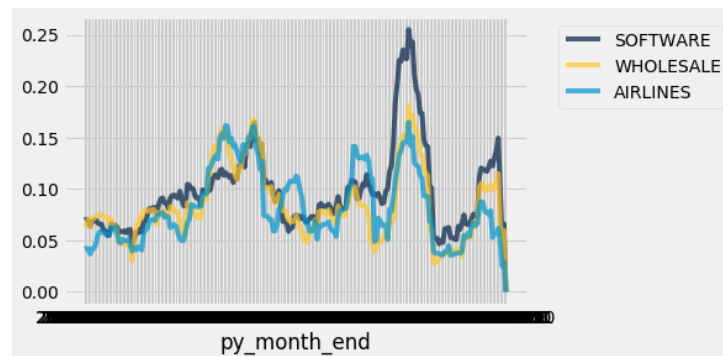


Figure 1: Rolling standard deviation across different industries

In order to identify specific industries that are more sensitive to volatility changes, we plotted the rolling standard deviation across different industries, from year 2001 to 2019. From *Figure 1*, software has higher volatility than other industries. Our client can use this information to justify their investment decisions while using our model for more informed investment experiences. In addition, we can see the sudden spikes in rolling standard deviation for year 2008 and 2015, which are caused by the financial crisis and Chinese stock market boom respectively.¹²

¹ Amadeo, K. (2019). When and Why Did the Stock Market Crash in 2008?. [online] The Balance. Available at: <https://www.thebalance.com/stock-market-crash-of-2008-3305535> [Accessed 11 Dec. 2019].

² Nytimes.com. (2019). Opinion | China's Unsettling Stock Market Boom. [online] Available at: <https://www.nytimes.com/2015/06/15/opinion/chinas-unsettling-stock-market-boom.html> [Accessed 11 Dec. 2019].

Modelling Approach 1: Regression

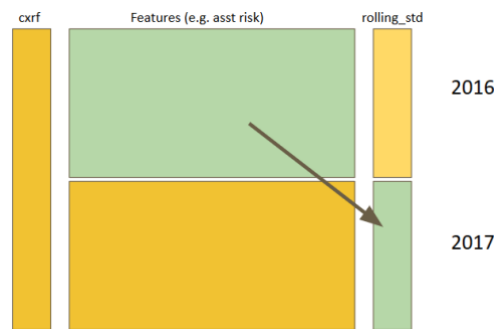


Figure 2: Initial regression modelling approach

Having established our dependent variable, our first approach was to identify similarly-behaving-variables by finding the correlation of volatility with other features (Figure 2). After iterating through each company, we created a correlation matrix heat map (Figure 3). In the table, green and red indicates a positive and negative relationship respectively, and the intensity of a colour's concentration depends on the correlation magnitude. We observed that the correlation varies across each company for each variable, making model building complicated as there are multiple correlation factors to consider. Since we wanted to create a predictive model that is easily replicable, we decided not to delve deeper into regression models and considered other modelling approaches.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	cxrf	12616	12707	9643	12698	12697	9644	12693	9645	9646	12689	12688	9647	12684	12683	9648	9649	9650	12674	9651	9652
2	cxrf																				
3	cxrf	0.094	-0.164	-0.152	0.0808	0.1362	0.0869	0.0513	-0.1066	0.0897	-0.2345	0.0759	-0.2425	-0.4346	-0.1863	-0.0967	-0.4616	-0.0668	-0.3962	0.1568	-0.122
4	cxrf	-0.018	-0.5328	-0.185	0.2933	-0.2184	-0.0236	-0.0537	-0.0922	-0.1902	-0.4075	-0.031	0.0508	-0.1843	0.1057	0.0172	0.1849	0.1314	0.232	0.0439	0.28
5	cxrf	0.3769	0.1951	0.3602	0.4489	0.1998	0.4676	0.1371	0.3088	0.1114	0.5051	0.3697	0.0816	0.4479	0.4178	0.4611	0.3456	0.3642	-0.1126	0.4892	0.773
6	cxrf_risk	0.2999	-0.3665	-0.0018	-0.0312	-0.1068	0.2174	0.1881	-0.203	0.3532	-0.4613	-0.1807	-0.0514	-0.0909	0.1696	0.2024	0.3326	0.4698	0.1185	0.2432	-0.003
7	cxrf	-0.1481	0.2637	0.2568	0.3996	0.4256	0.3912	0.5295	0.0301	0.2792	-0.1339	0.3548	0.0838	0.0612	0.0478	0.5106	-0.2717	0.2295	-0.0973	0.0401	-0.015
8	cxrf	0.2324	-0.303	-0.1005	-0.366	-0.0465	0.134	0.3991	-0.4231	-0.2128	-0.5066	-0.1003	-0.1203	-0.1295	0.1478	0.0135	-0.1034	0.1778	0.1133	0.01	0.15
9	cxrf_risk	-0.1189	-0.3757	-0.211	-0.0729	-0.3502	-0.0434	-0.0613	-0.5399	-0.4365	0.0709	-0.183	-0.0756	0.3793	-0.064	-0.2572	0.0596	-0.1018	0.015	0.2805	0.245
10	cxrf	0.2307	-0.1342	-0.0418	-0.2334	-0.1185	0.2018	-0.1446	-0.3827	-0.1071	-0.0529	0.0696	-0.1184	-0.0377	0.0021	0.6086	-0.0738	0.2796	0.0272	-0.3679	-0.628
11	cxrf	0.181	-0.3866	0.0388	-0.0507	-0.0209	0.0694	-0.0807	-0.3273	-0.4623	0.0121	-0.1079	-0.0811	0.2332	-0.0827	-0.0088	-0.0621	0.2121	-0.0646	0.3837	-0.1702
12	cxrf	0.0643	-0.0324	0.5362	0.3054	-0.0216	0.3058	-0.2858	-0.1743	0.0941	-0.1154	-0.3171	0.2862	-0.0612	0.1606	0.5648	0.3692	0.3301	0.39	0.0471	-0.047
13	cxrf	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578	-0.2578
14	cxrf	-0.2827	0.241	0.2386	0.4268	0.2861	0.424	0.361	0.2497	0.0784	-0.3854	0.2453	0.1184	0.1208	0.209	0.5288	-0.4061	0.1975	-0.1303	0.116	-0.085
15	cxrf	0.296	-0.2567	0.1807	-0.1631	-0.2444	0.0811	-0.122	-0.3419	0.1498	-0.1571	-0.2147	-0.0913	-0.1222	-0.0185	0.2448	-0.1504	0.2315	0.007	-0.4228	-0.199
16	cxrf	0.4419	-0.1861	0.4445	-0.2739	-0.1175	-0.1274	-0.0413	-0.3409	-0.2001	-0.2232	-0.1059	-0.0496	-0.3882	0.1343	0.2475	-0.1818	0.2022	-0.0005	0.4665	-0.1831
17	cxrf	0.2158	-0.3385	0.0635	-0.1271	-0.0722	0.0697	-0.1794	-0.2996	-0.1506	-0.2161	-0.147	0.136	-0.0767	0.0191	0.1814	-0.1155	0.2381	0.0216	-0.3193	-0.183
18	cxrf	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822	-0.0822
19	cxrf_risk	0.2084	-0.3405	0.0124	-0.1245	-0.0414	-0.0517	-0.2271	-0.2985	-0.3495	-0.1906	-0.1783	-0.0208	-0.0533	0.0056	0.1519	-0.1111	0.2203	0.0321	-0.4139	-0.161
20	cxrf	-0.2128	0.2127	0.3496	0.478	0.4361	0.3745	0.5922	0.5624	0.1302	-0.054	0.2017	0.1834	0.1151	0.1777	0.602	-0.2887	0.3268	-0.0943	0.2215	0.01
21	cxrf	0.183	-0.2134	0.3423	-0.4474	-0.4372	0.1308	0.3652	-0.5158	-0.0541	-0.14	-0.1534	-0.3809	-0.02	-0.2606	0.2151	-0.0443	0.1894	0.184	0.3605	-0.031
22	cxrf	0.3278	0.3634	0.2101	-0.184	-0.0481	0.2211	-0.0809	-0.225	0.1096	-0.2028	-0.0767	0.0118	0.1582	0.0368	-0.4583	0.1347	0.1668	-0.4397	0.027	-0.027
23	cxrf	0.3178	0.1684	0.1852	-0.184	-0.1794	0.2156	-0.1789	-0.184	0.1901	-0.2339	-0.0932	0.0118	-0.1222	0.1728	-0.1076	-0.4608	0.1215	0.1444	-0.4897	-0.076
24	cxrf	0.1394	-0.2144	0.1961	-0.3667	-0.1446	0.3381	-0.102	-0.24	0.1796	-0.0504	0.0904	-0.0531	-0.2603	0.1477	0.3617	-0.3697	-0.4628	0.4876	-0.4036	-0.275
25	cxrf	-0.0324	0.1281	-0.1183	-0.0315	-0.3044	-0.094	-0.0533	-0.4736	0.5228	0.1747	0.1461	-0.3728	0.145	-0.2927	0.206	0.3027	-0.5094	-0.5686	-0.3118	-0.167
26	cxrf	0.0928	0.1905	-0.3683	-0.0807	-0.1098	0.1981	0.1937	0.0761	0.4666	-0.3419	-0.6853	-0.2103	-0.2903	0.1438	0.1393	-0.1174	0.1807	0.1963	0.078	-0.078
27	cxrf	-0.205	0.0277	0.1031	-0.0363	0.1654	0.2872	-0.2473	-0.3187	0.0562	-0.2222	0.3954	0.1777	-0.0074	-0.0884	0.3963	-0.4189	0.191	0.4114	-0.5297	-0.141
28	cxrf	0.088	-0.4938	0.0542	0.1691	0.1035	0.0899	-0.0962	-0.3884	-0.2151	-0.1138	-0.0175	0.1763	0.2994	-0.0004	0.0018	-0.0723	0.2002	0.01	0.3793	-0.1875
29	cxrf	-0.1562	-0.1323	-0.1408	-0.0957	-0.061	0.0589	-0.0102	-0.0779	-0.4311	-0.282	-0.0477	0.1293	-0.1188	-0.2818	0.103	0.0241	-0.1819	0.1867	0.0207	-0.027
30	cxrf	-0.2408	-0.1301	0.0027	-0.1284	0.4928	-0.1985	0.4867	-0.4261	0.4413	0.2923	0.1085	-0.1753	-0.2742	0.1587	-0.1014	-0.0759	-0.1555	-0.1096	0.3142	-0.049
31	cxrf	0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643	-0.4643
32	cxrf	0.258	-0.2877	0.2186	-0.4337	-0.0281	-0.0718	-0.1364	-0.1048	0.0124	-0.2555	-0.2357	0.0811	-0.0672	-0.0909	0.0202	-0.1579	0.019	0.0935	-0.3758	-0.218
33	cxrf	0.4586	0.3026	0.0283	-0.1111	0.5184	0.3445	0.5547	0.1736	0.3025	0.1052	0.3881	0.3884	0.0376	0.2418	0.2205	0.3926	0.4648	0.5227	-0.0743	-0.1031
34	cxrf	0.0905	-0.3918	-0.0513	0.4923	-0.1879	-0.1129	-0.1426	-0.1277	-0.2917	-0.2648	-0.0909	0.139	-0.3276	0.0154	0.2147	-0.108	-0.054	0.1973	0.2508	-0.797
35	cxrf	0.406	0.4003	0.2983	0.4461	0.4566	0.5002	0.5007	0.2046	0.4043	0.1344	0.2949	0.2544	0.3236	0.3876	0.5591	0.4318	0.4784	0.5642	0.3794	0.2431
36	cxrf	-0.1114	-0.6905	-0.2005	-0.2462	-0.2383	-0.0987	-0.2912	-0.1091	-0.4538	-0.4053	-0.0117	0.0441	-0.2252	0.0115	-0.0206	0.1494	0.0951	0.0079	0.0114	-0.897
37	cxrf	0.0597	0.1532	-0.0308	-0.0649	-0.0309	0.1099	-0.0778	-0.0582	-0.1243	-0.0473	0.0778	-0.1362	0.0125	0.0148	-0.1065	0.1463	-0.1613	-0.091	0.1336	0.0603
38	cxrf	-0.2124	-0.4823	-0.222	-0.126	-0.1621	-0.0778	-0.1611	-0.3913	-0.3918	-0.4589	-0.3907	-0.0777	-0.3428	-0.0416	-0.1995	0.1245	-0.0864	0.0598	-0.1617	-0.3975
39	cxrf	-0.2028	-0.3075	0.181	0.133	0.0644	0.3876	0.3853	0.7385	0.1255	-0.2987	0.3388	0.1815	0.1017	0.0616	0.6346	-0.0257	0.2008	-0.0656	0.0711	0.1131
40	cxrf	-0.2141	-0.0015	0.1663	0.2583	-0.2414	0.0319	0.052	-0.4802	-0.3538	-0.2647	-0.1377	-0.1295	0.4887	0.0706	-0.1685	0.2506	0.2427	0.0471	0.2181	0.9311
41	cxrf	-0.0017	-0.3409	0.1345	-0.0711	-0.1885	0.2094	0.2128	-0.4811	-0.2602	-0.0157	-0.1295	-0.2471	0.0074	0.1	0.1017	0.3129	-0.0219	-0.4438	-0.1467	-0.1467
42	cxrf	-0.0781	-0.0494	0.5179	0.0506	-0.0528	0.2653	-0.3373	-0.2612	-0.018	-0.151	-0.3801	0.1008	-0.1172	0.0481	0.4739	0.3705	0.2205	0.1997	-0.041	-0.042

Figure 3: Correlation matrix heat map

Modelling Approach 2 : Time Series

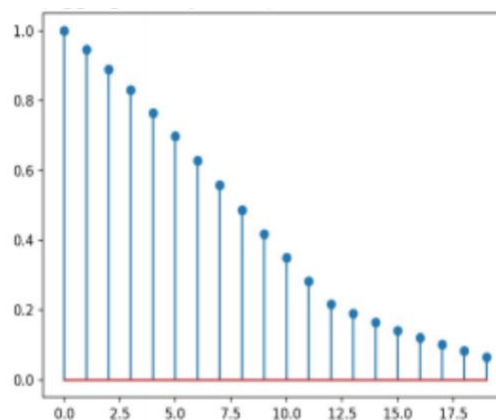


Figure 4: Combined ACF plots of volatility of all companies (aggregated by mean)

After moving out of the regression approach, we did an ACF analysis on the variance of all companies. The result of the ACF Plot (*Figure 4*) showed that a company's volatility is highly dependent on its previous values. Hence, we decided to build an ARIMA (Auto Regressive Integrated Moving Average) time series forecasting model.³

ARIMA is a forecasting algorithm that uses past values of the time series to predict the future values. The ARIMA model is able to take into account the lags of forecast errors between consecutive data and patterns in growth/decline in data, including the rate of the changes. This is achieved through the use of the autocorrelation function, which is easily controlled by differences, auto-regression and moving averages, without performing complicated transformations or using extra surrogate variables.

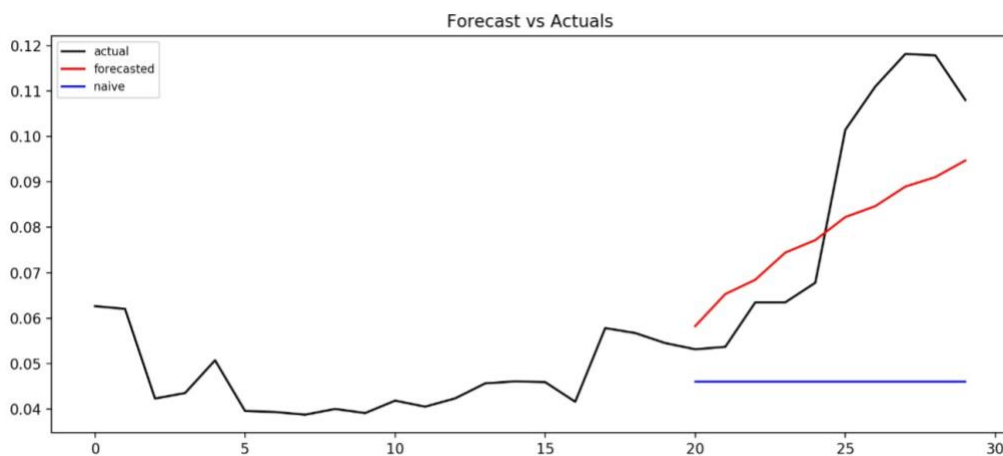
³ An ACF plot gives the values of auto-correlation function of any series with its lagged values, measuring the internal association between observations in a time series

Tools

In the exploration data analysis phase, we utilized *R* and *Tableau* to get a visual understanding of the data. To manipulate data for specific analysis (e.g. aggregating industries), we used the *data science* and *pandas* library from Python 3.x before exporting it into *Excel* to get a visual sensing of our data. Subsequently, we used *Gretl*, a cross-platform software package for econometric analysis, to experiment with different models for our data. Finally, we used the Python's *statsmodel* and *scikit-learn* to polish up our model for presentation. The scripts used can be found [here](#).

Main results

To compare the accuracy of the ARIMA model, we created a naive prediction model. This naive model set the future 12 months rolling standard deviation value to be the mean value of the previous 12 months value. *Figure 5* shows the ARIMA and naive model forecast for one company.



Legend : Red → ARIMA model, Blue → naive model, Black → Actual values

Figure 5: ARIMA and naive model forecast for one company.

After running tests on all full-data companies, the results are as below:

Average mean-squared-error of ARIMA model	0.001637108
Average mean-squared-error of naive model	0.003174031
Percentage difference	52%

Figure 6: Results of average mean-squared-error of ARIMA model, actual values and naive prediction model

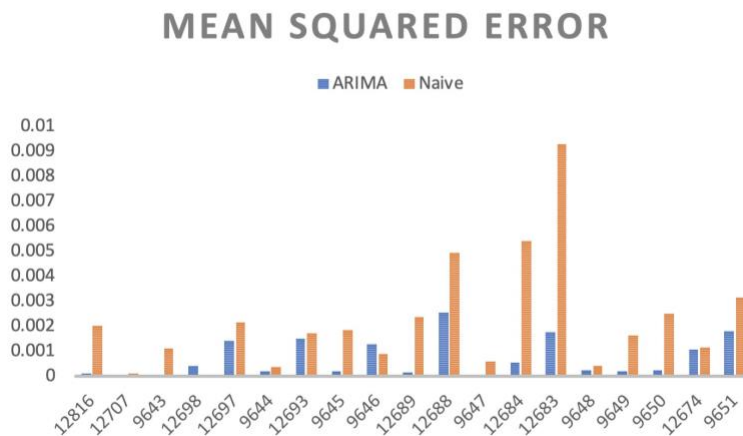


Figure 7: Mean Squared Error between Arima and Naïve model

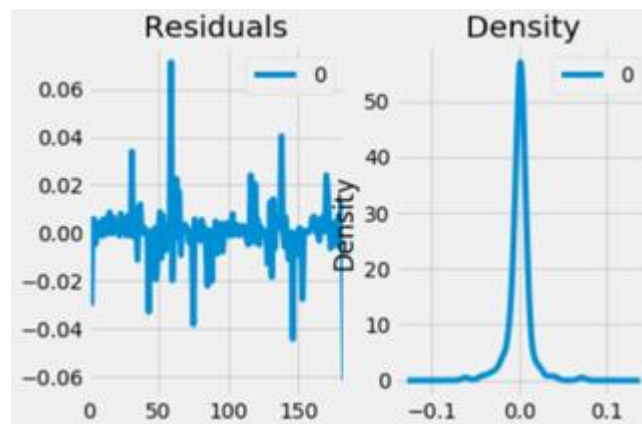


Figure 8: Residual plot

Figure 8 shows a random pattern with density centred at 0, indicating that the residuals are uncorrelated and have zero mean. This indicates that while our model may not be extremely precise, it is accurate.

Assumptions

- The time series is stationary – the mean and standard deviation is constant
- The data for the ARIMA model has to be univariate (works on a single variable). In this case, we are using rolling standard deviation as our variable.
- We assume that a rolling standard deviation with 12 months sliding window is a good representation of volatility

Limitations

One limitation of the model is that the model does not take into account large scale global happenings. Thus, the model is subject to world events like the 2008 financial crisis and the 2015 Chinese stock market boom.

Furthermore, since we are training our model with full-data companies, companies with less data points will have less accurate predicted volatility values.

Conclusion

ARIMA model has its advantages in time-series analyses. The trend, autocorrelation is easily controlled by autoregressive and moving average without performing complicated transformations. Rolling standard deviation is a good statistical measurement of market volatility. All in all, the entire methodology is well described and is made to be easily replicable.

References

1. Wei, Z. (2018). *How the low volatility factor has performed in China A shares - MSCI*. [online] Msci.com. Available at: <https://www.msci.com/www/blog-posts/how-the-low-volatility-factor/01132841955> [Accessed 10 Dec. 2019].
2. Machine Learning Plus. (2019). *ARIMA Model - Complete Guide to Time Series Forecasting in Python | ML+*. [online] Available at: <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/> [Accessed 10 Dec. 2019].
3. Brownlee, J. (2019). *How to Create an ARIMA Model for Time Series Forecasting in Python*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/> [Accessed 10 Dec. 2019].
4. Amadeo, K. (2019). *When and Why Did the Stock Market Crash in 2008?*. [online] The Balance. Available at: <https://www.thebalance.com/stock-market-crash-of-2008-3305535> [Accessed 11 Dec. 2019].
5. Nytimes.com. (2019). *Opinion | China's Unsettling Stock Market Boom*. [online] Available at: <https://www.nytimes.com/2015/06/15/opinion/chinas-unsettling-stock-market-boom.html> [Accessed 11 Dec. 2019].

Appendix

Appendix A: Data provided

Column Name	Description
cxrf	Company index number
returns	Returns of a company. (e.g. if a company loses half its worth, its return will be 0.5)
industry	Industry that company belongs to (e.g. service, F&B)
date	Month that data is recorded of the company
asset risk	45 different financial indexes of the company that we could possible use to predict its volatility
...	
cnc	

Appendix B: Year-by-year volatility

cxrf	returns	date
3544	0.5076	Jan 18
3544	0.8275	Feb 18
3544	0.3782	Mar 18
3544	0.4761	Apr 18
3544	0.9074	May 18
3544	0.3025	Jun 18
3544	0.1246	Jul 18
3544	0.9623	Aug 18
3544	0.1385	Sep 18
3544	0.2920	Oct 18
3544	0.3222	Nov 18
3544	0.7232	Dec 18



cxrf	year	std
3544	2018	0.2795

Appendix C: Rolling Window Volatility

cxrf	returns	date	rolling_std
3544	0.5076	Jan 18	
3544	0.8275	Feb 18	
3544	0.3782	Mar 18	
3544	0.4761	Apr 18	0.236268513
3544	0.9074	May 18	0.281560159
3544	0.3025	Jun 18	0.31146446
3544	0.1246	Jul 18	0.410351122
3544	0.9623	Aug 18	0.438084996
3544	0.1385	Sep 18	0.098505482
3544	0.2920	Oct 18	0.438084996
3544	0.3222	Nov 18	0.098505482
3544	0.7232	Dec 18	0.236268513