Finance And Analytics Club
Students' Gymkhana
Indian Institute of Technology, Kanpur

IMPLEMENTING MEAN REVERSION TRADING STRATEGY

Final Documentation

19th July 2020

By :

Akarsh Mittal
Akshat Goyal
Aman Verma
Dhruv Mittal
Ikjot Singh
Madhav Agarwal
Mohammad Imad Khan
Panshul Rastogi
Pralaykaveri Chaitanya
Shobhit Patel
Ashish Tiwari
Shubham Yadav
Sumit

Project Mentors :

Jatin Khandelwal
Jitesh Agrawal

Acknowledgments

We are grateful to the Science and Technology Council IIT Kanpur, and the Finance and Analytics Club for providing us with an opportunity and motivation to carry out this summer project, given the shortcomings and unprecedented conditions. We would also like to express our gratitude to the coordinator of the Finance and Analytics Club: Jatin Khandelwal and Jitesh Agrawal, for guiding and supporting us through this project.

We would also want to take the opportunity to thank the numerous researchers and authors to citations we referred to in the course of this project. It is because of them that the study of trading strategies and the concept of mean reversion is gaining importance and traders are more confident about their models and strategies.

Abstract

To learn about mean reversion as a concept and how it holds importance when one is forming a profitable trading strategy, along with other basic essentials of corporate finance and how to apply this knowledge in python to filter out the most promising stocks out there on the market. We first filter out stocks on the basis of their stationarity by making the use of the hurst exponent. This is followed by an in-depth analysis of these filtered stocks over three different time intervals to finally select the Top 25 stocks. Upon this selection, we use Machine Learning techniques, namely ARIMA models for the prediction of these 25 stocks and gauge our accuracy.

Contents :

1. Introduction

“Reversion to the mean is the iron rule of the financial markets”

- John C. Bogle

1.1 Overview

Mean Reversion trading is the theory which suggests that prices, returns, or various economic indicators tend to move to the historical average or mean over time. This theory has led to many trading strategies which involve the purchase or sale of a financial instrument whose recent performance has greatly differed from their historical average without any apparent reason.

“*History teaches us that when valuations are extreme, mean reversion, a move towards historical norms, is likely. Once value stocks turn, the recovery can be fast and intense.*”

- Robert D. Arnott

In this project, we used financial concepts of Hurst Values, stockVWAP, basics of mean-reversion theory and machine learning concepts of ARIMA(Auto Regressive Integrated Moving Average) modelling to build an algo-trading strategy on top-25 performing stocks.

2. Problem Statement:

Identify mean-reverting stocks by calculating their Hurst values, calculate the mean value and deviations using StockVWAP and price spread. Use ARIMA modelling to predict the future value of stock. Building a mean-reversion strategy which involves buying when the predicted value falls below mean value - deviation and selling when predicted value rises above mean value + deviation.

3. Understanding from the Project

3.1 Capital Asset Pricing Model (CAPM)

The Capital Asset Pricing Model (CAPM) describes the relationship between systematic risk and expected return for assets, particularly stocks.
CAPM is widely used throughout finance for pricing risky securities and generating expected returns for assets given the risk of those assets and cost of capital.
The formula for calculating the expected return of an asset given its risk is as follows:

$$ER_i = R_f + \beta_i(ER_m - R_f)$$

ER_i = Expected return of investment
R_f = Risk-free rate
β_i = Beta of the investment
ER_m = Expected return of market
$(ER_m - R_f)$ = Market risk premium

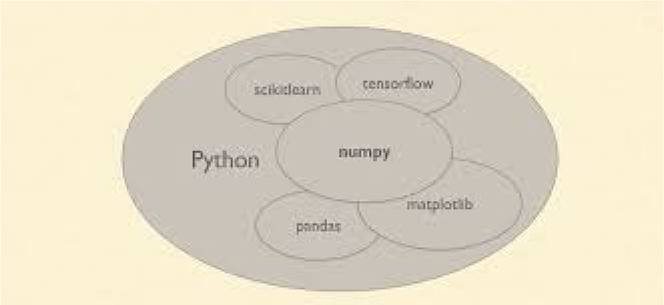
The risk-free rate in the CAPM formula accounts for the time value of money. The other components of the CAPM formula account for the investor taking on additional risk.
The ultimate goal of the CAPM formula is to evaluate whether a stock is fairly valued when its risk and the time value of money are compared to its expected return.

3.2 Central Concepts Of Algorithmic Trading

Algorithmic trading (also called automated trading, or algo-trading) uses a computer program that follows a defined set of instructions (an algorithm) to place a trade.
The algorithms are based on timing, price, quantity, or any mathematical model. Apart from profit opportunities for the trader, algo-trading renders markets more liquid and trading more systematic by ruling out the impact of human emotions and manual errors on trading activities. Also, Algo-Trading can be backtested using available historical and real-time data to check for its viability.
Frequently used Algo-Trading Strategies:

1) Trend-following Strategies
2) Arbitrage Opportunities
3) Mean Reversion
4) Volume-weighted Average Price (VWAP)
5) Index Fund Rebalancing

6) Mathematical Model-based Strategies
7) Time Weighted Average Price (TWAP)



Common library Stack used in Data Science

3.3 Python Libraries : NumPy and Pandas

This project also provided us a handy knowledge about the usage of various python libraries such as numpy, pandas, matplotlib, scikit learn and many more .
NumPy stands for 'Numerical Python' or 'Numeric Python'. It is an open source module of Python which provides fast mathematical computation on arrays and matrices. Since, arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit-learn, Pandas, Matplotlib, TensorFlow, etc. complete the Python Machine Learning Ecosystem.
Similar to NumPy, Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Dataframe. It is like a spreadsheet with column names and row labels.

3.4 Data

We were provided with minute-wise data containing stockVWAP (stock volume-weighted average price), bid Price, ask Price of 111 stocks spanning Energy, IT, Financials, HealthCare, Telecommunications and other sectors of the Indian market for the entire year of 2017. Stock traded for 375 minutes each day from 9:15-3:30 and for 160 days throughout the year. So, we have 90,000 data points to backtest our model and predict future stock price.

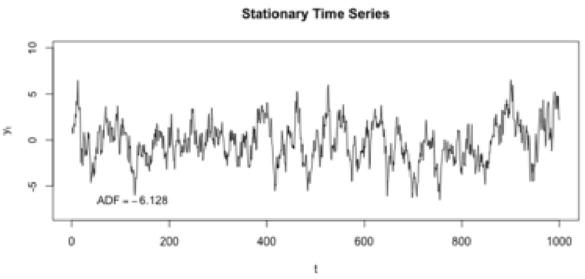
4. Mean Reversion Trading Strategies (MRTS)

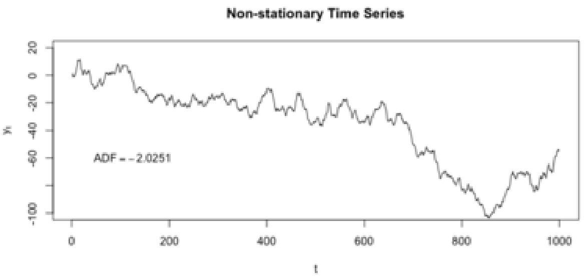
4.1 Hurst Exponent Calculation for checking stationarity of a time series:

Time Series: In simple language, time series is a sequence of observations in a certain period of time. Recording changes in various things such as temperature in a specific city, or as we make use of it, stock prices in regular intervals over a large period of time is a time series. We use a time series as initial data for our analysis and to experiment on to predict whether our models and strategies are profitable or not.

A time series of stock prices tell us a lot about how the stock has performed in different time intervals in the past and how it may perform in the future. Through the years, extensive analysis of time series by using sophisticated statistical tools have told traders a lot about a stock's traits and behaviours. Through time, the trend of the graph has given rise to the concept of Stationarity.

Stationarity: A time series is defined to be stationary if its joint probability distribution is mostly invariant under translations in time or space. In particular, and of key importance for traders, the mean and variance of the process do not change over time or space and they each do not follow a trend.





If a time series is stationary in nature, we observe that the probability distribution is invariant and hence a lot of factors somewhat constant remain in control and such a series is easier to work upon for statistical purposes. Hence, calculating the stationarity of the series becomes important.

To calculate the stationarity of a time series, we have made use of the concept of Hurst Exponent.

Hurst Exponent: Hurst Exponent aims to classify a time series into one of the following; mean reverting, random walking or trending.
The idea behind it is to look at the variance of log prices to assess the rate of diffusive behavior. For a time lag τ, the variance is given by:

$$\mathrm{Var}(\tau) = \langle |\log(t+\tau)-\log(t)^{2} \rangle$$

In case of random walking, or general brownian motion, we can conclude that the equation stated above directly depends on the time lag τ:

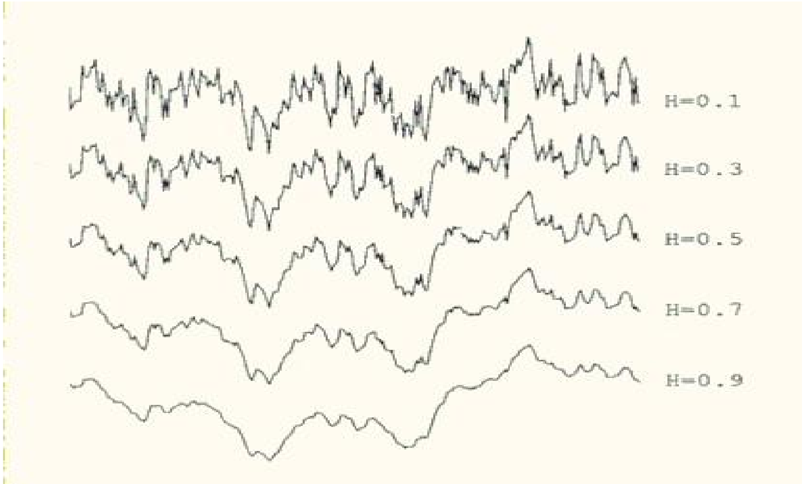
$$\mathrm{Var}(\tau) = \langle |\log(t+\tau)-\log(t)^{2} \rangle \sim \tau$$

Here, if autocorrelations exist (any sequential price movements possess non-zero correlation) this relation stated above does not stand. So to overcome this, we modify the τ variable to τ^{2H}. So we introduce an exponent factor of '2H', where H is the **Hurst Exponent.**

$$\mathrm{Var}(\tau) = \langle |\log(t+\tau)-\log(t)^{2} \rangle \sim \tau^{2H}$$

According to the Hurst Exponent we obtain, we conclude as following:

- **H < 0.5** - Time series is mean reverting
- **H = 0.5** - Time series is random walking o r in General Brownian Motion
- **H > 0.5** - Time series is trending



4.2 Theory of MRTS:

Mean-reversion strategies work on the assumption that there is an underlying stable trend in the price of an asset and prices fluctuate randomly around this trend . Therefore, values deviating far from the trend will tend to reverse direction and revert back to the trend. That is, if the value is unusually high, we expect it to go back down and if it is unusually low, go back up.

4.3 Time frames used for MRTS :

We have taken 1 day, 5 day and 2 weeks as time periods for mean reversion trading strategy.

4.4 Calculation of profits from the mean reverting stocks:

We started with 111 stocks and calculated their Hurst Component. 92 stocks had a Hurst Exponent below 0.5; they were mean reverting. We made a list of these stocks and proceeded to our profit analysis step.

We have split the data of each stock time series into train and test data frames and operate further on the train set. After splitting the data, we refine our data points according to the time frames mentioned to improve computation time.

After we have the final dataframes to operate upon, we begin computing our moving average data in a newly formed array. As a healthy habit in finance, to calculate the moving average, we should resort to using three times the data points than we are calculating for to maintain a more stable trend to compare data upon. Given below is the time intervals and the data points taken into consideration for the moving average.

- 1 Day - 75 Data Points - 225 taken for M.A.
- 5 Day - 185 Data Points - 555 taken for M.A.
- 14 Day - 84 Data Points - 252 taken for M.A.

The first 75, 185, 84 data points of 'stock volume weighted average price' go into the array as they are. The following data points are a moving average of the 75, 185, 84 data points preceding this position.

Now, we calculate the deviation of each data point's stock Volume weighted average price from the moving average taking the spread of ask and bid price, and brokerage into consideration.

We take note of all instances when the deviation is positive(i.e > 0) and count these instances as the number of trades and the deviations for Total profit and mean deviation. Further, we compile this data into a final data frame and also obtain a CSV file of the same.

Now when it comes to selecting the top performing 25 stocks across timeframes, there is no fixed approach needed. One of the approaches we used was loading the 3 profits CSVs in datadrames and adding the columns of total profit(avg.profit per trade* number of trades) and timeframe(1d,5d or 2w depending on the file). Now we selected the stocks whose totalprofit was more than mean value, and avgprofit and numstocks were greater than the lowest 25% values ,individually for the 3 dataframes. We then combined the three datasets into a single dataframe. Now, in the new dataframe, arrange the datapoints in decreasing order of total profit. Then, write a code that reads the stockname and corresponding timeframe and save it in a dictionary. Once a new stockname is entered in the dictionary, if it is encountered again, it will be ignored. Then your answer will be the first 25 entries of your dictionary.

5. Applying Machine Learning

5.1 Why Use Machine Learning here?

We have used the ARIMA model to predict the future prices of the stocks up to the 3 times timeframe so that we can calculate the mean expected price in the future and can trade accordingly.

5.2 ARIMA models

ARIMA, short for 'Auto Regressive Integrated Moving Average' is actually a class of models that 'explains' a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values.
A pure Auto Regressive (AR only) model is one where Yt depends only on its own lags. That is, Yt is a function of the 'lags of Yt'. Likewise a pure Moving Average (MA only) model is one where Yt depends only on the lagged forecast errors. An ARIMA model is one where the time series is differenced at least once to make it stationary and you combine the AR and the MA terms.

ARIMA model in words:
Predicted Yt = Constant + Linear combination Lags of Y (upto p lags) + Linear Combination of Lagged forecast errors (upto q lags)

An ARIMA model is characterized by 3 parameters : p, d, q where,
p is the order of the AR term
q is the order of the MA term
d is the number of differencing required to make the time series stationary

5.3 Parameter selection for best results

For selecting the parameters we iterated through the possible values of the parameters and selected the best model using The Akaike Information Criteria (AIC) which is a widely used measure of a statistical model. It basically quantifies

1) the goodness of fit, and

2) the simplicity/parsimony, of the model into a single statistic. When comparing two models, the one with the lower AIC is generally "better".

6. Problems faced

The stock- price data of each stock was 8-9MB in size as for each stock we had 90k data points. Moreover, working with data of 110 such stocks, it took time to compute the mean, deviation and total profit for each stock for 3 different timeframes (1d, 5d, 14d).

In search for the top 25 performing stocks, we applied various criteria to obtain the best performing stocks. In addition to sorting by total profit for each stock, we also needed to ensure that they were not thinly traded, so we applied additional constraints on 1) number of trades executed 2) average profit per trade

```
df_1 =   df  [ ( df [“avg_profit”]  >  df [“avg_profit”] . quantile(0.25)) &
               ( df [“num_trades”]  >  df [“num_trades”] . quantile(0.25) ) &
               ( df [“total_profit”]  >  df [“total_profit”] . mean() ) ]
```

7. Conclusion :

From the analysis we have found following 25 stocks to be most profitable:

Stock	TimeFrame
INFIBEAM	5d
RELIANCE	5d
YESBANK	5d

HDFCBANK	5d
GRASIM	5d
CESC	1d
RELCAPITAL	1d
BHARATFIN	5d
SUNTV	1d
SRTRANSFIN	5d
KOTAKBANK	5d
DHFL	5d
PCJEWELLER	1d
TATASTEEL	1d
HINDPETRO	1d
AUROPHARMA	1d
TVSMOTOR	1d
BPCL	1d
INFY	5d
JUSTDIAL	1d
M_MFIN	5d
IOC	5d
IBREALEST	5d
RELINFRA	5d
BEL	2w

We also made the trading model using the ARIMA model for these stocks suggesting when to buy, sell or hold the position.