A PROJECT ON TIME SERIES MODELLING OF BANK STOCK PRICE PAIR

Project submitted and prepared by Bishal Biswas under guidance of Prof. Arghya Roy

Project done as a part of assignments for
Post Graduate Diploma Program
From
Bombay Stock Exchange (BSE)
In collaboration with
Maulana Abul Kalam Azad University of Technology
(MAKAUT)

Certification of Approval

This document is hereby approved as credible study of the science subject carried out and represented in a manner to satisfy to the warrants of its acceptance as a prerequisite to the degree for which it has been submitted.

Moreover, it is understood that by this approval the undersigned does not necessarily endorse or approve any statements made, the opinion expressed or conclusion drawn therein but approved only for the sole purpose for which it has been indeed submitted.

Signatures of the Examiners with date.

×______________________________

×______________________________

×______________________________

Dated:

Countersigned by:

×______________________________

Prof. Arghya Roy

Acknowledgement

Our project and everything started during the ending rule of SARS – CoVID19, virtually crippling the society and world as a whole sending everything into a lockdown, although at better stage but still this course of Post Graduate Diploma in Data Science by BSE in collaboration with MAKAUT was made possible thanks to the diplomacy and steps taken by both institutes to combat the situation and make this course and project a possibility.

I want to take this opportunity of the project to thank the people at BSE and MAKAUT who provided us this opportunity to have an exposure to real life scenarios and the status of the present market. I also want to thank Prof. Arghya Roy for guiding with every step from imparting knowledge about the subject to the intricacies of the Data analytics, clearing doubts and issues faced.

I am also grateful to my batchmates and peers where our collective knowledgebase and doubt clearing helped a lot in completing this project. Lastly, I want to thank my family for the mental support they provided me that played a big part in completing this project.

×______________________________

Bishal Biswas.
PGDDSPJULY2020/1
b.biswas_94587@ieee.org

Contents

Objective and Purpose

Data is everywhere and part of our daily lives in more ways than most of us realize in our daily lives. The amount of digital data that exists—that we create—is growing exponentially. According to estimates, in 2021, there will be 74 zetabytes of generated data. That's expected to double by 2024.Hence, there is a need for professionals who understand the basics of data science, big data, and data analytics. These three terms are often heard frequently in the industry, and while their meanings share some similarities, they also mean different things.

Data science is the combination of statistics, mathematics, programming, problem-solving, capturing data in ingenious ways, the ability to look at things differently, and the activity of cleansing, preparing, and aligning data. This umbrella term includes various techniques that are used when extracting insights and information from data.

Now, Big data refers to significant volumes of data that cannot be processed effectively with the traditional applications that are currently used. The processing of big data begins with raw data that isn't aggregated and is most often impossible to store in the memory of a single computer. A buzzword that is used to describe immense volumes of data, both unstructured and structured, big data can inundate a business on a day-to-day basis. Big data is used to analyze insights, which can lead to better decisions and strategic business moves.

Gartner provides the following definition of big data: "Big data is high-volume, and high-velocity or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation."

Data analytics involves applying an algorithmic or mechanical process to derive insights and running through several data sets to look for meaningful correlations. It is used in several industries, which enables organizations and data analytics companies to make more informed decisions, as well as verify and disprove existing theories or models. The focus of data analytics lies in inference, which is the process of deriving conclusions that are solely based on what the researcher already knows.

Industries like IT, Retail, Manufacturing, Automobile, Financial Institute, E Commerce etc. are focusing in depth towards Big Data Concept because they have found out its importance, they know Data is Asset and its value will grow day by day and it can lead the Global business. Some benefits of it are:

- Data driven decision making with more accuracy.

- Customer active engagement.
- Operation optimization.
- Data driven Promotions.
- Preventing frauds & threats.
- Exploring new sources of revenue.
- Being ahead of your competitors.

Thus, the growth of big data analytics will also probably be good for data scientists, especially those who have strong backgrounds in big data. Based on the growth of the big data analytics market in the past few years, along with the rising number of job openings, it's likely that demand for these skills will continue to increase in the near future.

Introduction

Since the invention of computers, people have used the term data to refer to computer information, and this information was either transmitted or stored. But that is not the only data definition; there exist other types of data as well. So, what is the data? Data can be texts or numbers written on papers, or it can be bytes and bits inside the memory of electronic devices, or it could be facts that are stored inside a person's mind.

Now, if we talk about data mainly in the field of science, then the answer to "what is data" will be that data is different types of information that usually is formatted in a particular manner. All the software is divided into two major categories, and those are programs and data. Programs are the collection made of instructions that are used to manipulate data. So, now after thoroughly understanding what is data and data science, let us learn some fantastic facts.

Growth in the field of technology, specifically in smartphones has led to text, video, and audio is included under data plus the web and log activity records as well. Most of this data is unstructured.

The term Big Data is used in the data definition to describe the data that is in the petabyte range or higher. Big Data is also described as 5Vs: variety, volume, value, veracity, and velocity. Nowadays, web-based eCommerce has spread vastly, business models based on Big Data have evolved, and they treat data as an asset itself. And there are many benefits of Big Data as well, such as reduced costs, enhanced efficiency, enhanced sales, etc.

The meaning of data expands beyond the processing of data in computing applications. When it comes to what data science is, a body made of facts is called data science. Accordingly, finance, demographics, health, and marketing also have different meanings of data, which ultimately make up different answers for what is data.

Analysis is the process of breaking a complex topic or substance into smaller parts in order to gain a better understanding of it. The technique has been applied in the study of mathematics and logic since before Aristotle, though analysis as a formal concept is a relatively recent development. Implementing these ideas of analysis using statistical processes to determine the future or optimize situations using available data at the disposal or data obtained from a specific source is the very idea on which the Data Analytics and subsequently Big Data Analytics is based on.

Data Analytics

Data analytics is the science of analyzing raw data in order to make conclusions about that information. Many of the techniques and processes of data analytics have been automated into mechanical processes and algorithms that work over raw data for human consumption.

Data analytics techniques can reveal trends and metrics that would otherwise be lost in the mass of information. This information can then be used to optimize processes to increase the overall efficiency of a business or system.

Why Data Analytics?

Data analytics is a broad term that encompasses many diverse types of data analysis. Any type of information can be subjected to data analytics techniques to get insight that can be used to improve things.

For example, manufacturing companies often record the runtime, downtime, and work queue for various machines and then analyze the data to better plan the workloads so the machines operate closer to peak capacity.

Data analytics can do much more than point out bottlenecks in production. Gaming companies use data analytics to set reward schedules for players that keep the majority of players active in the game. Content companies use many of the same data analytics to keep you clicking, watching, or re-organizing content to get another view or another click.

The process involved in data analysis involves several different steps:

1. The first step is to determine the data requirements or how the data is grouped. Data may be separated by age, demographic, income, or gender. Data values may be numerical or be divided by category.
2. The second step in data analytics is the process of collecting it. This can be done through a variety of sources such as computers, online sources, cameras, environmental sources, or through personnel.
3. Once the data is collected, it must be organized so it can be analyzed. Organization may take place on a spreadsheet or other form of software that can take statistical data.
4. The data is then cleaned up before analysis. This means it is scrubbed and checked to ensure there is no duplication or error, and that it is not incomplete. This step helps correct any errors before it goes on to a data analyst to be analyzed.

Hence, key takeaways from the definition of Data Analytics can be summarized into:

- Data analytics is the science of analyzing raw data in order to make conclusions about that information.
- The techniques and processes of data analytics have been automated into mechanical processes and algorithms that work over raw data for human consumption.
- Data analytics help a business optimize its performance.

Why Data Analytics Matters

Data analytics is important because it helps businesses optimize their performances. Implementing it into the business model means companies can help reduce costs by identifying more efficient ways of doing business and by storing large amounts of data.

A company can also use data analytics to make better business decisions and help analyze customer trends and satisfaction, which can lead to new—and better—products and services.

Types of Data Analytics

Data analytics is broken down into four basic types.

- Descriptive analytics describes what has happened over a given period of time.

    - Examples:
        - Have the number of views gone up?
        - Are sales stronger this month than last?
- Diagnostic analytics focuses more on why something happened. This involves more diverse data inputs and a bit of hypothesizing.
    - Examples:
        - Did the weather affect beer sales?
        - Did that latest marketing campaign impact sales?
- Predictive analytics moves to what is likely going to happen in the near term.
    - Examples:
        - What happened to sales the last time we had a hot summer?
        - How many weather models predict a hot summer this year?
- Prescriptive analytics suggests a course of action.
    - Examples:
        - If the likelihood of a hot summer is measured as an average of these five weather models is above 58%, we should add an evening shift to the brewery and rent an additional tank to increase output.

Data analytics underpins many quality control systems in the financial world, including the ever-popular Six Sigma program. If you aren't properly measuring something—whether it's your weight or the number of defects per million in a production line—it is nearly impossible to optimize it.

Applications of Data Analytics

- Healthcare

  The main challenge for hospitals is to treat as many patients as they efficiently can, while also providing a high. Instrument and machine data are increasingly being used to track and optimize patient flow, treatment, and equipment used in hospitals. It is estimated that there will be a one percent efficiency gain that could yield more than $63 billion in global healthcare savings by leveraging software from data analytics companies.
- Travel

  Data analytics can optimize the buying experience through mobile/weblog and social media data analysis. Travel websites can gain insights into the customer's preferences. Products can be upsold by correlating current sales to the subsequent browsing increase in browse-to-buy conversions via customized packages and offers. Data analytics that is based on social media data can also deliver personalized travel recommendations.

- Gaming

  Data analytics helps in collecting data to optimize and spend within and across games. Gaming companies are also able to learn more about what their users like and dislike.
- Energy Management

  Most firms are using data analytics for energy management, including smart-grid management, energy optimization, energy distribution, and building automation in utility companies. The application here is centered on the controlling and monitoring of network devices and dispatch crews, as well as managing service outages. Utilities have the ability to integrate millions of data points in the network performance and gives engineers the opportunity to use the analytics to monitor the network.

Data Analysis Tools

There are several data analysis tools available in the market, each with its own set of functions. The selection of tools should always be based on the type of analysis performed, and the type of data worked. Here is a list of a few compelling tools for Data Analysis.

1. Excel

It has a variety of compelling features, and with additional plugins installed, it can handle a massive amount of data. So, if you have data that does not come near the significant data margin, then Excel can be a very versatile tool for data analysis.

2. Tableau

It falls under the BI Tool category, made for the sole purpose of data analysis. The essence of Tableau is the Pivot Table and Pivot Chart and works towards representing data in the most user-friendly way. It additionally has a data cleaning feature along with brilliant analytical functions.

3. Power BI

It initially started as a plugin for Excel, but later on, detached from it to develop in one of the most data analytics tools. It comes in three versions: Free, Pro, and Premium. Its PowerPivot and DAX language can implement sophisticated advanced analytics similar to writing Excel formulas.

4. Fine Report

Fine Report comes with a straightforward drag and drops operation, which helps to design various styles of reports and build a data decision analysis system. It can directly connect to all kinds of databases, and its format is similar to that of Excel. Additionally, it also provides a variety of dashboard templates and several self-developed visual plug-in libraries.

5. R & Python

These are programming languages which are very powerful and flexible. R is best at statistical analysis, such as normal distribution, cluster classification algorithms, and regression analysis. It also performs individual predictive analysis like customer behavior, his spend, items preferred by him based on his browsing history, and more. It also involves concepts of machine learning and artificial intelligence.

6. SAS

It is a programming language for data analytics and data manipulation, which can easily access data from any source. SAS has introduced a broad set of customer profiling products for web, social media, and marketing analytics. It can predict their behaviors, manage, and optimize communications.

Data Analysis Techniques

There are different techniques for Data Analysis depending upon the question at hand, the type of data, and the amount of data gathered. Each focuses on strategies of taking onto the new data, mining insights, and drilling down into the information to transform facts and figures into decision making parameters. Accordingly, the different techniques of data analysis can be categorized as follows:

1. Techniques based on Mathematics and Statistics

- Descriptive Analysis: Descriptive Analysis takes into account the historical data, Key Performance Indicators, and describes the performance based on a chosen benchmark. It takes into account past trends and how they might influence future performance.
- Dispersion Analysis: Dispersion in the area onto which a data set is spread. This technique allows data analysts to determine the variability of the factors under study.
- Regression Analysis: This technique works by modeling the relationship between a dependent variable and one or more independent variables. A regression model can be linear, multiple, logistic, ridge, non-linear, life data, and more.

- Factor Analysis: This technique helps to determine if there exists any relationship between a set of variables. In this process, it reveals other factors or variables that describe the patterns in the relationship among the original variables. Factor Analysis leaps forward into useful clustering and classification procedures.
- Discriminant Analysis: It is a classification technique in data mining. It identifies the different points on different groups based on variable measurements. In simple terms, it identifies what makes two groups different from one another; this helps to identify new items.
- Time Series Analysis: In this kind of analysis, measurements are spanned across time, which gives us a collection of organized data known as time-series.

2. Techniques based on Artificial Intelligence and Machine Learning
    - Artificial Neural Networks: a Neural network is a biologically-inspired programming paradigm that presents a brain metaphor for processing information. An Artificial Neural Network is a system that changes its structure based on information that flows through the network. ANN can accept noisy data and are highly accurate. They can be considered highly dependable in business classification and forecasting applications.
    - Decision Trees: As the name stands, it is a tree-shaped model that represents a classification or regression models. It divides a data set in smaller subsets simultaneously developing into a related decision tree.
    - Evolutionary Programming: This technique combines the different types of data analysis using evolutionary algorithms. It is a domain-independent technique, which can explore ample search space and manages attribute interaction very efficiently.
    - Fuzzy Logic: It is a data analysis technique based on probability which helps in handling the uncertainties in data mining techniques.

3. Techniques based on Visualization and Graphs
    - Column Chart, Bar Chart: Both these charts are used to present numerical differences between categories. The column chart takes to the height of the columns to reflect the differences. Axes interchange in the case of the bar chart.
    - Line Chart: This chart is used to represent the change of data over a continuous interval of time.

- Area Chart: This concept is based on the line chart. It additionally fills the area between the polyline and the axis with color, thus representing better trend information.
- Pie Chart: It is used to represent the proportion of different classifications. It is only suitable for only one series of data. However, it can be made multi-layered to represent the proportion of data in different categories.
- Funnel Chart: This chart represents the proportion of each stage and reflects the size of each module. It helps in comparing rankings.
- Word Cloud Chart: It is a visual representation of text data. It requires a large amount of data, and the degree of discrimination needs to be high for users to perceive the most prominent one. It is not a very accurate analytical technique.
- Gantt Chart: It shows the actual timing and the progress of activity in comparison to the requirements.
- Radar Chart: It is used to compare multiple quantized charts. It represents which variables in the data have higher values and which have lower values. A radar chart is used for comparing classification and series along with proportional representation.
- Scatter Plot: It shows the distribution of variables in the form of points over a rectangular coordinate system. The distribution in the data points can reveal the correlation between the variables.
- Bubble Chart: It is a variation of the scatter plot. Here, in addition to the x and y coordinates, the area of the bubble represents the 3rd value.
- Gauge: It is a kind of materialized chart. Here the scale represents the metric, and the pointer represents the dimension. It is a suitable technique to represent interval comparisons.
- Frame Diagram: It is a visual representation of a hierarchy in the form of an inverted tree structure.
- Rectangular Tree Diagram: This technique is used to represent hierarchical relationships but at the same level. It makes efficient use of space and represents the proportion represented by each rectangular area.
- Map
  - Regional Map: It uses color to represent value distribution over a map partition.
  - Point Map: It represents the geographical distribution of data in the form of points on a geographical background. When the points are the same in size, it becomes meaningless for single data, but if the points are as a bubble, then it additionally represents the size of the data in each region.

- Flow Map: It represents the relationship between an inflow area and an outflow area. It represents a line connecting the geometric centers of gravity of the spatial elements. The use of dynamic flow lines helps reduce visual clutter.
- Heat Map: This represents the weight of each point in a geographic area. The color here represents the density.

A little intro to the problem at hand

A standard data of NSE (abbreviation for National Stock Exchange) stocks is to be taken for two companies competing in the same field and over a certain amount of time whereby the task of the analyst is to figure out the real market scenario these two companies are into and give an insight to someone investing their money on each of these companies' stocks. The analysis and modelling are at the final leg of its operation should be able to figure out an identity of the market as well as forecast the potential customer who is eyeing these two companies with intension as to when and where should this customer invest or sell out and by how much or little should the transaction be. The solution is also to be assessed on the metrics of its accuracy of working out the problem of prediction primarily with a manual check of the tendencies via graphical analysis then moving on to the defined processes of AR, MA, ARMA and ARIMA pertaining to time-series to tackle the objective. Also, since the data we have is a historical data which binds the fact we need time-series analysis

Historical Data

Historical data, in a broad context, is collected data about past events and circumstances pertaining to a particular subject. By definition, historical data includes most data generated either manually or automatically within an enterprise. Sources, among a great number of possibilities, include press releases, log files, financial reports, project and product documentation and email and other communications.

Storage capacities have increased significantly in recent years and cloud storage has taken some of the burden of storage administration from many enterprises. Businesses are collecting more data than ever and often storing it for longer, both for their own purposes and to satisfy compliance requirements.

Not all historical data is old and much of it must be retained for a significant amount of time. However, a study of more than 3,000 corporate storage infrastructures indicated that up to 40 percent of the capacity of every disk drive within a business holds data that hasn't been referenced in the last month, six months or one year. Because that data storage requires resources to maintain, data life cycle management (DLM) is recommended to try to ensure that data is not maintained without good reason or for longer than necessary and that it is properly archived or disposed of as appropriate.

In the arena of active trading, market participants dedicate substantial time and effort to gaining insight into how a market's past behavior relates to its future. The acquisition of timely market data and relevant news garners large capital allocations, with firms around the world spending nearly US$27 billion on market-related information annually. No matter if one's approach to the marketplace is rooted in fundamental or technical analysis, profitability depends on the recognition of future opportunities and the elimination of past mistakes.

Historical data analysis is the study of market behavior over a given period of time. The phrase "market behavior" is used in reference to the many different facets of the market and their interactions. Recorded market-related data such as price, volatility and volume are able to be quantified and studied over a defined period. Through detailed examination of a market's past behavior, traders and investors can gain perspective on the inner workings of that market. The information obtained over the course of the process may prove useful in developing a viable trading plan or improving an existing methodology.

Historical data analysis pertaining to an individual security or market can be useful in several ways:

- Market insight: Extensive study of the past behavior of a financial instrument or market can provide the trader with an idea of which exhibited characteristics are normal and which are extraordinary.
- System development: A clear definition of when, what, and how to trade a given market are the starting points for the creation of a trading system. Through historical data analysis, a statistical "edge" may be identified and developed for active trade.
- Consistency: The selection of trades with a predefined expectation can give the trader confidence in the potential outcome. Through understanding how a given trade has performed over time, unexpected results can be reduced.

It has been said that those who do not understand history are doomed to repeat it. The discipline of historical data analysis aspires to not only avoid the mistakes of the past, but establish a working advantage moving into the future.

Financial Data Mining

Data mining is the process of analyzing large, and sometimes-unrelated, data sets for useful information. As technology has evolved, the ability to conduct a data mining operation has become readily accessible to anyone with computing power and a database. This ability to quickly sift through large amounts of information in an attempt to identify relationships and patterns hidden within the data is extremely valuable in the financial markets.

Historical data analysis is essentially a data mining project that focuses on data sets related to the past behavior of a specific market or financial instrument. Recorded market-related statistics such as price, volume, open interest and assorted volatility measures are a few types of market data that can provide cause and context for seemingly erratic market moves.

In order to conduct a data mining operation with focus upon a specific market or security, the following inputs are required:

- Computing power: Access to a personal computer with an adequate processor, hard-drive space and RAM is required. For instance, use of trading platforms such as Metatrader4 and Trading Station require a minimum of a 300 MHz processor, 256MB of RAM and 60 MB of available hard-drive space. Hardware requirements vary depending upon the trading software package, but as a general rule, the more power the better.
- Data set: Selection of a specific time period, or quantity of data to be analyzed, is a key element of a useful study. Many broker-provided software packages furnish complimentary market data to the user, in addition the ability to purchase specialized data sets.
- Query: Basic questions, typically in the form of customized algorithms, are necessary to begin deciphering data.

A study of historical data pertaining to a security or market may prove to have predictive value. Concealed patterns, relationships and tendencies within the data may be identified and capitalized upon by future trading activities.

Market Data: Price

Market-relevant data comes in many different varieties. As mentioned earlier, volatility measures, volume and open interest are all examples of market data. However, the most referenced form of any market-related information is pricing data. Pricing data, or simply price, is the exact value at which both the buyer and seller of a security agree to conduct an exchange. By law, pricing data must be factual and independently verifiable. Because traders and investors are largely concerned with pricing fluctuations as they pertain to a specific market or security, historical pricing data is meticulously inspected for information useful in the prediction of future price variances.

There are two major classifications of pricing data:

- End-of-day (EOD) data: This data is gathered and reported at the trading session's end. It is used by long-term investors, swing traders and true day traders to gain perspective on a trading session's action. EOD data can be grouped in terms of weeks, months and years.
- Intraday data: The traded prices of a security over the course of a trading session are known as intraday data. It focuses on the pricing fluctuations occurring within a single trading session. It may be obtained in real-time, or in historical context using time-based increments or tick-by-tick format (known as tick data). Typically, intraday data is more costly than EOD data, and its availability varies depending upon the instrument or market desired.

For chart-based technical analysts and traders, pricing data is deciphered through the use of automated charting software applications. No matter which classification of pricing data one selects, the software program commissioned with deciphering the data will use predefined parameters to sort and compile the data set. Each desired parameter—delineated in terms of days, minutes, or number of ticks—will represent a unique period.

For each period, there are four key aspects of price that prove valuable in the analysis of historical data:

- Open: The open is the first price traded at the beginning of a given period.
- Close: The close is the last price traded at the end of a given period.
- High: The high is the greatest price traded during a given period.
- Low: The low is the smallest price traded during a given period.

The open, close, high and low-price values often play an important role in chart construction and analysis, and serve as the basis for many trading strategies. It is important to remember that any historical data study needs to have a defined time horizon. The trading approach itself has great bearing upon which time parameters are most relevant to the data analysis.

For instance, if one is looking to invest in blue-chip stocks for retirement, then a 20-year study of S&P 500 daily closing prices based upon EOD data may be the most appropriate. Likewise, if one is involved in the scalping of currencies on the forex, study of a currency's intraday price action in increments of 5, 15, and 30 minutes, will prove much more useful than its weekly closing prices.

In the current electronic marketplace, the availability of historical market data has improved greatly. Trading service companies and brokerage firms offer different types of market data at varying costs to the trader.

Back-testing

Perhaps the most commonly implemented form of historical data analysis is back-testing. Back-testing is the application of a trading method or strategy to a selected historical data set. Automated trading systems, algorithmic trading and more traditional trading approaches often rely upon statistical data compiled through an extensive back-testing study. In order to conduct a back-test, one must have a defined trading strategy and access to a relevant data set. After both are in place, the strategy is used as an overlayment upon the data, and a simulation of the strategy's performance is conducted. Back-testing studies can be simple or intricate, and largely depend upon the sophistication of the trading approach.

Upon completion of the testing, performance metrics can be applied to the results and used to determine the viability of the strategy. Several key statistics are quantified through a comprehensive back-testing study:

- Number of opportunities: The extent and frequency of trade setups created by a strategy over a specified period of time is a crucial piece of information.
- Success rate: A strategy's win/loss percentage, or probability of success, can be useful in determining whether it is a suitable means of trade for a given product or market. It can also shed some light upon the optimal time and product to engage.
- Risk vs reward: A back-testing study can determine the necessary amount of capital needed to properly execute a trading approach upon a market or product. The diagnosis of a market's inherent volatility can be useful in identifying the degree of risk facing the trading strategy.

In earlier days, back-testing was an arduous task performed manually with pencil and paper. Fortunately for modern-day traders, automation has streamlined the procedure, exponentially improving efficiency. Trading platforms provide software functionality capable of executing detailed strategy back-testing operations.

Challenges and Pitfalls

Although historical data analysis is a powerful tool in both system development and strategic fine-tuning, there are also a few pitfalls of which to be aware:

- Hindsight bias: Hindsight bias can be a major problem affecting the accuracy of a back-testing study. Also known as the "I knew it all along" bias, it is the tendency for individuals to assume that unpredictable events can be forecasted ahead of time. Hindsight bias is severely detrimental to historical data analysis because certain results may be perceived avoidable and disregarded. It actively compromises the objectivity of the study, thereby producing skewed results.
- Data omissions and errors: The physical accuracy of the historical data set is of paramount importance to the back-testing study. Even a relatively small number of data errors can impact a study's results greatly over time. This factor is especially important in the examination of intraday data. When considering small time frames or tick-by-tick intervals, precision in the recording of pricing data can be elusive. The quality of the historical data set is crucial to the accuracy of the back-test, and small mistakes can compromise the integrity of study results.
- Software performance: A software "glitch" can destroy the credibility of test results. Strategy testing software is the filter by which market data is sifted. If there is any discrepancy between the software's desired function and its actual function, the results of the back-test are inaccurate. It can be extremely difficult to spot software errors. Manual checks and automated diagnostics are both needed to ensure accuracy.
- Underestimation of randomness: Random chance plays an important role in the marketplace. A trading strategy may produce outstanding results during a back-test, yet struggle in live market conditions. Factors such as slippage, enhanced volatility and periodic fundamental changes in market structure can be impossible to account for, serving to compromise the viability of a trading strategy.

Human psychology and technological failure can affect the relevance of any back-test or study of market history. Inevitably, it serves the trader well to be aware of the old axiom: "past performance does not guarantee future results."

Time Series: an intro

Many statistical methods relate to data which are independent, or at least uncorrelated. There are many practical situations where data might be correlated. This is particularly so where repeated observations on a given system are made sequentially in time. Data gathered sequentially in time are called a timeseries. Examples

Here are some examples in which time series arise:

- Economics and Finance
- Environmental Modelling
- Meteorology and Hydrology
- Demographics
- Medicine
- Engineering
- Quality Control

The simplest form of data is a longish series of continuous measurements at equally spaced time points. That is

- observations are made at distinct points in time, these timepoints being equally spaced
- and, the observations may take values from a continuous distribution.

The above setup could be easily generalized:

for example, the times of observation need not be equally spaced in time, the observations may only take values from a discrete distribution, ... If we repeatedly observe a given system at regular time intervals, it is very likely that the observations we make will be correlated. So, we cannot assume that the data constitute a random sample. The time-order in which the observations are made is vital. Objectives of time series analysis:

- description - summary statistics, graphs
- analysis and interpretation - find a model to describe the time dependence in the data, can we interpret the model?
- forecasting or prediction - given a sample from the series, forecast the next value, or the next few values
- control - adjust various control parameters to make the series fit closer to a target
- adjustment - in a linear model the errors could form a time series of correlated observations, and we might want to adjust estimated variances to allow for this.

Definition of Time-Series

Assume that the series X_t runs throughout time, that is (X_t) t=0, ±1, ±2, ..., but is only observed at times t= 1, …, n.

So, we observe $(X_1, \ldots, X_n)$. Theoretical properties refer to the underlying process $(X_t)_{t \in Z}$. The notations X_t and $X(t)$ are interchangeable. The theory for time series is based on the assumption of 'second-order stationarity'. Real-life data are often not stationary: i.e., they exhibit a linear trend over time, or they have a seasonal effect. So, the assumptions of stationarity below apply after any trends/seasonal effects have been removed.

Stationarity and autocovariances

The process is called weakly stationary or second-order stationary if for all integers t, τ.

$$E(X_t) = \mu$$
$$\mathrm{cov}(X_{t+\tau}, X_\tau) = \gamma_t$$

where μ is constant and γ_t does not depend on τ.

The process is strictly stationary or strongly stationary if,

$$(X_{t_1}, \ldots, X_{t_k}) \quad \text{and} \quad (X_{t_1+\tau}, \ldots, X_{t_k+\tau})$$

have the same distribution for all sets of time points $t_1, \ldots, t_k$ and all integers τ.

Notice that a process that is strictly stationary is automatically weakly stationary. The converse of this is not true in general. However, if the process is Gaussian, that is if $(X_{t_1}, \ldots, X_{t_k})$ has a multivariate normal distribution for all $t_1, \ldots, t_k$, then weak stationarity does imply strong stationarity.

Note that var $(X_t) = \gamma_0$ and, by stationarity, $\gamma_{-t}=\gamma_t$. The sequence (γ_t) is called the autocovariance function. The autocorrelation function(acf)(ϱ_t) is given by

$$\rho_t = \mathrm{corr}(X_{t+\tau}, X_\tau) = \frac{\gamma_t}{\gamma_0}.$$

The acf describes the second-order properties of the time series. We estimate γ_t by c_t, and ϱ_t by r_t, where,

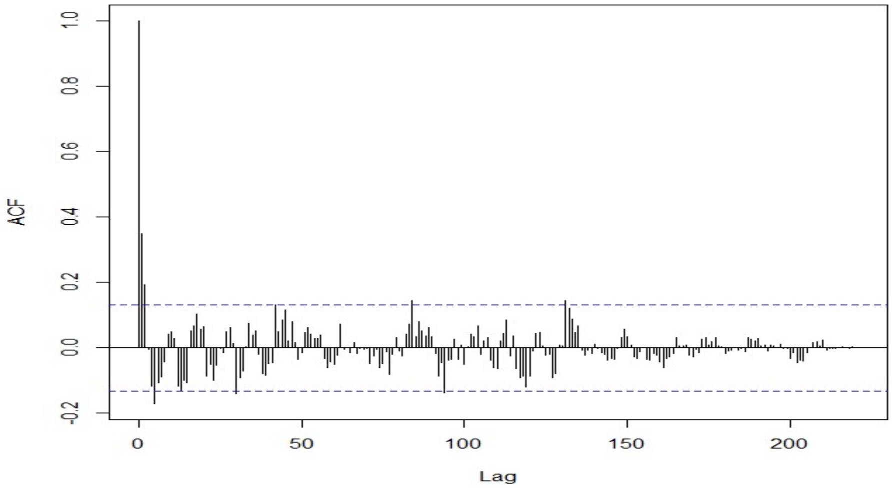
$$c_t = \frac{1}{n} \sum_{s=\max(1,1-t)}^{\min(n-t,n)} [X_{s+t} - \overline{X}][X_s - \overline{X}] \qquad \text{and} \qquad r_t = \frac{c_t}{c_0}.$$

For t >0, the covariance cov ($X_{t+\tau}$, X_τ) is estimated from the $n-t$ observed pairs

$$(X_{t+1}, X_1), \ldots, (X_n, X_{n-t}).$$

If we take the usual covariance of these pairs, we would be using different estimates of the mean and variances for each of the subseries (X_{t+1}, . . ., X_n) and (X_1, . . ., X_{n-t}), whereas under the stationarity assumption these have the same mean and variance. So, we use $\overline{X}$ (twice) in the above formula.

A plot of r_t against t is called the correlogram. A series (X_t)is said to be lagged if its time axis is shifted: shifting by τ lags gives the series ($X_{t-\tau}$). So r_t is the estimated autocorrelation at lag t; it is also called the sample autocorrelation function. An example of such curve is shown below.



Models of stationary processes

Assume we have a time series without trends or seasonal effects. That is, if necessary, any trends or seasonal effects have already been removed from the series. How might we construct a linear model for a time series with autocorrelation?

Linear processes

The process (X_t)is called a linear process if it has a representation of the form

$$X_t = \mu + \sum_{r=-\infty}^{\infty} c_r \epsilon_{t-r}$$

where μ is a common mean, $\{c_r\}$is a sequence of fixed constants and$\{\varepsilon_t\}$ are independent random variables with mean 0 and common variance.

We assume$\sum c^2_r<\infty$ to ensure that the variance of X_t is finite. If the$\{\varepsilon_t\}$ are identically distributed, then such a process is strictly stationary. If c_r= 0 for r <0 it is said to be causal, i.e., the process at time t does not depend on the future, as yet unobserved, values of ε_t. The AR, MA and ARMA processes that we are now going to define are all special cases of causal linear processes

Autoregressive processes

Assume that a current value of the series is linearly dependent upon its previous value, with some error. Then we could have the linear relationship

$$X_t = \alpha X_{t-1} + \epsilon_t$$

where ε_t is a white noise time series. [That is, the ε_t are a sequence of uncorrelated random variables (possibly normally distributed, but not necessarily normal) with mean 0 and variance σ^2.]

This model is called an autoregressive (AR) model, since X is regressed on itself. Here the lag of the autoregression is 1.

More generally we could have an autoregressive model of order p, an AR(p)model, defined by

$$X_t = \sum_{i=1}^{p} \alpha_i X_{t-i} + \epsilon_t.$$

At first sight, the AR(1) process

$$X_t = \alpha X_{t-1} + \epsilon_t$$

is not in the linear form $X_t=\mu+\sum c_r\varepsilon_{t-r}$. However, note that

$$\begin{aligned} X_t &= \alpha X_{t-1} + \epsilon_t \\ &= \epsilon_t + \alpha(\epsilon_{t-1} + \alpha X_{t-2}) \\ &= \epsilon_t + \alpha\epsilon_{t-1} + \alpha^2\epsilon_{t-2} + \cdots + \alpha^{k-1}\epsilon_{t-k+1} + \alpha^k X_{t-k} \\ &= \epsilon_t + \alpha\epsilon_{t-1} + \alpha^2\epsilon_{t-2} + \cdots \end{aligned}$$

which is in linear form.

If ε_t has variance σ^2, then from independence we have that

$$Var(X_t) = \sigma^2 + \alpha^2\sigma^2 + \cdots + \alpha^{2(k-1)}\sigma^2 + \alpha^{2k}Var(X_{t-k}).$$

The sum converges as we assume finite variance.

But the sum converges only if|∝|<1. Thus|∝|<1 is a requirement for the AR (1) process to be stationary.

Moving average processes

Another possibility is to assume that the current value of the series is a weighted sum of past white noise terms, so for example that

$$X_t = \epsilon_t + \beta\epsilon_{t-1}.$$

Such a model is called a moving average (MA) model, since X is expressed as a weighted average of past values of the white noise series.

Here the lag of the moving average is 1.   We can think of the white noise series as being innovations or shocks: new stochastically uncorrelated information which appears at each time step, which is combined with other innovations (or shocks) to provide the observable series X. More generally we could have a moving average model of order q, an MA(q)model, defined by

$$X_t = \epsilon_t + \sum_{j=1}^{q} \beta_j\epsilon_{t-j}.$$

If ε_t has variance σ^2, then from independence we have that

$$Var(X_t) = \sigma^2 + \sum_{j=1}^{q} \beta_j^2\sigma^2.$$

ARMA processes

An autoregressive moving average process ARMA (p, q) is defined by

$$X_t = \sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{j=0}^{q} \beta_j\epsilon_{t-j}$$

where β_0= 1.

A slightly more general definition of an ARMA process incorporates a non-zero mean value μ, and can be obtained by replacing X_t by $X_{t-\mu}$ and X_{t-i} by $X_{t-i-\mu}$ above.

From its definition we see that an MA (q) process is second-order stationary for any$\beta_1, \ldots, \beta_q$. However, the AR (p) and ARMA (p, q) models do not necessarily define second-order stationary time series. For, example, we have already seen that for an AR (1) model we need the condition$|\alpha|<1$. This is the stationarity condition for an AR (1) process. All AR processes require a condition of this type.

Define, for any complex number z, the autoregressive polynomial

$$\phi_\alpha(z) = 1 - \alpha_1 z - \cdots - \alpha_p z^p.$$

Then the stationarity condition for an AR (p) process is: "all the zeros of the function $\varphi_\alpha(z)$ lie outside the unit circle in the complex plane".

This is exactly the condition that is needed on $\{\alpha_1, \ldots, \alpha_p\}$to ensure that the process is well-defined and stationary.

The backshift operator

Define the backshift operator B by

$$BX_t = X_{t-1}, \quad B^2X_t = B(BX_t) = X_{t-2}, \quad \ldots$$

We include the identity operator $IX_t=B^0X_t=X_t$.

Using this notation, we can write the AR(p) process $X_t=\sum_{i=1}^{p} \alpha_i X_{t-i} + \varepsilon_t$ as

$$\left(I - \sum_{i=1}^{p} \alpha_i B^i\right) X_t = \epsilon_t$$

or even more concisely,

$$\phi_\alpha(B)X = \epsilon.$$

Recall that an MA(q) process is Xt=$\varepsilon_t + \sum_{j=1}^{q} \beta_j \varepsilon_{t-j}$.

Define, for any complex number z, the moving average polynomial

$$\phi_\beta(z) = 1 + \beta_1 z + \cdots + \beta_q z^q.$$

Then, in operator notation, the MA(q) process can be written

$$X_t = \left(I + \sum_{j=1}^{q} \beta_j B^j\right) \epsilon_t$$

or

$$X = \phi_\beta(B)\epsilon.$$

For an MA(q) process we have already noted that there is no need for a station-arity condition on the coefficients β_j, but there is a different difficulty requiring some restriction on the coefficients.

Consider the MA (1) process

$$X_t = \epsilon_t + \beta\epsilon_{t-1}.$$

As ε_t has mean zero and variance σ^2, we can calculate the autocovariances to be

$$\begin{aligned}
\gamma_0 &= Var(X_0) = (1+\beta^2)\sigma^2 \\
\gamma_1 &= Cov(X_0, X_1) \\
&= Cov(\epsilon_0, \epsilon_1) + Cov(\epsilon_0, \beta\epsilon_0) + Cov(\beta\epsilon_{-1}, \epsilon_1) + Cov(\beta\epsilon_{-1}, \beta\epsilon_0) \\
&= Cov(\epsilon_0, \beta\epsilon_0) \\
&= \beta\sigma^2, \\
\gamma_k &= 0, \quad k \geqslant 2.
\end{aligned}$$

So, the autocorrelations are

$$\rho_0 = 1, \quad \rho_1 = \frac{\beta}{1+\beta^2}, \quad \rho_k = 0 \quad k \geqslant 2.$$

Now consider the identical process but with β replaced by 1/β. From above we can see that the autocorrelation function is unchanged by this transformation: the two processes defined by β and 1/β cannot be distinguished. It is customary to impose the following identifiability condition: “all the zeros of the function $\varphi_\beta(z)$ lie outside the unit circle in the complex plane”.

The ARMA (p, q) process

$$X_t = \sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{j=0}^{q} \beta_j \epsilon_{t-j}$$

where $\beta_0 = 1$, can be written

$$\phi_\alpha(B)X = \phi_\beta(B)\epsilon.$$

The conditions required are

1. the stationarity condition on $\{\alpha_1, \ldots, \alpha_p\}$
2. the identifiability condition on $\{\beta_1, \ldots, \beta_q\}$
3. an additional identifiability condition: $\varphi_\alpha(z)$ and $\varphi_\beta(z)$ have no common roots.

Condition 3 is to avoid having an ARMA (p, q) model which can, in fact, be ex-pressed as a lower order model, say as an ARMA (p−1, q−1) model.

Differencing

The difference operator ∇ is given by

$$\nabla X_t = X_t - X_{t-1}$$

These differences form a new time series ∇X (of length n−1 if the original series had length n). Similarly

$$\nabla^2 X_t = \nabla(\nabla X_t) = X_t - 2X_{t-1} + X_{t-2}$$

and so on.

If our original time series is not stationary, we can look at the first order difference process ∇X, or second order differences $\nabla^2 X$, and so on. If we find that a differenced process is a stationary process, we can look for an ARMA model of that differenced process.

In practice if differencing is used, usually d= 1, or maybe d= 2, is enough.

ARIMA processes

The process X_t is said to be an autoregressive integrated moving average process ARIMA (p, d, q) if its *d*th difference $\nabla^d X$ is an ARMA (p, q) process.

An ARIMA (p, d, q) model can be written

$$\phi_\alpha(B)\nabla^d X = \phi_\beta(B)\epsilon$$

or

$$\phi_\alpha(B)(I - B)^d X = \phi_\beta(B)\epsilon.$$

Second order properties of MA(q)

For the MA(q) process $X_t=\sum_{j=0}^{q} \beta_j \varepsilon_{t-j}$, where β_0= 1, it is clear that $E(X_t)=0$ for all t. Hence, for k>0, the autocovariance function is

$$\begin{aligned}\gamma_k &= E(X_tX_{t-k}) \\ &= E\left[(\sum_{j=0}^{q}\beta_j\epsilon_{t-j})(\sum_{i=0}^{q}\beta_i\epsilon_{t-k-i})\right] \\ &= \sum_{j=0}^{q}\sum_{i=0}^{q}\beta_j\beta_i E(\epsilon_{t-j}\epsilon_{t-k-i}).\end{aligned}$$

Since the ε_t sequence is white noise, $E(\varepsilon_{t-j}\,\varepsilon_{t-k-i}) = 0$ unless $j=i+k$.

Hence the only non-zero terms in the sum are of the form $\sigma^2\beta_i\beta_{i+k}$ and we have

$$\gamma_k = \begin{cases}\sigma^2\sum_{i=0}^{q-|k|}\beta_i\beta_{i+|k|} & |k| \leqslant q \\ 0 & |k| > q\end{cases}$$

and the acf is obtained via $\varrho_k=\gamma_k/\gamma_0$.

In particular notice that the acf if zero for $|k|>$ q. This 'cut-off' in the acf after lag q is a characteristic property of the MA process and can be used in identifying the order of an MA process.

Second order properties of AR(p)

Consider the AR(p) process

$$X_t = \sum_{i=1}^{p}\alpha_i X_{t-i} + \epsilon_t.$$

For this model fixing $E(X_t) = 0$.

Hence, multiplying both sides of the above equation by X_{t-k} and taking expectations gives

$$\gamma_k = \sum_{i=1}^{p}\alpha_i\gamma_{k-i}, \qquad k > 0.$$

In terms of the autocorrelations $\varrho_k=\gamma_k/\gamma_0$

$$\rho_k = \sum_{i=1}^{p}\alpha_i\rho_{k-i}, \qquad k > 0$$

These are the Yule-Walker equations.

The population autocorrelations ϱ_k are thus found by solving the Yule-Walker equations: these autocorrelations are generally all non-zero.

Our present interest in the Yule-Walker equations is that we could use them to calculate the ϱ_k if we knew the α_i. However, later we will be interested in using them to infer the values ofα_i corresponding to an observed set of sample autocorrelation coefficients.

To identify an AR(p) process:

The AR(p) process has ϱ_k decaying smoothly as k increases, which can be difficult to recognize in a plot of the acf.

Instead, the corresponding diagnostic for an AR(p) process is based on a quantity known as the partial autocorrelation function(pacf). The partial autocorrelation at lag k is the correlation between X_t and X_{t-k} after regression on X_{t-1}, . . ., X_{t-k+1}.

To construct these partial autocorrelations, we successively fit autoregressive processes of order 1,2, . . . and, at each stage, define the partial autocorrelation coefficient a_k to be the estimate of the final autoregressive coefficient: so a_k is the estimate of α_k in an AR(k) process. If the underlying process is AR(p), then α_k= 0 for k > p, so a plot of the pacf should show a cutoff after lag p.

The simplest way to construct the pacf is via the sample analogues of the Yule-Walker equations for an AR(p)

$$\rho_k = \sum_{i=1}^{p} \alpha_i \rho_{|k-i|} \qquad k = 1, \ldots, p$$

The sample analogue of these equations replaces ϱ_k by its sample value r_k:

$$r_k = \sum_{i=1}^{p} a_{i,p} r_{|k-i|} \qquad k = 1, \ldots, p$$

where we write $a_{i,p}$ to emphasize that we are estimating the autoregressive coefficients α_1, . . ., α_p on the assumption that the underlying process is autoregressive of order p.

So we have p equations in the unknowns $a_{1,p}$, . . ., $a_{p,p}$, which could be solved, and the pth partial autocorrelation coefficient is $a_{p,p}$.

Calculating the pacf

In practice the pacf is found as follows.

Consider the regression of $X_t$on X_{t-1}, . . ., X_{t-k}, that is the model

$$X_t = \sum_{j=1}^{k} a_{j,k} X_{t-j} + \epsilon_t$$

with ε_t independent of $X_1, \ldots, X_{t-1}$.

Given data$X_1, \ldots, X_n$, least squares estimates of $\{a_{1,k}, \ldots, a_{k,k}\}$are obtained by minimizing

$$\sigma_k^2 = \frac{1}{n} \sum_{t=k+1}^{n} \left(X_t - \sum_{j=1}^{k} a_{j,k} X_{t-j} \right)^2 .$$

These $a_{j,k}$ coefficients can be found recursively in k for k= 0,1,2, ...

For k= 0: σ^2_0=c_0; $a_{0,0}$= 0, and $a_{1,1}$=$\varrho(1)$.

And then, given the $a_{j,k-1}$ values, the $a_{j,k}$ values are given by

$$a_{k,k} = \frac{\rho_k - \sum_{j=1}^{k-1} a_{j,k-1}\rho_{k-j}}{1 - \sum_{j=1}^{k-1} a_{j,k-1}\rho_j}$$

$$a_{j,k} = a_{j,k-1} - a_{k,k} a_{k-j,k-1} \qquad j = 1, \ldots, k-1$$

and then

$$\sigma_k^2 = \sigma_{k-1}^2 (1 - a_{k,k}^2).$$

This recursive method is the Levinson-Durbin recursion. The $a_{k,k}$ value is the kth sample partial correlation coefficient.

In the case of a Gaussian process, we have the interpretation that

$$a_{k,k} = \operatorname{corr}(X_t, X_{t-k} \mid X_{t-1}, \ldots, X_{t-k+1}).$$

If the process X_t is genuinely an AR(p) process, then $a_{k,k}$= 0 for k > p.

So, a plot of the pacf should show a sharp drop to near zero after lag p, and this is a diagnostic for identifying an AR(p).

Tests on sample autocorrelations

To determine whether the values of the acf, or the pacf, are negligible, we can use the approximation that they each have a standard deviation of around1/√n.

So, this would give ±2/√n as approximate confidence bounds (2 is an approximation to 1.96). In R these are shown as blue dotted lines.

Values outside the range ±2/√n can be regarded as significant at about the 5% level. But if a large number of r_k values, say, are calculated it is likely that some will exceed this threshold even if the underlying time series is a white noise sequence.

Interpretation is also complicated by the fact that the r_k are not independently distributed. The probability of any one r_k lying outside ±2/√n depends on the values of the other r_k.

Statistical Analysis

Fitting ARIMA models: The Box-Jenkins approach

The Box-Jenkins approach to fitting ARIMA models can be divided into three parts:

- Identification
- Estimation
- Verification

Identification

This refers to initial preprocessing of the data to make it stationary, and choosing plausible values of *p* and *q* (which can of course be adjusted as model fitting progresses).

To assess whether the data come from a stationary process we can

- look at the data
- consider transforming it (e.g., by taking logs;)
- consider if we need to difference the series to make it stationary.

For stationarity the acf should decay to zero fairly rapidly. If this is not true, then try differencing the series, and maybe a second time if necessary. (In practice it is rare to go beyond d= 2 stages of differencing.)

The next step is initial identification of *p* and *q*. For this we use the acf and the pacf, recalling that

- for an MA(q) series, the acf is zero beyond lag *q*
- for an AR(p) series, the pacf is zero beyond lag *p*.

We can use plots of the acf/pacf and the approximate±2/√n confidence bounds.

Estimation: AR processes

For the AR(p) process

$$X_t = \sum_{i=1}^{p} \alpha_i X_{t-i} + \epsilon_t$$

we have the Yule-Walker equations $\varrho_k = \sum_{i=1}^{p} \alpha_i \varrho_{|i-k|}$, for k >0.

We fit the parameters $\alpha_1, \ldots, \alpha_p$ by solving

$$r_k = \sum_{i=1}^{p} \alpha_i r_{|i-k|}, \qquad k = 1, \ldots, p$$

These are *p* equations for the *p* unknowns $\alpha_1, \ldots, \alpha_p$ which, as before, can be solved using a Levinson-Durbin recursion.

The Levinson-Durbin recursion gives the residual variance

$$\widehat{\sigma}_p^2 = \frac{1}{n} \sum_{t=p+1}^{n} \left(X_t - \sum_{j=1}^{p} \widehat{\alpha}_j X_{t-j} \right)^2 .$$

This can be used to guide our selection of the appropriate order *p*. Define an approximate log likelihood by

$$-2 \log L = n \log(\widehat{\sigma}_p^2).$$

Then this can be used for likelihood ratio tests. Alternatively, *p* can be chosen by minimizing AIC where

$$AIC = -2 \log L + 2k$$

and *k*=*p* is the number of unknown parameters in the model.

If $(X_t)_t$is a causal AR(p) process with i.i.d. WN(0, σ^2_ε), then (see Brockwell and Davis (1991), p.241) then the Yule-Walker estimator $\widehat{\alpha}$ is optimal with respect to the normal distribution.

Moreover (Brockwell and Davis (1991), p.241) for the pacf of acausal AR(p)process we have that, for *m* > *p,*

$$\sqrt{n}\hat{\alpha}_{mm}$$

is asymptotically standard normal. However, the elements of the vector $\hat{\alpha}_{\mathrm{m}}$= ($\hat{\alpha}_{1\mathrm{m}}$, . . ., $\hat{\alpha}_{\mathrm{mm}}$) are in general not asymptotically uncorrelated.

Estimation: ARMA processes

Now we consider an ARMA(*p, q*) process. If we assume a parametric model for the white noise – this parametric model will be that of Gaussian white noise – we can use maximum likelihood.

We rely on the prediction error decomposition. That is, $X_1, \ldots, X_n$ have joint density

$$f(X_1, \ldots, X_n) = f(X_1) \prod_{t=2}^{n} f(X_t \mid X_1, \ldots, X_{t-1}).$$

Suppose the conditional distribution of X_t given $X_1, \ldots, X_{t-1}$ is normal with mean $\hat{X}_{\mathrm{t}}$ and varianceP_{t-1}, and suppose that$X_1 \sim N(\hat{X}_1, P_0)$. (This is as for theKalman filter– see later.)

Then for the log likelihood we obtain

$$-2 \log L = \sum_{t=1}^{n} \left\{ \log(2\pi) + \log P_{t-1} + \frac{(X_t - \hat{X}_t)^2}{P_{t-1}} \right\}.$$

Here $\hat{X}_{\mathrm{t}}$ and P_{t-1} are functions of the parameters $\alpha_1, \ldots, \alpha_{\mathrm{p}}, \beta_1, \ldots, \beta_{\mathrm{q}}$, and so maximum likelihood estimators can be found (numerically) by minimizing −2×log(L) with respect to these parameters.

The matrix of second derivatives of −2×log(L), evaluated at the mle, is the observed information matrix, and its inverse is an approximation to the covariance matrix of the estimators. Hence, we can obtain approximate standard errors for the parameters from this matrix.

In practice, for AR(p) for example, the calculation is often simplified if we condition on the first m values of the series for some small m. That is, we use a conditional likelihood, and so the sum in the expression for −2×log(L) is taken over t=m+ 1to n.

For an AR(p) we would use some small value of *m*, $m \geq p$.

When comparing models with different numbers of parameters, it is important to use the same value of m, in particular when minimizing AIC−2×log(L) + 2(*p* + *q*).

Verification

The third step is to check whether the model fits the data.

Two main techniques for model verification are

Overfitting: add extra parameters to the model and use likelihood ratio or *t* tests to check that they are not significant.

Residual analysis: calculate residuals from the fitted model and plot their acf, pacf, 'spectral density estimates', etc., to check that they are consistent with white noise.

Python Language

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was created in the late 1980s, and first released in 1991, by Guido van Rossum as a successor to the ABC programming language. Python 2.0, released in 2000, introduced new features, such as list comprehensions, and a garbage collection system with reference counting, and was discontinued with version 2.7 in 2020. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3. With Python 2's end-of-life (and pip having dropped support in 2021), only Python 3.6.x and later are supported, with older versions still supporting e.g., Windows 7 (and old installers not restricted to 64-bit Windows).

Python interpreters are supported for mainstream operating systems and available for a few more (and in the past supported many more). A global community of programmers develops and maintains CPython, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

As of January 2021, Python ranks third in TIOBE's index of most popular programming languages, behind C and Java, having previously gained second place and their award for the most popularity gain for 2020.

Python’s rise to prominence: a WIRED article

(Klint Finley Business 03.02.2020 02:15 PM)

Python is one of the world’s most popular programming languages. In fact, it’s more so than ever. Python climbed from third place to tie for second in the latest ranking of programming language popularity published by the analyst firm RedMonk. It’s the first time that a language other than JavaScript, which remains number one in the firm’s ratings, or Java, the other runner-up, has entered the top two since RedMonk started compiling its rankings in 2012.

That milestone is all the more significant given a sometimes rocky transition from the second version of Python, which the language's developers stopped supporting this year, to the third version.

RedMonk usually doesn't make much of small rankings changes, cofounder Stephen O'Grady writes in the report, but it's rare to see any sort of movement in the top tier of programming languages, which consists of well-established languages. JavaScript is the primary way developers run code inside web browsers and is also increasingly used for other purposes, ranging from mobile and desktop app development to programming drones. Java, meanwhile, is the standard language for writing Android apps and is a corporate software development mainstay.

Python's continued success wasn't a given. The long transition from Python 2 to Python 3 in particular could have shunted developers elsewhere. Python 3 was first released in 2008, and the team initially planned to stop supporting Python 2 in 2015, meaning there would be no further bug fixes and security updates from the official project. But they extended that deadline to 2020 when it became clear that many developers would need more time to update their code to the newer version.

During that time, developers could have opted to switch to a newer programming language, like Mozilla's Rust, Google's Go, or JetBrains' Kotlin. Meanwhile, newer programmers could have opted to learn the more widely used JavaScript, or been turned off by confusion over what tutorials, code samples, and open-source code libraries would work with each version of Python. But if Python lost any developers due to the transition, it appears to have more than made up for them in new converts.

O'Grady cites Python's versatility as one reason for its ongoing popularity. Companies like Google, Dropbox, and Instagram all rely heavily on Python, as do countless smaller ventures. It also has a home in academia as the preferred data-crunching language of many scientists and mathematicians.

RedMonk ranks programming languages based on two criteria: the number of questions asked about each language on the question-and-answer site StackOverflow, and the number of projects based on each language hosted on the Microsoft-owned service GitHub. The idea is to spot trends in the software development profession.

RedMonk's assessment, at least as it relates to Python, is consistent with other measures. According to a survey by StackOverflow, Python is the third most widely used programming language, not counting HTML, behind only JavaScript and the database query language SQL. The survey found Python was the respondents' second-favorite language, after Rust. Meanwhile, the Tiobe index, which measures the number of search engine results for particular languages, shows Python has grown in popularity in recent years and now ranks third in the index, after Java and C.

There was little movement in the top 20 languages in RedMonk's latest report. But O'Grady did flag one rising star further down in the ranks: Dart, a language developed by Google, jumped nine places, from 33rd to 24th in the past 18 months.

Dart is a language for writing software that runs inside web browsers. Dart code is translated into JavaScript, which is supported by practically all modern browsers. O'Grady writes that its surge in popularity is probably due to its use in Google's open-source programming framework Flutter, which was released in December, 2018.

Python and Data Science

Python's popularity in the datascience field is owning to the following reasons:

- It's Flexible

If you want to try something creative that's never done before; then Python is perfect for you. It's ideal for developers who want to script applications and websites.

- It's Easy to Learn

Thanks to Python's focus on simplicity and readability, it boasts a gradual and relatively low learning curve. This ease of learning makes Python an ideal tool for beginning programmers. Python offers programmers the advantage of using fewer lines of code to accomplish tasks than one needs when using older languages. In other words, you spend more time playing with it and less time dealing with code.

- It's Open Source

Python is open-source, which means it's free and uses a community-based model for development. Python is designed to run on Windows and Linux environments. Also, it can easily be ported to multiple platforms. There are many open-source Python libraries such as Data manipulation, Data Visualization, Statistics, Mathematics, Machine Learning, and Natural Language Processing, to name just a few (though see below for more about this).

- It's Well-Supported

Anything that can go wrong will go wrong, and if you're using something that you didn't need to pay for, getting help can be quite a challenge. Fortunately, Python has a large following and is heavily used in academic and industrial circles, which means that there are plenty of useful analytics libraries available. Python users needing help can always turn to Stack Overflow, mailing lists, and user-contributed code and documentation. And the more popular Python becomes, the more users will contribute information on their user experience, and that means more support material is available at no cost. This creates a self-perpetuating spiral of acceptance by a growing number of data analysts and data scientists. No wonder Python's popularity is increasing!

Hence, to sum up, these points, Python isn't overly complex to use, the price is right (freeware software), and there's enough support out there to make sure that you won't be brought to a screeching halt if an issue arises. That means that this is one of those rare cases where "you get what you pay for" most certainly does not apply.

NSE – National Stock Exchange of India Limited

The National Stock Exchange of India Limited (NSE) is India's largest financial market. Incorporated in 1992, the NSE has developed into a sophisticated, electronic market, which ranked fourth in the world by equity trading volume. Trading commenced in 1994 with the launch of the wholesale debt market and a cash market segment shortly thereafter.

Key Takeaways about NSE

- The National Stock Exchange of India Limited (NSE) is India's largest financial market and the fourth largest market by trading volume.

- The National Stock Exchange of India Limited was the first exchange in India to provide modern, fully automated electronic trading.
- The NSE is the largest private wide-area network in India.
- The NSE has been a pioneer in Indian financial markets, being the first electronic limit order book to trade derivatives and ETFs.

Understanding the National Stock Exchange of India Limited (NSE)

Today, the National Stock Exchange of India Limited (NSE) conducts transactions in the wholesale debt, equity, and derivative markets. One of the more popular offerings is the NIFTY 50 Index, which tracks the largest assets in the Indian equity market. US investors can access the index with exchange-traded funds (ETF), such as the iShares India 50 ETF (INDY).

The National Stock Exchange of India Limited was the first exchange in India to provide modern, fully automated electronic trading. It was set up by a group of Indian financial institutions with the goal of bringing greater transparency to the Indian capital market.

Special Considerations

As of June 2020, the National Stock Exchange had accumulated $2.27 trillion in total market capitalization, making it one of the world's largest stock exchange. The flagship index, the NIFTY 50, represents the majority of total market capitalization listed on the exchange.

The total traded value of stocks listed on the index makes up almost half of the traded value of all stocks on the NSE for the last six months. The index itself covers 12 sectors of the Indian economy across 50 stocks. Besides the NIFTY 50 Index, the National Stock Exchange maintains market indices that track various market capitalizations, volatility, specific sectors, and factor strategies.

The National Stock Exchange has been a pioneer in Indian financial markets, being the first electronic limit order book to trade derivatives and ETFs. The exchange supports more than 3,000 Very Small Aperture Terminal (VSAT) terminals, making the NSE the largest private wide-area network in the country. Girish Chandra Chaturvedi is the Chairman of the Board of Directors and Vikram Limaye is the Managing Director and CEO of the exchange.

Benefits of the NSE

The National Stock Exchange is a premier marketplace for companies preparing to list on a major exchange. The sheer volume of trading activity and application of automated systems promotes greater transparency in trade matching and the settlement process.

This in itself can boost visibility in the market and lift investor confidence. Using cutting-edge technology also allows orders to be filled more efficiently, resulting in greater liquidity and accurate prices.

Google Colaboratory

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Problem Statement as provided

The problem statement as was provided during project briefing.

Project 2: Time Series Modelling of Bank Stock Price Pair

Data Set: Last two years daily trade data of HDFCBANK and ICICIBANK. Data can be obtained from

https://www.nseindia.com/get-quotes/equity?symbol=HDFCBANK

https://www.nseindia.com/get-quotes/equity?symbol=ICICIBANK

Data Length: 273 trading days data consisting of LTP and Close price

Fields to be considered: LTP and Close Price.

Compute the daily price ratio using LTP and Close Price

Model the Ratio variable using suitable Time Series Modelling

Problem Statement:

Devise a trading strategy using the stock price data as given. Specifically to identify the duration (or trading signal) when a stock will be kept in long position and the another one in short position.

The trade set up should be market neutral and the positions will be taken simultaneously.

Workaround:

Identify the values of p, q and d, in case we are using ARIMA(p,d,q) model. In this way, we identify the mean reversion of the ratio. In other words we need to find out the speed of mean reversion.

Speed of Mean Reversion = HL = ln(2)/kappa where kappa is estimated from an AR(p) process.

Secondly once we have identified the speed of mean reversion, trade is to be set up. In this case we need to identify using following formula the number of FUTURE contract we should go long and short.

Number of Contract * Lot Size * FutureContract_Price_StockA = Number of Contract * Lot Size * FutureContract_Price_StockB

In the above Lot Size and FutureContract_Price are to be obtained from NSE site. Solve for Number of Contract.

Expected Output:

- Identify the values of p, d and q
- Performance Metrics of the ARIMA Model i.e RMSE.
- Number of Contracts required to set up the trade
- Mean reversion speed.

Interpretation and Workflow

Pulling in ideas as was imparted in the brief intro into problem as well as the problem statement itself. We followed the following steps to workout our way to the end solution of our problem.

Data Collection

Data required for the project was specified in the problem to be obtained from NSE according to the given links shared to us as the problem statement. But here is a hurdle as NSE gives out complete lengths of data of any specified lengths to only companies subscribed to them. For non-subscribed users, data is limited to 1-year lengths.

Hence to work around for this restriction where for each of the banks HDFC and ICICI we need 2-year data of stock value changes we took the data for each banks' in two recurring shifts and joined them one after the other by date as specifier to identify anomaly.

This was a completely manual process where we joined four separate data sheets into two datasheets

1. 2019 to 2020 and 2020 to 2021 for the HDFC
2. 2019 to 2020 and 2020 to 2021 for the ICICI

Next step we copied the relevant fields as was informed in the project briefing (Date, LTP, Close, VOLUME) to a new worksheet

Excel Graphical

1. As a first step to the problem since we need to consider LTP and Close price but LTP being more significant in our case (Specifically discussed on the project briefing classes) we kept Close price but didn't involve in calculation.
2. Next step to the problem we calculated the ratio of LTP for HDFC: ICICI.
3. Now, we plotted the curve of the LTPs' of both banks' stocks as a reference of time as well as the Ratio of LTPs' as a function of time.
4. The visualization of these curves revealed that they have two very prominent seasons. This can be said for all of them viz., LTP of HDFC, LTP of ICICI, Ratios of LTPs.
5. With the ideas obtained from the curves we can proceed as well as later reference about the trends of the dataset we have in hand.

Excel Calculations – Time Series

1. The very start of this process we need to operate the AR(p) hence we need to create lags in the data we are working on.
2. Taking the ratios of LTPs' calculated we generate Lags for L1, L2, L3; this is also factored from the dual seasonality of the ratio curve
3. This is to be noted that the lags setup doesn't involve any mathematical operations on the ratio data, rather in fact for each onward factor of L_i for i= 1, 2, …, the ratios value each start from the beginning of the ratio column or from the previous lag factor but just shifted a cell downward than being parallel (which would waste the purpose of a lag or won't generate lag).
4. With the Lags generated we proceed to calculate the Differences – D1, D2, D3, from the ratio information with the same reason as we reasoned in the option2 of this methodology list.
5. The differences contrary to the lags is calculated from the ratios in a manner where for each D_i for i= 1, 2, …, we calculate a difference for the ratios by literally calculating the differences between the ratios but with the increasing '*i*' the separation between the elements of ratios will increase by that number.
6. To provide as example D_i being the difference term and R_i being the ratio term the differences are calculated as

   $D_1 = R_2 - R_1, R_3 - R_2, R_4 - R_3, R_5 - R_4$, …. and so on.

   Consequently, we also do the others as

   $D_2 = R_3 - R_1, R_4 - R_2, R_5 - R_3, R_6 - R_4$, …. and so on

   and $D_3 = R_4 - R_1, R_5 - R_2, R_6 - R_3, R_7 - R_4$, …. and so on.
7. Now before we do anything, we calculate the Auto-Correlation Function from the Lags and Difference data. Hence, we implement a correlation function with offset functions to calculate the ACF elements.
8. Next up we generate a plot of ACF vs Lags, which triggered us which value to assume for the AR(p) and MA(q) order relevance.
9. With the Lags safely in place we calculate the regression from the Regression module as supported in the Data Analysis Add-Ins supplied and supported by Microsoft© and Office™ products.
10. The regression as obtained we get a set of information and coefficients for setting the Lag equation of AR(p) model.
11. With the values of Differences in place we calculate the Exponentially Weighted Moving Average (EWMA). The first value of EWMA is a simple average of three consecutive primary values of D_2. At this stage we fix a value for alpha as 0.7 which was shared in the project detailing and briefing

session. All following values of EWMA are calculated from the equation required respectively (as was shared in project briefing).

12. Having the EWMA in place we find the Error column elements which are nothing but absolute values of the differences between corresponding elements from EWMA and Difference D_2 columns.
13. Now, having generated the Error column we find the Squared Errors SE(q=i) where i = 1, 2, 3, this directly roots from the reason of taking the lags and differences.
14. SE(q=1) is obtained by directly equating the squares from each error terms.
15. For the consecutive SE(q=2) and SE(q=3) we shift each values of SE(q=1) by 1 and 2 places respectively.
16. Now, with the Squared Errors in place we calculate the regression from the Regression module as supported in the Data Analysis Add-Ins supplied and supported by Microsoft© and Office™ products.
17. The resultant regression details reveal us another set of coefficients which stands for setting up the MA(q) model.
18. With AR(p) and MA(q) models generated we configure the ARIMA solutions (as was required in the problem statement), which also stems out from the two regression solutions we computed and obtain the [p, d, q] order values.
19. As an additional metric as well as for further studies and visualization purposes we plot D_2 as a function of time to see how the values have fared. In a way also noting an anomalous spike.
20. From the project detailing session, we were noted of the value of $\varkappa$ (kappa) as was obtained from the AR(p) process. With this value in hand the speed of mean reversion is calculated using the formulae as given in the problem statement.
21. Next up we calculate the required performance metrics as was required in the problem namely the RMSE viz., Root Mean Squared Error. Obtained in the way the nomenclature suggests from the values of SE(q=1) column using the two-step process:
    a. Taking the mean of the values of Squared Errors from SE(q=1) column.
    b. Calculating the square root of this mean calculated in the previous point (a).
22. At this point we set up the equation for number of contacts according to the values that was shared during the project discussion session provided, where we obtained specific values for lot sizes and future contact prices for each banks' shares. The equation was set up for finding a suitable solution but due to the equation being a linear equation with trivial solutions the minimization required failed repetitively for the Solver operation using the

Solver tool as supported in the Data Analysis and Solver Add-Ins supplied and supported by Microsoft© and Office™ products.

23. This setback faced in the previous step compelled us to pick up a coding platform to recursively find out in a crude manner any suitable answers to the problem.

Python recursive calculation coding

1. With the equation we had found in the Excel™, that failed in the Solver module, we set up the primary coefficient values of the equation.
2. But the Solver having even failed still gave insight into the problem where we found that the minimum permissible difference calculated by the solver is at 447(approx.), pointing erroneous situation in addition to failing most of the times. But random objectives revealed a value of around 50 running at the objective cell updating simulations.
3. With the found insight we tried our luck by fixing a limit around half our random objective simulation.
4. Using nested loops running from 1 to 1 lakh (100,000) thereby operating a combined cycle of $100{,}000^2$ steps.
5. At each value of the loop variable, we check for the equation objective and any set values if matching our objective margin to be printed.
6. Using high configured machines as provided from Google Colaboratory, since this process though small is cumbersome for small computers and is sometimes treated as malwares or worm like codes while running since even being small in addition to superior memory management of python the operation consumes a lot of memory, triggering Antivirus software to spring to action and stop function immediately.
7. In Google Colaboratory the solution takes around 1.7 hours to complete the process since we are using the freeware version of the Colaboratory.
8. From the solutions obtained, we matched to find the least possible solution which as found is much lower than the target we got in the Solver operation hence as conclusion to this equation we took in the solutions as given by the python code as final and binding.

Post Process

The tasks in the post process mainly required us to compile back and collect all the solutions we are required to provide as the expected outputs from the problem statement, and compile them in a presentable manner. This also stands causal to the case that although we created the model, we weren't required to determine the efficacy of the model in the longer run or testing it with time. Also owning to the fact this being a Time Series and the data is for a very small amount of time and taken for a time where the COVID-19 disaster effective disturbed a major portion of market.

Sample of ICICI data

A 16-row sample of the ICICI data as was obtained from NSE website. N.B.: Data in this documentation is limited due to size constraints.

Table is drawn on page 46

Sample of HDFC data

A 16-row sample of the HDFC data as was obtained from NSE website. N.B.: Data in this documentation is limited due to size constraints.

Table is drawn on page 47

Sample of Extracted data

A 10-row sample of the extracted data of HDFC and ICICI as was filtered out from the mother datasets.

N.B.: Data in this documentation is limited due to size constraints.

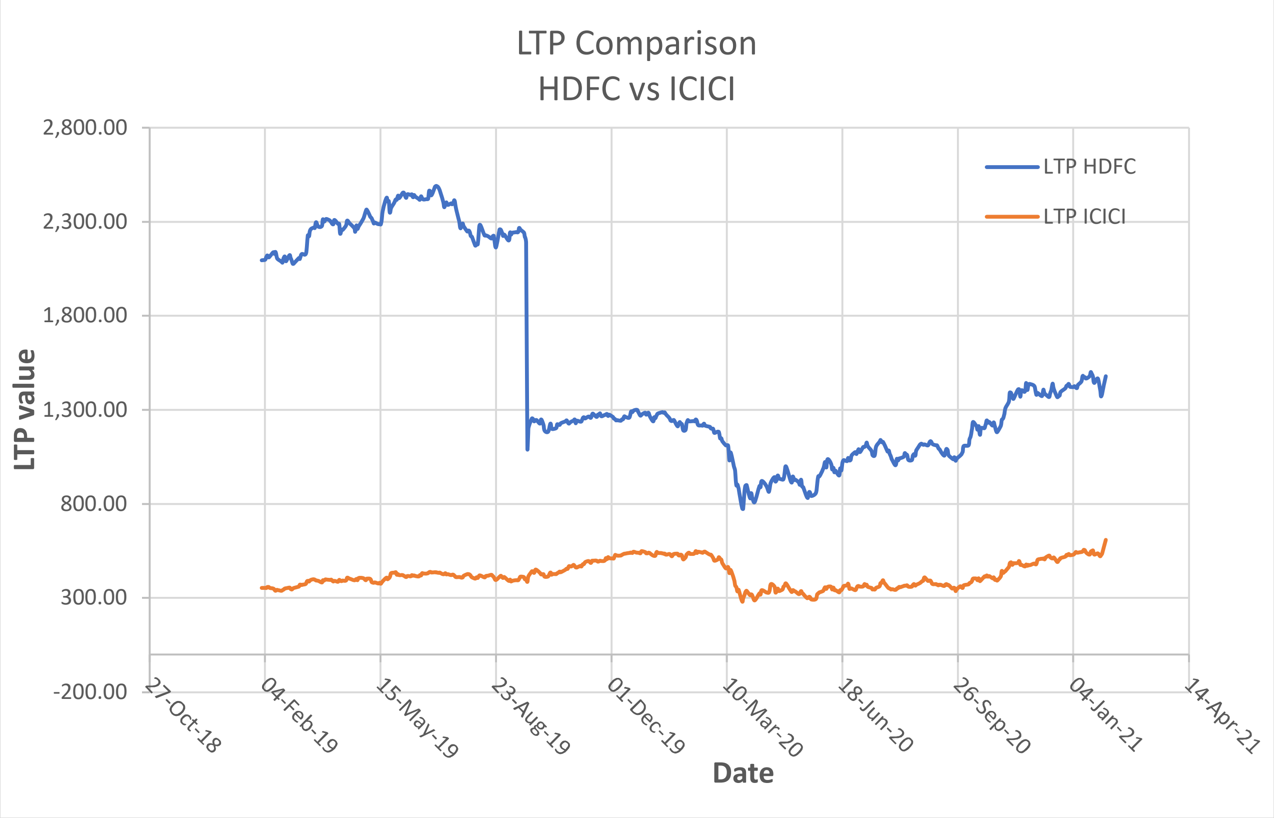
HDFC					ICICI			
Date	**LTP**	**Close**	**VOLUME**		**Date**	**LTP**	**Close**	**VOLUME**
01-Feb-21	1,480.00	1,476.75	13185272		01-Feb-21	608.75	603.8	69354326
29-Jan-21	1,391.00	1,390.50	14352251		29-Jan-21	538.95	537	33172808
28-Jan-21	1,372.00	1,371.45	21352223		28-Jan-21	528.25	528.25	29836591
27-Jan-21	1,411.25	1,409.60	11778138		27-Jan-21	522.75	522.35	28049286
25-Jan-21	1,467.00	1,462.85	10172359		25-Jan-21	537.3	538.05	23415249
22-Jan-21	1,444.35	1,443.55	7696182		22-Jan-21	532	533.8	24515997
21-Jan-21	1,476.00	1,474.80	13166527		21-Jan-21	553.1	552.7	18189277
20-Jan-21	1,490.00	1,492.00	6673026		20-Jan-21	550.75	551	16097138
19-Jan-21	1,501.85	1,503.85	8680127		19-Jan-21	547	546.45	20575358
18-Jan-21	1,480.00	1,483.10	21412816		18-Jan-21	530.45	533.15	21580313

Date	series	OPEN	HIGH	LOW	PREV. CLOSE	ltp	close	vwap	52W H	52W L	VOLUME	VALUE	No of trades
01-Feb-21	EQ	552	609.45	551	537	608.75	603.8	580.42	609.45	268.3	69354326	40,25,47,48,010.30	677016
29-Jan-21	EQ	533.35	544.95	529.45	528.25	538.95	537	536.18	561	268.3	33172808	17,78,66,97,557.65	299110
28-Jan-21	EQ	514.6	529.85	512	522.35	528.25	528.25	520.72	561	268.3	29836591	15,53,63,78,116.80	340760
27-Jan-21	EQ	537.8	539.2	519.1	538.05	522.75	522.35	528.25	561	268.3	28049286	14,81,70,89,783.20	346606
25-Jan-21	EQ	536.55	542.3	531.2	533.8	537.3	538.05	536.87	561	268.3	23415249	12,57,08,91,207.30	230605
22-Jan-21	EQ	553.1	553.5	530.6	552.7	532	533.8	540.16	561	268.3	24515997	13,24,26,69,845.85	286973
21-Jan-21	EQ	555	561	549.7	551	553.1	552.7	556.1	561	268.3	18189277	10,11,51,12,199.80	239999
20-Jan-21	EQ	546.45	554.6	546	546.45	550.75	551	551.04	561	268.3	16097138	8,87,01,39,857.90	203783
19-Jan-21	EQ	539	547.75	533.9	533.15	547	546.45	540.53	561	268.3	20575358	11,12,16,15,616.55	173894
18-Jan-21	EQ	544.5	548.4	528.9	543	530.45	533.15	539.34	561	268.3	21580313	11,63,90,68,880.85	242319
15-Jan-21	EQ	550	551	541.5	553.3	541.95	543	545.13	561	268.3	15708956	8,56,34,45,988.75	256949
14-Jan-21	EQ	554.05	558.45	550.6	556.5	553.6	553.3	553.49	561	268.3	13330898	7,37,85,01,384.15	204427
13-Jan-21	EQ	551.5	561	548.25	548	556	556.5	554.8	561	268.3	21920297	12,16,13,43,003.50	228055
12-Jan-21	EQ	541	550.65	537.1	544.7	547.95	548	544.09	554.4	268.3	16388310	8,91,66,50,749.60	225207
11-Jan-21	EQ	545.15	546	535	542.05	545.2	544.7	540.62	554.4	268.3	19394393	10,48,50,70,657.75	201986
08-Jan-21	EQ	547	547.1	536.35	541.1	542	542.05	541.01	554.4	268.3	21937965	11,86,87,28,979.40	169218

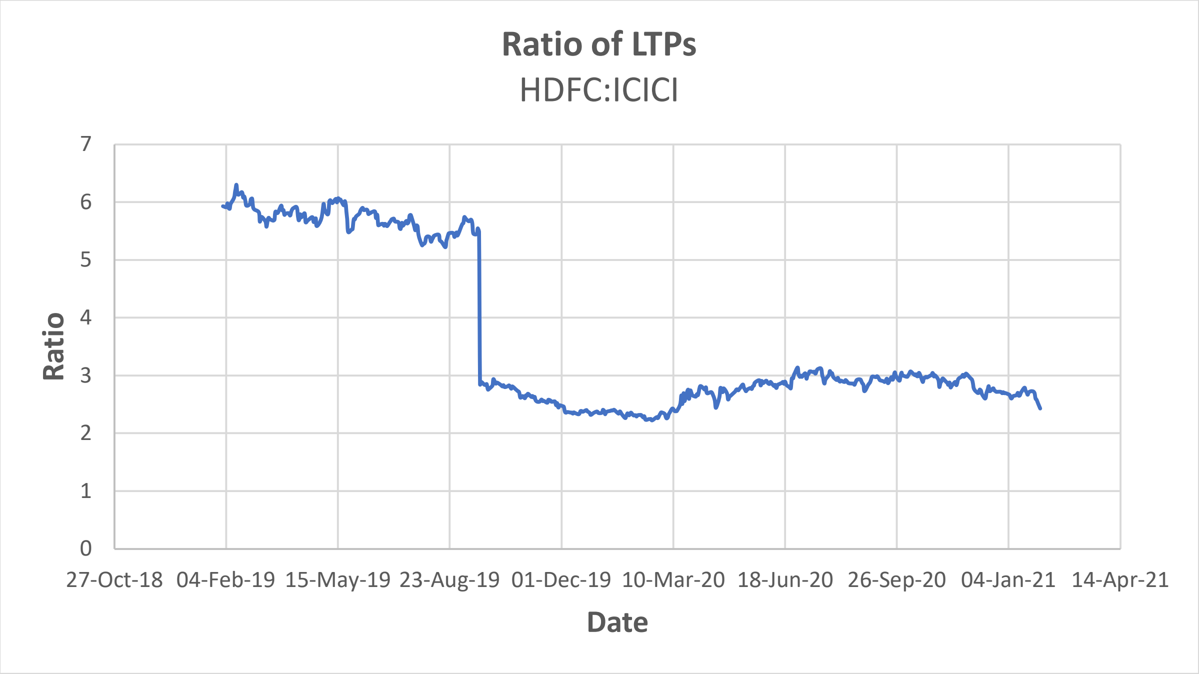
Date	series	OPEN	HIGH	LOW	PREV. CLOSE	ltp	close	vwap	52W H	52W L	VOLUME	VALUE	No of trades
01-Feb-21	EQ	1,410.25	1,482.50	1,401.00	1,390.50	1,480.00	1,476.75	1,439.43	1,511.65	738.75	13185272	18,97,92,31,489.45	301952
29-Jan-21	EQ	1,391.35	1,408.75	1,364.50	1,371.45	1,391.00	1,390.50	1,388.21	1,511.65	738.75	14352251	19,92,38,91,702.85	358077
28-Jan-21	EQ	1,389.90	1,401.30	1,342.00	1,409.60	1,372.00	1,371.45	1,364.16	1,511.65	738.75	21352223	29,12,78,86,352.35	452148
27-Jan-21	EQ	1,468.00	1,471.90	1,406.15	1,462.85	1,411.25	1,409.60	1,437.02	1,511.65	738.75	11778138	16,92,54,45,420.15	308009
25-Jan-21	EQ	1,465.10	1,481.00	1,455.15	1,443.55	1,467.00	1,462.85	1,465.91	1,511.65	738.75	10172359	14,91,18,05,741.75	231078
22-Jan-21	EQ	1,467.90	1,467.90	1,440.15	1,474.80	1,444.35	1,443.55	1,454.40	1,511.65	738.75	7696182	11,19,33,63,698.15	222926
21-Jan-21	EQ	1,492.00	1,494.35	1,468.15	1,492.00	1,476.00	1,474.80	1,479.22	1,511.65	738.75	13166527	19,47,61,38,889.35	262288
20-Jan-21	EQ	1,501.00	1,501.00	1,486.00	1,503.85	1,490.00	1,492.00	1,493.75	1,511.65	738.75	6673026	9,96,78,40,647.30	137443
19-Jan-21	EQ	1,491.80	1,511.65	1,467.00	1,483.10	1,501.85	1,503.85	1,490.12	1,511.65	738.75	8680127	12,93,44,47,133.90	197402
18-Jan-21	EQ	1,469.90	1,502.85	1,467.00	1,466.65	1,480.00	1,483.10	1,488.94	1,502.85	738.75	21412816	31,88,24,88,250.45	483314
15-Jan-21	EQ	1,469.10	1,471.65	1,445.00	1,468.75	1,467.00	1,466.65	1,458.15	1,496.90	738.75	7082618	10,32,75,07,402.40	168203
14-Jan-21	EQ	1,471.15	1,488.00	1,456.00	1,470.65	1,474.00	1,468.75	1,468.80	1,496.90	738.75	6148583	9,03,10,62,462.40	187531
13-Jan-21	EQ	1,492.90	1,496.90	1,462.10	1,481.00	1,473.65	1,470.65	1,476.17	1,496.90	738.75	8467325	12,49,91,97,280.65	179332
12-Jan-21	EQ	1,452.45	1,487.70	1,449.10	1,451.45	1,480.55	1,481.00	1,469.44	1,487.70	738.75	10194078	14,97,95,63,595.90	226594
11-Jan-21	EQ	1,450.00	1,464.90	1,436.30	1,431.65	1,452.60	1,451.45	1,452.80	1,464.90	738.75	8665696	12,58,94,82,027.00	213318
08-Jan-21	EQ	1,432.00	1,442.00	1,423.10	1,416.25	1,433.00	1,431.65	1,432.81	1,464.40	738.75	6884382	9,86,39,97,390.15	162375

Graphical Pre-Analysis

The graphs as generated for the data as a precursor to operating the timeseries operations to the data extracted out and the ratios calculated.



↑ Graphs of LTPs for HDFC and ICICI as a comparison with respect to time



↑ Graph of the Ratio of LTPs with respect to time

Sample of the Ratio, Lags, Differences, EWMA, Error, and Squared Errors

A 13-row sample of the various working columns as an example.

Ratio	Lags			Differences			EWMA	Error	SE(q)		
	L1	L2	L3	D1	D2	D3			SE(q=1)	SE(q=2)	SE(q=3)
2.431211499											
2.580944429	2.431211499			0.14973293							
2.597255088	2.580944429	2.431211499		0.016310659	0.166043589						
2.699665232	2.597255088	2.580944429	2.431211499	0.102410144	0.118720803	0.268453733					
2.730318258	2.699665232	2.597255088	2.580944429	0.030653026	0.13306317	0.149373829	0.139275854	0.006212684	3.85974E-05		
2.714943609	2.730318258	2.699665232	2.597255088	-0.015374649	0.015278377	0.117688521	0.137412049	0.122133672	0.014916634	3.85974E-05	
2.668595191	2.714943609	2.730318258	2.699665232	-0.046348418	-0.061723067	-0.031070041	0.100771947	0.162495015	0.02640463	0.014916634	3.85974E-05
2.705401725	2.668595191	2.714943609	2.730318258	0.036806534	-0.009541884	-0.024916533	0.052023443	0.061565327	0.003790289	0.02640463	0.014916634
2.745612431	2.705401725	2.668595191	2.714943609	0.040210707	0.077017241	0.030668822	0.033553845	0.043463396	0.001889067	0.003790289	0.02640463
2.790083891	2.745612431	2.705401725	2.668595191	0.04447146	0.084682166	0.1214887	0.046592864	0.038089303	0.001450795	0.001889067	0.003790289
2.70689178	2.790083891	2.745612431	2.705401725	-0.083192111	-0.038720652	0.001490055	0.058019654	0.096740306	0.009358687	0.001450795	0.001889067
2.662572254	2.70689178	2.790083891	2.745612431	-0.044319525	-0.127511637	-0.083040177	0.028997563	0.156509199	0.024495129	0.009358687	0.001450795
2.65044964	2.662572254	2.70689178	2.790083891	-0.012122614	-0.056442139	-0.139634251	-0.017955197	0.038486942	0.001481245	0.024495129	0.009358687

ACF calculation and curves

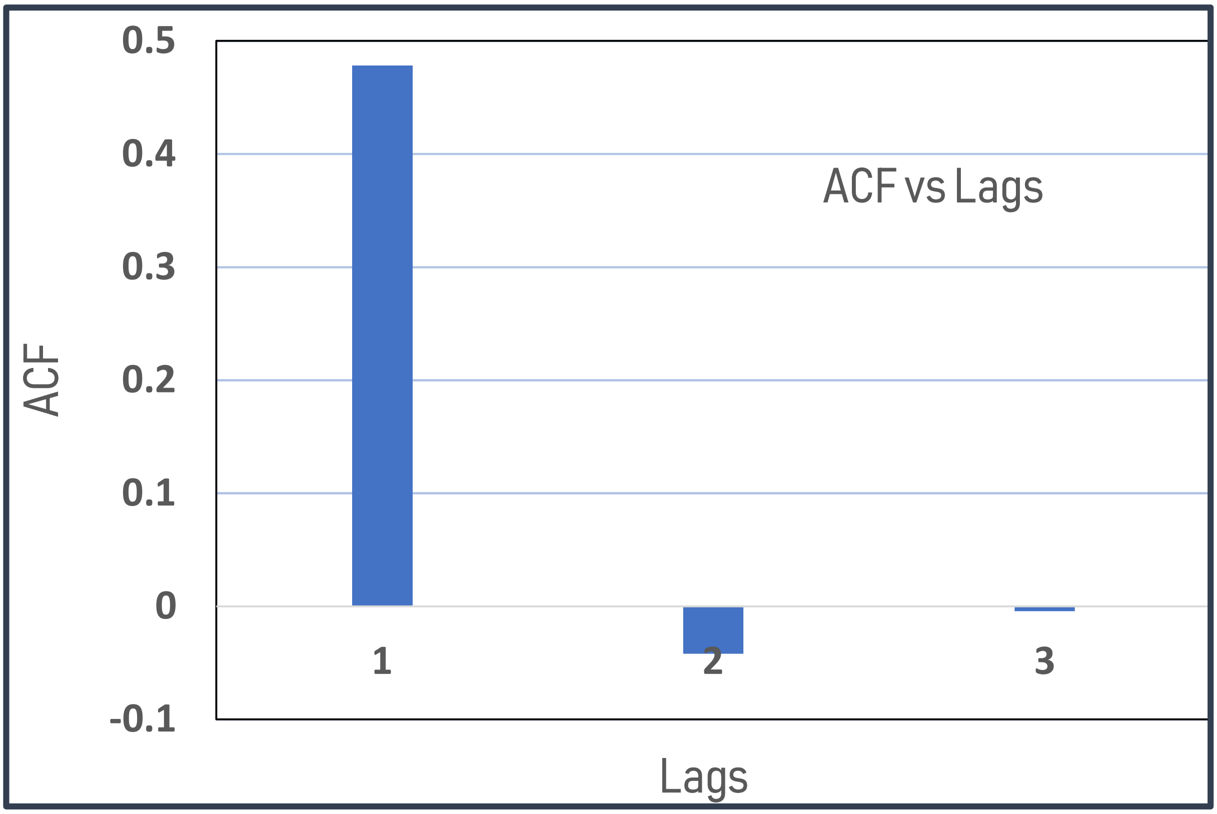
The ACF is calculated in Excel using the formula as

ACF = CORREL(OFFSET(<difference D_2>,0,0,COUNT(<difference D_2>)-<Lag value>,1),OFFSET(<difference D_2>,<Lag value>,0,COUNT(<difference D_2>)-<Lag value>,1))

The corresponding results obtained are:

Lags	ACF
1	0.47814258
2	-0.04191717
3	-0.00440235

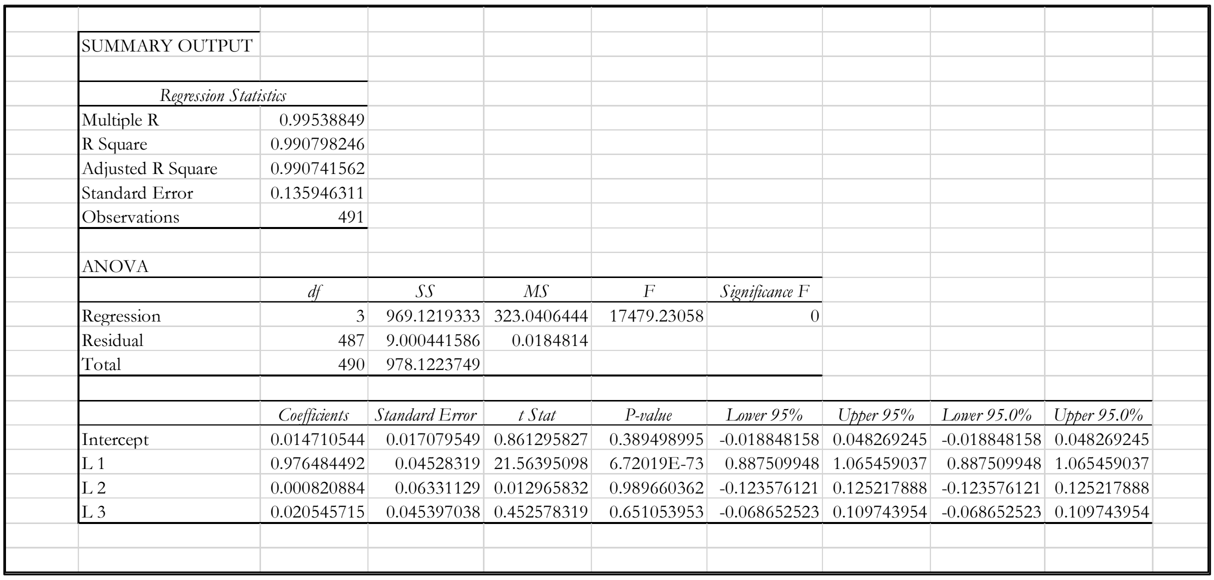
The resultant curve obtained is as given below,



Regression metrics

The regressions as obtained using the regression tool

1. Input = Lags L1, L2, L3 and Output = Ratio



SUMMARY OUTPUT

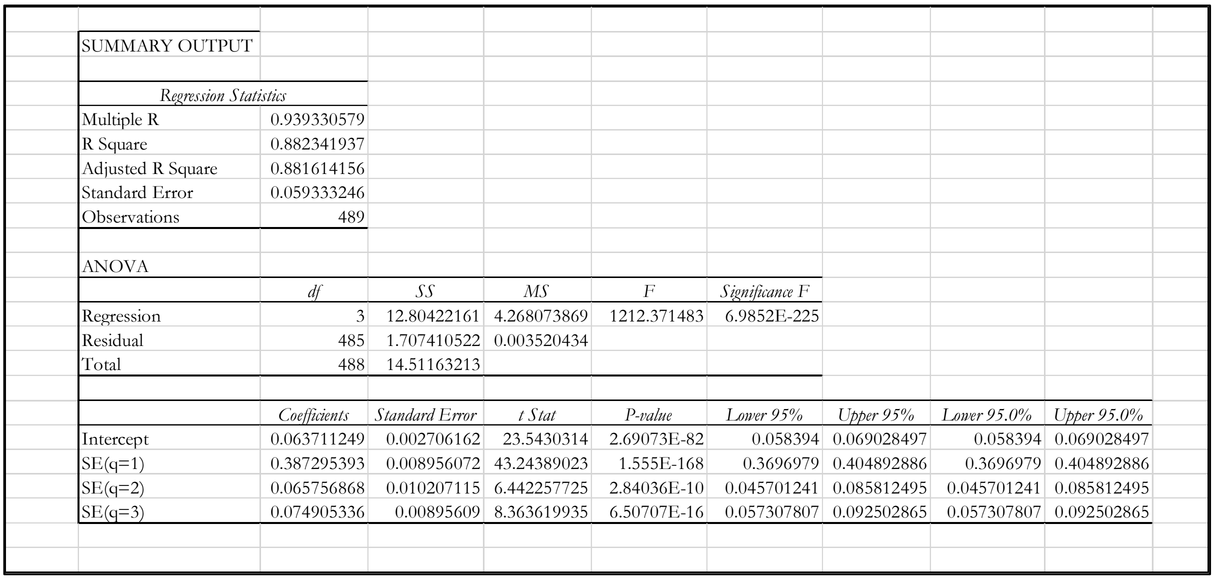
Regression Statistics	
Multiple R	0.99538849
R Square	0.990798246
Adjusted R Square	0.990741562
Standard Error	0.135946311
Observations	491

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	3	969.1219333	323.0406444	17479.23058	0
Residual	487	9.000441586	0.0184814		
Total	490	978.1223749			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	0.014710544	0.017079549	0.861295827	0.389498995	-0.018848158	0.048269245	-0.018848158	0.048269245
L 1	0.976484492	0.04528319	21.56395098	6.72019E-73	0.887509948	1.065459037	0.887509948	1.065459037
L 2	0.000820884	0.06331129	0.012965832	0.989660362	-0.123576121	0.125217888	-0.123576121	0.125217888
L 3	0.020545715	0.045397038	0.452578319	0.651053953	-0.068652523	0.109743954	-0.068652523	0.109743954

2. Input = Squared Errors SE(q=1), SE(q=2), SE(q=3) and Output = Error



SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.939330579
R Square	0.882341937
Adjusted R Square	0.881614156
Standard Error	0.059333246
Observations	489

ANOVA

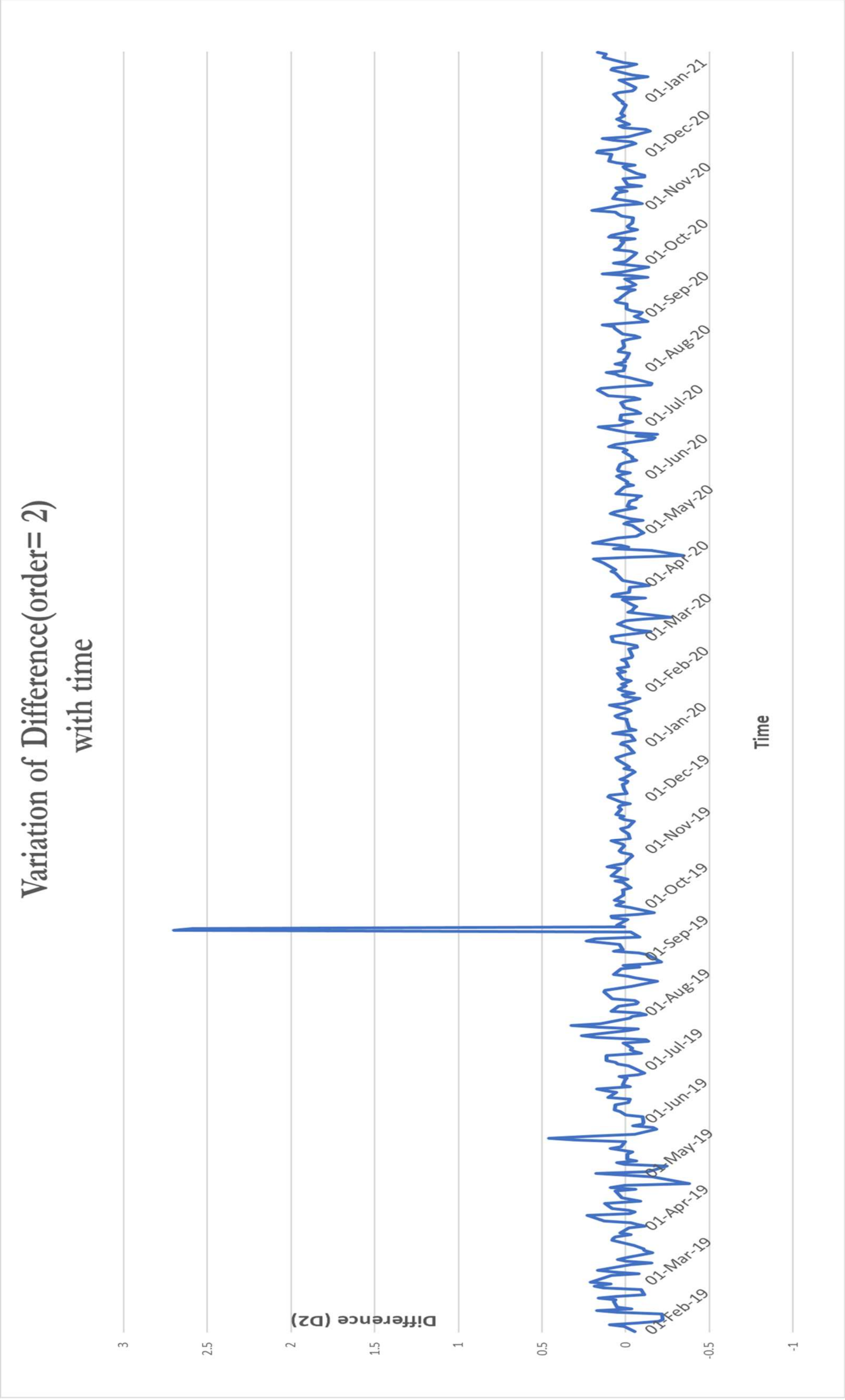
	df	*SS*	*MS*	*F*	*Significance F*
Regression	3	12.80422161	4.268073869	1212.371483	6.9852E-225
Residual	485	1.707410522	0.003520434		
Total	488	14.51163213			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	0.063711249	0.002706162	23.5430314	2.69073E-82	0.058394	0.069028497	0.058394	0.069028497
SE(q=1)	0.387295393	0.008956072	43.24389023	1.555E-168	0.3696979	0.404892886	0.3696979	0.404892886
SE(q=2)	0.065756868	0.010207115	6.442257725	2.84036E-10	0.045701241	0.085812495	0.045701241	0.085812495
SE(q=3)	0.074905336	0.00895609	8.363619935	6.50707E-16	0.057307807	0.092502865	0.057307807	0.092502865

Thus, we get the AR(p) model as = 0.014710544 + 0.976484492*L1 + 0.000820884*L2 +0.020545715*L3; and the MA(q) model as = 0.063711249 + 0.387295393*SE(q=1) +0.065756868*SE(q=2) +0.074905336*SE(q=3)

Curve of D_2 with time

Curve for the variation of D2 with time is given below.



Calculation of the Number of contacts

The values supplied is given in the table as

Static values supplied		
	Lot size	Future Contact Price
HDFC	550	1591.55
ICICI	1375	617.65

As per the equation in the problem statement

Number of Contract * Lot Size * FutureContract_Price_StockA = Number of Contract * Lot Size * FutureContract_Price_StockB

Thus, we set up the sides of the equation as

LHS:	No of Contacts of ICICI * 1375 * 617.65
RHS:	No of Contacts of HDFC * 550 * 1591.55

Given that we are at the situation where Solver operation failed, the solution is obtained from python code.

Python code for calculating Number of contacts

```
#!usr/bin/python
 #~ Code to define and solve a linear equation with triviality in solutions
 #~ in a manner so a to compute in a serial and crude way than use
 #~ mathematical solver functions of various packages as Numpy or Scipy.
#code
a = 550*1591.55 #raw initializes
b = 1375*617.65
# loop for calculating the values
for i in range(100000):
    for j in range(100000):
        # operative to calculate
        # i is the icici operative and j is the hdfc operative
        c = b*i - a*j
        # checking suitability of solution
        if(abs(c)<25):
            print(c,"\t",i,"\t",j)


# end of code
```

Solution obtained from the python code

The solution obtained from the python code is collected as

LHS - RHS	Absolute (LHS - RHS)	No of contacts	
		ICICI	HDFC
0.00	0	0	0
13.75	13.75	30841	29922
-13.75	13.75	32821	31843
0.00	0	63662	61765
13.75	13.75	94503	91687
-13.75	13.75	96483	93608

Where we find the minima at 0 for number of contacts ≠ 0 at both HDFC and ICICI ends.

Required Metrics

The required metrics as a solution for the problem statement is given in sequence as was asked for

ARIMA (p, d, q)	
Seasonality Orders	**values**
Auto Regression (p)	**1**
Differences (d)	**2**
Moving Average (q)	**2**

Performance Metrics (RMSE)		
MSE	**RMSE**	
0.036450244	**0.190919471**	

Number of contacts		
Minimum difference found	0.00	
No of contacts for ICICI	63662	
No of contacts for HDFC	61765	

Speed of Mean Reversion Calculations		
kappa [ϰ]	1	
ln(2)	0.693147181	
Speed of Mean Reversion ≡[ln(2)/kappa [ϰ]]	0.693147	

Conclusion and Inferences

Owning to the information imparted to us by the metrics and its related calculation with the model design we can infer and/or conclude the various points

1. With a detailed knowledge of the year 2020 which just passed by and the massacre it created along with the details of the curves we obtained in the beginning, we can quite well say that the way the year fared and the lockdown created a new seasonality to the curves of the LTP which corresponds to the dates around the beginning to the middle of March, 2020 when the countrywide lockdown was announced in India.
2. From the LTP curves of HDFC we can visibly detect a sudden anomaly at around days from 16th to 19th September, 2019.
3. From the curve D_2 vs time we can conclude vaguely but with surety that the day of 17th September, 2019 was quite eventful since we got a very large spike. This also matches up with the fact we found in point (2).
4. Although out of scope for the problem addressed in this project analysis of events of the day in all fields of news is bound to reveal data or information of events.
5. This spike is also visible in the ratio curve which kind of translated to the D_2 vs time curve. But comparing all curves HDFC bank had faced some extraordinary events during the time period from 16th to 19th September, 2019.
6. Looking back into how we completed the job, using Excel™ and the corresponding RMSE obtained we could have done it in a better way if specific software that are made to handle these kinds of jobs of handling

AR, MA, ARMA, ARIMA models are used like MATLAB, R, Python and its related packages like NumPy, SciPy, Scikit, Pandas, Quantax, PyTAF, etc., that are exactly built to handle tasks of solving these kinds of problems.

7. We could have taken a larger dataset than the one we have taken, since the number of datapoints is positively proportional to accuracy of a model or prediction generated from that datapoint. As lesser datapoints creates a bias in the model structuring.
8. In spite of restrictions to the usage of python or any programming language for that matter of fact (forbidden to use for calculating in the main portion of the project with dire consequences if disobeyed) we had to use python to conclude the equation and the solution of the problem of finding the number of contacts. Since the Solver module which was to solve the problem failed every way possible religiously.
9. As a final piece to this discussion, it would be incomplete if not pointed out the wide ranges of studies that could be done from these models created, most of which remained unexplored since those completely fell out of the scope of the problem statement explored in the project.

Bibliography

The following materials were referenced for bringing the project to fruition.

- https://www.investopedia.com/terms/d/data-analytics.asp
- https://web.archive.org/web/20201205235717/https://www.ibm.com/in-en/analytics/hadoop/big-data-analytics
- https://www.zarantech.com/blog/importance-of-data-science/
- https://www.import.io/post/business-data-analysis-what-how-why/
- https://www.analyticsvidhya.com/blog/2020/09/time-series-forecasting-ms-excel-exponential-smoothing/
- https://people.stat.sc.edu/wang528/Stat%20720/STAT720%20Notes.pdf
- https://www.stat.berkeley.edu/~bartlett/courses/153-fall2005/lectures/1notes.pdf
- https://www.stat.auckland.ac.nz/~ihaka/726/notes.pdf
- http://www.stats.ox.ac.uk/~reinert/time/notesht10short.pdf
- http://home.iitj.ac.in/~parmod/document/introduction%20time%20series.pdf
- https://www.python.org/doc/

- https://towardsdatascience.com/8-reasons-why-python-is-good-for-artificial-intelligence-and-machine-learning-4a23f6bed2e6
- https://www.wired.com/story/python-language-more-popular-than-ever/
- https://brochure.getpython.info/media/releases/python-brochure-current
- https://www.simplilearn.com/why-python-is-essential-for-data-analysis-article
- https://www.investopedia.com/terms/n/national_stock_exchange.asp
- https://math.unm.edu/~ghuerta/tseries/week6_1.pdf
- https://colab.research.google.com/notebooks/intro.ipynb#scrollTo=5fCEDCU_qrC0

* * *

Complete collection of the project files is safely kept at

https://github.com/WolfDev8675/RepoSJX7/tree/Assign2_2

* * *