A rough and ready guide to

Algorithmic Trading

Vivek Krishnamoorthy
Ashutosh Dave

A QuantInsti® Publication

A rough and ready guide to Algorithmic Trading

Version 1.0

Vivek Krishnamoorthy
Ashutosh Dave

A QuantInsti Publication

January 2020

CONTENTS

Preface

The trading industry, like virtually every other industry in sight, has gone through a drastic technological shift in the last few decades. More people now have access to the markets than ever before. However, succeeding consistently in the financial wild is a different story.

With the advent of quantitative trading, it is imperative that we traders, whether greenhorns or seasoned players, whether institutional or retail, get a wide understanding of the modern financial marketplace. In order to do that, using contemporary tools and adding a quantitative dimension to our trading style is essential.

It is our endeavor here at QuantInsti to bring the knowledge and tools to anyone who wants to learn about and be a part of the algorithmic and quantitative trading industry.

We hope that this book will serve as an introductory guide for such curious readers and inspire them to take their first steps towards it.

What Is This Book?

The backstory first:

Until mid-2019, we had a collection of essays on quantitative trading compiled into a book titled 'A Beginner's Guide to Learn Algorithmic Trading'. It was well-received, but we felt that it didn't go far enough or deep enough. As content creators in the domain that literally justifies our existence, we had a lot more to say. So, we took some parts of our older book, added a lot more updated and relevant material to weave it together into a (hopefully!) coherent story. And that's what this book is, really.

The book provides an initiation into the principles, practices and components of algorithmic trading. It also discusses the career pathways to be a part of this industry.

Who Is This Book For?

This book has been written for anyone who wants to learn about the field of algorithmic trading. From our experience, we imagine that our readers would be

- University students,
- Technology professionals,
- Retail traders of different hues (e.g. professional traders, or hobbyists who like to actively manage their personal portfolio),
- Anyone eager to know more about applied quantitative finance

What Are the Prerequisites?

We write assuming our readers do not have a background in programming. While an understanding of finance, mathematics or computer science is not necessary, having a moderate grasp on any/some/all of them will make this book an easier read.

Book Structure

We first introduce the reader to the domain of algorithmic trading by briefly exploring its history and then its terminology. We then proceed to discuss the pros and cons of automated trading. Further, we elaborate, with illustrative examples, on the components needed to create a robust trading system. We also briefly cover some key algorithmic trading strategies. to give you a taste of what's in store for the more interested among you. We dwell on the skill sets you need to build a career in this domain or to start your own desk. Finally, we close out our work with a recommended reading list and resources for diving deeper.

What This Book Is Not

We do not discuss advanced algorithms or quantitative strategies in any measure of detail; our aim in this book is more modest viz. to give you a taste of the quantitative way of trading. We also do not teach any programming here. Instead, we will shamelessly self-promote and point you to the book on Python programming co-written by one of us (Vivek Krishnamoorthy) if that's what you're looking for. Or many other interesting resources (like blogs/webinars/free courses) on the QuantInsti portal.

Copyright License

Acknowledgments

We owe a great deal to many pioneers from the financial markets and trading community for taking time out of their busy lives and reviewing our work at short notice. Their comments on our book draft have immensely helped us improve its quality. We include the Twitter handles of each of them in parentheses here.

Robert Carver (@investingidiocy) and Wayne Himelsein (Twitter handle @WayneHimelsein) conducted an extraordinarily thorough review and pointed out typos, errors and inconsistencies in certain sections. We found their comments to be extremely helpful.

We were stoked to receive suggestions and words of encouragement from Larry Tabb (@ltabb), James Long (@CMastication) and Tom Basso (@basso_tom).

Harry Tucker's (@harrytucker) feedback made us reconsider and eventually change the title of our book to reflect its contents more accurately.

The Options Cafe (@future_nifty), Andreas Clenow (@clenow), R Balakrishnan (@balakrishnanr), Alok Dharia (@alok_dharia), Dave Bergstrom (@dburgh), Brian (@quantfiction), wantonwallet (@wantonwallet), Alok Churiwala (@alokgbc), Suri Duddella (@surinotes), Matt Davio (@MissTrade), fiquant (@fiquant), Subhadip Nandy (@SubhadipNandy), Rob Smith (@robintheblack), and Tim Racette (@eminimind) all provided valuable feedback on the draft.

We are also grateful to our friends and colleagues from QuantInsti and iRage whom we mention below.

For old content which we are reusing: Many colleagues and ex-colleagues from iRage & QuantInsti

For creating the book design and cover: Paresh Ramtekkar and Shaival Diwan

For proofreading and helpful comments: Rekhit Pachanekar, Viraj Bhagat and Smiti Khandelwal

For overseeing the digital marketing and distribution of this book to far-flung corners of the internet:

Anupriya Gupta and the Digital Marketing team at QuantInsti

For showing patience to ignore missed deadlines: Nitesh Khandelwal

Our other colleagues at QuantInsti deserve a mention here for creating a wonderful atmosphere at our office with a mix of their great lunches (which we mooched on), their bad jokes (which we bore the brunt of) and their infectious energy and enthusiasm (which we fed off).

Finally, we are thankful to the hundreds of students and traders (often the same people) we regularly interact with that helped shape this book.

Vivek's acknowledgements:

I would like to acknowledge the love and support of my family - my parents, Krishnamoorthy and Vanaja, my brother, Vishwesh, and my spouse Neha's parents, Drs. Rupa and Nitin Pandit.

They often ask me about the field that I work (read as 'am obsessed with') in and would like a detailed answer.

The book you're reading is that answer, and I'd like to dedicate it to them. I can now show them the book and shame them into reading it. :)

A special shout-out to my friend, Anirban Sanyal ("Ban") whose suggestions have benefited me in ways that are hard to express on paper (and I hope he knows how much it has).

And last, but by no means the least, I'd like to thank my spouse, Neha, who puts up with my atrocious work habits (at home) among other things. She mostly does this with affection and a lightness of heart that I cannot fathom. I love her more than I think she realizes and certainly more than I say it. There. I've said it in a book in front of my readers.

Ashutosh's acknowledgements

I would first like to thank my family, especially my parents, Dr Shubhada and Deepak, my sister, Radhika and brother-in-law, Devesh for the continuous love and support they have given me through thick and thin; I could not have done it without them.

Second, I would like to thank my dear wife, Deepika, for all the love, comfort, and guidance she has given me. I would like to dedicate this work to her.

I want to thank Nitesh Khandelwal for creating and leading the excellent knowledge-sharing platform that is QuantInsti. This book would not be a reality without it.

I want to express my gratitude to Vivek Krishnamoorthy, for his incredible advice and guidance (academic and otherwise) through the course of writing this book.

Lastly, my fellow content creators at QuantInsti needs to be acknowledged for making my experience in QuantInsti truly pleasurable, for offering continuous assistance and for enriching me as a member of the community.

About the Authors

Vivek Krishnamoorthy is the Head of Content & Research at QuantInsti. He teaches Python for data analysis, building quant strategies and time series analysis to our students across the world. He comes with over a decade of experience across India, Singapore and Canada in industry, academia and research. He has a Bachelors' in Electronics & Telecom Engineering from VESIT (Mumbai University), an MBA from NTU Singapore and a Graduate Certificate in Public Policy from The Takshashila Institution. You can reach out to Vivek on LinkedIn.

Ashutosh Dave is a quant researcher with more than a decade of experience in financial derivatives trading and quant finance. Currently, he leads a team of quants in a prop trading firm engaged in alpha research and strategy development. Previously, he has worked as a derivatives trader specializing in trading fixed income and commodities with a proprietary trading firm in London where he worked for several years before relocating to India where he later worked as a Senior Associate, Content & Research at QuantInsti. His key areas of interest include applying advanced data science and machine learning techniques to financial data. Ashutosh holds a Masters in Statistics with distinction from the London School of Economics (LSE) and is a Certified FRM (GARP). You can reach out to Ashutosh on LinkedIn.

1 A Really Brief History of Financial Trading

As of 2018, an estimated 60% to 80% of daily traded US equities (by volume) on average were accounted for by Automated Trading.[1] Algorithmic trading accounts for more than a third of the total volume on Indian cash shares and almost half of the volume in the derivatives segment.[2]

In this chapter, we will take a little peek at the history of financial trading and at the events that shaped the current trading and investing landscape.

1.1 The Beginnings: Setting up of the Exchange

To start from the very beginning of trading history, we go back four centuries to 1602. The secondary market for VOC (Dutch East India Company or Vereenigde Oost-Indische Compagnie) shares started in the first decade of the seventeenth century. The Dutch East India Company in 1602 initiated Amsterdam's transformation from a regional market town into a dominant financial center. With the introduction of easily transferable shares, within days buyers had begun to trade them. Soon the public was engaging in a variety of complex transactions, including forwards, futures, options, and bear raids, and by 1680, the techniques deployed in the Amsterdam market were as sophisticated as any we practice today.

New asset classes began to be traded over time encompassing stocks, bonds, currencies and commodities. By the eighteenth and nineteenth century, these practices spread across continents and into the major financial capitals of the world.

1.2 The Quest for Faster Access to Information

The speed of getting news about firms and geopolitical events has always mattered. This only gained more importance as trading moved to exchange 'pits' in an 'open outcry' setup.

This setup consisted of brokers and traders being physically present in the pit and shouting prices at which they were willing to buy and sell. Participants used hand signals to convey their intentions to other traders and execute the trades.

[1] CNBC report: https://www.cnbc.com/2018/12/05/sell-offs-could-be-down-to-machines-that-control-80percent-of-us-stocks-fund-manager-says.html

[2] NIFM study on Algorithmic trading in Indian capital markets.

1.3 Growth of Financial Markets in the Twentieth Century

The story of the financial markets is the story of the changing economy. The 'open-outcry' model gradually began to give way to telephone trading and eventually to electronic trading.

Computerization of the order flow in financial markets began in the early 1970s, with some landmarks being the introduction of the New York Stock Exchange's "designated order turnaround" system (DOT, and later SuperDOT), that routed orders electronically to the proper trading post, to execute them manually. The "opening automated reporting system" (OARS) aided the market specialist in determining the market clearing opening price (SOR; Smart Order Routing).

In 1981, Michael Bloomberg, who was a general partner of Salomon Brothers, was given $10 million as partnership settlement. Having designed in-house computerized financial systems for Salomon Brothers, Bloomberg built his own Innovative Market Systems (IMS). Merrill Lynch invested $30 million in IMS to help finance the development of the Bloomberg terminal computer system and by 1984 IMS was selling machines to all Merrill Lynch clients. This led to the development of the famous Bloomberg terminal that is being widely used by traders till date.

1.4 The Start of Algorithmic Trading

Financial markets with fully electronic execution and similar electronic communication networks developed in the late 1980s and 1990s. In the U.S., decimalization changed the minimum tick size from 1/16 of a dollar (US $ 0.0625) to US $ 0.01 per share. This encouraged algorithmic trading as it changed the market microstructure by permitting smaller differences between the bid and offer prices. It also resulted in a decrease in the market-makers' trading advantage and increased market liquidity.

By 1998, the US Securities and Exchange Commission (SEC) authorized electronic exchanges paving the way for computerized High Frequency Trading (HFT). HFT was able to execute trades more than a thousand times faster than a human.

1.5 The Boom of High Frequency Trading (HFT)

In the early 2000s HFT accounted for less than 10% of equity orders, but this has grown rapidly. According to NYSE, HFT volume increased by 164% between the years 2005 and 2009.[3] The global algorithmic trading market size will grow at estimated 10% CAGR during 2018-2022, according to a study by Technavio.[4]

The year 2011 took the latency game in trading to another level. A firm called Fixnetix developed a microchip that could process orders in 740 nanoseconds (one nanosecond is one billionth of a second).

A $300 million transatlantic cable was built in 2015 just to shave 0.006 seconds off transaction times between New York City and London.

[3] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3743071/

[4] https://www.technavio.com/report/global-algorithmic-trading-market-analysis-share-2018

With HFT came the concept of co-location. Co-location essentially means that computers owned by HFT firms and proprietary traders are kept in the same premises as the exchange's computer servers. This enables HFT firms to access price data a split second faster than other market participants. Co-location has become a lucrative business for exchanges, which charge HFT firms millions of dollars for the privilege of "low latency access".

To cater to the demands of the HFT industry, companies like *CoreSite* offer a service where traders can install "co-located" computers right in the heart of Washington DC.

The idea is to get access to federal data milliseconds faster than those traders waiting patiently for it to travel at the speed of light up the fiber optic lines to markets in New York, New Jersey and Chicago. All of it—the information's transmission, translation, and trading in a journey from Washington DC to market servers in New Jersey, New York and Chicago happens faster than the speed of human thought.

It takes a person 300 milliseconds to blink an eye. But the firms involved in this telecommunications arms race view a single millisecond as a margin of victory or defeat.

1.6 Use of Social Media and Twitter for Trading

By September 2012, information solution provider company *Dataminr* had launched a brand-new service to turn social media streams into actionable trading signals. This helped report the latest business news up to 54 minutes faster than conventional news coverage. The platform was able to identify several distinct "micro-trends" which helped clients predict what the world may soon be focused on. Some of these signals included – on-the-ground chatter, consumer product reactions, discussion shifts in niche online communities, and growth and decay patterns in public attention.

The monitoring of social media by the FBI and the virtually instant impact of social media reactions on security prices led the SEC to place restrictions on public company announcements through social media in April 2013.

Just two days after this, Bloomberg Terminals incorporated live tweets into its economic data service. Bloomberg now even has a tracker called Bloomberg Social Velocity (BSV) that tracks abnormal spikes in chatter about specific companies on social media.

A noteworthy example of the impact of tweets on stock markets is from 23rd April 2013. On that day, a tweet sent from a hacked *Associated Press* Twitter account said "Breaking: Two Explosions in the White House and Barack Obama is injured". Immediately the Dow Jones index plummeted 143 points (1%) in 3 minutes. *Associated Press* issued an urgent clarification about their account being hacked and the news being fake after which the markets recovered in six minutes.

In 2019, a note by Bank of America said that since 2016, there is a high correlation between negative stock market returns and the days on which US President Trump tweets frequently.[5]

[5] https://www.cnbc.com/2019/09/03/on-days-when-president-trump-tweets-a-lot-the-stock-market-falls-investment-bank-finds.html

2 Terminology

A critical part of understanding and engaging with any field of study is to learn its field-specific vocabulary. This is applicable to quantitative trading just as much as it is to medicine or quantum mechanics or a programming language. Once we understand the semantics of quantitative trading, we will be able to do two things. One, to read and appreciate books, articles and other works on it without getting too bogged down with unfamiliar jargon—two, to reason critically and communicate meaningfully with members of the quant trading community.

This chapter introduces some basic terminology and lays a foundation for subsequent chapters and your life in quantitative trading.

2.1 Algorithmic Trading and Automated Trading

Algorithmic trading (or simply algo-trading) is a method of trading where we use computer programs to follow a defined set of instructions or rules to calculate the price, quantity, timing and other characteristics of the orders. In other words, trading signals (buy/sell decisions) are generated based on a set of instructions.

In algorithmic trading, the trades can be placed into the market either manually or by semi/fully automating the order placement and execution process.

The key feature of algorithmic trading is the lack of human intervention in making trading decisions.

Automated trading (or fully automated trading) is a subset of algorithmic trading wherein computers are used to generate the trading signals and to manage the flow of orders in the markets without human intervention.

2.2 Quantitative Trading

Quantitative trading is any method of trading that employs mathematical/statistical techniques to make trading decisions (the decisions can be realized i.e. trades can be executed either manually or automatically). Cheap computing power helps us to apply such techniques and to thoroughly test our strategies on historical data rather than relying on our instincts. The frequency at which we trade can be high or low as per the strategy.

One of the major benefits of using a quantitative approach is the removal of subjectivity, as the decisions are based on quantifiable information.

Quantitative models can incorporate elements from both technical as well as fundamental analysis. For instance, one can design a quantitative model based on the cross-over of technical indicators such as RSI or Bollinger bands. Someone else with a good understanding of fundamental data such as stock earnings can design a quantitative model to incorporate these numbers. Examples of quantitative systems include various momentum, mean-reversion and statistical arbitrage-based strategies. We will see a couple of examples of a quantitative trading strategy later in this book.

More often than not, quantitative strategies are employed in an algorithmic or automated way. Additionally, the usage of these terms and the meanings ascribed to them by different writers have been a little uneven. So, the terms ‘algorithmic trading’, ‘quantitative trading’ and ‘automated trading’ are used interchangeably in many places including here.

2.3 High Frequency Trading (HFT)

High Frequency Trading (HFT) is a special category of algorithmic trading characterized by unusually brief position-holding periods, low-latency response times, and high trading volumes in a day. Algorithms are written so as to utilize trading opportunities which appear in very brief time periods as short as milli- or micro- seconds. The profit margin of each trade is small, which is compensated by fast speeds and large volumes.

To be competitive in the HFT space, the participants need to minimize the time it takes to send orders and execute trades, also called ‘latency’. This means heavy investments in the infrastructure in terms of hardware, network connectivity and co-locating the servers in the exchange premises. Co-location means that your server is in the same premises and on the same local area network as that of the exchange. Most exchanges provide colocation facility nowadays.

Higher costs of HFT presents barriers that are simply too high for some retail traders. This type of trading is particularly popular with trading and market-making firms with substantial budgets at their disposal.

2.4 What is a Trading System?

A trading system, more commonly referred to as a ‘trading strategy’ is nothing but a set of rules, which when applied to the given input data generate entry and exit signals (buy/sell).

When talking about a trading system, we try to be specific and answer the following questions with respect to entering and exiting a trade:

What and Why to Buy or Sell?

There are thousands of financial instruments available to be traded in different exchanges.

For example, if a trader wants to be invested in the technology sector, she may decide to buy certain individual stocks in that sector or buy a technology index. The same could also be achieved by buying derivatives such as options and futures on the stocks of technology companies.

All these different instruments carry different risk and reward profiles. One has to select what to trade in, based on prior knowledge, risk/reward expectations and risk appetite.

When to Buy/Sell?

One has to specify the conditions under which a trade will be entered into. The conditions and timing have to be clearly specified before trading starts. For example, a trader might decide to buy the S&P 500 index at the start of the fourth day, if it has closed negatively for the previous three successive days.

How Much to Buy/Sell?

This is called position-sizing and is a key element of overall risk management. This is determined based on factors like investment criterion (ex. not more than 10% of our capital will be invested in any one sector), Kelly criterion (a money allocation technique), etc.

What Price or What Time to Take Profit?

The price at which we will take profit if the trade goes as per plan is called the 'take-profit' price (TP). This price is decided by taking into consideration the amount of risk taken in the trade. In addition, we can also specify the time at which we book the profit.

Stop Loss Price

If the trade does not work out as expected and moves adversely, then at what price will we decide to book a loss and get out of the trade? This price is called the Stop loss (SL) price and is another important element of risk management.

2.5 Quants, Traders and Market Makers

Creating a profitable trading strategy requires considerable work, and the ones involved in it are known as **'quants'** in the quantitative trading world. Within an algorithmic trading set-up specifically, we can define a quant as someone who applies mathematical models to create alpha-seeking strategies.

By an alpha-seeking strategy, we mean a profitable trading strategy that can consistently generate returns over and above the risk-adjusted returns. To the layperson, quantitative trading strategies can appear inscrutable and hence algo trading is often called 'black box' trading.

Quants comes in different stripes depending on the role and the organization they work for. They are usually employed to create, validate and optimize different pricing and risk management models by investment banks, hedge funds and other financial institutions.

There are many trading styles and hence many types of **traders**. For example, a technical trader is one who relies on past patterns in the data to make trading decisions whereas a fundamental trader bases her decisions on fundamental factors such as the quality of a firm's management, its earnings etc. Manual or discretionary traders execute their trades themselves based on a mix of trading rules and experience. Proprietary traders (or prop traders), in general, are traders who use the capital of a proprietary trading firm or a brokerage firm to trade and take a cut from the profits. Retail traders usually sit in the comfort of their homes and take positions in the markets.

A key market participant in an exchange's trading structure is the **market maker**.

Market makers are agents who stand ready to buy and sell securities in the financial markets. Other market participants are therefore guaranteed a counterparty for their transactions and this increases the overall liquidity in the market. This is called ‘liquidity provision’ and is looked at as a service provided by the market-makers to the exchange.

Market makers profit by charging higher offer prices than bid prices. This difference is called the spread. The spread compensates the market makers for the risk inherited in such trades. To be clear, the risk they ride is the price movement against their own held position.

For example, a market maker may purchase 1000 shares of IBM for $100 each (the ask price) and then offer to sell them to a buyer at $100.05 (the bid price). The difference between the ask and bid price is only a nickel, but by trading millions of shares a day, she manages to pocket a significant chunk of change to offset her risk.

Traditional market makers are usually under contractual arrangements with the stock exchange to provide these services. This increases liquidity and makes that exchange an attractive platform for transacting. High frequency traders can also act as market makers and play a vital role in the overall ecosystem.

Market makers are known by different names. Earlier in the eighties, market makers were called jobbers on the London Stock Exchange. On the New York Stock Exchange, they were known as 'specialists’ earlier but are now referred to as Designated Market Makers.

A high frequency trader could also be a market maker and perform quantitative analysis at the same time. So, a trading strategy could be using quantitative methods, but the trader may be executing it using automated systems or semi-automated systems or manually.

There can be a significant overlap or distinction in terms of who does what while trading and hence knowing the context is important.

3 Why Go Algo: The Case for Algorithmic Trading

A natural question that crosses the minds of budding and even expert discretionary traders is why one should choose automation and develop quantitative trading skills.

In this chapter, we will discuss the advantages and drawbacks of algorithmic/automated trading and answer the above question.

3.1 Advantages of Automation in Trading

Speed

Even a skilled trader will take at least a couple of seconds to place an order. In the age of machine trading and HFT, that's a lot of time in which the price can move significantly. The computer will have placed and closed hundreds of orders in that time frame. Human limitations on speed and accuracy can cost tremendous opportunities.

Accuracy

Machines are accurate every single time it comes to dealing with operational aspects of trading. For example, while filling in the order details, humans can commit errors due to loss of concentration or other factors. Some of the most significant losses in trading history have happened due to what traders call a 'fat-finger', which is just another name for sending in an incorrect or unintended order ticket by mistake.

Scalability and Up-Time

Given the vast amount of computing power available today, we can run multiple strategies on thousands of instruments simultaneously. Computers can simultaneously scan thousands of signals for trade opportunities. However, for us humans, it is nearly impossible to focus on many assets at the same time for long durations due to natural constraints. A human trader suffers from mental and physical exhaustion and needs rest. Compare this to a computer which can work for days and months with no downtime.

Trading Minus Emotions

Machines do not have emotions (at least not yet!), and we can use that to our advantage. In manual trading, this is a huge detriment. Fear and greed often prevent human traders from doing what needs to be done. Machines don't cloud their decisions based on any emotional factors as they just follow the preset rules that we've programmed.

Backtesting

Automation is not only limited to the execution of trades but is widely used for validating strategies. Any strategy used in live markets is tested and tried on historical data to evaluate its performance. This is called backtesting the strategy. Backtesting gives vital feedback about the past performance of the strategy.

3.2 Drawbacks & Constraints

Like with any approach, automated trading also presents a few challenges for its users.

Now let us discuss some drawbacks and constraints of automated trading to get a fuller picture.

The Skill Gap

The skills required for algo trading can act as a barrier to enter the domain. It takes time, effort and mentorship to develop and update these skills. Apart from programming, an algo trader needs to be at ease with other facets of the field like market microstructure, math and statistics (more on this later), machine learning and computer architecture.

Infrastructure Costs

The infrastructure required for an automated setup can act as the other barrier to entry.

High performance computers, robust network connectivity, power backups, and maintaining reliable hardware including servers can lead to substantial overheads. For HFT, the costs are even higher due to the additional cost of colocation, leased line & other types of network connectivity, etc.

With the advent of cloud computing and affordable managed private servers, these costs are dropping. However, this is an overhead one should be cognizant of.

Mental Makeup

A lot of traditional discretionary traders are habituated to controlling their trades or acting on their instincts. They are unable to trust a computer program to do the job. This makes it tougher for them to make the transition.

Some think that algo trading involves just setting it all up and going on a vacation. With a heavy heart, we must inform you that this is a misconception, as an algo trading system requires considerable oversight. An algo trader still must monitor the performance of the algorithm periodically and needs to tweak it if the market conditions change significantly. It is essential to monitor any major deviations between the model and the actual performance and to ensure that the system is behaving as expected.

Underestimating the Role of Humans in an Automated Trading Environment

Given the advantages of automated trading, one might conclude that machines have made humans completely obsolete in an automated trading setup, but this is far from the truth. Algorithmic trading is a means of efficient allocation of tasks to resources that are most capable of performing them.

Although humans can't beat computers in many tasks related to efficiency, speed or accuracy, there are certain tasks where humans are superior.

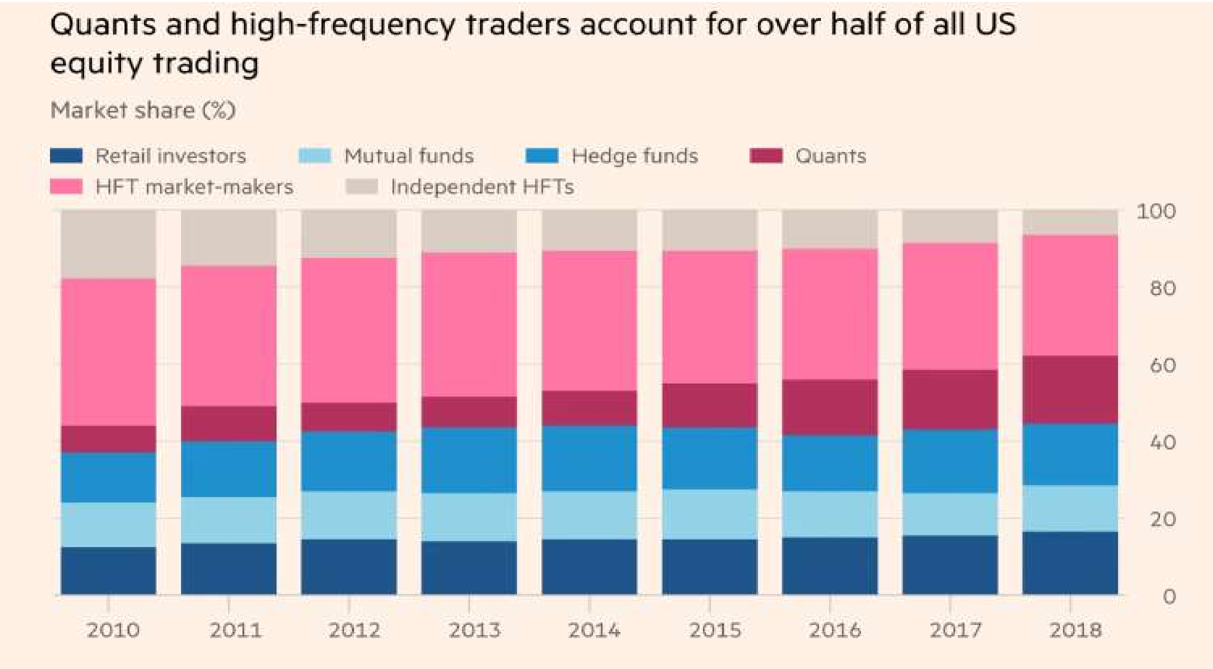
Humans have the creativity to look at the markets in ways a machine cannot. They can respond to situations which have not been anticipated or planned for and hence not programmed into the computer.

Impact on Financial Markets as a Whole

It has been observed and argued that the use of machines and algorithms in trading has exacerbated and amplified market movements which cause spikes in market volatility. Fortunately, the exchanges and regulators are always working to find new ways to handle such scenarios and ensure that markets operate smoothly.

3.3 Closing the Case: The Shifting Paradigm

Even though there are a few drawbacks, as outlined in the previous section, the advantages and benefits of using an automated approach more than outweigh them. In the twenty-first century, not utilizing the power of technology and quantitative analysis is not an option as one has to compete with market participants who do.



Source: Tabb Group © FT, as of Dec. 31, 2018

Various key players have widely adopted algorithmic trading in the financial markets, including:

- Major trading firms
- Brokerage firms
- Hedge Funds
- Multinational investment banks

This has had a domino effect on each other as these institutions are interconnected which makes it even more imperative for retail and discretionary traders to start acquiring quantitative and technical skills.

4 System Architecture of an Algorithmic Trading System

Before we start trading in an automated setup, let us understand how it operates.

This chapter can be a little heavy on technical jargon for some of our readers. We still recommend going through the whole chapter once and revisiting it occasionally to further your understanding.

We will first start by looking at the constituents of a traditional trading system.

4.1 System Architecture of a Traditional Trading System

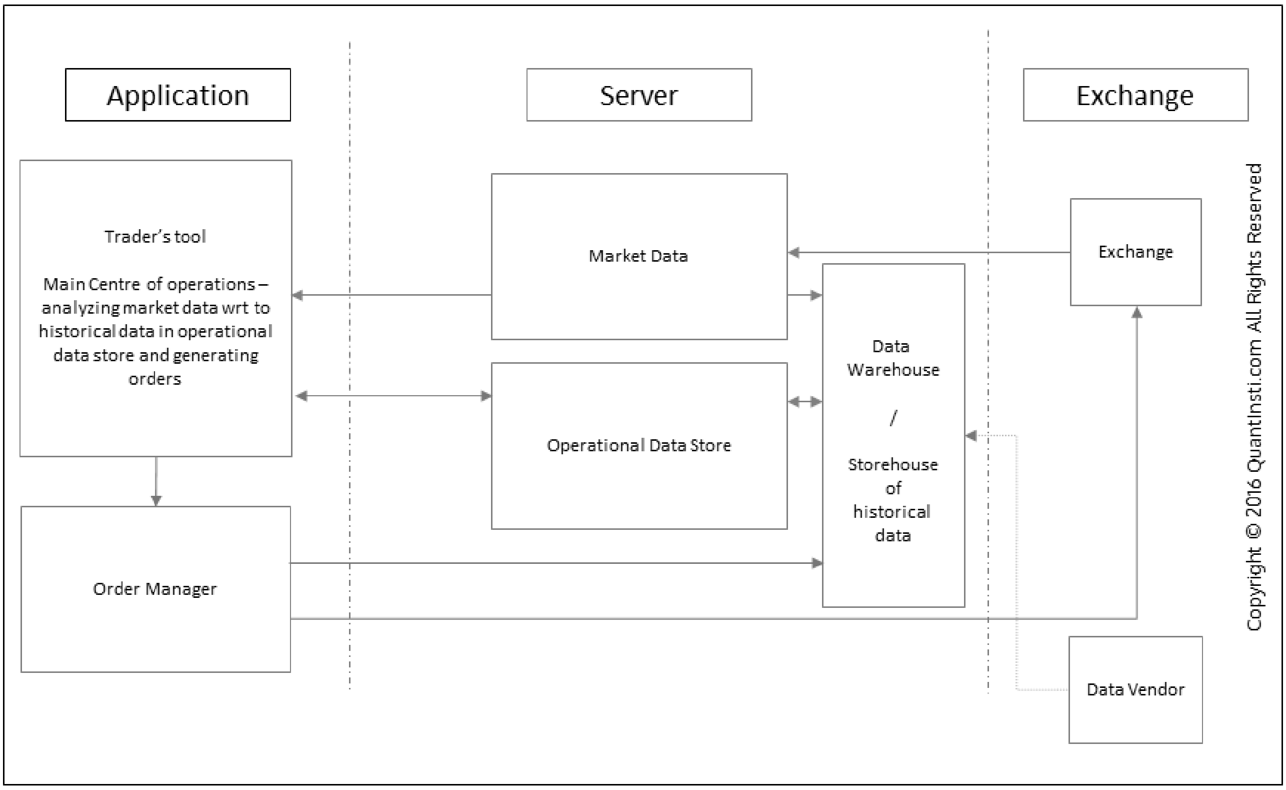
In a traditional trading system, the trader takes buying and selling decisions and also herself manages order flow at the trading application level.

The data from the exchange or external data vendors is received by the market data adapter (MDA) module. The market data that is received typically provides the system with the latest snapshot of the order book. It might contain some additional information like the volume traded until then, the last traded price and quantity for a stock. However, to make a trading decision, the trader might need to look at historical values or derive certain indicators from past data. For such needs, a conventional system would have a database to store past market data and tools to analyze it. It would have another database for storing the trading decisions. Lastly, there is a GUI interface for the trader to view all this information on the screen.

The entire trading system can be broken down into three parts:

- The exchange(s) – the external world
- The server
    - Market Data Adapter
    - Store market data
    - Store orders generated by the user
- The trader-side Application part
    - Take inputs from the user including the trading decisions
    - Interface for viewing the information including the data and orders
    - An order manager sending orders to the exchange.

Thus, the traditional system architecture would look like:



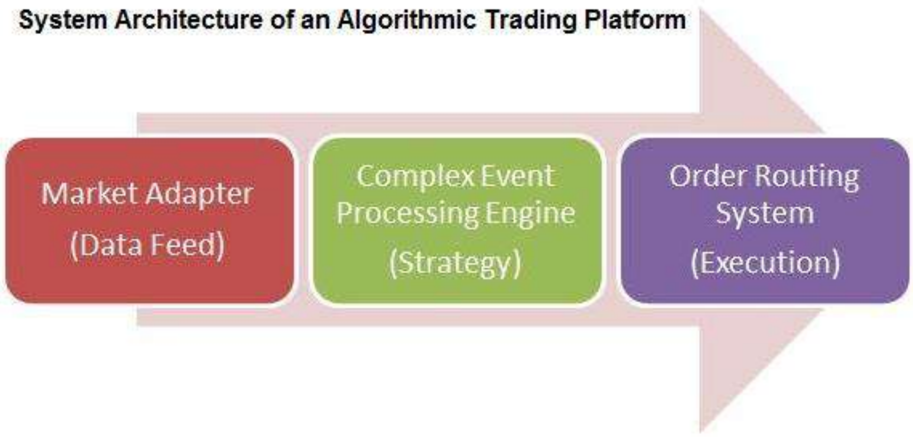
The architecture shown above is often sufficient to carry out low to medium frequency automated trading. The difference, however, is that the human trader is replaced by the machine, which makes its own trading and execution decisions.

4.2 Evolution of the Architecture of an HFT System

The traditional architecture cannot scale up to the needs and demands of automated trading in case of HFT, as the amount of data to be processed and the rate at which decisions are made increase exponentially. The latency between the origin of the event to the order generation goes beyond human control and enters the realm of milliseconds and microseconds.

So, there is a need to adapt and develop various parts of the architecture to reduce this latency and deal with the increased scale and complexity. Let us now see how this process unfolds.

Any algorithmic trading system, at its core, is a computational block that interacts with the exchange on two different streams, first, to receive market data from the exchange and second, to send order requests and receive replies from the exchange. All the computations related to the strategy must happen between these two interactions. Thus, any automated trading system consists of three major components which are shown in the diagram below:



Market Data Adapter (MDA) for Data Feed

The Market Data Adapter receives the data from the exchange or the data vendor in their own respective formats. The algorithmic trading system may not understand that format. The exchange provides an API which allows the user to program and create her own adapter to convert the data format into one that the system can understand.

Complex Event Processing Engine (CEP) for Strategy

Once you have the data, the next step would be the analysis that often involves calculations, comparisons with historical data and taking trading decisions. We will now see how this is achieved.

In the algorithmic trading system, we define a complex event as a set of incoming events that cause the order book to change. These include stock trends, geopolitical events, news or change in market conditions etc. The HFT algorithm would usually react to such events and place or cancel orders based on them. Complex event processing is performing computational operations on the incoming events in a short time. The operations can include detecting patterns, building correlations and relationships such as causality and timing between many incoming events.

In a fully automated HFT architecture, the decision-making moves from the trader side application to a complex event processing server (CEP). CEP systems process events in real time. The faster the processing of events, the better a CEP system is.

The two primary components of any CEP system are the CEP engine and the set of CEP rules. The CEP engine processes incoming events based on rules. These rules and the events that go as an input to the CEP engine are determined by the trading system (trading strategy) applied.

A separate calculation engine can be attached to the CEP to which it offloads computationally intensive calculations. For example, suppose a hundred different trading strategies are being run over a single market event. In this case there might be computations common to all of them such as the calculation of greeks for options. If each strategy were to run independently, each unit would do the same computation and wastefully using up processor resources. In order to cut redundancies of this nature, such computations are typically hived off to the said calculation engine.

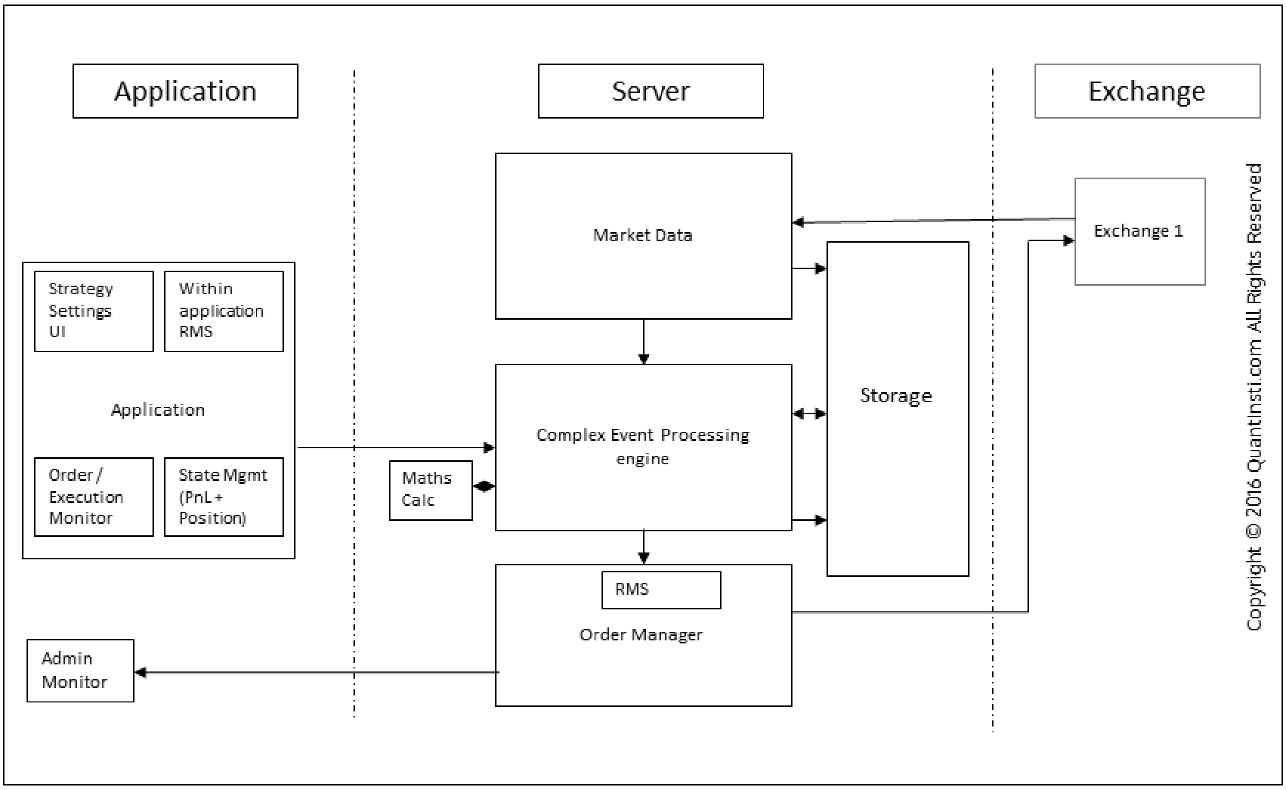
A quant will spend most of her time on the CEP system block formulating trading strategies, running backtests, optimizing parameters, and position-sizing among other things. This ensures the viability of the trading strategy in real markets.

No single strategy can guarantee everlasting profits. Hence, quants and algo traders are required to monitor and fine-tune existing strategies on a regular basis and to come up with new ones when needed to maintain an edge in the markets.

Order Routing System (ORS) for Execution

The signals generated by an algorithmic system can be either executed manually or automatically. When the signals are executed in an automated manner, we call the entire system an "automated trading system". Automation of the orders is done by the "Order Manager" module also called the Order Routing System (ORS) or the Order Management System (OMS). The ORS needs to be robust and capable of handling many orders per second. Since the time frame is so small compared to human reaction time, risk management also needs to handle orders in real time and in a completely automated way.

Let's now take a slightly different and more detailed view of the architecture of an automated trading system. We will then closely inspect some of its key blocks.



Risk Management System (RMS) in Automated Trading Systems

Although the trader-side application layer is primarily a view, some of the preliminary risk checks can be managed in it, especially those that are to do with sanity of user inputs like 'fat finger' errors. The rest of the risk checks are performed now by a separate risk management system (RMS) within the Order Manager (OMS), just before releasing an order.

Thus, there are two places where Risk Management is handled in algo trading systems:

1) Within the trader-side application – We need to ensure that wrong parameters are not set by the trader.

2) Before generating an order in OMS – Before the order flows out of the system we need to make sure it goes through some risk management system. This is where the most critical risk management check happens.

The New Role of the Trader-side Application

The Application part now, is little more than a user interface for viewing and providing parameters to the CEP, however basic risk management is also handled by it as mentioned above.

The Admin Monitor

The Admin monitor residing in the application layer gives an overall view and control of orders to the risk department or other stakeholders and is linked directly to the OMS. This helps the risk department to monitor the order size limits and act swiftly in case of breaches.

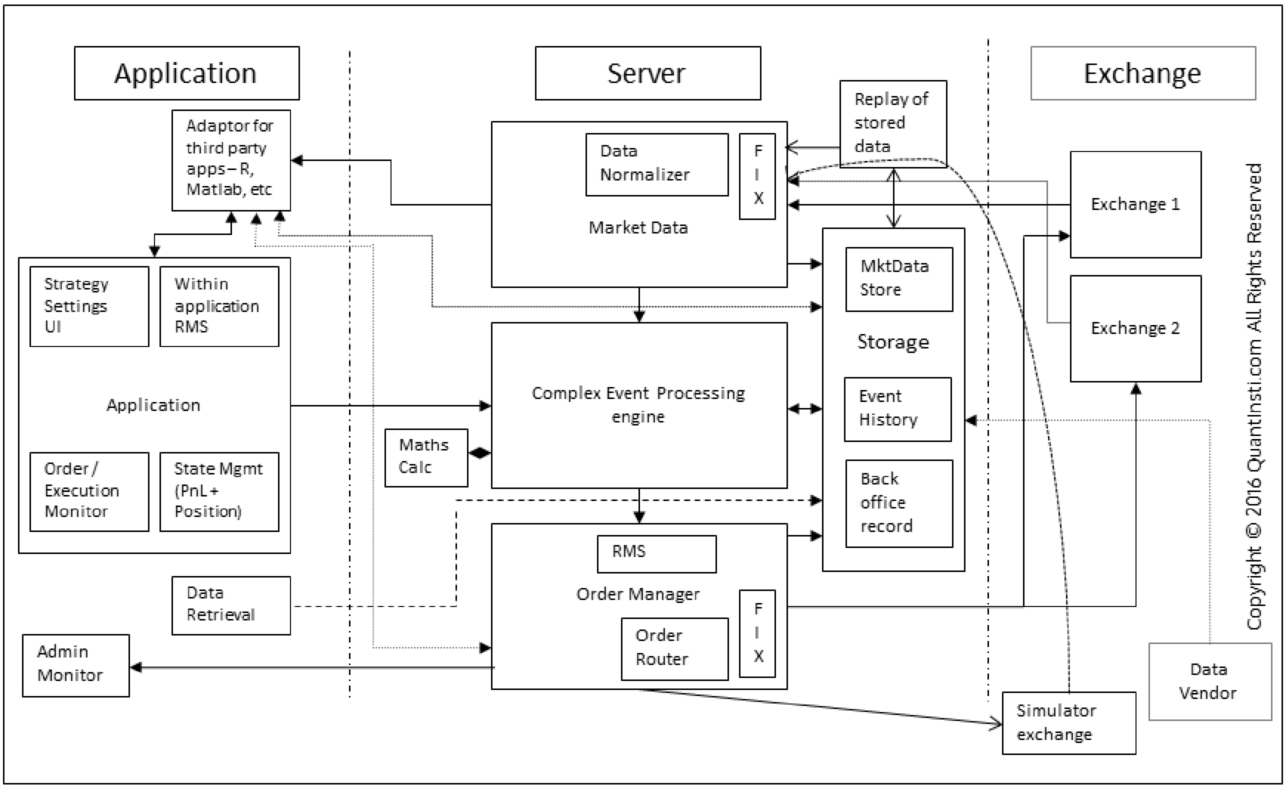
Protocol Adapters and Simulator

Once the CEP has decided what actions need to be taken (sending new orders, canceling existing orders, etc.), the OMS comes into action. A new order to be sent is encrypted in a format the exchange can understand, again using the APIs which are provided by the exchange. There are two kinds of APIs provided by the exchange: native API and FIX API. Native APIs are those which are specific to a certain exchange. The FIX (Financial Information Exchange) protocol is a set of rules used across different exchanges to make the data flow in security markets easier and effective. OMS can be designed to send orders to exchanges or non-exchanges and to be able to handle orders to different destinations.

The presence of standard protocols makes it easy to integrate with third party vendors, for analytics or market data feeds as well. As a result, the market becomes very efficient as integrating with a new destination/vendor is no more a constraint.

In addition, simulation becomes as easy as receiving data from the real market and sending orders to a simulator is just a matter of using the FIX protocol to connect to a simulator. The simulator itself can be built in-house or procured from a third-party vendor. Similarly, recorded data can be replayed with the adaptors being agnostic to data being a live feed or pre-recorded.

Incorporating these two aspects, our system architecture now looks like this:



We show above an indicative design of an automated trading system. Organizations create their own custom designs based on their requirements and the exchanges they are dealing with.

5 A Stepwise approach to Algorithmic Trading

Developing a robust and consistently profitable strategy is a multi-step process.

This chapter takes you through the various steps in the development of an algorithmic trading strategy.

5.1 Developing a Hypothesis

This stage involves gaining a deep understanding of the financial market/instrument to come up with a hypothesis to base your trades. You need to have/develop some knowledge-based edge in any market where you wish to invest and make risk-adjusted profits. Also, sometimes, traders have a hunch about a certain strategy being profitable in a particular market. In this case also, they can develop a hypothesis around it and scientifically test its efficacy.

To design a hypothesis which can be tested, one needs to determine the specifics such as

- The markets (Equities/ Equities Index/ Bonds/ FX or Commodities) and instruments (asset itself/ Futures/ Options etc.) to trade in.
- The rules or the strategy itself: These involve the trigger that will cause us to enter or exit a trade, the reasoning for setting stop loss and take profit levels.

For example, a trader might think that following certain rules, money can be made in the crude oil market. This might involve buying crude oil futures in one exchange based in London and selling them on another exchange based in New York. Such specifics need to be predetermined.

5.2 Formalizing the Strategy Programmatically

This is where we code the strategy using a suitable programming language. We translate the strategy into a set of logical statements that the computer can execute. An HFT trader is concerned about reducing the latency and would probably opt for a low-level programming language like C or C++. However, for a retail level algorithmic trader, a high-level language like Python or R serves her needs quite well. She could use some of the dozens of amazing free libraries available for both these languages.

In case you don't program, you can always hire someone to do it for you. It is also possible to use software tools that allow you to build your strategy, within defined complexity constraints, with minimal code writing. Such functionalities are increasingly being integrated within trading platforms, which may be offered to you by your broker or through a software vendor.

5.3 Backtesting

Backtesting is the process of validating our strategy by testing its performance on historical data. It helps us to gauge the past performance of the strategy. Backtesting also gives us the opportunity to optimize the strategy parameters.

Through backtesting, we can measure strategy performance based on certain metrics such as dollar PnL, percentage of profitable trades, Sharpe ratio (a measure of risk-adjusted returns), maximum drawdown (maximum fall in the value of the asset from a peak value), etc.

Algorithmic traders spend most of their time researching and backtesting their trading strategies using historical market data and other datasets as required by the strategy.

One of the aspects to be considered in backtesting is the 'backtesting window' or how far back in the past should we go to test our strategy. In the case of HFT, we generally don't go back to even the past 5 or 10 years of data because the market microstructure changes much faster. For example, if you are trying to backtest a simple strategy on data from 2007 when algorithmic trading was not allowed in India, then the results that we get will not be very useful. Typically, the backtesting window for high frequency trading strategies is short compared to the backtesting window for low frequency strategies that may range from a few years to few decades.

These days there are various platforms available which provide functionalities to perform backtesting on historical data. The important points to consider before selecting a backtesting platform are:

- knowing which asset classes and markets the platform supports,
- knowing about its sources of market data feeds and,
- figuring out which programming languages it supports.

Backtesting allows us to test our strategies before actually implementing them in the live market. However, the maxim 'past performance does not necessarily guarantee future returns' should be kept in mind while backtesting.

5.4 Demo Trading/Paper Trading and Parameter Optimization

A natural result of backtesting and validating is that it will either lead you to completely discard your hypothesis (happens very often!) or to extract actionable signals from the pool of data you started with.

In the latter case, our next goal is to optimize the strategy parameters. We need to keep in mind that our strategy should work well on out-of-sample data (new data or live trading data) and avoid 'overfitting'. Overfitting means that our strategy parameters have been fine-tuned to maximize profits on historical data that we used to build and backtest our strategy but will perform poorly on new or live data.

To avoid this issue, we need to forward-test our strategy on real market data (but NOT in the real market!) in a demo account. Most of the algorithmic trading platforms come with a simulated environment that you can use for this kind of forward testing. This is the same as doing paper trading using real or live market data. Paper trading involves simulating the performance of the strategy by noting down positions and PnL, instead of actual buying and selling. Once you're satisfied with the strategy performance, you can finalize it along with the optimized parameters.

5.5 Live Execution and Risk Management

Once you are satisfied with your algorithm, let it do its job in live markets! Deployment in the real time environment requires multiple aspects to be managed. Although the computer takes care of the execution part, an algo trader needs to keep an eye on the following aspects:

- ### Market Risk

  Monitor the performance of your algo continuously in terms of profit and loss (PnL). If the algorithm is not performing as expected, then you would need to review the logic used or tweak the strategy parameters.

- ### Operational Risk

  Operational factors such as the connection with the broker/exchange API and robust hardware play an important role in our success, so we should keep an eye on these factors and perform regular checks. We should plan to deal with all the things that can conceivably go wrong. For example, if the power goes out when a large number of orders are pending, then how should we go about handling this situation? Also, there can be a situation where an individual or a group of individuals try to manipulate the markets by sending orders that are declared to be illegal by the exchange. A good system must have risk checks for all such scenarios and raise flags/alerts in such cases.

- ### Regime Changes

  We have to keep an eye on the larger economy/sector for structural shifts, also called regime changes. In case of a regime change we might have to alter or scrap our strategy altogether. For example, say, a trader wants to test a strategy based on the notion that Internet IPOs outperform the overall market. If you were to test this strategy during the dotcom boom years in the late 90s, the strategy would outperform the market significantly. However, trying the same strategy after the bubble burst would result in dismal returns.

5.6 Continuous Research and Development

Financial trading is a competitive industry and some of the sharpest minds on the planet are engaged in this domain. Top financial organizations expend massive resources on developing their proprietary algorithms. The good news is that with the advent of data sciences, many such methods are becoming open source, i.e. they have become accessible to retail and small traders, especially the ones that are being created in Python. You can use such research tools and packages to come up with institutional grade ideas.

In short, we need to invest time, effort and resources in research and development to make our algorithms and systems more competitive and up to date. We should look to enhance and update both our domain knowledge and technical skills required to act on that knowledge/information.

6 The Elements of Algorithmic Trading

Success in algorithmic trading is determined not only by your quantitative skills but also the process and the tools you select for analyzing, devising, and executing your strategies.

In this chapter, we discuss some of the essential aspects one needs to take care of before starting to trade quantitatively.

6.1 Data Quality and Sources

The first and perhaps the most important element of algo trading is the data. A trader needs access to market data for validating the strategy / hypothesis, to back-test, and to execute the strategy in real markets.

It is essential that the data we get and use in our analysis is 'high quality' which means it should be:

- Consistent
- Reliable
- Unbiased

Trading and market data can be obtained from the broker and in some cases directly from the exchange.

Historical data for backtesting and analysis can be obtained from the exchanges, data vendors or from the financial portals that give access to historical market data.

For real-time market data, depending on geography and exchange, a trader can get it from the broker for free or at a cost. In case one can't get it from the broker, it can also be obtained from the exchange/vendors. Live market data from the broker or vendor can be streamed using their respective APIs. But one needs to check for the lag that might occur due to network latency. Also, some exchanges in certain geographies charge an additional fee (called exchange data fee) to access or stream market data.

Some data vendors provide data-feed only, while others provide charting platforms and other analytics for creating watch lists, tracking different markets, strategy development, generating buy/sell signals, etc. A trader can connect to such platforms with her broker's platform via a bridge, and have the orders executed. Data vendors usually list the broker partners on their websites, and the compatibility of their feed with different charting platforms.

Although there are a few sources for free financial data, one should use them with caution. There is the problem of data granularity. Most free sources provide daily price data only, whereas we might require data at much shorter time intervals to conduct our analysis and backtest our strategies.

Here is a list of some free and paid data vendors for reference:

Free Sources:

- Yahoo Finance

- Google Finance
- FXCM
- Quandl (free)

External Vendors (Paid Data):

- Bloomberg
- Global Datafeed
- Trading Economics
- Thomson Reuters
- Quandl (paid)

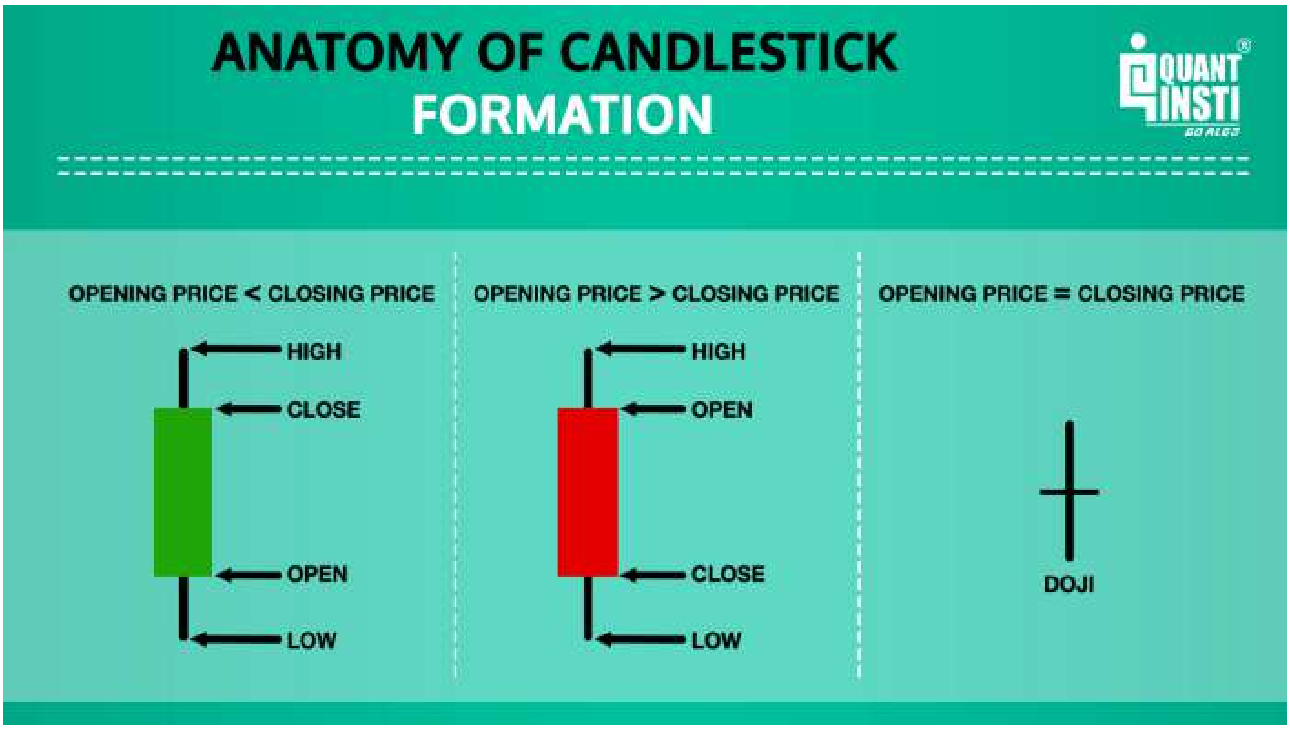
6.2 Data Formats

6.2.1 OHLC Data

OHLC is a standardized format in which historical or live data is provided by the vendors for backtesting and trading. OHLC stands for Open, Low, High and Close prices in a particular timeframe such as an hour or a day. In addition, some data sources also provide adjusted prices.

OHLC data can be easily visualized in the form of 'candles' for trading or analysis purposes. Each candlestick in a candlestick chart is used to describe high, low, opening and closing price movements of a security, derivative, or currency for a specific period.

The color scheme of the candle represents whether the price has gone up or down in the period for which candles are formed. For e.g. if we are looking at a chart of 5-minute candles for the price of a stock, then a green candle represents that the stock has closed at a higher price compared to the opening price in that 5-minute window. The color schemes may vary in charts from different sources.



6.2.2 Levels of Data for Trading

The market data that the exchange shares with market participants for trading typically contains the following basic information:

- Best bid price: highest price among the various participants bidding to buy
- Best ask price: lowest price among the various participants offering to sell
- Bid size: the total cumulative quantity at a quoted ask price
- Ask size: the total cumulative quantity at a quoted bid price

Level 1 data includes the Best bid and Best ask, plus the Bid size and the Ask size.

Level 2 provides market depth data up to 5 best bid and ask prices and level 3 provides market depth data up to 20 best bid and ask prices.

The following is an example of level 2 data:

5th Best Ask Price	Cumulative Quantity at 5th Best Ask Price
4th Best Ask Price	Cumulative Quantity at 4th Best Ask Price
3rd Best Ask Price	Cumulative Quantity at 3rd Best Ask Price
2nd Best Ask Price	Cumulative Quantity at 2nd Best Ask Price
1st Best Ask Price	Cumulative Quantity at 1st Best Ask Price
1st Best Bid Price	Cumulative Quantity at 1st Best Bid Price
2nd Best Bid Price	Cumulative Quantity at 2nd Best Bid Price
3rd Best Bid Price	Cumulative Quantity at 3rd Best Bid Price
4th Best Bid Price	Cumulative Quantity at 4th Best Bid Price
5th Best Bid Price	Cumulative Quantity at 5th Best Bid Price

6.2.3 Snapshot Data

The snapshot of the top 'n' buy and sell prices in each instrument are provided to all market participants every 't' time frame. The market participants are thus totally in the dark between two snapshots. However, if the time frame is very small like half of a second, it would be good enough for basic level of trading.

6.2.4 Tick-by-Tick Data

Tick-By-Tick (TBT) data includes each and every order or a change in the order. In case of TBT, information is provided immediately after the occurrence of the event. This type of data is generally used by arbitrageurs, market makers and high frequency trading firms/ institutions.

6.3 Brokers & Trading Platforms

The next aspect in algorithmic trading is choosing the right broker. Unless you have your own membership, you would need to access the exchange through a broker to trade.

Considerations that go into choosing the right broker include:

- Segments and instruments offered
- Brokerage cost
- Speed and reliability of the trading platform (if offered by the broker)
- Leverage and the margin requirements
- Compatibility of charting software with the broker's platform
- Gateway APIs offered by the broker

Some brokers provide APIs in Python and other programming languages to connect with them after authenticating your credentials.

As an algorithmic trader, you can execute your strategies in live markets via charting platforms that connect to your broker or through the gateway APIs offered. The available APIs are usually listed by the broker on their websites.

Some brokers offer platforms which are a set of simple HTTP APIs built on top of their exchange-approved web based trading platform. This enables users to gain programmatic access to data such as profile and funds information, order history, positions, live quotes, etc.

In addition, it enables users to place orders and manage portfolio at their convenience using any programming language of their choice (such as Excel VBA, Python, Java, C#).

Thus, for a prospective trader it is essential that she gets herself acquainted with the workings of an API and other relevant features offered by the broker's platform.

Some of the popular brokers and vendors for the Indian markets include:

- Interactive Brokers
- MasterTrust
- Presto ATS by Symphony Fintech
- Composite Edge
- Zerodha

External Platforms

Your broker might provide you with its own trading platform as discussed in the previous section. If that is not the case, then you can use an external trading platform and pay for it separately. These days there are various trading platforms available with advanced charting, analytics and backtesting features. Some popular trading platforms among traders include:

- NinjaTrader
- Trading Technologies (TT)
- AmiBroker
- TradeStation

Before subscribing to a platform, it is also vital that a trader understands the pricing policy, as these platforms in addition to the software charges can also charge for data-feed, exchange fees, and third-party add-ons separately. A trader should choose a platform based on her trading style, features and pricing.

6.4 Programming

Algorithmic trading involves devising & coding strategies by analyzing the historical/real-time data which is procured from the data vendors. Some of the trading platforms mentioned above have their own scripting language which can be used for coding & backtesting strategies in the platforms itself.

Algorithmic traders around the globe use Python, R, Java and MATLAB extensively for trading on trading platforms. There are hundreds of external analytical packages that can be used in these languages, which aid in developing various trading strategies. Traders use external wrappers to implement codes into the trading platform.

It is vital to have sound programming knowledge to trade successfully in the markets. Algo trading aspirants must learn not only the basics of programming, but also to devise different strategies for various markets using these languages.

6.5 System Configuration & Software

By now we have discussed that as an algorithmic trader we will be working with different applications (trading platforms, programming tools, broker terminal, charting software, news feed, etc.). We will be dealing with huge data for backtesting and multi-tasking in live markets. So, it is essential to have the right computer system that fulfills all these needs without going on occasional breaks and strikes.

A trader can purchase a suitable workstation based on her requirements (i.e. trading frequency, strategy complexity, etc.), or by consulting someone with sound knowledge of computer hardware & technology.

One can add multiple screens to the system if required. In case of HFT, it is advisable to go for best-in-class equipment only.

6.6 Regulatory Approvals

Every country (and in some cases economic bloc like the EU) has its own regulatory agencies which set the guidelines and rules that need to be followed before you can go for automation. These regulatory agencies try to set the rules of the game and act as a referee in case of conflict or non-compliance.

All market participants (banks, investment firms, funds, agents, brokers, traders, systems, etc.) operating in a given country must adapt to them, and the country's regulatory agency is responsible for enforcing and supervising the participants' compliance.

For example, In India, there are well-defined rules that are laid down by SEBI and the exchanges relating to algorithmic trading. In Europe, the regulators have introduced new regulations in the last few years which have had a considerable impact on the trading industry as a whole. These regulations include MiFID2 (Markets in Financial Instruments Directive), MiFIR (Markets in Financial Instruments Regulation) and MAR (Market Abuse Regime). They are an attempt to update the regulation to new technological advances and although the scope is Europe, it has common aspects to other countries or serves as a reference to others as well.

In case a trader is trading through a broker, she must consult with the broker and get the required approvals before automating strategies using any platform.

Here is a non-exhaustive list of some regulatory agencies around the world:

European Union (EU)

- European Securities Markets Agency (ESMA)
- Financial Market Authority of Germany (FMA)
- UK’s Financial Conduct Authority (FCA)
- National Securities Market Commission of Spain (CNMV)

United States (US):

- Securities and Exchange Commission (SEC)
- Commodity Futures Trading Commission (CFTC)

Asia

- India: Securities and Exchange Board of India (SEBI)
- China: China Securities Regulatory Commission (CSRC)
- Singapore: Monetary Authority of Singapore (MAS)

7 Algorithmic Strategies

In this chapter we give you a high-level understanding of the concepts behind some popular algorithmic strategies along with some examples. We begin with momentum trading and the moving average crossover strategy. Then we discuss the statistical concept of mean reversion and how it can be used to devise trading strategies like pairs trading. Lastly, we briefly discuss the role of machine learning and Artificial intelligence in trading.

7.1 Classification of Algorithmic Trading Strategies

A lot of algorithmic trading strategies that are being used today can be classified broadly into the following two categories:

- Momentum based Strategies or Trend Following Strategies
- Mean reversion based strategies

Let's get to know them better.

7.2 Momentum Based Strategies

Momentum trading involves trying to profit from the trends in asset prices by taking a position in the direction of the trend. The rationale behind such a strategy is that once the asset price gains momentum in a particular direction, it keeps going in that direction.

Traders spend a lot of time and effort to determine the strength of these trends, before they take a position to ride them.

There are various technical indicators that have been designed to gauge the strength and direction of these trends using different approaches. Examples of such indicators include:

- Moving averages (both simple and exponential)
- Relative strength index (RSI)
- Moving average convergence divergence (MACD)
- Stochastic oscillator

Different automated trading strategies can be designed using these indicators and the current price series of the asset. We will now discuss one of the most common examples of momentum based algorithmic trading strategy called "Moving average crossover" strategy.

7.2.1 Simple Moving Average

Simple Moving Averages are the averages of a series of numeric values. They have a predefined length for the number of values to average at each step called the ‘window size’. Given a series of numbers and a fixed window size, the first element of the MA series is obtained by taking the average of the first subset of the number series whose size equals the window.

The next value in MA series is obtained by averaging the set of values when the window moves by one place i.e. excluding one data point from the left and including one from the right and so on.

Consider the example mentioned below to understand the calculation of simple moving averages. Let the average be calculated for five data points (fixing the window size as 5).

Suppose we want to find the MA for the following price series:

Daily price series for an asset: 7, 12, 2, 14, 15, 16, 11, 20, 7, 10, 23

Then, the first value of the MA series is the average of first 5 data points= (7 + 12 + 2 + 14 + 15) / 5 = 10

Second value of the MA series is calculated as: (12 + 2 + 14 + 15 + 16) / 5 = 11.8

Similarly, the third value of the MA series is: (2 + 14 + 15 + 16 + 11) / 5 = 11.6

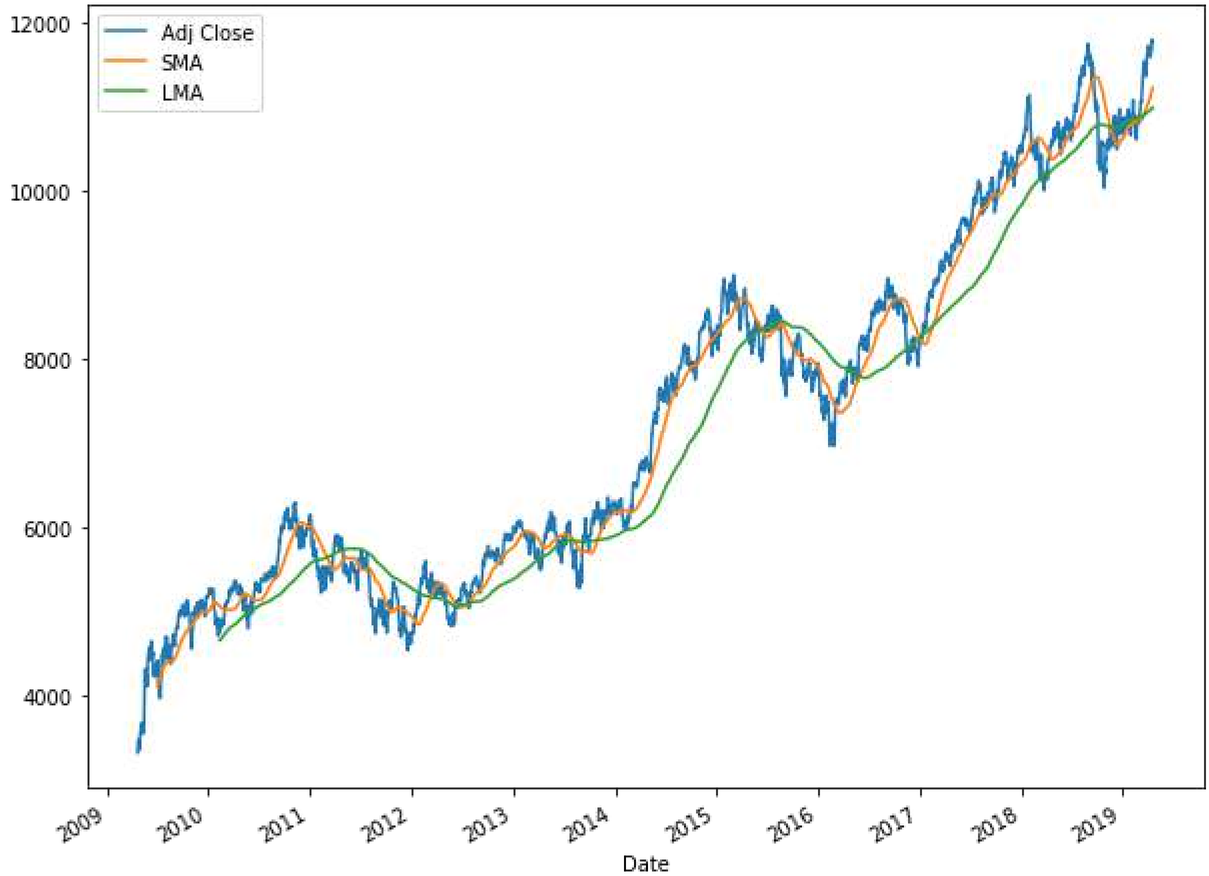
and so on.

7.2.2 The Simple Moving Average Crossover Strategy

When calculated for an asset price series, the MA series represents the overall momentum or trend by ‘smoothing’ what can appear to be ‘noisy’ or erratic price movements on short-term charts, into more easily understood visual trend lines.

Let us look at a simple example of a moving average crossover strategy on the NIFTY index (a major Indian equity index).

The strategy involves moving average indicators of different durations. An average of the shorter look-back window is called SMA and the one with the longer look-back window is called LMA. Popularly used SMA-LMA pairs include 20-40, 20-60 and 50-200. We show below the SMA-LMA plot along with the adjusted close price for Nifty.



Trading rules are simple. Buy the asset when the SMA crosses above LMA and sell the asset when SMA crosses the LMA from above. The idea is to capture the trend and profit from it as discussed above.

In terms of the coding logic, we shall buy the asset when SMA(today) > LMA(today) and SMA(yesterday) < LMA(yesterday). Similarly sell the asset when SMA(today) < LMA(today) and SMA(yesterday) > LMA(yesterday).

The parameters of this strategy are the SMA and LMA lookback windows that need to be optimized. For example, 20-day SMA and 50 day LMA pair (20-50) may perform better than a 15-50 pair. Decisions about strategy parameters can only be taken after backtesting the strategy on past data and forward testing it on real time data.

Such strategies can be easily coded, visualized, backtested and executed using programming languages such as Python.

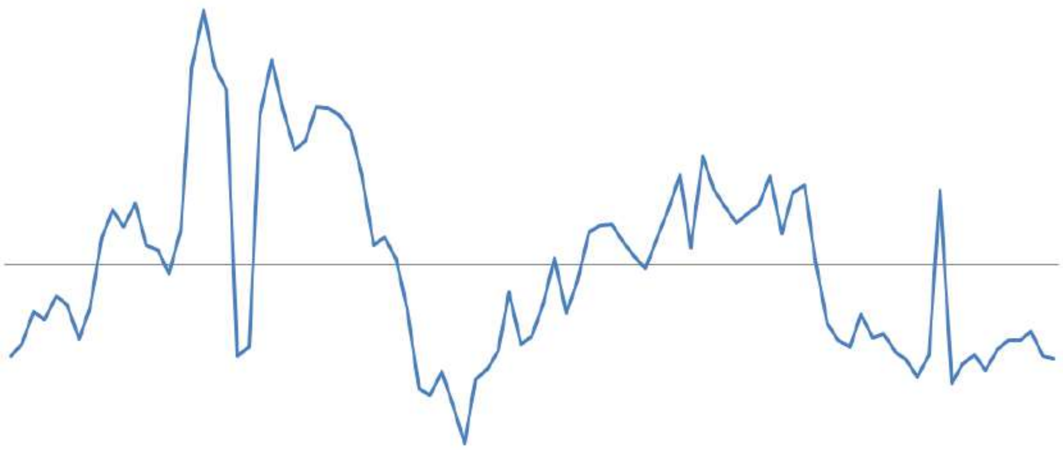
7.3 Mean Reversion Based Strategies

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

Consider a hypothetical example of a commodity whose price has stayed around USD 100 per tonne for the last ten years. Now, if one day it's price increases by USD 40 without any significant news or factor behind this rise, then by the principle of mean reversion, we can expect the price to fall in the coming days. In such a case, the mean reversionist would sell the commodity, in anticipation of the price falling in the following days. She would make profits by buying back when the price has fallen back to its mean of USD 100. The risk involved in this trade is that the price may keep drifting away from the mean or stay away from the mean longer than a trader can hold her position.

7.3.1 Time Series and Stationarity

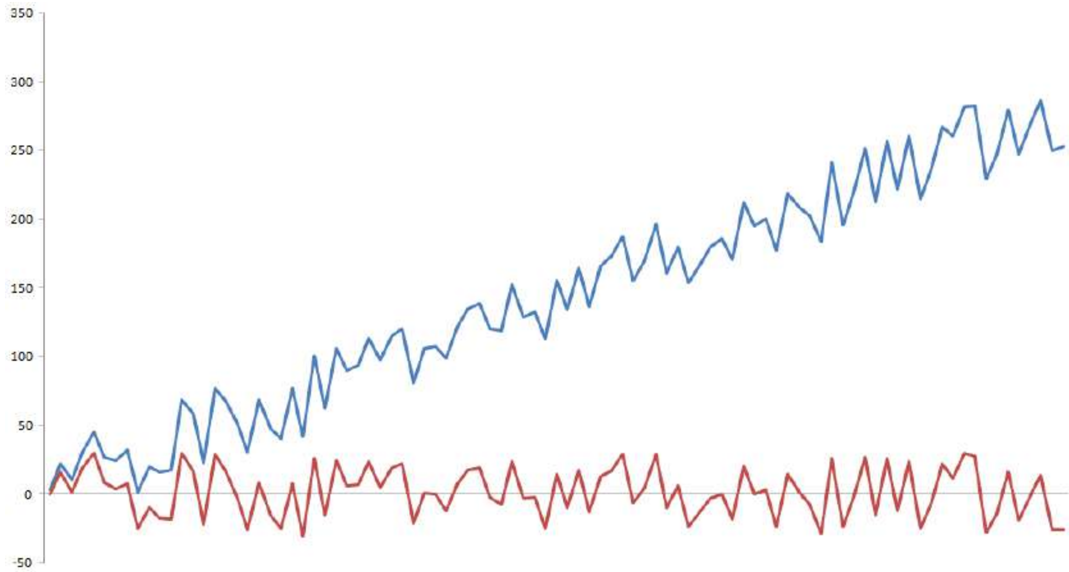
Time series is the data on a particular variable (for e.g. an asset prices or returns), arranged in the order of time. A mean-reverting time series has been plotted below where the horizontal black line represents the mean and the blue curve is the time series observations which tends to revert back to the mean.



Loosely speaking, a time series is stationary if its mean and variance are time invariant (constant over time). A stationary time series will be mean reverting in nature, i.e. it will tend to return to its mean and fluctuations around the mean will have roughly equal amplitudes. A stationary time series will also not drift too far away from its mean.

A non-stationary time series, on the contrary, will have a time varying variance or a time varying mean or both, and will often not revert back to its mean.

In the following diagram the blue line represents a non-stationary time series, whereas the red line represents a stationary time series.



In the financial industry, traders take advantage of stationary time series by placing orders when the price of a security deviates considerably from its historical mean, speculating the price to revert back to its mean.

7.3.2 Simple Mean Reversion Strategy

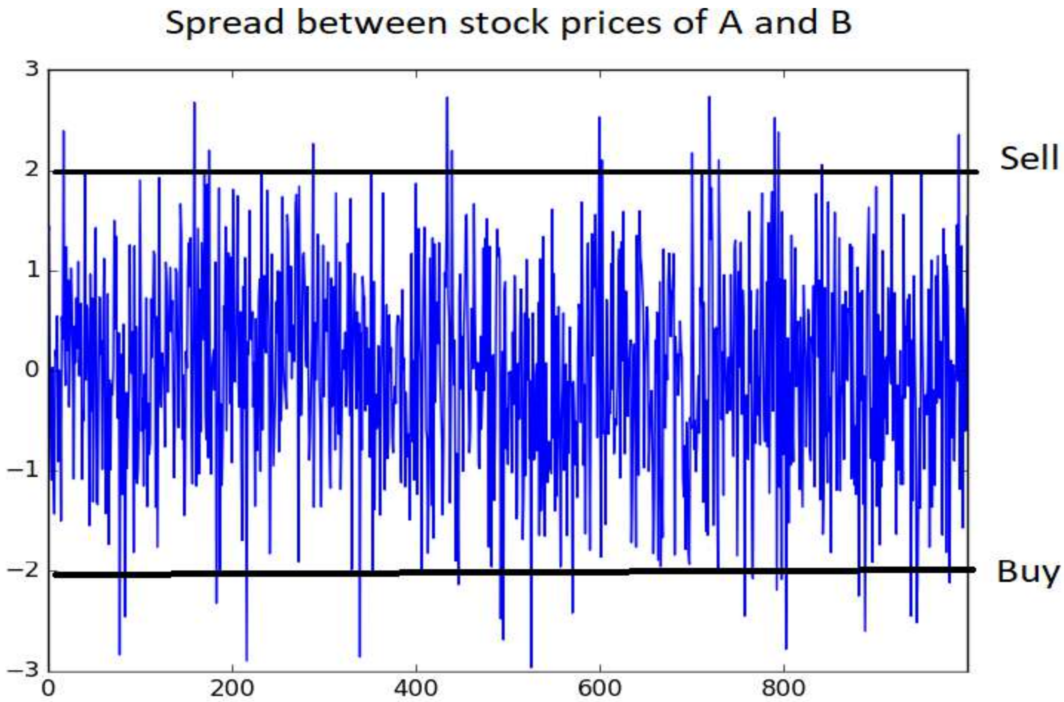
One of the simplest mean reversion trading related trading strategies is to find the average price over a specified period, followed by determining a high-low range around the average value from where the price tends to revert back to the mean. The trading signals will be generated when these ranges are crossed - placing a sell order when the range is crossed on the upper side and a buy order when the range is crossed on the lower side. The trader takes contrarian positions, i.e. goes against the movement of prices (or trend), expecting the price to revert back to the mean.

7.3.3 Pairs Trading Strategy: Mean Reversion of Spreads

Pairs Trading is another strategy that relies on the principle of mean reversion trading. Two assets are said to be co-integrated if the difference (spread) of their price series is stationary and hence mean reverting. Assets whose prices move closely together often exhibit co-integration, but not always and we need to confirm this using statistical tests.

For example, let us consider the stocks of two oil companies A and B, which are fairly similar in terms of size, company structures, markets and risk exposures. After quantitative analysis, we have observed that the spread between the prices of these stocks has remained fairly constant with a mean of zero in the past five years, with occasional divergences. We have performed statistical tests that have confirmed that the two price series are cointegrated i.e. the spread is mean reverting.

spread=(Price of stock A) - (Price of stock B)

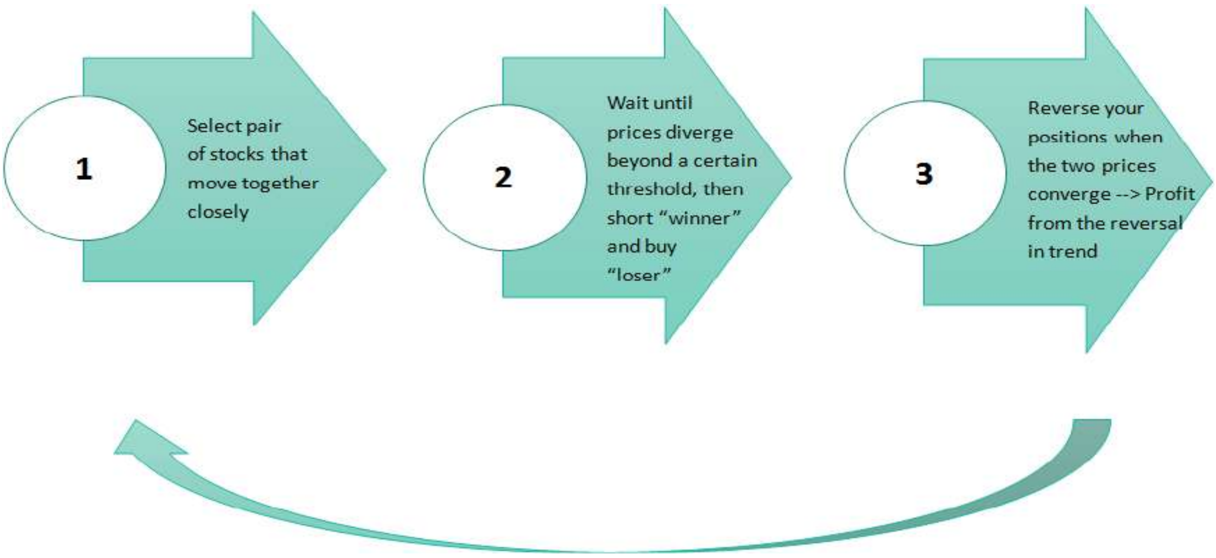


We devise our strategy such that buy and sell signals (for the spread) will be generated if the value of breaches is -2 and +2 respectively.

Now suppose suddenly the price of stock A increases considerably without any apparent reason or market news, whereas stock B continues to trade around its original price. This increases the spread between the two securities, significantly away from its mean value of zero to a value of +2.5.

According to our strategy we need to sell the spread. To do so, we can then sell the overpriced asset (A) and buy the underpriced asset (B) to enter into a pairs trade and wait for the spread to come back to its historic mean value of 0.

The following diagram summarizes the steps in pairs trading:



This strategy can easily be automated using a programming language such as Python. We can write a program to fetch price data for different stocks and conduct co-integration checks using available libraries and functions. Then, we can generate trading signals to buy the underpriced security and sell the overpriced security, when the spread breaches some threshold values. We can also set appropriate stop-losses and profit booking commands in place.

8 Careers in Algorithmic Trading

This chapter addresses some common concerns of individuals who want to make a career in algorithmic trading. It also discusses ways to build a career in the field.

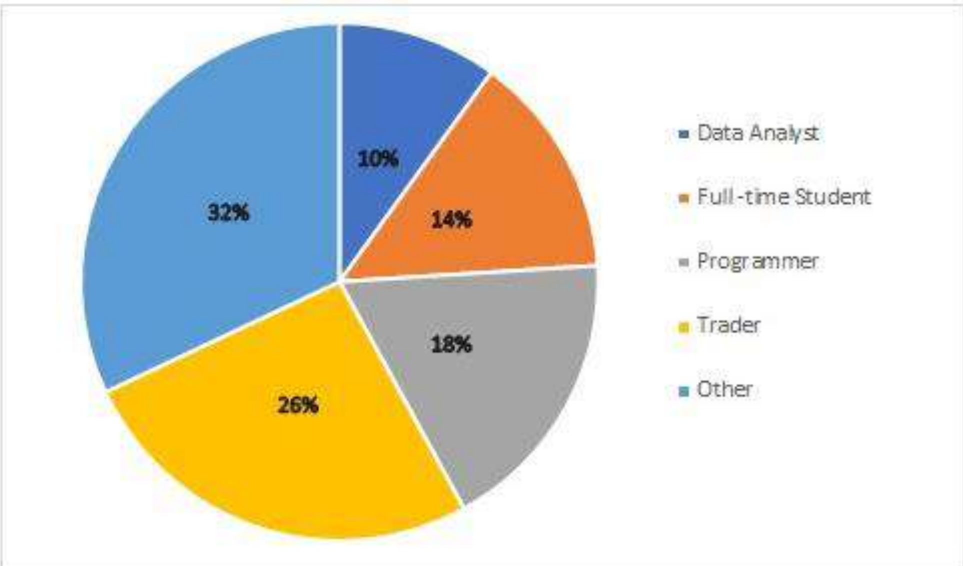
8.1 Do I Need a PhD to Break into Algorithmic Trading?

The short answer to this question is No. Nevertheless, we would like to add that it will take concerted effort to attain the requisite skills needed to be a quant trader.

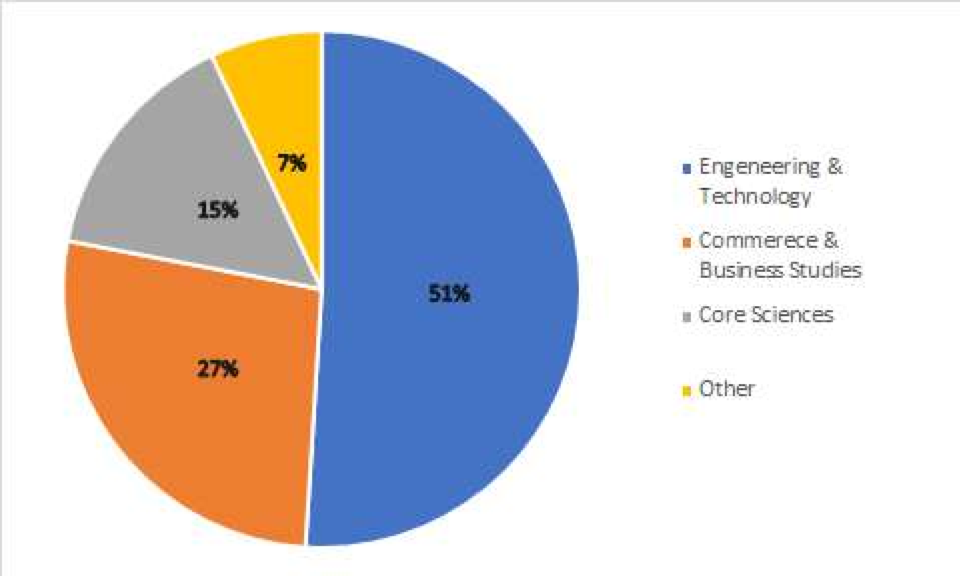
In the past, entry into the quantitative trading domain used to be heavily tilted towards PhDs in the hard sciences like Physics, Mathematics, or Engineering.

However, in recent years, there are a surfeit of alternative choices that are available. You have Masters' programs in quant finance, financial engineering, etc. offered by universities worldwide and similarly structured online courses offered elsewhere. The latter has the added benefit of flexibility and affordability.

At QuantInsti, we have crafted the EPAT specifically to train aspiring algorithmic traders and quants. The algorithmic trading domain has now opened up for individuals from diverse backgrounds. This is also reflected in the professional backgrounds of the participants who regularly enroll for the EPAT. Here's a breakdown of the professional background of our EPAT participants:



And here's a breakdown of domain expertise of our students:



8.2 Avenues of Employment

Here are some types of companies that employ quantitative finance professionals:

- Commercial Banks (e.g. RBS, HSBC, Natwest etc.)
- Investment Banks (e.g. Goldman Sachs, Credit Suisse, UBS etc.)
- Hedge Funds (e.g. Bridgewater, AQR Capital, Blackrock etc.)
- High Frequency Trading firms (e.g. IRage capital, DRW, XTX, Citadel etc.)
- Asset management companies (e.g. Allianz, PIMCO etc.)
- Financial technology companies (e.g. FinTech startups such as a-Quant)

8.3 Job Profiles

Here we list down a few profiles to understand what types of roles are available in the industry for individuals equipped with Quantitative skills.

- Algorithmic trader/HFT trader- Generate and execute profitable strategies for proprietary trading firms and hedge funds
- Execution Trader - Execute trades for financial institutions such as Investment banks
- Desk Quant - Implement pricing models that are directly used by traders
- Front Office Quants (FOQs) - Develop and manage models for calculating the price of assets on the markets
- Investment/Asset Management Quants - Develop models for mitigation of losses in investments

- Research Quant - Research and create new approaches for pricing
- Model Validation Quant - Implement pricing models to validate Front Office models
- Quant Developer - A developer/programmer from the field of finance
- Statistical Arbitrage Quant - Identify data patterns and suggest automated trades based on the findings
- Capital Quant - Model the bank's credit exposures and capital requirements

In addition, quantitative skills are also in huge demand for risk management profiles in various financial institutions.

8.4 Starting Your Own Algo Trading Desk or Firm

One needs domain knowledge, skilled resources, technology & infrastructure in the form of hardware and software for setting up an algo desk. The requirements, especially in terms of regulations, infrastructure and cost estimates can vary depending on the country you plan to set up your desk. Irrespective of your location, there are some common steps to be taken which we describe below.

Registering the Company

The first step is to register the firm. One can register a trading firm (for proprietary trading) as a Company, Partnership, LLP or even as an Individual. If, however, the aim is to set up a Hedge Fund with investors, other approvals from regulators (For e.g. SEBI in India and MAS in Singapore) are also required and the compliance rules and regulations are generally much stricter.

Capital Required for Trading and for Operations

Broadly speaking, trading capital required for High Frequency Trading is usually relatively less than that required for Low-Frequency Trading. LFT is scalable and can absorb much more trading capital. But the capital required for trading operations is typically far higher in case of HFT as compared to LFT given the infrastructure and technology requirements in HFT.

Trading Philosophy

We need to decide on the trading philosophy we'll adapt. The most common trading philosophies include execution-based strategies where the focus is to get the best price for execution rather than focusing on Alpha. Then there are High frequency strategies which are extremely latency sensitive and mainly include market making, scalping, and arbitrage. Then there is market sentiment based, machine learning based and news-based trading algorithms which can be relatively less sensitive to latency as compared to HFT.

Access to Market

There are different kinds of memberships which exchanges offer- clearing members, trading members, trading cum clearing members, professional clearing members, etc. If we don't want to go for direct membership in the exchange, we can also go through a broker. This involves lesser compliance rules and regulatory requirements. However, the flip side is that we have to pay brokerage and most HFT strategies are highly sensitive to transaction cost.

Infrastructure Requirements

Main focus areas under this head are colocation, hardware, network equipment and network lines. Colocation means that your server is in the same premises and on the same local area network as that of the exchange. Most exchanges provide colocation facility now. In some cases when exchanges do not provide colocation facility, there are vendors who provide co-location or proximity hosting facility. Since everything related to individual retail trading is connected over the internet, one needs to pay attention to opt for the best network connections to trade fast that are capable of providing really fast speeds, without any breaks, disturbances or shortfalls. For professional traders, it is advisable to go for leased line connections with the relevant exchange(s).

Risk Management

While manual trading mostly deals with market risk, algorithmic trading has a high degree of operational risk in it. The primary reason being that the machines do not possess the power of common sense that a human mind has. So, we have to make sure that there is common sense in algorithms before we take them to the market. Also, exchanges give a detailed checklist of risk management parameters that one must adhere to before starting to trade algorithmically.

Audit and Compliance

Each country has different rules with respect to Trading, and they differ a lot. Countries like the USA, or Japan where algorithmic trading is widely practiced, and countries like India where it is still developing, are entirely different examples of such a situation. It is highly advised to check with the broker for detailed guidelines before one starts.

9 Learning Algorithmic Trading

The global algorithmic trading market is growing rapidly, with a rising need for individuals familiar with its inner workings.

This chapter will serve as a beginners' guide to the skills one needs to develop to trade algorithmically. We will also discuss the means and resources that can be used to do so.

Key Skills

Algorithmic trading is an interdisciplinary field which requires knowledge in multiple domains. Here is a list of vital skills and knowledge you need to develop.

- Quantitative Analysis
- Programming
- Financial Markets Acumen
- Data Management
- Risk Management
- Machine Learning (ML) and Artificial Intelligence (AI)
- Knowledge of System Architecture
- Knowledge of Regulatory and Compliance Issues

Let us now look at each one in more detail:

Quantitative Analysis

With so much of financial data being generated each day, a good grip on key topics (we spell that out further ahead) in math and statistics are a prerequisite to any kind of quantitative analysis.

The algorithmic/systematic approach to investments incorporates a scientific method of inquiry. That is one reason you find a number of Ph.D.'s (or people who have done a Masters' in a quantitative discipline like math, engineering, physics, computer science, etc.) in the field. Fortunately, acquiring such degrees is by no means a necessity. However, in order to be competent at the task at hand, you would need to acquaint yourself with models used to analyze and forecast price movements, statistical properties of financial markets, etc.

Some of the topics that you should build on in no particular order are : Probability, Calculus, Linear Algebra, Inferential Statistics, Financial Econometrics including Time Series Analysis.

Programming

In 2019, if one is associated in any shape or form with the knowledge economy, not having basic programming skills can be a disadvantage. A quantitative trader uses programming at one or more stages (such as formulating and testing hypotheses, data download and manipulation, back-testing, execution) in the trading workflow depending on her comfort with the tool. In our land of quantitative trading, Python is arguably the most popular programming language and a great place to get initiated into programming if you've never done it before.

That being said, the requirements of the job profile and the trading system one uses will affect the choice of programming language one would prefer to learn. For a retail level algo trader, an open source language like Python or R are the preferred choices. On the other hand, for a trader working in HFT, lower level languages like C or C++ are the way to go.

The best way to learn to program is by writing code. We enlist some of our favorite resources in the recommended reading section to get you started.

Financial Markets Acumen

Any type of quantitative trader needs to be well versed with the financial markets. Aside from the above-mentioned areas, you need to develop acumen for financial markets in general and trading in particular. What type of concepts are we talking about here?

One way to consider this is to learn the ideas pertaining to what we want to trade in. The list below should give you a flavor of what we mean:

- Types of asset classes (equities, fixed income, currencies, commodities etc.)
- Types of trading instruments (stocks, options, futures etc.)
- Types of strategies (trend following, mean reverting, arbitrage etc.)
- Asset and derivative pricing models (for e.g. Black Scholes option pricing model, CAPM etc.)
- Risk management (metrics to monitor risk such as VaR limits, position limits etc.)
- Macroeconomics (understanding of the impact of economic factors like GDP, unemployment, monetary policy decisions etc. on markets)

Data Management

Getting access to quality data is important for any kind of trader. While there are ample sources of data available for daily price data, access to historical intraday data can be restricted. It is important to understand the patterns of data from across various markets and exchanges across the globe. One of the most important tasks for a trader/analyst is cleaning the data, structuring it to be uniform with the database (e.g. converting integers, floating decimals, etc.), and then using it to identify patterns, create and optimize strategies.

Risk Management

Trading strategies will help us to make money, but the risk management system is the one that is responsible for preserving it. In any kind of trading, market risk is something that a trader tries to mitigate through various means. In the case of automated trading, mitigating operational risk takes center-stage. We often tell our EPAT course participants that when you are trading manually, you have a very powerful risk management tool with you, which is common sense. When it comes to a machine, that critical part is missing, and you have to take care of that by carefully adding the appropriate risk checks in your strategy and overall code base.

Machine Learning & Artificial Intelligence

Machine learning and artificial intelligence are the buzzwords of today's technological landscape and with good reason. These tools are widely being used for business decision making and naturally their use is being extended to the markets.

Simply speaking, machine learning is the science of deriving insights from data using statistical models. For example, linear regression is a low-level machine learning algorithm in which the machine learns the relationship between two variables based on some statistical criterion.

Today we have much more complex methods and algorithms, such as artificial neural networks (ANN) and ensemble learning methods such as random forests.

Machine Learning algorithms are used in trading for different purposes.

Some of these include:

- Analyzing historical market behavior using large data sets
- Determining optimal inputs (predictors) into a strategy
- Determining the optimal set of strategy parameters
- Making trade predictions etc.

ML and AI are increasingly being used in designing quantitative strategies for trading.

These tools are being used in new and creative ways to model the market movements. For example, a collection of satellite images of oil tankers at sea can be used to predict movements of crude oil prices. Social media feeds data is increasingly being used to gauge the sentiments about a particular stock using the power of Natural Language Processing (NLP), a subset of machine learning.
ML and AI are vast topics unto themselves and one needs to put time and effort to appreciate and gain expertise in them.

Knowledge of System Architecture

As important as it is to know how to use a trading system, it is also important to know the insides of a trading system. An algorithmic trader needs to know about various components of a trading system including the infrastructure aspects from both the hardware and the network perspective. In the case of medium or low-frequency strategies, a basic understanding of these things should do.

In the case of HFT, it is also important to know about the components such as adapters, Complex Event Processing (CEP) engine, etc.

Regulatory and Compliance Knowledge

Every country and region have its own set of regulations and compliance requirements that it must adhere to trade in the respective trading destinations. There are rules related to short-selling, co-location, system approvals, etc. that one needs to know about. For example, some exchanges would need the approval to be taken at an overall system level, while in other exchanges (in India as well as a few other countries), you need to get approval at each strategy level. Additionally, if you intend to trade/manage other people's money, knowledge of regulations becomes even more critical.

10 Conclusion

In this final chapter, we bust some common myths about algorithmic trading and leave you with some parting thoughts.

10.1 The Myths

Myth 1: Algorithmic trading is rocket science, and all algorithmic strategies are insanely complicated.

Not quite! Some of the most successful strategies have very simple ideas like averages and standard deviations behind them.

However, it's still worth your while to get acquainted with statistics, programming and system architecture gradually to expand your repertoire as an algorithmic trader. Most long-term thriving practitioners start slowly with what they know and keep picking up skills along the way to improve the quality and sophistication of their strategies.

Myth 2: Retail Traders can never compete against the technical advantages of big HFT firms.

The reality is that High Frequency Traders do not compete with retail traders; instead, they compete amongst themselves. Given that most of the markets are made (market-making) by HFT desks, and since they target collecting a few pennies on average per trade, any sudden event/news can cause significant losses. Being in a colocation facility and using other forms of technology ensures that they can update their orders to the fair price within a very short time. This ensures that they can offer much better quotes, resulting in significant savings in transaction cost for an average retail trader.

Thus, the presence of HFT firms potentially benefits the retail traders as the bid-ask spread is reduced, and they can execute their orders at a better price in general.

Myth 3: Big players exploit every possible market opportunity leaving little scope for retail investors.

Large firms usually do not seek to exploit market inefficiencies or short-term opportunities if they are not scalable enough. This creates an opening for retail level traders to design strategies to exploit them.

Also, keep in mind that a big institution whether an HFT firm or a hedge fund has to comply with investment mandates and regulatory concerns which prevents them from profiting from certain anomalies they spot. Fortunately, retail investors can capitalize on them.

10.2 Parting Thoughts

"A journey of a thousand miles begins with a single step."

-Lao Tzu (600 BC)

Over the course of the book, we've attempted to provide you a tour of the algorithmic trading domain. We learned how trading systems have evolved over the years to eventually arrive at the modern quantitative trading system. While doing so, we walked through what it takes to create viable trading strategies and touched upon the knowledge and skills needed to be good quant traders. We explained how the quant trading approach combines contemporary advances in applied computer science (like high level programming languages, machine learning, artificial intelligence and the like) with financial markets know-how and big data.

The Reading List, Research Papers and other links that we share below and elsewhere are for you to see how deep this rabbit hole runs.

Please feel free to reach out to us with your thoughts, feedback, questions or even a hello on contact@quantinsti.com. We'd love to hear from you.

We hope this book has inspired you to get started on a journey to consistent alpha returns!

11 Reading List

11.1 Books

Algorithmic Trading/Quantitative Trading Strategies

- *Algorithmic Trading: Winning Strategies and Their Rationale* by Chan, E.
- *Quantitative Trading – How to Build Your Own Algorithmic Trading Business* by Chan, E.
- *Algorithmic Trading and DMA: An introduction to direct access trading strategies* by Johnson, B.
- *High frequency trading: A practical guide to algorithmic strategies and trading systems* by Aldridge, I.
- *Systematic Trading: A unique new method for designing trading and investing systems* by Carver, R.
- *Trading Evolved: Anyone can Build Killer Trading Strategies in Python* by Clenow, A.
- *The Evaluation and Optimization of Trading Strategies* by Pardo, R.
- *Quantitative Methods for Trading and Investment* by Dunis, C.L, Laws, J, Naim, P.
- *Algorithmic and high frequency trading* by Cartea, Á., Jaimungal, S., and Penalva, J.
- *Python Basics, With Illustrations from the Financial Markets* by Krishnamoorthy, V., Parmar, J., and Pena, P. M.

Market Microstructure

- *Trading and Exchanges: Market Microstructure for Practitioners* by Harris, L.
- *Market Microstructure Theory* by O'Hara, M.

Statistics and Econometrics

- *Schaum's Outline of Statistics and Econometrics* by Salvatore, D.
- *Analysis of Financial Time Series* by Tsay, R.
- *The Handbook of Portfolio Mathematics – Formulas for Optimal Allocation and Leverage* by Vince, R.

Technical Analysis

- *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications* by Murphy, J.
- *Technical Analysis Explained: The Successful Investor's Guide to Spotting Investment Trends and Turning Points* by Ping, M.

- *Evidence-Based Technical Analysis – Applying the Scientific Method and Statistical Inference to Trading Signals* by Aronson, D.

Derivatives

- *Options, Futures, and Other Derivatives* by Hull, J.
- *Dynamic Hedging: Managing Vanilla and Exotic Options* by Taleb, N.

Machine Learning & AI

- *Applied Predictive Modelling* by Kuhn, M. and Johnson, K.
- *Machine Learning with R* by Lantz, B.
- *Financial Decision Making Using Computational Intelligence* by Doumpos, M., Zopounidis, C. and Pardalos, P.
- *The Elements of Statistical Learning* by James, G., Witten, D., Hastie, T. and Tibshirani, R.
- *Statistically Sound Machine Learning for Algorithmic Trading of Financial Instruments* by Aronson, D. and Masters, T.

11.2 Research Papers

- Martiny, K. (2012). Unsupervised Discovery of Significant Candlestick Patterns for Forecasting Security Price Movements. In KDIR (pp. 145-150).
- Krauss, C. (2017). Statistical arbitrage pairs trading strategies: Review and outlook. *Journal of Economic Surveys,* 31(2), 513-545.
- Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of Finance,* 48(1), 65-91.
- Okunev, J., & White, D. (2003). Do momentum-based strategies still work in foreign currency markets? *Journal of Financial and Quantitative Analysis*, 38(2), 425-447.
- Balvers, R. J., & Wu, Y. (2006). Momentum and mean reversion across national equity markets. *Journal of Empirical Finance*, 13(1), 24-48.
- Di Graziano, G. (2014). Optimal trading stops and algorithmic trading. Available at SSRN 2381830.
- Jiang, Z. (2016). Currency Returns in Different Time Zones. Available at SSRN 2613592.
- Cartea, Á., Donnelly, R., & Jaimungal, S. (2017). Algorithmic trading with model uncertainty. *SIAM Journal on Financial Mathematics,* 8(1), 635-671.
- Takeuchi, L., & Lee, Y. Y. A. (2013). Applying deep learning to enhance momentum trading strategies in stocks. In Technical Report. Stanford University.

- Zemke, S. (2002, July). On developing a financial prediction system: Pitfalls and possibilities. In Proceedings of the First International Workshop on Data Mining Lessons Learned (DMLL-2002) (pp. 8-12). sn.

11.3 Online Resources

- QuantInsti Blogs
  https://blog.quantinsti.com/

- Quantstart.com
  https://www.quantstart.com/

- Quantocracy.com
  https://quantocracy.com/

- Dr E. P. Chan's Blog
  http://epchan.blogspot.com/

- Dr Jonathan Kinlay's Blog
  http://jonathankinlay.com/home/

- Robert Carver's Blog
  https://qoppac.blogspot.com/p/about-me.html

- PyQuant News
  http://pyquantnews.com/

12 References

https://www.cnbc.com/2018/12/05/sell-offs-could-be-down-to-machines-that-control-80percent-of-us-stocks-fund-manager-says.html

https://dea.gov.in/sites/default/files/NIFM%20Report%20on%20Algo%20trading.pdf

https://faculty.haas.berkeley.edu/odean/papers/Day%20Traders/Day%20Trading%20and%20Learning%20110217.pdf

https://www.technavio.com/report/global-algorithmic-trading-market-analysis-share-2018

https://www.cnbc.com/2019/09/03/on-days-when-president-trump-tweets-a-lot-the-stock-market-falls-investment-bank-finds.html

QuantInsti® is one of the pioneer algorithmic trading research and training institutes across the globe. With its educational initiatives, QuantInsti is preparing financial market professionals for the contemporary field of algorithmic and quantitative trading. QuantInsti has also designed education modules and conducted knowledge sessions for/with various exchanges in South and South-East Asia and for leading educational and financial institutions.

QuantInsti's flagship programme 'Executive Programme in Algorithmic Trading' (EPAT®) is designed for professionals looking to grow in the field algorithmic and quantitative Trading. It inspires individuals towards a successful career by focusing on derivatives, quantitative trading, electronic market-making, financial computing and risk management. This comprehensive certificate offers unparalleled insights into the world of algorithms, financial technology and changing market microstructure with its exhaustive course curriculum designed by leading industry experts and market practitioners.

Quantra® is an e-learning portal by QuantInsti that specializes in short self-paced courses on algorithmic and quantitative trading. Quantra offers an interactive environment which supports 'learning by doing' through guided coding exercises, videos and presentations in a highly interactive fashion through machine enabled learning.

Blueshift helps you turn your ideas into trading strategies. You can research your ideas, backtest them, and take your strategies live with a broker of your choice. It is a fast, flexible and reliable platform to research and trade systematic investment strategies in Python. It is asset-class, instruments and style agnostic - we support multiple asset classes and instruments like FX, Equities, Futures.