Chapter 1 Introduction

Quantitative trading is where mathematics and computer science meet.

- Dr. Haksun Li

1.1. What is Quantitative Trading?

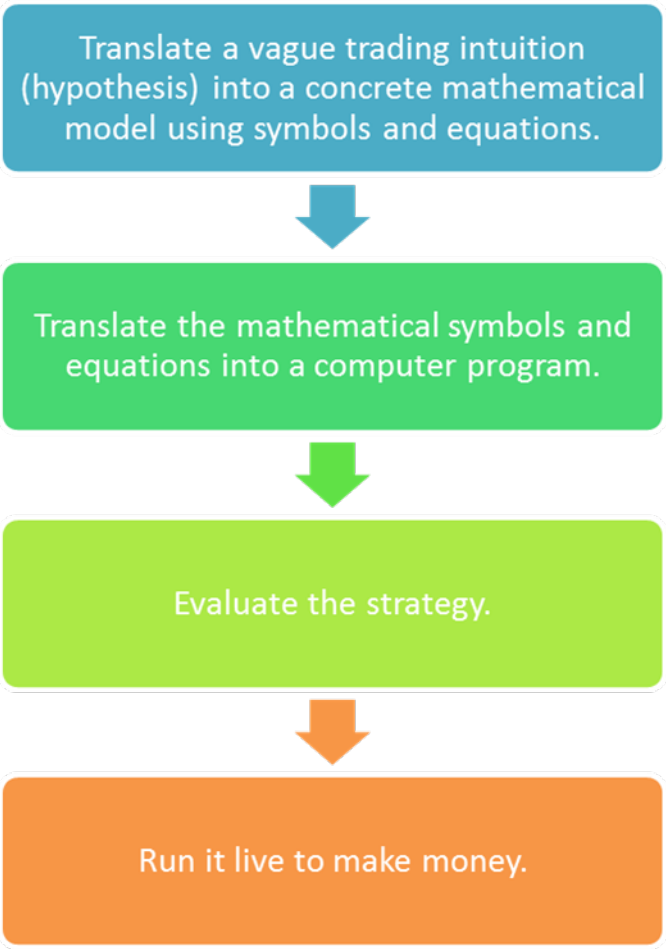
Quantitative trading (quantitative investment, quantitative wealth management, algorithmic trading) is the buying and selling of assets following the instructions computed from a set of mathematical models that are proven under certain assumptions. It is an industry where mathematics and computer science meet. Not only is the execution and trading of models systematic but the research process to create them must also be scientific. Every step, e.g., backtesting, in the research process and every decision made, e.g., parameters chosen, in building models must be justified by mathematical reasoning. The key differentiation from other investment approaches is on justifying them in certain scientific ways (the how) but not necessarily on producing different trading strategies (the what). This is the main and recurring theme of this book. We will see how each and every trading strategy discussed in this book is mathematically proven, meaning that they are guaranteed to be profitable on average, if certain assumptions are true. No other investment analysis gives such level of confidence.

There are many other ways to do investment analysis in the capital market to make (or lose) money. For example, there are value investing, (macro-) economic models, fundamental analysis, technical analysis and astrology[2] (that's why Wall Street hires so many physicists, no?[3]). These approaches may use mathematics in some steps, e.g., data analysis, but not in the entire process of constructing trading strategies. For example, an economist runs a regression model on a data set against a set of factors to build a return prediction model (Fama & French, 1992). Regrettably, the justification of the model is often by ad-hoc historical backtesting. When you backtest a model using last 10 years of data, all you get is how much money you *could* have made if you traded the model in the best scenario, assuming no slippage, no technical problem and nothing unexpected. We, traders, however care only about how much bonus we can pocket this year, after taxes.

Quantitative trading applies (rigorous) mathematics in all steps during trading strategy construction from start to finish: (1) hypothesis/modeling, (2) analysis/justification, (3) evaluation and (4) execution. We recommend these four steps as the standard procedure in quantitative trading research.

[2] I meant to type 'astronomy'.
[3] Which one is easier for the physicists, building a time machine or predicting the stock market?

Figure 1-1



Step 1: Modeling

Any trading strategy starts with an intuition or a hypothesis from the trader. He may get it from his creativity, imagination, day (or night) dreaming, books, newspapers, TV shows and hearsays. He may even "borrow" it from his (ex-) colleagues. This step turns a vague trading idea into a concrete trading strategy. A concrete trading strategy is completely specified with the entry and exit signals, the positions to take, etc.

Mathematically, a quantitative trading strategy is a math function, f, that at any given time, t, takes as inputs any information that the strategy cares about and that is available as of time t, $\mathcal{F}_t$, and gives as output the positions of the assets to take, $f(t, \mathcal{F}_t)$. The symbol, $\mathcal{F}_t$, is called the filtration at time t. It means the (set of all) information available at time t (Klebaner, 2012). For example, $\mathcal{F}_t$ may include the current positions, V_t, and the current price, x_t, of the assets. A strategy that makes decisions based on the current positions and the current prices is therefore denoted as $f(t, V_t, x_t)$.[4]

This step is challenging because it requires the trader to have mathematical skill that is mature enough to create models from ideas and to use symbols and equations to

---

[4] V_t and x_t may be uni- or multi- variates.

describe rules and situations. The mathematical skill we need here is imagining and modeling something innovative, solving complex equations and deriving properties from abstraction. These demand experiences and more than textbook mathematics. We need more than mechanically punching in numbers and following formulae as in, for example, technical analysis. As we shall see, writing down f is by no mean a trivial task.

Step 2: Programming

Only after the quantification of a trading strategy by writing down the equations can we code it up in, say, Java. The code serves two purposes. First, the code enables us to study and analyze the strategy. Most of the time, we cannot investigate a strategy analytically except for the very simple ones. We have to resort to computer simulation. For example, to find out the historical P&L of trading a strategy in the last 10 years, we need to do a backtesting by simulating it using 10 years of data on a computer. Pen and paper would take practically forever.

Second, ideally, the same piece of code used for research and investigation can go straight into production for live trading. This practice eliminates the possibility of translation error between research and IT (which unfortunately and usually are done by two different (groups of) people). Otherwise, there is no guarantee that the strategy being traded in production is *exactly* the same as the strategy being studied and backtested in research. The research code should not be just a simplified version of the production code. We should, especially in high frequency and intraday trading, simulate also production problems and details in backtesting as well, such as partial order fills, connection errors, position overflow.

This step is challenging because it requires the trader to have professional programming skill to code up a strategy correctly and, no less important, elegantly. Unlike a programmer who usually has specifications and knows exactly what to code up, a trader often has no idea because he would not know what a profitable trading strategy looks like when doing research.[5] He will need to do a lot of trials and errors to explore different twists, ways and ideas. If all he could program in is R/MATLAB, essentially imperative programming using a script language,[6] then he would need to keep track of numerous changes and versions manually and with a lot of copy-and-pastes. His code would look very much like spaghetti and would be impossible to manage when it grows. A nightmare! Instead, a trader needs to be familiar with data structures, design patterns and algorithms to do elegant programming. This can help ensure that the code is effective and that the organization and tracking the many versions is efficient. There are many features in a modern programming language like Java that are designed for effective and efficient programming (Bloch, 2008). For example, an interface facilitates experimenting different methods of trading without

[5] If he did know, e.g., being told a model by his boss, he would be a programmer not a trader.

[6] Yes, they have objects in MATLAB/R but they hardly match up to other real object-oriented languages. They are not often used by MATLAB/R programmers anyway.

having to change the main program; inheritance helps keeping track of the many (subtle) differences in ideas.

Step 3: Justification

With the mathematical model and the computer code of a trading strategy, we are ready to evaluate it. We can ask a lot of questions about the strategy and answer them. For instance, the most single important measure of a trading strategy is probably how much money we *will* make. As discussed, the historical P&L, as a look-back indicator, does not answer this question nor is it very relevant. We want a look-forward indicator. The amount of money we will make in the future is not one number (as with historical P&L) because it is uncertain. An uncertain quantity is a random variable. It is best described by a probability distribution. The correct question to ask, as a scientist, is therefore: what is the probability distribution of the strategy's future P&L? One way to compute the distribution is by bootstrapping. When we ask questions like this, think about them in scientific terms and solve them using mathematics, we are doing quantitative trading research.

Other important properties to compute for a quantitative trading strategy are for example: holding time distribution[7], Value at Risk, stop-loss, max drawdown. You can find a list of performance measures at:

http://redmine.numericalmethod.com/projects/public/repository/svn-algoquant/show/core/src/main/java/com/numericalmethod/algoquant/execution/performance

It would take a lot of luck if the first idea that comes to mind turns out to be a good strategy, passing all performance criteria such as a high expected (not historical) Sharpe Ratio and a small expected (not historical) max drawdown. We never feel that lucky! Otherwise, we iterate many cycles between steps 1, 2 and 3 to refine and improve a candidate strategy until we find something satisfying. Often we totally discard an idea and start afresh with another new idea back to step 1.

Step 4: Trading

The ultimate goal is to put a good trading strategy found in live execution to print money. We start with a small capital that is enough to test the strategy to make sure things run smoothly: the strategy is correctly implemented; the IT system supports the execution without much delay; safety precautions are in place so we don't bankrupt the company; the middle and back offices are ready to process the trades; the bosses are happy with the results and etc. As we gain experience playing the strategy and feel confident with it, we can increase the capital to the full capacity allocated.

[7] A holding time is how long we will hold the position for after we enter a trade before we exit. Because the duration is about the future and is uncertain, it is a random variable. It is therefore best described by a probability distribution, not a number.

It is easy to be “confident” about a strategy when we are making money with it. When the strategy loses money, we want to find out whether the losses are “expected”. In fact, we should do the same for profits as well. Put differently, we want to know if the live P&L agrees with our backtesting. For example, when we lose money 5 days in a roll, do we stop the strategy or do we continue, assuming that the stop loss is not yet hit? Again, as quantitative traders, we want to think in terms of mathematics. The mathematical question to post is therefore: given a strategy f, what is the likelihood of losing money 5 days in a roll? If that probability is very small, we may[8] be playing the strategy in an uncharted territory. The regime that we assume for the strategy in our research may be different from the regime that we play the strategy in production. This may be a good signal to early stop the strategy. By the same reasoning, even if we are making money 5 days straight but that is not expected in our calculations, we may be simply stretching our luck. That is just gambling not trading.

1.2. Technical Analysis is Not Quantitative Trading

Many would deride this trading strategy as nonsense: when Apollo (son of Zeus) brings his golden ram to his grandfather Cronus and the three dance into an equilateral triangle, the E-Mini S&P will be at 864. However, many would believe when the same statement is rewritten in (financial) astrology terms: when the Sun (Apollo) is at 24 degrees of Aries (the golden ram) and is at 60 degree aspect to the planet Saturn (Cronus), the E-Mini S&P will be at 864. This “strategy” was invented by W. D. Gann and there is even software to compute it (Ted, 2003).

Yet a lot more people would believe and put their life savings in a similar statement: when this line crosses that line, S&P will go up. Sound familiar? This is the very prevalent strategy called “Moving Average Crossover” made popular not by scientists but by news reporters, TV show hosts and journalists. Many unwary traders follow suit as if they found some treasures.

Most technical analysis books describe the strategy like this: a crossover occurs when a fast moving average (i.e. a short period moving average) crosses above/below a slow moving average (i.e. a long period moving average), then you buy/sell. It is then illustrated with an example of applying this strategy to a stock for a period of time to show the profits. However, none of the technical analysis books provides a scientific explanation *why* the strategy may work. At the very best, they simply appeal only to common sense. The fast moving average tracks the recent trend and the slow moving average tracks the long term trend. When it is recently trending up, we buy. Otherwise, we sell. “Justification” by common sense, however, is nonsense rather than science.[9] It is problematic in a number of ways.

First, a justification together with an illustration for a technical analysis rule is more like wishful thinking than a scientific proof. When a bullish crossover happens, you

[8] In statistics or probability theory, every statement is probabilistic, hence “may”.
[9] It is the deposit of prejudice laid down in the mind before the age of eighteen. – Albert Einstein.

just wish that the trend will continue to go up. This going-up "conclusion" is not a necessary consequence of any assumptions followed by a logical deduction process. In fact, even the notion of going-up after crossovers is not well defined. What exactly is a trend in the Moving Average Crossover strategy?[10] Presumably if the indicator is worth of anything, then when a bullish crossover happens, the (conditional) probability of the price going up is bigger than the (conditional) probability of the price going down. The conditional vs. unconditional probabilities of outcomes after crossovers have never been computed as far as we know. Only a few scholarly works have investigated the conditional probabilities of technical indicators (Lo, Mamaysky, & Wang, 2000).

It is very worth pointing out that the P&L shown for a technical indicator in a technical analysis book merely illustrates the mechanics of the indicator. It does not prove that the strategy is profitable or any way useful. In fact, because of the sheer number of assets and a long history of data available, for just about any strategy it is almost always possible to find an asset and a carefully chosen period of time that the strategy is profitable during that historical period. In other words, it is almost surely impossible to find a strategy that does not work for all assets during all historical periods of time.

Second, the application of a technical indicator is more of an art (if it is at all) than a science. Warren Buffett explained, "I realized that technical analysis didn't work when I turned the chart upside down and didn't get a different answer." For example, a technical indicator itself provides no guidance in how the parameters should be chosen. It is not clear how the fast and slow moving average windows should be picked. When you pick a set of parameters, it "works"; when you pick another, it does not. A common practice by practitioners of technical analysis is to optimize for the two parameters with respect to historical data. Not only does this method run into the over fitting problem but it also tells nothing about how the optimized parameters will work in the future.

In summary, technical analysis provides only indicators that measure some aspects of the traded asset. These indicators are not proven in a scientific way to have any predictive power the way that chartists present them.[11] We do not know when they are applicable or what their expected P&Ls are. They at best remain as hypotheses in Step 1 in our quantitative trading research process in Figure 1-1. Therefore, we do not see technical analysis as quantitative trading research. Nor do we see technical indicators such as Moving Average Crossover as quantitative trading strategies.

[10] How would you define "trend" mathematically? Please try it as an exercise before reading the several definitions we shall provide.
[11] OK, there are some rich chartists. But there are also rich psychics and astrologists.

1.3. Moving Average Crossover as a Quantitative Trading Strategy

In the last section, we dismiss Moving Average Crossover as a quantitative trading strategy because technical analysis provides no scientific justification for it. In this section, we are going to show that Moving Average Crossover is a quantitative trading strategy. “What? Are you outta your mind?” you may complain. We cannot emphasize enough that whether a strategy is quantitative or not depends not on the strategy (mechanics) itself but entirely on the process to construct it. That is, whether there exists a scientific justification to “prove” the strategy. In what follows, we will illustrate the quantitative trading research process in Figure 1-1 using Moving Average Crossover as an example following Kuo’s work (Kuo, 2002).

Step 1 in Quantitative Trading Research

We start with a simple hypothesis: when a bullish crossover happens, we buy; when a bearish crossover happens, we sell. The difficulty is to translate this not well defined and imprecise idea into precise math equations. There is differentiation between highly skilled scientists who can apply mathematical thinking to be creative and lowly skilled technicians who can merely apply formulae. Here we need to make concrete a few ideas such as: what are bullish and bearish crossovers, how much to buy/sell. Before you read on, the reader is encouraged to challenge himself to see if you can come up with a model for the strategy.

(Kuo, 2002) proposes the following. Suppose n and m are the lengths of two moving windows with $n > m$. The Moving Average Crossover indicator is defined as the difference between two moving averages:

$$B_t = \frac{1}{m}\sum_{j=0}^{m-1} P_{t-j} - \frac{1}{n}\sum_{j=0}^{n-1} P_{t-j} \qquad \text{Equation 1-1}$$

The first term is the fast moving average and the second term the slow moving average. When $B_t > 0$, the fast moving average crosses the slow moving average above from below; our position is to long 1 unit of the asset. When $B_t < 0$, the fast moving average crosses the slow moving average below from above; our position is to short 1 unit of the asset. Equation 1-1 is a significant progress in quantifying a trading strategy. It is our first quantitative trading strategy! We have translated a vague trading idea in English into a well-defined mathematical model in symbols. Only with this equation can we do algebra and mathematical manipulations on this trading strategy. Only with this equation can we deduce the mathematical properties of this trading strategy, such as the P&L distribution and holding time distribution. It is magical![12]

[12] It is worth noting that not every translation or modeling is as simple as this example. Very often, advanced mathematics is involved as we shall see in the future chapters.

To make our analysis easier in Step 3, we simplify Equation 1-1. First, we reduce this two dimensional problem to a one dimensional problem by getting rid of one variable, m. Intuitively, m should be small because the fast moving average tracks the short term trend. The smallest number we can assign to m is 1. We try that first.

Second, as we shall see in Section 4.2 there are a number of good reasons to work with log returns than prices. To rewrite Equation 1-1 in terms of log returns, we use geometric moving average instead of the popular arithmetic moving average. Putting together these two modifications, we have:

$$B_t = \left(\prod_{j=0}^{m-1} P_{t-j}\right)^{\frac{1}{m}} - \left(\prod_{j=0}^{n-1} P_{t-j}\right)^{\frac{1}{n}} = P_t - \left(\prod_{j=0}^{n-1} P_{t-j}\right)^{\frac{1}{n}} \qquad \text{Equation 1-2}$$

When $m = 1$,

$$B_t = P_t - \left(\prod_{j=0}^{n-1} P_{t-j}\right)^{\frac{1}{n}} \qquad \text{Equation 1-3}$$

We long the asset when

$$B_t > 0, \text{iff } P_t > \left(\prod_{j=0}^{n-1} P_{t-j}\right)^{\frac{1}{n}} \qquad \text{Equation 1-4}$$

Notice that log-return is defined as:

$$R_t = \log \frac{P_t}{P_{t-1}} = \log P_t - \log P_{t-1} \qquad \text{Equation 1-5}$$

Taking log on both sides to convert prices to log returns, we have:

$$\log P_t > \log\left(\prod_{j=0}^{n-1} P_{t-j}\right)^{\frac{1}{n}}$$

$$\log P_t > \frac{1}{n}\sum_{j=0}^{n-1} \log P_{t-j}$$

$$n \log P_t > \log P_t + \log P_{t-1} + \cdots + \log P_{t-n+1}$$

$$\begin{aligned}\log P_t - \log P_{t-1} \\ &> (\log P_{t-2} - \log P_t) + \cdots \\ &+ (\log P_{t-n+1} - \log P_t)\end{aligned}$$

Equation 1-6

$$\begin{aligned}R_t > \big((\log P_{t-2} - \log P_{t-1}) + (\log P_{t-1} - \log P_t)\big) + \cdots \\ &+ \big((\log P_{t-n+1} - \log P_{t-n+2}) + \log P_{t-n+2} \\ &- \cdots - \log P_{t-1} + (\log P_{t-1} - \log P_t)\big)\end{aligned}$$

$$R_t > (-R_{t-1} - R_t) + \cdots + (-R_{t-n+2} - \cdots - R_t)$$

$$(n-1)R_t > -\sum_{j=1}^{n-2} [n-(j+1)]R_{t-j}$$

Alternatively, we have:

$$R_t > -\sum_{j=1}^{n-2} \frac{[n-(j+1)]}{(n-1)} R_{t-j}$$

Equation 1-7

What this strategy says is that when the current period return is bigger than (the negative of) the weighted sum of the returns in the last $(n-2)$ periods, we buy. Otherwise, we sell.

We have many choices for the slow moving average window length. It is not obvious what the optimal n is. We can start with the two extremes: $n = 2$ and $n = \infty$.[13] When $n = 2$, we use the smallest amount of information possible, i.e., today's return or last two days' prices, so the strategy reacts quickly to recent development. When $n = \infty$, we use all available information, i.e., all prices available, to make a trading decision. Both choices seem plausible and make sense. Common sense tells us that the more information there is the better decision we make. Let's see where our common sense takes us.

[13] $n > m = 1$. So the smallest n is 2.

Step 3 in Quantitative Trading Research

After building a quantitative model for a trading strategy, we can ask and answer a lot of questions about it. For a complicated model, we often resort to computer simulation. For a simple model like this one, we can analytically compute certain properties, such as expected holding time and expected return. Here we report the expected returns of the two extremes. The details can be found in (Kuo, 2002) and some of the mathematics are reproduced in Section 5.1.

GMA(2, 1):

This is the Geometric Moving Average Crossover with $n = 2, m = 1$. Plugging the parameters in Equation 1-7 and combining it with Equation 1-4, we have:

$$B_t > 0, \text{iff } R_t > 0 \quad \text{Equation 1-8}$$

The trading strategy simplifies to that we go long in the next period when the current return is positive; otherwise we go short. Put differently, if the price goes up today, we buy; otherwise we sell. This is a (surprisingly) simple trend following strategy. (Kuo, 2002) shows that the expected return of making one (long) trade is:

$$\mathrm{E}(RR_t) = \frac{1}{1-p}[\Pi p \mu_\varepsilon - (1-p)\mu_\delta] \quad \text{Equation 1-9}$$

We will provide a proof for this equation in a later chapter. For now, these symbols mean the following. p is the "strength" or "persistence" of the uptrend between 0 and 1. Π is a constant between 0 and 1. μ_ε and μ_δ are the means of the positive and negative shocks (the daily returns) of the traded asset respectively. All these parameters can be estimated from historical data of the asset.

With Equation 1-9 we can deduce the exact circumstances when we should play this trend following strategy. For the expected return to be positive, we require:

$$\mathrm{E}(RR_T) = \frac{1}{1-p}[\Pi p \mu_\varepsilon - (1-p)\mu_\delta] > 0$$

$$\Pi p \mu_\varepsilon - (1-p)\mu_\delta > 0$$

$$\Pi p \mu_\varepsilon > (1-p)\mu_\delta \quad \text{Equation 1-10}$$

$$\mu_\varepsilon > \frac{1-p}{\Pi p}\mu_\delta$$

This condition says that when the average positive shock is bigger than the average negative shocks by a predetermined factor, we expect the strategy to be profitable. The amount of money expected to be made per trade is computed by Equation 1-9. This condition is sufficiently satisfied when:

$$\mu_\varepsilon \gg \mu_\delta \qquad \text{Equation 1-11}$$

Furthermore, when $\mu_\varepsilon \approx \mu_\delta$, i.e., when the positive and negative shocks are more or less the same, the expected return is also positive when:

$$\Pi p - (1 - p) > 0$$

$$(\Pi + 1)p > 1 \qquad \text{Equation 1-12}$$

$$p > \frac{1}{\Pi + 1}$$

This condition says that when the uptrend is strong or persistent enough, bigger than a predetermined factor, the strategy is expected to be profitable. In other words, we should stop playing this strategy when neither Equation 1-10 nor Equation 1-12 is satisfied.

Behold the magic of quantitative trading! Before we even look at any data, before we even bet our first dollar, we already know exactly when we should play the strategy, when we should stop, how we choose the parameters, how long we expect to keep the trades for and how much money we expect to make. This is nothing that other investment approaches offer. Value investing tells you to wait; technical analysis tells you nothing. Quantitative trading tells us everything about the strategy before we start trading!

GMA(∞, 1):

This is the Geometric Moving Average Crossover with $n = \infty, m = 1$. We use all information available – the whole past history of asset prices – to compute trading signals. We can repeat a similar calculation as in the GMA(2, 1) case. (Kuo, 2002) shows that the expected return of a long trade is:

$$\mathrm{E}(RR_T) = -[1 - p(1 - \Pi)][\mu_\varepsilon + \mu_\delta] \qquad \text{Equation 1-13}$$

The wary reader may already smell something unusual. The little minus sign makes the expected return always negative! As $0 \leq p \leq 1$, $0 \leq \Pi \leq 1$, the first term is positive. As $\mu_\varepsilon > 0$, $\mu_\delta > 0$, the second term is also positive. The product of two positive numbers is positive. That seemingly innocuous minus sign in the front turns the whole expression negative. The equation states that when we play GMA(∞, 1), we expect to always lose money for all assets we trade and for all parameters we use!

The conclusion is very surprising as it stands in sharp contrast with what common sense tells us the more information the better. For this particular trend following strategy and a particular assumption on the underlying price process[14], we make more

[14] We assume that the price process follows the general Knight-Satchell-Tran process. (Knight, Satchell, & Tran, 1995). See Section 5.1.

money by using as little information as possible. In fact, it can be shown that GMA(∞, 1) is most profitable (or least negative) when the price process is a random walk, i.e., no information. Comparing to what newspapers and TV shows suggest, for instance, 250-day line vs. 20-day line (1 year vs. 1 month) or 250-day line vs. 5-day line (1 year vs. 1 week) and etc., we now know in theory that they are recipes for going bankrupt. These suggestions come not from scientists but from reporters and journalists.

Once again, we want to emphasize that although both technical analysis and quantitative trading may recommend the same trading strategy, in our example Moving Average Crossover, technical analysis is not quantitative trading. The former only proposes a strategy (mechanics) and rules without any scientific justification or evaluation (other than historical backtesting); the latter provides a theoretical framework to prove and analyze a strategy for its properties. This book is about quantitative trading and the mathematics.

1.4. Why Quantitative Trading?

We have chosen to do trading, investment or wealth management among many alternatives like value investing or technical analysis, in a mathematical way. The very reason is our passion for seeking the truth. Scientist Joseph Needham concluded that it was the zeal for truth that sparked and fueled the European advancement of science. (Chincarini & Kim, 2006) argues that truth triumphs Gordon Gekko's greed in the financial world. "Mathematics is the language with which God wrote the universe," wrote Galileo Galilei. It is the supreme scientific truth that our civilization has achieved so far. In fact, as many philosophers such as Plato believe, it is the only scientific truth. For example, the laws of natural numbers or the value of π are fundamentally true or unchangeable and do not require any specific context. Newton's laws are not like that. They do not apply to very big or very small worlds. Therefore, we want to use exactly the same language that describes the physical Universe so amazingly well to discover the truths in the financial world. On the other hand, if mathematics did not work, what would?

During our journey to learn about quantitative trading, we have created a four step process to generate a trading strategy (see Figure 1-1). This process requires three essential skills: (1) mathematics, (2) programming and (3) creativity. Mathematics is what translates a trading idea or intuition into well-defined meaningful symbols. Starting from the assumptions, we can derive the properties of a made-concrete trading strategy. Before betting our first $1, we can compute the expected return (or P&L distribution) and the expected holding time of a trade. Programming is what translates the mathematical symbols into lines of code for trading research and execution. An effective programming skill is like an effective communication skill. We collaborate with our research tools by "talking" to them. An effective usage of the tools increases the probability of generating effective trading strategies. At the very least, it reduces in the execution systems the number of bugs that could cause millions of dollars of losses.[15] It is easy to hire good mathematicians; you look for them in New York City. It is easy to hire good programmers; you look for them in the Silicon Valley. However, it is extremely difficult to hire someone who can come up and code up advanced mathematical trading strategy. This book focuses on teaching these two skills.

1.5. Technology as Weapon in Quantitative Trading Research

There needs an arsenal of technologies to streamline the quantitative trading research process. Typically, quants and traders, from idea generation to strategy deployment, may take weeks if not months. This means not only loss of trading opportunity (the opportunity could disappear after you finish backtesting), but also a lengthy, tedious, erroneous process marred with ad-hoc decisions and primitive tools. There is usually very little standardization in terms of the quantitative trading research process even within one firm, let alone in the industry. Most small to medium quantitative fund houses do not invest in building research technology.

This is best illustrated by the languages quantitative traders use to do backtesting. When there are six traders, they could be using: MATLAB, R, VBA, C++, Java, C#. The first consequence of using-their-best-language is that there is absolutely no sharing of code. The firm would write the same volatility calculation function six times. Suppose trader A comes up with a new way of measuring volatility, trader B could not leverage on this. Trader C could not quickly prototype a new trading idea by combining the mean-reverting signal from trader D and the trend following signal from trader E. The productivity is very low. Trader F will therefore fail miserably in the chaos due to the lack of effective research tools.

The management is not able to compare strategies from two traders. The six traders all make different "simplifications" in backtesting. For instance, they would clean the same data set in different ways; they would use their "proprietary" bid-ask, execution, (market vs. limit) order book models for computing P&L; they would make different assumptions about slippage, liquidity and market impacts. Mainly due to making coding simpler and/or faster, they would make all sorts of guesses and unfortunate "approximations" about the details in executing their strategies live. The management does not have the time to question every single detail in their backtesting, hence the lack of understanding and confidence. They would simply resort to "trusting" the reports but they should not. This is one major reason why real and paper P&L's often do not agree – one is upward sloping and the other one downward sloping.

Worst, while quantitative traders may be excellent mathematicians, they are usually bad programmers. It is very rare to see a quantitative trader who understands modern programming concepts such as interface vs. inheritance, memory model, thread safety, design patterns, testing and software engineering. They usually produce code that is

[15] For instance, Knight Capital Group suffered $440 million electronic trading loss due to a programming bug in August 2012.

very long, unstructured and poorly documented, hence spaghetti code. The code must have bugs. Ideally traders should spend their time on what they are supposed to be best at doing – imagining new trading ideas or strategies. Realistically they waste too much time on getting the details right on coding and backtesting (and still get them wrong).

Our solution is a technological process that standardizes and automates most of the mundane and mechanical steps in quantitative trading research. First, we mandate that all traders do their backtesting in the same language, e.g., Java. Second, we mandate that all traders contribute their code to a research library. Third, the firm invests in a common backtesting infrastructure by expanding this research library. The advantages are the following.

1. The traders can focus on what they are supposedly good at doing – producing innovative trading strategies. They no longer bother with the IT grunt work.
2. They can quickly prototype a trading idea by putting together components, such as signals, filters, modules, from the research library.
3. They can share code with colleagues and be understood because all conform to the same standard.
4. They can compare strategies because the performance measures are computed from simulations making the same assumptions.

Over time, as the quantitative trading firm invests, creates as well as expands the research library and backtesting system, this IT infrastructure will become the most valuable asset. Suppose a star trader[16], using R, takes 3 months to test a good idea and make it profitable. With this standardized process and technology, a good trader[17] is able to rapidly prototype 30 strategies in 3 days in parallel on a cluster of hundreds or thousands of computers, automating all the backtesting, sensitivity analysis, P&L distribution and risks computations. Technology changes the game by enhancing productivity. The profitability of the firm depends not on hiring Einstein but on good and hardworking people leveraging on using the infrastructure.

As in any battles, weapon is a decisive factor in winning. The state-of-the-art financial weapon for quantitative trading is still at its infancy, like in Stone Age. Using script languages like R is at best very cumbersome and is prone to errors. We foresee that in the future those successful funds that top the ranking are those who have the best quantitative trading research technologies at their disposal.

1.6. AlgoQuant and SuanShu

There are many vendors that sell backtesting tools. However, many of these backtesters are no more than augmented versions of for-loops – looping over historical data. Some better ones come with various features, e.g., importing data,

[16] E.g., Einstein
[17] You and me

cleaning data, statistics and signal computation, optimization, real-time paper trading. Some even go one step further to provide brokerage services for live trading. The major problem with all these backtesters is that you cannot code, and thus simulate, non-trivial mathematical trading strategies with them. Suppose you want to replicate the pair trading strategy in (Elliott, van der Hoek, & Malcolm, 2005), you will need to do in one strategy stochastic differential equation (SDE), expectation-maximization (EM) algorithm, maximum likelihood estimation (MLE), and Kalman filter. Most commercial backtesters are not built to support advanced mathematics and simply cannot do it.

There are a few professional backtesters that provide link to R/MATLAB for data analysis. They may be alright for low frequency trading but are definitely inappropriate for intraday or high frequency trading. Many traders use R only because they cannot code. (It is very rare to find someone who is mathematically talented and can professionally code.) The problems are:

1. They are very slow. These scripting languages are interpreted line-by-line. They are not built for parallel computing. (OK. R can do parallel computing but who use them in practice?[18] Which traders understand immutability to write good parallel code?)
2. They do not handle a lot of data well. EUR/USD tick by tick data from Thomson-Reuters is about 1GB per day. One year of data is 250GB. You cannot even load all of them into the memory in R let alone analysis.
3. There is no modern software engineering tools built for R. For example, it is cumbersome to do automatic unit testing to run a suite of regression tests to ensure new code does not break the old code. There is no way to make sure that the code is correct. Many traders just “hope” that their code is correct because they are not trained programmers.
4. R code cannot be debugged easily. There is no breakpoint, no variable watcher, no stack, no step-in and no step-out. It does not compare to debuggers in NetBeans, Eclipse or IntelliJ IDEA. How do you debug a large R application with 50 pages of scripts anyway? I usually just give up.

AlgoQuant and SuanShu

AlgoQuant[19] by Numerical Method Incorporation Limited is a materialization of the quantitative trading research philosophy and process that we discussed. AlgoQuant is a suite of quantitative trading research tools (with source code for most parts) for quick prototyping of trading ideas. Like other professional backtesting tools, AlgoQuant supports also data import, data filtering and cleaning, in-samples calibration, out-samples simulation and performance analysis. Beyond these, the main competitive advantage that AlgoQuant has over other software is the incorporation of

[18] Recently, there is some effort from Google to use R on a massively parallel computational infrastructure for statistical computing.
[19] You may download a trial copy here: http://numericalmethod.com/algoquant/.

SuanShu, a powerful modern library of mathematics, statistics, optimization and data mining.

AlgoQuant has a large collection of mathematical models, many from top academic journal publications, which a trader can use as building blocks to create his own trading models. For example, to build a mean reversion model, the trader can combine (D'Aspremont, 2011) to construct a maximally mean reverting portfolio of two assets and then trade the pair using (Elliott, van der Hoek, & Malcolm, 2005) pair trading strategy. Both mathematical modules are available in the library. In addition, AlgoQuant has a large number of algorithms that the trader can apply. For example, in the area of portfolio optimization, AlgoQuant covers Quadratic Programming (Markowitz or Modern Portfolio Theory), Second Order Conic Programming as well as Differential Evolution. In other words, using AlgoQuant, the trader does not need external math software like Excel, R, or MATLAB, and there are many readily available modules and algorithms to use. He can quickly build up very complicated mathematical strategies by combining together the components from the library. Thorough this book, we will use AlgoQuant and SuanShu to do demonstrations.