

Wiley Trading Series

MACHINE TRADING

DEPLOYING COMPUTER **ALGORITHMS**
TO CONQUER THE MARKETS

ERNEST P. CHAN

WILEY

MACHINE TRADING

Founded in 1807, John Wiley & Sons is the oldest independent publishing company in the United States. With offices in North America, Europe, Australia, and Asia, Wiley is globally committed to developing and marketing print and electronic products and services for our customers' professional and personal knowledge and understanding.

The Wiley Trading series features books by traders who have survived the market's ever-changing temperament and have prospered—some by reinventing systems, others by getting back to basics. Whether a novice trader, professional, or somewhere in between, these books will provide the advice and strategies needed to prosper today and well into the future.

For more on this series, visit our website at www.WileyTrading.com.



MACHINE TRADING

Deploying Computer Algorithms to Conquer
the Markets

Ernest P. Chan

WILEY

Copyright © 2017 by Ernest P. Chan. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600, or on the Web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993, or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Cataloging-in-Publication Data is available:

ISBN 978-1-119-21960-6 (Hardcover)

ISBN 978-1-119-21967-5 (ePDF)

ISBN 978-1-119-21965-1 (ePub)

Cover Design: Wiley

Cover Images: Wave © kdrkara90/Shutterstock; Abstract background © Marina Koven/Shutterstock; Fractal Realms © agsandrew/Shutterstock

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

*To my mom, Ching, my spouse, Ben, and to the memory
of my beloved father, Hung Yip.*

CONTENTS

Preface	ix	
CHAPTER 1 The Basics of Algorithmic Trading	1	
CHAPTER 2 Factor Models	27	vii
CHAPTER 3 Time-Series Analysis	59	
CHAPTER 4 Artificial Intelligence Techniques	83	
CHAPTER 5 Options Strategies	119	
CHAPTER 6 Intraday Trading and Market Microstructure	159	
CHAPTER 7 Bitcoins	201	
CHAPTER 8 Algorithmic Trading Is Good for Body and Soul	215	
Bibliography	227	
About the Author	235	
Index	237	

PREFACE

The best way to learn something really well is to teach it to someone else (Bargh and Schul, 1980). So I confess that one major motivation for my writing this book, the third and the most advanced to date in a series, is to force myself to study in more depth the following topics:

ix

- The latest backtesting and trading platforms and the best and most cost-effective vendors for all manners of data (Chapter 1);
- How to pick the best broker for algorithmic executions and what precautions we should take (Chapter 1);
- The simplest way to optimize allocations to different assets and strategies (Chapter 1);
- Factor models in all their glory, including those derived from the options market, and why they can be useful to short-term traders (Chapter 2);
- Time series techniques: ARIMA, VAR, and state space models (with hidden variables) as applied to practical trading (Chapter 3);
- Artificial intelligence/machine learning techniques: particularly methods that will reduce overfitting (Chapter 4);
- Options and volatility trading strategies, including those that involve portfolios of options (Chapter 5);

- Intraday and higher frequency trading: market microstructure, order types and routing optimization, dark pools, adverse selection, order flow, and how to backtest intraday strategies with tick data (Chapter 6);
- Bitcoins: bringing some of the techniques we covered to this new asset class (Chapter 7);
- How to keep up with the latest knowledge (Chapter 8);
- Transitioning from a proprietary trader to an investment advisor (Chapter 8).

I don't know if these topics will excite you or bring you profits, but my study of them has certainly improved my own money management skills. Besides, sharing knowledge and ideas is fun and ultimately conducive to creativity and profits.

You will find most of the materials quite accessible to anyone who has some experience in a quantitative field, be it computer science, engineering, or physics. Not much prior knowledge of trading and finance is assumed (except for the chapter on options, where we do assume basic familiarity). However, if you are completely new to trading, you may find my more basic treatments in *Quantitative Trading* (Chan, 2009) and *Algorithmic Trading* (Chan, 2013) easier to understand. This book can be treated as a continuation of my first two books, with coverage on topics that I have not discussed before, but it can also be read independently.

Although many prototype trading strategies have been included as examples, one should definitely not treat them as shrink-wrapped products ready to deploy in live trading. As I have emphasized in my previous books, nobody should trade someone else's strategies without a thorough, independent backtest, removing all likely sources of biases and data errors, and adding various variations for improvement. Most, if not all, the strategies I describe contain hidden biases in one way or another, waiting for you to unearth and eliminate.

I use MATLAB for all of my research in trading. I find it extremely user-friendly, with constantly improving and new features, and with an increasing number of specialized toolboxes that I can draw on. For example, without the Statistics and Machine Learning Toolbox, it would take much longer to explore using AI/ML techniques for trading. (See why Google scientist and machine learning expert Kevin Murphy prefers MATLAB to R for AI/ML research in Murphy, 2015.) In the past, readers have complained about the high price of a MATLAB license. But now, it costs only \$150 for a

“Home” license, with each additional toolbox costing only \$45. No serious traders should compromise their productivity because of this small cost. I am also familiar with R, which is a close relative to MATLAB. But frankly, it is no match for MATLAB in terms of performance and user-friendliness. A detailed comparison of these languages can be found in Chapters 1 and 6. If you don’t already know MATLAB, it is very easy to request a one-month trial license from mathworks.com and use its many free online tutorials to learn the language. One great advantage of MATLAB over R or other open-source languages is that there is excellent customer support: If you have a question, just email or call the staff at Mathworks. (Often, someone with a PhD will answer your questions.)

I have taught many of these topics to both retail and institutional traders at my biannual workshops in London, as well as online (www.epchan.com). In order to facilitate lecturers who would like to use this as a textbook for a special topics course on Algorithmic Trading, I have included many exercises at the end of most chapters. Some of these exercises should be treated as suggestions for open-ended projects; there are no ready-made answers.

Readers will also find all of the software and some data used in the examples on epchan.com/book3. The userid and password are embedded in Box 1.1. But unlike my previous books, some of the data involved in the example strategies are under strict licensing restrictions and therefore are unavailable for free download from my website. Readers are invited to purchase or rent them from their original sources, all of which are described in Chapter 1.

I have benefited from tips, ideas, and help from many people in putting the content together. An incomplete list would include:

- Stephen Aikin, a renowned author (Aikin, 2012) and lecturer, who helped me understand implied quotes due to calendar spreads in the futures markets (Chapter 6).
- David Don and Joseph Signorelli of Lime Brokerage, who corrected some of my misunderstanding of the market microstructure (Chapter 6).
- Jonathan Shore, infinitely knowledgeable about bitcoins, who helped compile some order book data in that market and shared that with me (Chapter 7).
- Dr. Roger Hunter, CTO at our firm, QTS Capital Management, who reviewed my manuscript and who never failed to find software bugs in my codes.

- The team at Interactive Brokers (especially Joanne, Ragini, Mike, Greg, Ian, and Ralph) whose infinite patience with my questions about all issues related to trading are much appreciated.

I would like to thank Professor Thomas Miller of Northwestern University for hiring me to teach the Risk Analytics course at the Master of Science in Predictive Analytics program. In the same vein, I would also like to thank Matthew Clements and Jim Biss at Global Markets Training for organizing the London workshops for me over the years. Quite a few nuggets of knowledge in this book come out of materials or discussions from these courses and workshops.

Trading and research have been made a lot more interesting and enjoyable because I was able to work closely with our team at QTS, who contributed to research, ideas, and general knowledge, some of which find their way into this book. Among them, Roger, of course, without whom there wouldn't be QTS, but also Yang, Marcin, Sam, and last but not least, Ray.

Of course, none of my books would come into existence without the support of Wiley, especially my long-time editor Bill Falloon, development editor Julie Kerr, production editor Caroline Maria, and copy editor Cheryl Ferguson (from whom no missing “end” to a “for”-loop can escape). It was truly a great pleasure to work with them, and their enthusiasm and professionalism are greatly appreciated.

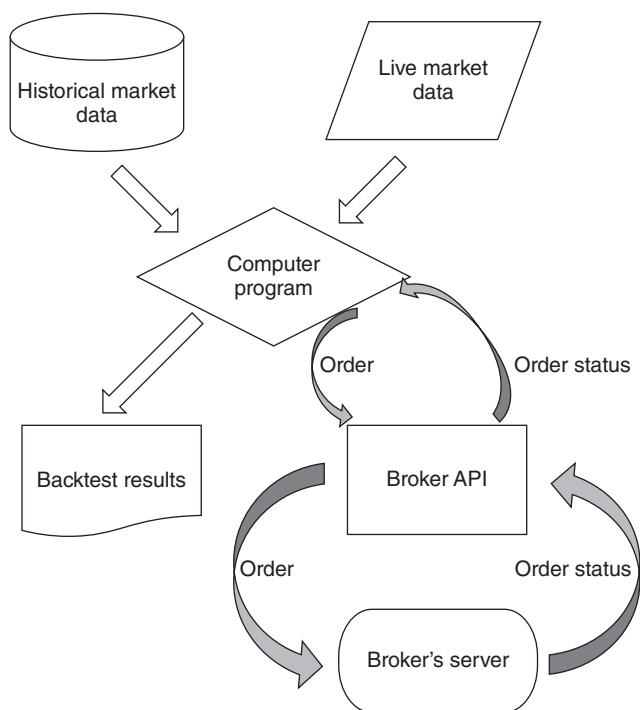
MACHINE TRADING

The Basics of Algorithmic Trading

An algorithmic trading strategy feeds market data (historical or live) into a computer (backtest or automated execution) program. The program then submits orders to the broker through an API, and receives order status notifications back from the broker. The flowchart in Figure 1.1 illustrates this process.

Notice that I deliberately use the same box to indicate the computer program that generates backtest results and live orders: This is the best way to ensure we are trading the exact same model that we have backtested.

In this chapter, I will discuss the latest services, products, and their vendors applicable to each of the blocks in Figure 1.1. In addition, I will describe my favorite performance metrics, the way to determine the optimal leverage, and the simplest asset allocation method. Though I have touched on many (but not all) of these issues in my previous books, I have updated them here based on the state of the art. The FinTech industry has not been standing still, nor has my understanding of issues ranging from brokers' safety to subtleties of portfolio optimization.

FIGURE 1.1 Algorithmic trading at a glance

■ Historical Market Data

For **daily historical data in stocks and futures**, I have been using CSI (csidata.com) for a long time. CSI has a very flexible, and robust, desktop application. The beauty of this application is that we can set a time in the evening when the data are automatically updated through an Internet connection with CSI's server. Also, the data can be stored in various convenient formats such as .txt, .csv, or .xlsx. We can ask it to automatically adjust historical stock (and ETF) prices for splits and dividends. For a little extra, CSI can also provide delisted stocks' historical data, so that you can have a survivorship-bias-free data set.¹ (By the way, CSI data powers Yahoo! Finance's historical stock data.) For futures, we can choose different rollover methods to create continuous contracts. Or not, since the original contract prices are also available, and many professional traders prefer to backtest futures using original contract prices instead of back-adjusted continuous contract prices. This is because the latter depends on a particular roll method, and may have look-ahead bias embedded (see Chan, 2013,

for a detailed exploration of this issue). Finally, CSI has excellent customer support through email and phone.

An alternative to CSI is Quandl.com, which is a consolidator of many kinds of data from many different vendors. It also provides an API in different languages (including MATLAB, which I use in this book, or Python, which many other traders use) that we can use for data selection and download. Some of Quandl's data are free (daily data for stocks is one example), and others require payment. I have purchased, for example, fundamental stock data from them (see Chapter 2, Factor Models), and they are much more economical than established vendors such as Compustat.

Serious traders or academic finance researchers may prefer stock, ETF, and mutual fund data from CRSP (www.crsp.com). Their historical data are carefully compiled to be survivorship-bias-free, and dividends and splits are provided separately so you can decide how to utilize them in a backtest. But most importantly, they provide the best bid and offer (BBO) prices at the close. These are important because, as is explained in Box 6.2 in Chapter 6, using the usual consolidated closing prices from CSI or Quandl can inflate backtest performances for certain strategies. A similar issue arises from using the consolidated opening prices. The best open and close prices to use are the auction prices from the primary exchange. (See also Box 6.2 for an explanation of how we can extract such auction prices from tick data.) The second best open and close prices to use are the BBO prices that can be used to compute the midprices at the open and close. Unfortunately CRSP does not provide the BBO prices at the open, so one must use intraday data for that purpose. For academic researchers, CRSP data can be obtained at a lower cost through WRDS (wrds-web.wharton.upenn.edu), which is a consolidator of many high quality historical databases for scholarly research.

Of course, those serious traders who can afford to buy data from CRSP may also be able to afford a Bloomberg terminal subscription. One advantage of a Bloomberg terminal is that we can download the “primary exchange” close price for US stocks. Of course, a Bloomberg subscription also includes access to many historical and live data spanning a wide variety of instruments and, importantly, breaking news on every stock. I have found Bloomberg's news service to be superior to many other vendors'. Often, we will see a stock moves suddenly, and are not able to find the reason anywhere else but on Bloomberg's news feed. They do capture the most obscure news on the most obscure stocks in the shortest time frame, which is important when you have an event-driven strategy. Bloomberg's historical US stock data are also survivorship-bias-free. (To be fair to Thomson Reuters's Eikon platform,

which is a keen competitor to Bloomberg's, I have not tested its news feed. So it is possible that it provides just as wide and timely coverage as well. There is one feature on Eikon that impressed me in a demo: I was able to see the geographical locations of individual oil tankers and where they were headed. Apparently, this is useful to oil traders to predict short-term oil inventory, supply, and demand.)

For **futures traders**, daily data does not present much of a problem. CSIs and the free data on Quandl are as good as any.

Daily options data can be obtained from ORATS.com as well as ivolatility.com. Both offerings are survivorship-bias-free. The institutional trader or academic researcher may also purchase from Option Metrics, which is often part of the WRDS package (see above). One nice feature of all these databases: They do not include just option closing prices, but also the bid-ask quote at the close as well. This is important because some options, especially ones that are out-of-the-money or have long tenor, may be traded infrequently. So the last trade price of the day may be very different from the bid-ask quotes at the close, and is not indicative of the price we can actually trade at or near the close. These databases also include auxiliary but important information such as the Greeks and implied volatilities. Furthermore, they may include an implied volatility surface. This uses implied volatilities computed on actual options and interpolates them to yield implied volatilities for strikes and tenors that did not actually exist.

Options historical data tend to be more expensive than stock or futures data, due to their voluminous nature. However, it can be cheaper to rent *intraday* option prices from QuantGo.com than to buy *daily* option prices from other vendors. We will talk more about QuantGo when we discuss intraday data in general. It would be a trivial programming exercise to extract the end-of-day quotes from the intraday data, by looking for the quotes with timestamps at or just before the daily closing time.

Beyond daily price data, there are, of course, **fundamental financial data** for companies. I already mentioned that Quandl carries such data. Institutional traders would most likely look to Compustat for such data. For corporate earnings estimates by analysts, Thomson Reuters' IBES database is the standard. Both Compustat and IBES are available from WRDS. Meanwhile, crowd-sourced earnings estimates are available from Estimote. There is some research that suggests Estimote's contributors can more accurately forecast actual earnings than traditional sell-side analysts (Wang et al., 2014). An example strategy using Estimote's data is discussed in Deltix (2014). Short interest data are available from Compustat and

SunGard's Astec database. SunGard's data have a lot more details culled from stock lenders and prime brokers around the Street than a simple short interest number. In addition, their data are available on an intraday basis as a live feed, though the historical data do not have historical time stamps.

News data is another type of data that is becoming fashionable. Many vendors sell elementized news feeds (i.e., news that is machine-readable, which makes it easier to capture keywords, categories, and the like), including Bloomberg, Dow Jones, and Thomson Reuters. If it is too much trouble for your strategy to generate trading signals from raw news data, you can also buy news sentiment data from Ravenpack, Thomson Reuters News Analytics, MarketPsych, or Accern. (AcquireMedia's NewsEdge database is similar, but they provide only impact scores. This is a kind of unsigned sentiment score that doesn't tell you which way the stock will move, only that it will move, which may be suitable for options traders.) However, there is one problem for sentiment data: Different vendors have different ways to compute sentiment scores from the raw news. So a trading model depends to some extent on which vendors' sentiment scores are most predictive.

We will leave the topic of buying or renting intraday data to Chapter 6 on Intraday Trading, because the features associated with intraday data are intimately tied to the market microstructure. Here, we will just note that some of the historical intraday data vendors include tickdata.com, nanex.net, CQG, QuantGo.com, kibot, and of course, the various exchanges themselves.

Finding, buying, or renting data is both expensive and time-consuming, though consolidators like Quandl and QuantGo have made it much less so. Another way to avoid dealing with acquiring data directly is to adopt a trading platform that comes integrated with data (though you may have to pay for the data separately). A good example is Quantopian.com, which provides free US stock trades data with one-minute bars, together with many other forms of fundamental and news data at lower frequency. (I have been told that futures data will be available soon.) We will talk more about platforms like these in the section "Backtesting and Trading Platforms."

■ Live Market Data

Most if not all brokerages provide live market data to their clients, and if you are trading a daily strategy (i.e., you trade only at the market open or close), such data are usually more than sufficient. However, if you engage

in intraday trading, then the quality of data becomes a bigger issue. As we will discuss more thoroughly in Chapter 6, low latency market data can be quite expensive to obtain. Vendors that provide data suitable for intraday trading that can tolerate a latency of more than 25 ms (ms = millisecond) include eSignal, IQFeed, CQG, Interactive Data, Bloomberg, and many others. But vendors that provide data feed with latency of below 10 ms are far fewer: They include S&P Capital IQ (formerly QuantHouse), SR Labs, and Thomson Reuters. Of course, you can also subscribe to the direct feeds from the exchange, but that is strictly for high frequency traders who find the high expense justified by the high return. (That is true unless you are after currency data, where most FX exchanges will give their customer a free direct feed.)

As with historical market data, many trading platforms also include live market data feeds. These will be discussed in the following section.

■ Backtesting and Trading Platforms

Traditionally, we would backtest our trading strategy on one platform (e.g., R) and once successful, write a different program to automate execution of the strategy that utilizes a broker's API. However, this proves to be quite bug-prone: there is no way to ensure that the backtest and the live trading program encapsulate exactly the same trading logic. Fortunately, most backtest platforms nowadays have extended their ability to execute live as well; hence, we will combine the discussions on backtesting and trading platforms here.

As I mentioned in the Preface, MATLAB has been my favorite backtesting platform. It has a very comprehensive and user-friendly interface for developing and debugging programs, and it has a wide array of toolboxes that cover almost every arcane mathematical or computational technique you will likely encounter in trading strategy development. One of these toolboxes, the Trading Toolbox, enables a MATLAB program to connect to a number of brokerages' APIs to receive market data, submit orders, and receive order status notifications. If you prefer not to buy the Trading Toolbox, there are at least three adaptors developed by third-party vendors that enable you to do the same: exchangeapi.com's `quant2ib`, undocumentedmatlab.com's `IB-Matlab`, and Jev Kuznetsov's free MATLAB-to-IB API available at MATLAB's File Exchange. I have discussed these options in some depth in Chan (2013). Finally, MATLAB is fast. (See Chapter 6 for

a comparison of performance speed among MATLAB, R, and Python.) The only drawback for this platform is that it isn't free, but the "Home" license costs only \$150, with each additional toolbox costing an extra \$45. If you plan to buy MATLAB's Toolboxes, here are the three I recommend (in decreasing order of importance): Statistics and Machine Learning, Econometrics, and Financial Instruments (for options traders).

For those who prefer free, open-source platforms, there are always R and Python.

R is very similar to MATLAB: It is an array-processing language, and it has a large variety of specialized "packages" (the analogue of MATLAB's toolboxes), many of them perhaps more sophisticated than MATLAB's due to the large number of academic researchers who use R. There is a GUI development platform called RStudio, but I find its user interface to be quite crude compared to that of MATLAB, and hence debugging productivity is lower. R is also the slowest among the three languages, and the slowness is all the more problematic because, unlike MATLAB or Python, it cannot be compiled into C or C++ code for speeding up. (You can, however, use the Rcpp package to access compiled C++ code from within R.) As for automating executions, you can connect an R program to Interactive Brokers through a package called "IBroker."

Python is a language in the ascendant, though I know of quants who used it for backtesting back in 1998. Aside from being a standalone language of choice for many quantitative traders, platforms such as Quantopian also use it as their strategy specification language. Native Python is not an array processing language (though one can use SciPy packages which do have this feature). While array processing is convenient for backtesting a large number of instruments simultaneously (e.g., portfolio of stocks), it is not useful for writing an automated execution program. Hence, if we were to insist on using the same program for both backtesting and live execution, we can disregard this feature altogether. One major advantage of Python is that its codes can be developed and debugged using Microsoft's Visual Studio, thus piggybacking on the full power of this well-polished development environment. Another integrated development environment² (IDE) for Python that received good reviews is PyCharm. Python's pandas library is a data analysis package similar to R, and the rpy2 package provides an interface to access all R packages. Python isn't as fast as MATLAB, though it is faster than R, and can be compiled into C or C++.³ You can connect a Python trading program to Interactive Brokers for executions via IBPy or a number of other packages.

TABLE 1.1 Ranking of Three Programming Languages for Quant Trading
(Ranked from 1 to 3 where 1 = best ranking and 3 = poorest ranking.)

Feature	MATLAB	R	Python
Ease of use	1	2	2
IDE polish	1	3	1
Speed	1	3	2
Toolboxes	2	1	1
Compilation to C/C++	1	N/A	1
Connectivity to brokers	1	2	2
Customer support	1	N/A	N/A
Price	2	1	1

In Table 1.1, I provide my personal, arguably subjective, ranking of the various features and aspects of the three scripting languages most widely used in trading strategy development.

You may be wondering why I have left out some of the most common programming languages such as C/C++, Java, or C#. The fact is, the most productive way to develop trading strategies is to quickly build prototype programs to test a lot of them, and to quickly discard them if they fail to live up to their promises. Scripting languages (also called REPL⁴ languages) like MATLAB, R, or Python allow us to shorten this research cycle. However, productivity in research is not the same as productivity in building an automated trading system. The latter usually involves the usual bells and whistles of software development such as object-oriented design, exception handling, version control, memory and speed optimization, and so forth. Such bells and whistles are actually quite cumbersome to implement in a scripting language, and, furthermore, it is quite easy to introduce bugs without a robust and extensible software architecture. Typically, once object-oriented design is imposed on a scripting language, it will run too slowly to be useful as an execution system.

To benefit from the best of both worlds, in our firm we do our initial research mostly in MATLAB or Python (though I often ask our research associates to test their strategies on Quantopian first, just to make sure their codes do not have look-ahead bias). After we settle on a strategy, our CTO, Roger, will then independently build a system in C# that can both backtest

and execute live the same strategy, as a way to confirm the correctness of the initial backtest. Once it is confirmed, a figurative turn of the key will allow us to trade live, with much lower latency than if we were to execute using a scripting language. In building the execution system, we can often reuse existing classes that we have written for other strategies. This way, we have compressed the research life cycle without sacrificing software correctness, component reuse, or execution efficiency.

There are now many backtesting and automated execution platforms available that purport to make it easier to develop and deploy automated trading strategies both quickly and robustly, just like what I described in the last paragraph, but without using two different languages. I have written extensively about this in Chan (2013), so here I will restrict myself to those platforms that fit the following criteria:

1. It has integrated historical and live market data, or provides adaptors for popular data vendors.
2. It allows maximum flexibility in strategy design by relying on generic programming languages such as Python or Java.
3. It allows connection to popular broker APIs.
4. It allows backtesting and live trading in US equities, among other instruments.

The platforms that satisfy these criteria are Quantopian, QuantConnect, Ninjatrade, Marketcetera, AlgoTrader, OpenQuant, Deltix, and QuantHouse. (This is, of course, not an exhaustive list.) Some of these platforms, such as Quantopian, are free, while others, such as QuantHouse, have a price tag suitable for institutional traders.

If you are trading a high frequency strategy, it is possible that many of these platforms will not be adequate: either because they may be too slow, or because they are missing level 2 quotes. But there are platforms such as Lime Brokerage's Strategy Studio that are designed specifically for such demanding tasks. More about this will be discussed in Chapter 6.

Though these platforms have certainly made it easier to backtest and automate trading strategies, they usually come with some constraints on the exact type of strategies, data, or instruments that we are allowed to test. Not surprisingly, the more expensive a platform is, the more flexible it is. But if absolute flexibility coupled with low development cost is required, you will probably have to do what we do in our firm.

■ Brokers

In this day and age, practically every broker will offer customers an API where they can subscribe to market data, submit orders, and receive order status notifications: See the last two blocks on the right of Figure 1.1. Of course, the ease of use and comprehensiveness of their APIs differ widely. If you would rather not deal directly with the vagaries and low-level details of each broker's API, you can use one of the automated execution platforms that I described in the previous section.

Meanwhile, commissions are generally so low that it also isn't a crucial factor in deciding who to trade with. Brokers have other ways to make money from you besides commissions. One popular way, for some stockbrokers, is "payment for order flow." Essentially, when you send your order to this broker, it will forward your order to a particular market maker or a specific exchange for a rebate (e.g., a penny per share), and let them execute this order. Brokers who do this must inform their customers of this practice when they open their accounts. If you prefer not to let your broker earn this rebate, and instead want to pocket it yourself, you can use brokers with Direct Market Access (e.g., Interactive Brokers or Lime Brokerage). For some FX brokers, a popular way to earn money is to widen the bid-offer spread on a currency pair. This way, they earn the difference between the spread they offer you and the spread they have to pay in the interbank market where they execute your trade. This can make it hard to compare costs among different FX brokers, since this extra spread may change over time, but of course this is precisely the reason why some brokers adopt the practice. If you prefer transparency, you can restrict your search to FX brokers that only charge commissions, not spread.

If you are a futures or currency trader, there is one more item about your broker that you have to worry about: financial stability. For US securities (i.e., stocks trading) accounts, many traders know that their cash⁵ is insured up to \$250,000 by the Securities Investor Protection Corporation (SIPC). Many brokers also provide additional insurance through a private insurance company such as Lloyd's of London. However, none of this is applicable to commodities futures trading accounts—hence the furor in wake of the MF Global and PFGBest bankruptcies (MarketWatch, 2012). It would be pointless to advise you to ascertain the financial strength of a commodities broker before opening an account: If the National Futures Association (NFA) couldn't detect that some of these large brokers (called Futures Commission Merchants, or FCM) are not meeting capital requirement or are

committing fraud, what chance do we have? However, there is one way out of this conundrum: If a securities broker is also an FCM, it might offer a way to automatically sweep the excess cash⁶ from an uninsured commodities account into an insured securities account. (For example, Interactive Brokers offers such a cash sweep service.)

Currency traders may have heard of *Herstatt risk*, or settlement risk (Galati, 2002). To illustrate, let's say we have sold 1 million EURUSD to a bank, and have delivered 1 million EUR to it. But before this bank can send us the corresponding amount of USD, it collapses (as was the case with Bankhaus Herstatt in Germany, 1974). We won't be getting our USD any time soon, and have lost 1 million EUR. It is a risk that is not solved by having an account at a reputable FX broker, because your counterparty may be some bank in a faraway country, and your broker isn't liable for your loss. This scenario is often used by some uninformed financial advisors to scare customers off investing in foreign currency strategies or funds. However, since the establishment of the CLS bank in 2002, an institution owned by some of the largest banks in the world, this risk has been largely eliminated. The CLS bank is a US financial institution regulated by the US Federal Reserve, and it facilitates simultaneous settlement of both legs of a currency transaction. It can do so because it maintains accounts with 18 central banks around the world. In our example above, we will receive the USD payment at the same time our counterparty receives our EUR payment, all via the CLS bank. It is almost like transacting on a stock exchange, where we are guaranteed that if we sold some shares, payment for those shares would be forthcoming, even if the trader who bought our shares went bankrupt. The 18 central banks whose currencies settle in CLS are listed at www.cls-group.com/About/Pages/Currencies.aspx, and these are the currencies that have little or no settlement risk.

We have been discussing the risks that other parties (brokers or counterparties) fail to pay us in a transaction. But what if we are the ones who fail? More specifically, what if in a levered transaction, we lost more money in the brokerage account than our account equity? Are we liable to the negative equity? Can we lose more than what we invested in the account? The answer is yes and yes, if we invested as individuals. In a particularly poignant example, on January 15, 2015, the Swiss National Bank (Switzerland's central bank) suddenly announced that the Swiss Franc would no longer be pegged to the euro (Economist, 2015). EURCHF plummeted by about 10 percent in seconds, and ultimately lost 29 percent in a day. Many FX traders' account equities went negative. Some smaller FX