

Appendix C — Development Platforms

Technical analysts who plot price and volume, inspect the chart, and use their experience and expert judgement to trade, can use one of the many excellent charting packages to prepare their data.

Quantitative technical analysts use precisely defined logical and mathematical expressions to identify signals and trades. Whether simple, such as moving averages, or complex, such as support vector machines using non-linear kernels, the mathematics is invariably implemented as a computer program.

Your success as a quantitative technical analyst depends very much on your ability to understand the mathematics and logic that represent these techniques, and to accurately translate and implement them as computer programs.

There is no alternative. You must understand the techniques and the associated math. You must be able to read, write, and execute computer programs.

Two Approaches

One uses the traditional trading system development platform, computes indicators and identifies patterns, and observes how prices change over the period of time that follows. Call this the *indicator and what follows* approach.

The other uses data science and machine learning. In it, we first identify the trades that have the desired characteristics, then look for persistent patterns that precede those trades. Call this the *notable trade and what precedes it* approach.

Figure C.1 shows the development process and illustrates that the two approaches occupy the same position in the development process. You may decide to do all your development using one platform, or you may decide to use both.

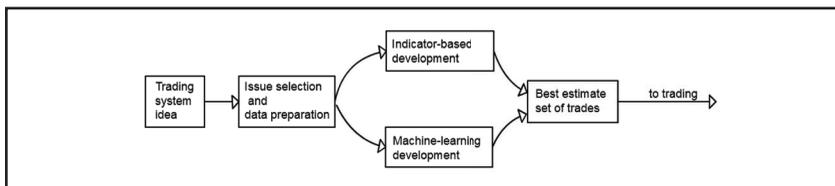


Figure C.1 Two development paths

Trading System Development Platform

For traditional technical analysis, chart-based, and indicator-based systems my choice has long been, and continues to be, AmiBroker. I recommend AmiBroker as being the best available to individuals and small trading companies based on its superior capabilities alone. A nice bonus is its reasonable cost.

AmiBroker is distributed from its website.¹ A free and fully functional trial is available. There is extensive support and reference material on the AmiBroker site. Complete instructions for installing AmiBroker and setting up several sources of data, as well as a set of ten graduated exercises that guide new users through the capabilities of the program, can be found in my book, *Introduction to AmiBroker*,² available free in downloadable pdf format.³

My earlier books, *Quantitative Trading Systems*,⁴ *Mean Reversion Trading Systems*,⁵ and *Quantitative Technical Analysis*⁶ address the design, testing, and validation of trading systems from the indicator-based perspective. They all use AmiBroker to illustrate and implement the ideas and techniques.

AmiBroker runs under Windows. Both 32 bit and 64 bit operating systems are supported, and AmiBroker itself comes in both 32 bit and 64 bit versions.

All versions of AmiBroker—trial and registered, standard and professional, end-of-day and real-time—begin with a visit to the AmiBroker web site and downloading the installation file. As of this writing, the latest version is 6.10. Click the Download AmiBroker link, choose the 32 bit or 64 bit version, and download the installation file. The file for the 64 bit version is about 9 MB. Save it to your hard disc. Double click the file to begin installation. Accept all defaults.

When the installation is complete, launch AmiBroker. The installation will have created an AmiBroker icon on your desktop. Double-click that. When AmiBroker starts, it displays a message that it is Standard

1 <https://www.amibroker.com/>

2 Bandy, Howard, *Introduction to AmiBroker, Second Edition*, Blue Owl Press, 2012.

3 <http://www.introductiontoamibroker.com/>

4 Bandy, Howard, *Quantitative Trading Systems, Second Edition*, Blue Owl Press, 2011.

5 Bandy, Howard, *Mean Reversion Trading Systems*, Blue Owl Press, 2013.

6 Bandy, Howard, *Quantitative Technical Analysis*, Blue Owl Press, 2015.

Edition and Unregistered. This is the trial version, but will be instantly converted to Registered (and Professional Edition, if you requested that) when your AmiBroker Registration file is processed. In the mean time, AmiBroker is ready for your use in Trial mode. The installation includes a small database suitable for testing.

The User's Guide is included in the download, is indexed and searchable, and is accessible by pressing the F1 key when in AmiBroker.

Data Science and Machine Learning

For the data science approach, there are literally hundreds of computer languages that could be used. Ideally, the one we use will have these characteristics and capabilities:

- Readily available
- Inexpensive
- Widely supported
- Easily learned
- Flexible input and output
- Flexible display of graphical output
- Efficient at handling financial time series
- Efficient numeric computation
- Efficient matrix computation
- Library of machine learning applications
- Library of statistical analysis applications

As recently as 2010, the choice was not clear, and the solution involved multiple languages and applications packages inefficiently cobbled together. Components typically included, among many others:

- General programming: C++, Java, Visual Basic, Python.
- Financial time series: AmiBroker, TradeStation.
- Statistical analysis: R, SPSS, Stata.
- Scientific library: C, C++, FORTRAN, MATLAB, Excel.

Python was just one of the contenders to be a single-application platform for quantitative analysis.

Python

In 2011, with the publication of *pandas*⁷ by Wes McKinney, it became crystal clear that Python had risen to be the top choice. McKinney developed *pandas* for his own use while a quantitative analyst at AQR Capital Management. *Pandas* provides several capabilities—most importantly the DataFrame data structure and the set of functions that handle financial time series. His book⁸ is essential reading.

7 <http://pandas.pydata.org/>

8 McKinney, Wes, *Python for Data Analysis*, O'Reilly, 2012.

The Python language was developed, beginning in 1989, by Guido van Rossum in the Netherlands. It is open source. It is free. It runs under Windows, Mac OS, UNIX / Linux, and even Raspberry Pi, among others.

It was released to the public in February 1991; Version 2.0 was released in October 2000, and Version 3.0 in December 2008. By design, Python is highly readable and easily programmed. It uses English keywords and indentation to define program structure. It is highly extensible. When I first wrote this paragraph, January 2014, Python was supported by over 38,000 libraries and packages. Currently, July 2016, there are over 84,000 in PyPI alone⁹.

Python, with the scientific stack of NumPy, SciPy, pandas, and matplotlib, is exceptionally easy to work with, well supported, powerful, and inexpensive. Many packages, such as scikit-learn, are available to provide classification, regression, dimension reduction, and other modeling and machine learning capabilities. If the analysis requires faster execution, critical routines can be written in compiled languages such as FORTRAN or C++ and linked in transparently. If the analysis requires functions specific to R, MATLAB, or Excel, data can be prepared in Python, processed in those programs, with results returned to Python for further analysis. It meets all of the requirements to be a single-language, data science-based, quantitative technical analysis platform.

The components you will need, all of which are free, are:

- Python—base language
- NumPy—support for multidimensional arrays
- SciPy—modules for scientific computing
- pandas—time series library
- matplotlib—plotting library
- scikit-learn—machine learning

Environments

Your Python environment can be on your machine, or you can use one of the many web-based or cloud-based service providers. Some of the on-line choices are free services whose main application is on-line tutorials, such as:

- Learn Python
www.learnpython.org
- Code Academy
www.codecademy.com
- Online Python Tutor
www.pythontutor.com

⁹ <https://pypi.python.org/pypi>

Others are powerful cloud-based sites, intended to support always-on computation, such as Python Anywhere

www.pythonanywhere.com

And joint-effort program development, such as cloud9

<https://c9.io>

On-line services are easy to use, accessible from any computer, and do not require installation of software on your computer.

Trading system development is computationally intensive, and you will soon reach the limits of the free on-line services provide. Assuming you might want to install a Python development environment on your own computer, there are some excellent alternatives.

In order to make easy use of Wes McKinney's book, I recommend using the operating environment he uses. That is Python version 2.7, iPython, NumPy, SciPy, pandas, and matplotlib.

One reference that stands out is Dr. Allen Downey's book, *Think Python*.¹⁰ It is an introduction to both Python and to computer science. The book is available in print, and also as a free e-book.¹¹ Downey's website provides downloadable code for programs in his book.¹²

Spyder

Spyder is an open source cross-platform interactive development environment for scientific programming in Python. Spyder integrates Python with the support libraries, a program editor and file manager, and the iPython execution environment. The Spyder website¹³ has documentation, including links to several installation options.

Installation

I use and recommend the Continuum Analytics Anaconda implementation of the Python environment. It is free. All of the support libraries we need, as well as Spyder, are installed in a single operation using the default Anaconda installer. Begin with a visit to their website.

<https://store.continuum.io/cshop/anaconda/>

Should you eventually need it, Anaconda supports *Accelerate* and *CUDA*, add-ons for Anaconda that make use of multi-core CPUs and GPUs.

¹⁰ Downey, Allen, *Think Python*, O'Reilly, 2012.

¹¹ <http://www.greenteapress.com/thinkpython/>

¹² <http://www.greenteapress.com/thinkpython/code/>

¹³ <https://pythonhosted.org/spyder/>

Reference

My *Quantitative Technical Analysis* book includes an introduction to machine learning. It demonstrates how to translate trading systems written in AmiBroker into Python, outlines data preparation for machine learning, and gives complete and ready-to-run code that implements safe-f and dynamic position sizing.

Tutorials

Please read some of the books and visit some of the sites that specialize in Python tutorials. While you do eventually need to become a competent programmer, you can continue reading this book while you practice Python.