Programming and Backtesting Quantitative Trading Strategies

AlgoQuant – A Quantitative Trading Research Toolbox

Haksun Li

haksun.li@numericalmethod.com
www.numericalmethod.com
Speaker Profile

- Dr. Haksun Li
- CEO, Numerical Method Inc.
- (Ex-)Adjunct Professors, Industry Fellow, Advisor, Consultant with the National University of Singapore, Nanyang Technological University, Fudan University, the Hong Kong University of Science and Technology.
- Quantitative Trader/Analyst, BNPP, UBS
- PhD, Computer Sci, University of Michigan Ann Arbor
- M.S., Financial Mathematics, University of Chicago
- B.S., Mathematics, University of Chicago
The Role of Technology in Quantitative Trading
Step 2 - Coding

- After modeling, we code up the quantitative trading strategy for:
  - backtesting (in-sample and out-sample),
  - computing the properties, e.g., expected P&L, max drawdown, using simulation,
  - calibrating parameters,
  - analyzing sensitivity,
  - trading live.
Ideal

- A trader dreams of a trading idea.
- He can quickly code it up to produce a prototype.
- He feeds the prototype to a computer system to automatically produce a report about:
  - backtesting (in-sample and out-sample)
  - computing the properties, e.g., expected P&L, max drawdown, using simulation,
  - calibrating parameters,
  - analyzing sensitivity,
- When he is happy with the report, he can trade the model live.
Building Blocks

- Moving average crossover, Acar & Satchell 2002
- Bull/bear market probabilities, Dai 2011
- Cointegration
- Pairs trading model calibration, Elliott 2005
- Mean reverting portfolio construction, d'Aspremont 2008
- Mean-variance portfolio optimization, Lai 2009
- Cone optimization of portfolio
- Factor models
- Many more...
Creating Strategy Like Building LEGO®

- Moving average crossover
- Cointegration
- Stoploss
- Portfolio optimization

$
Reality

- Clean data
- Align time stamps
- Read Gigabytes of data
  - Reuters’ EURUSD, tick-by-tick, is 1G/day
- Extract relevant information
  - PE, BM
- Handle missing data
- Incorporate events, news and announcements
- Code up the quant. strategy
- Code up the simulation
  - Bid-ask spread
  - Slippage
  - Execution assumptions
- Wait a very long time for the simulation to complete
- Recalibrate parameters and simulate again
- Wait a very long time for the simulation to complete
- Recalibrate parameters and simulate again
- Wait a very long time for the simulation to complete
- Debug
- Debug again
- Debug more
- Debug even more
- Debug patiently
- Debug impatiently
- Debug frustratingly
- Debug furiously
- Give up
- Start to trade
Research Tools – Very Primitive

- Excel
- MATLAB/ R/ other scripting languages...
- MetaTrader/ Trade Station
- RTS/ other automated trading systems...
R/ Scripting Languages Advantages

- Most people already know it.
  - There are more people who know Java/C#/C++/C than Matlab, R, etc., combined.
- It has a huge collection of math functions for math modeling and analysis.
  - Math libraries are also available in SuanShu (Java), NMath (C#), Boost (C++), and Netlib (C).
R Disadvantages

- TOO MANY!
Some R Disadvantages

- Way too slow
  - Must interpret the code line-by-line
- Limited memory
  - How to read and process gigabytes of tick-by-tick data
- Limited parallelization
  - Cannot calibrate/simulate a strategy in many scenarios in parallel
- Inconvenient editing
  - No usage, rename, auto import, auto-completion
- Primitive debugging tools
  - No conditional breakpoint, disable, thread switch and resume
- Obsolete C-like language
  - No interface, inheritance; how to define $f(x)$?
R’s Biggest Disadvantage

- You cannot be sure your code is right!
Productivity
Research Tools As Weapon in Trading Warfare

bare hand star trader → Excel → R/MATLAB

MT/TS → AlgoQuant
A Good Trading Research Toolbox (1)

- Allow easy strategy coding
- Allow plug-and-play multiple strategies
- Simulate using historical data
- Simulate using fake, artificial data
- Allow controlled experiments
  - e.g., bid/ask, execution assumptions, news
A Good Trading Research Toolbox (2)

- Generate standard and user customized statistics
- Have information other than prices
  - e.g., macro data, news and announcements
- Auto parameter calibration
- Sensitivity analysis
- Quick
Free the Trader!

- debugging
- calibrating
- data extracting
- waiting
- backtesting
- data cleaning
- programming
Basic Math Programming in Java
Downloads

- JDK
  - [http://docs.oracle.com/javase/tutorial/](http://docs.oracle.com/javase/tutorial/)
- NetBeans
- AlgoQuant
  - [http://www.numericalmethod.com/trac/numericalmethod/wiki/AlgoQuant](http://www.numericalmethod.com/trac/numericalmethod/wiki/AlgoQuant)
Procedural Programming

- The program is a series of computational steps to be carried out.
- The order of execution is linear from the first statement to the second and so forth
  - with occasional loops and branches.
Debugging and Testing

- F8/F7
- JUnit
Object-Oriented Programming

- Represent “Concepts” as “Objects”.
- Objects have
  - data fields,
  - methods to handle the data fields.
- Inheritance: a hierarchical relationships among objects.
Strategy Programming
Backtesting in AlgoQuant

1. Define a data source, e.g., the asset that you want to trade.
2. Construct an instance of the strategy to be backtested.
4. Run the simulation.
5. Analysis the performance statistics.
Event-Driven Programming

- An object reacts to the events that it listens to.
  - E.g., prices but not volumes.
- A trading strategy code is a set of event handlers that
  - update the internal states,
  - send orders to the (mock) market.
Strategy Programming

- GMA(2,1)
Backtesting

- Historical data
- Bootstrapped data
- Simulated data