CQF Course on Algorithmic Trading

James Isilay & Patrick Shami

27 October, 2016

Agenda

- **9:00 – 10:30 – Introduction to Algorithmic Trading**
  - Our backgrounds.
    - LPPL Project with Didier Sornette.
  - What is Algorithmic Trading
    - Some different trading styles and timeframes
  - Technology stack
    - Platform Pros & Cons
- **10:30 - 11:00 - Morning Break + Software Install**
- **SESSION 2 - 11:00 - 13:00**
  - Research and alpha discovery
    - Order Types
    - Preparing data
    - Back testing
    - Performance metrics
    - Optimisation
      - Exhaustive versus Genetic
    - Back testing – issues
    - Deltix demonstration – strategy session
- **13:00 – 14:00 – Lunch**
- **SESSION 3 - 14:00 - 16:00**
  - **PRACTICAL EXERCISE**
    - Trend Follower
    - Portfolio Trader
    - Experimentation
- **16:00 - 16:20 – Afternoon Break**
- **16:20 -17:00 - QUESTIONS**
  - Other challenges in Algorithmic Trading

Goals of this Course

- Provide a useful overview of algorithmic trading
  - Based on our experiences in the field
- Breadth
  - Touch on the major considerations for an algo trading project
  - Show some alternative platforms and approaches
    - You need to make tradeoffs based on your own situation
  - Discuss the research and alpha discovery process
  - Point out some common mistakes
- Depth
  - Design and code a simple algorithm yourselves
    - Follow our instructor-led template
    - Or, if you are more confident, test your own idea today with our help

Backgrounds

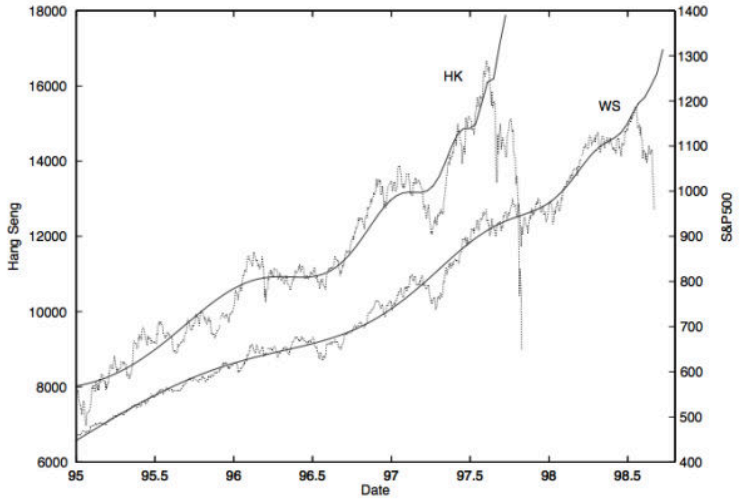
- James Isilay
- Information Systems Engineering, MEng
- CQF 2007
- Algorithmic Trader 2011-2016
  - Initially small systems built on Matlab and executed via CQG on Exchange.
  - Moved to trade on Deltix platform from 2013.
  - Traded ICE Brent, GasOil, WTI fully automated with order execution algorithm.
    - 100 lots per Oil System.
  - Traded German Power and Carbon with trend following strategies
    - 10 MW German Power and 300 lots Carbon.
  - Desk built from scratch with support from Patrick Shami and Peter Osagie.
  - Made ~$350,000 in 2014-2015 , ~$500,000 2015-2016 – which were my targets for the year.
  - Split role also spent 50% of time on quantitative fundamental research i.e. Research Carbon and trade large positions (i.e. 7000 lots) on long term fundamental trade with the desk.
- Founded Startup 2016 – Cognism

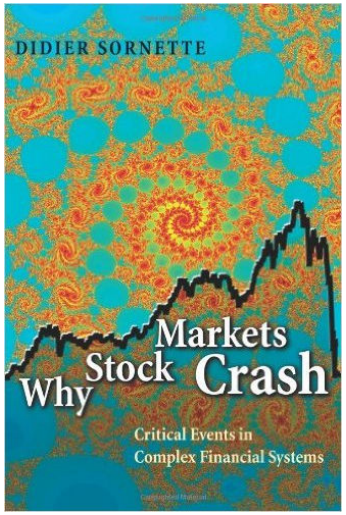
Backgrounds

- Patrick Shami
- Education: BS & MS Electrical Engineering, BA Economics
- CAIA 2008
- Early work experience
  - Business Analyst at McKinsey & Company (Texas office)
    - Served clients in energy and technology industries
  - Engineering roles at multiple California tech companies
    - Satellite communications and Wireless communications industries
    - Signal processing, Systems engineering, Software implementation
- Recent work experience
  - Independent consultant in computational finance
    - Developed trading strategies and quant models for multiple clients
  - Co-founded and co-managed a small commodity trading advisor (CTA)
    - Developed, tested and deployed automated algorithmic trading strategies for clients in an SMA structure

LPPL Strategy

- Professor Didier Sornette, ETH Zurich
  - https://www.ted.com/talks/didier_sornette_how_we_can_predict_the_next_financial_crisis?language=en
- Log-Periodic Power Law (LPPL), which is a log-periodic oscillation model for describing the characteristic behaviour of a speculative bubble and predicting its subsequent crash.

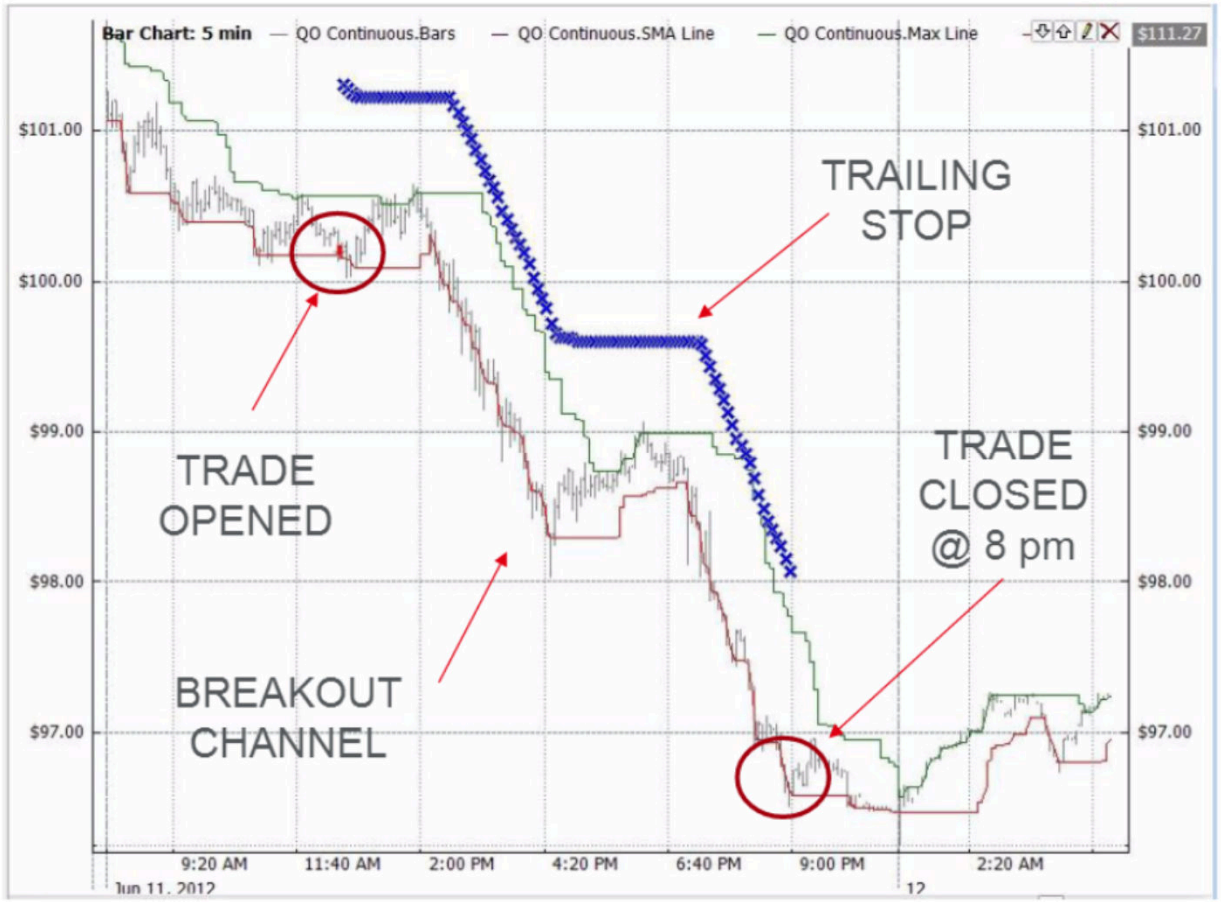




LPPL Strategy

- LPPL on ICE Brent Oil
- Strategy
  - Detect LPPL signal on intraday Oil Price move
  - Wait for Range Breakout – do not take signal if range already breached
  - Trade in the direction of the range breakout – excessive move would continue
  - Trailing stop
  - Close trade before US Oil market close
- Complications
  - Calculation of LPPL Signal
  - Back testing
  - Selecting the Parameters – LPPL = 7 parameter model. Different parameters for different “bubbles” needed a common signature

LPPL Strategy



LPPL Strategy – 10 lot – 2007-2012

4 cent slippage per trade + commission

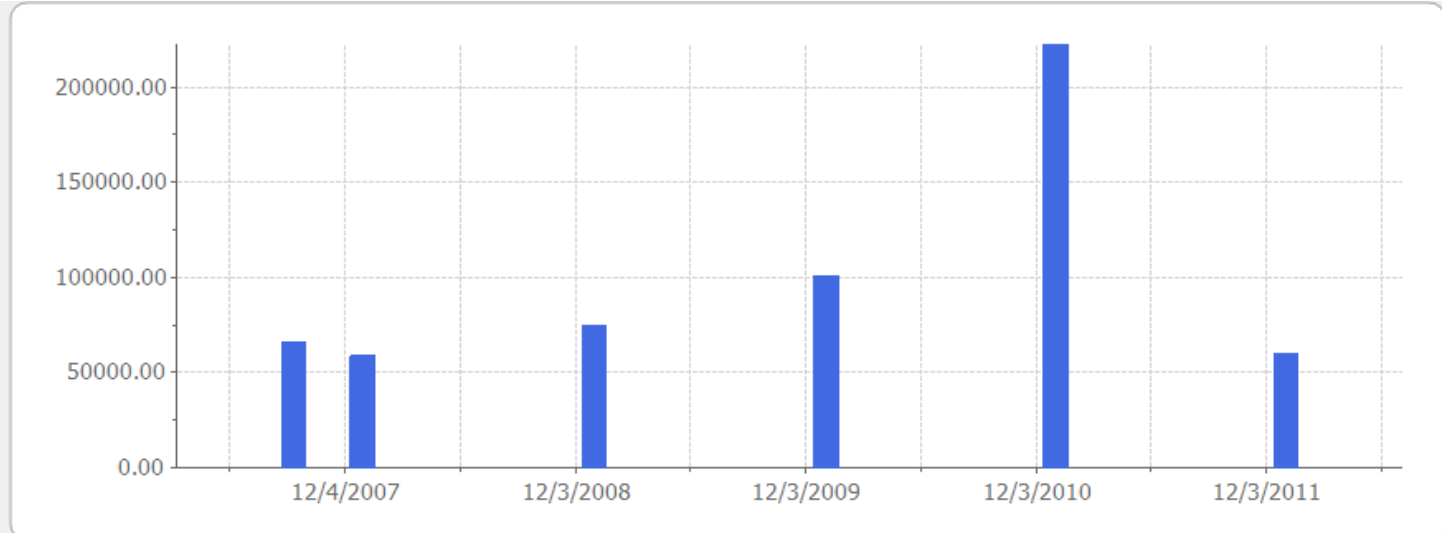
Parameter	All Trades	Long Trades	Short Trades
Net Profit/Loss	828,700.00	355,500.00	473,200.00
Total Profit	2,853,100.00	1,243,500.00	1,609,600.00
Total Loss	-2,024,400.00	-888,000.00	-1,136,400.00
Cumulated Profit %	414.35 %	177.75 %	236.60 %
Max Drawdown	-177,700.00	-138,200.00	-116,500.00
Max Drawdown %	-56.34 %	-40.79 %	-54.88 %
Max Drawdown Date	5/25/2008	6/30/2009	5/25/2008
Return/Drawdown Ratio	4.66	2.57	4.06
Drawdown Days %	83.22 %	83.08 %	85.34 %
Max Drawdown Duration	303	423	180
CAGR	36.42 %	21.38 %	25.88 %
Sharpe Ratio	1.34	0.93	0.96
Annualized Volatility	58.51	36.21	46.75
Sortino Ratio	2.69	1.91	1.96
UPI	0.44	0.14	0.28
Information Ratio	1.34	0.93	0.96
Optimal f	1.06	1.63	1.18
All Trades #	614	290	324
Profitable Trades Ratio	0.46	0.44	0.48
Winning Trades #	283	128	155
Losing Trades #	331	162	169
Average Trade	1349.67	1225.86	1460.49
Average Winning Trade	10081.63	9714.84	10384.52
Average Losing Trade	-6116.01	-5481.48	-6724.26
Avg. Win/ Avg. Loss Ratio	1.65	1.77	1.54
Average Profit per Share	134.97	122.59	146.05
Max Conseq. Winners	11	6	8
Max Conseq. Loosers	10	15	10

LPPL Strategy – 10 lot – 2007-2012

4 cent slippage per trade + commission

LPPL Strategy – 10 lot – 2007-2012

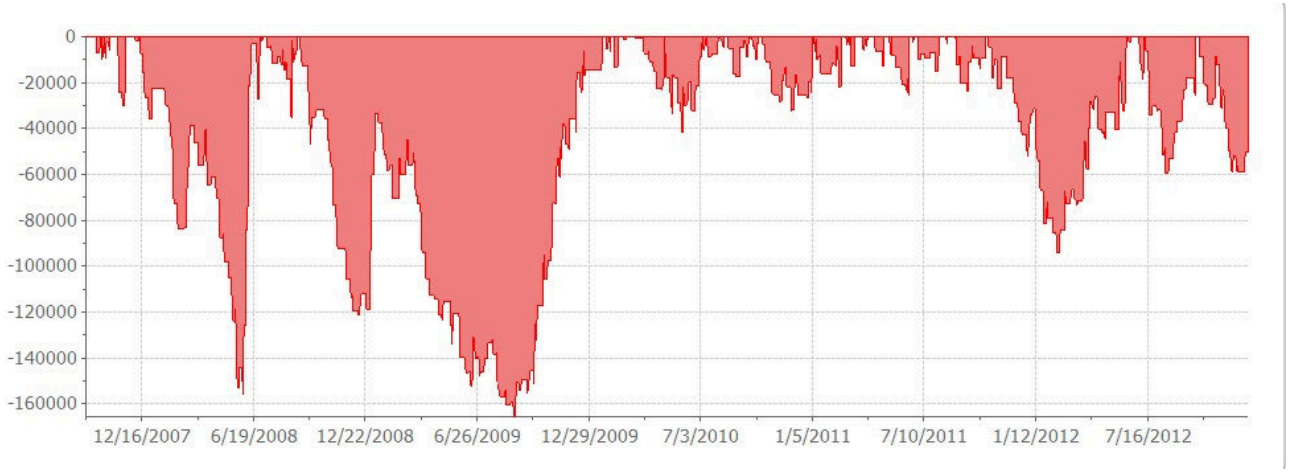
4 cent slippage per trade + commission



#	Date	Return
1	2007	65,892.00
2	2008	58,867.00
3	2009	74,297.80
4	2010	100,932.40
5	2011	222,168.40
6	2012	59,796.40

LPPL Strategy – 10 lot – 2007-2012

4 cent slippage per trade + commission

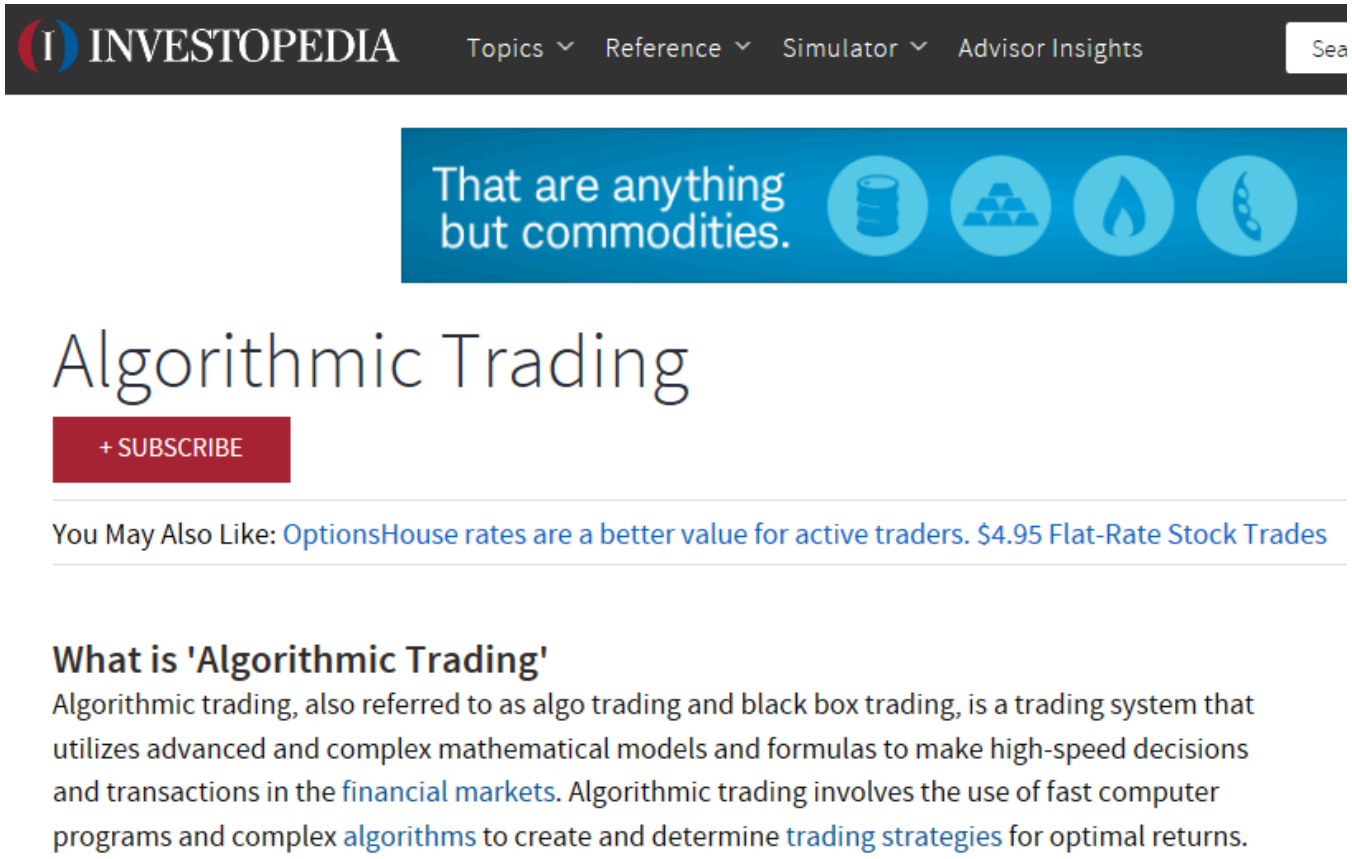


What is Algorithmic Trading?

- Nasdaq's definition
  - An attempt to execute at the best possible price
  - Looking for small trading opportunities

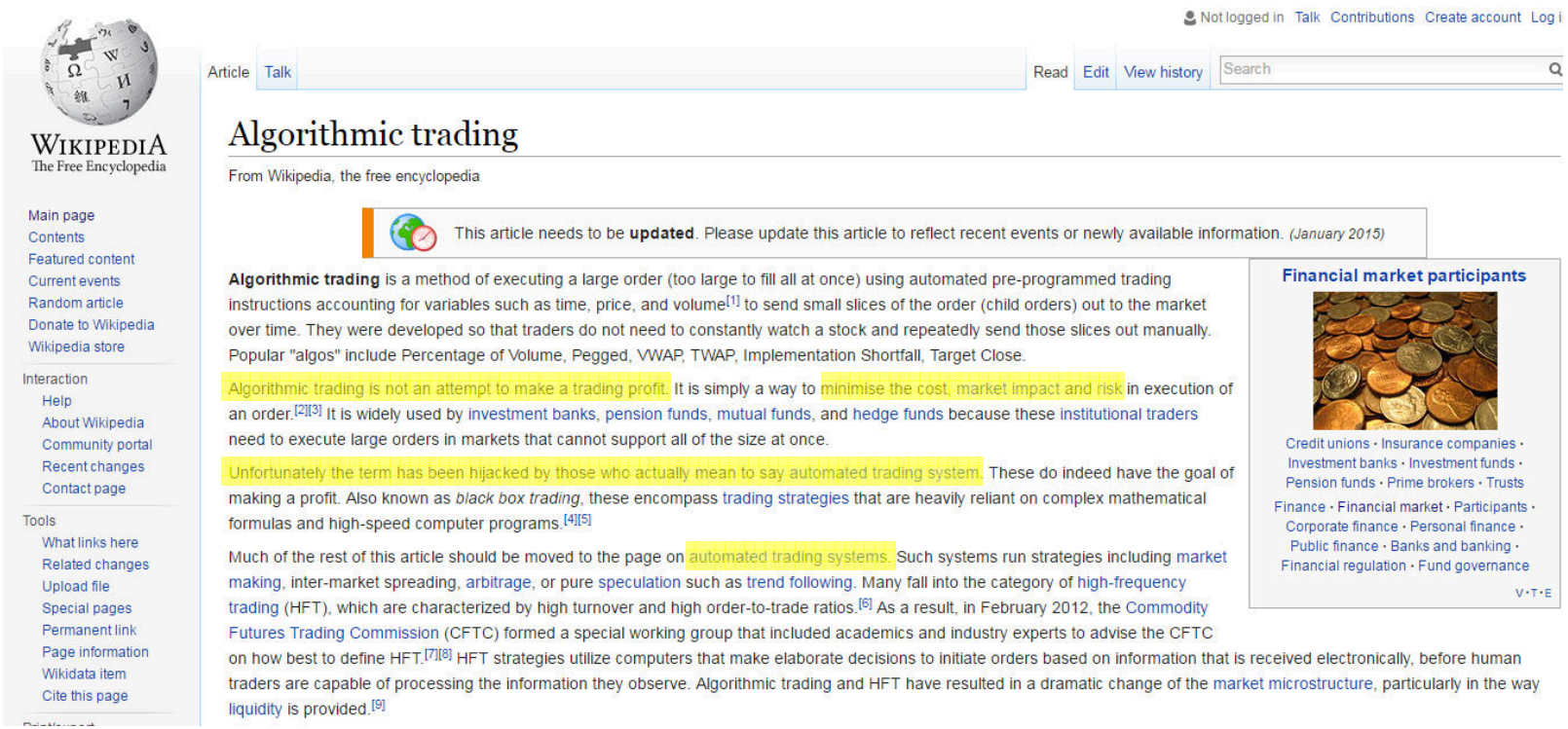
What is Algorithmic Trading?

- Investopedia's definition
  - Advanced and complex mathematical models
  - High-speed decisions
  - Trading strategies for optimal returns



What is Algorithmic Trading?

- Wikipedia definition
  - Large orders
  - Reduce trade cost, market impact, and risk
  - Not profit-seeking



WIKIPEDIA
The Free Encyclopedia

Not logged in Talk Contributions Create account Log i

Article Talk | Read Edit View history | Search

Algorithmic trading

From Wikipedia, the free encyclopedia

This article needs to be **updated**. Please update this article to reflect recent events or newly available information. *(January 2015)*

Algorithmic trading is a method of executing a large order (too large to fill all at once) using automated pre-programmed trading instructions accounting for variables such as time, price, and volume[1] to send small slices of the order (child orders) out to the market over time. They were developed so that traders do not need to constantly watch a stock and repeatedly send those slices out manually. Popular "algos" include Percentage of Volume, Pegged, VWAP, TWAP, Implementation Shortfall, Target Close.

Algorithmic trading is not an attempt to make a trading profit. It is simply a way to minimise the cost, market impact and risk in execution of an order.[2][3] It is widely used by investment banks, pension funds, mutual funds, and hedge funds because these institutional traders need to execute large orders in markets that cannot support all of the size at once.

Unfortunately the term has been hijacked by those who actually mean to say automated trading system. These do indeed have the goal of making a profit. Also known as *black box trading*, these encompass trading strategies that are heavily reliant on complex mathematical formulas and high-speed computer programs.[4][5]

Much of the rest of this article should be moved to the page on automated trading systems. Such systems run strategies including market making, inter-market spreading, arbitrage, or pure speculation such as trend following. Many fall into the category of high-frequency trading (HFT), which are characterized by high turnover and high order-to-trade ratios.[6] As a result, in February 2012, the Commodity Futures Trading Commission (CFTC) formed a special working group that included academics and industry experts to advise the CFTC on how best to define HFT.[7][8] HFT strategies utilize computers that make elaborate decisions to initiate orders based on information that is received electronically, before human traders are capable of processing the information they observe. Algorithmic trading and HFT have resulted in a dramatic change of the market microstructure, particularly in the way liquidity is provided.[9]

Financial market participants

Credit unions · Insurance companies · Investment banks · Investment funds · Pension funds · Prime brokers · Trusts

Finance · Financial market · Participants · Corporate finance · Personal finance · Public finance · Banks and banking · Financial regulation · Fund governance

V·T·E

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

What is Algorithmic Trading?

- Our definition: Algorithmic trading ***is*** Automated trading
  - You wouldn't be taking this course just to learn about VWAP
  - It's a very broad field, with widely varying goals/methods/constraints suited to different situations
- Show of hands, what do you want to cover today?
  - Trading platforms – Tech capabilities and tradeoffs
  - Alpha discovery – Workflow for discovering tradable profit-seeking systems
  - Anecdotes – A sampling of things that have actually happened to us
  - Common pitfalls – Mistakes you want to avoid
  - Other topics?
- What we don't intend to cover
  - Fundamentals of programming
  - Advanced mathematical theory or applications
  - A holy-grail trading system that is guaranteed to win

Algorithmic Trading – Phases and Tasks

- Alpha Discovery
  - Idea generation & research
  - Sourcing historical data
  - Backtesting/simulation
  - Optimization
  - Diversification
  - Statistical tests and reality checks
- Automated Execution
  - Integration with live data-feed
  - Integration with execution venue
  - Event-driven programming
  - Server colocation
  - Server administration
  - Server monitoring
  - Online risk management
- All Phases
  - Programming
  - Debugging
  - Data management
  - System integration

Types of Algorithmic Trading Systems

- Classes of System
  - Hedging an intrinsic risk
    - Banks hedging credit & interest rate risk
    - Airlines hedging fuel risk
    - Utilities hedging power & fuel risk
    - Agri-businesses hedging crop risk
  - Quantitative Active Portfolio Management
    - Filtering & and ranking based on quant factors
    - Constructing an “optimal” portfolio
  - Market-making
    - Capturing the rebate + bid/ask spread over and over again using a set of a high-frequency, low-latency, risk-controlled systems
  - Profit-seeking
    - Single- or multi-instrument trading systems with potentially large swings in profit/loss
    - Includes stat-arb, trend following, mean reversion, prediction, AI, technical analysis, chart patterns, sentiment, text analytics, etc.

- Asset Classes – Show of hands
  - Equities
  - ETF / ETN
  - Futures
  - Currencies
  - Vanilla Options
    - Equities
    - Futures
    - Currencies
  - OTC Derivatives
    - Power contracts
    - Interest rates
    - CDS
    - Currencies

Trading Systems & Time Frame

Weekly/Daily	Intraday	HF
Sharpe Ratio = 1+	Sharpe = 2-3	Sharpe = 4+
Diversification	Execution	Technology
i.e. Trend Following	i.e. Spread Trading	i.e. Liquidity Making

Selecting a Technology Stack

Selecting a "Technology Stack"

- Your firm may dictate what platform to use, in which case no decision is necessary
- Otherwise, you have many choices available
  - Turnkey packages that try to do everything
    - Datafeed, Broker connection, Charting, Backtest engine, Account manager, Live trading
    - Easiest path, provided your system can fit into their framework
  - Multiple tools may be required to address your workflow
    - From alpha discovery to live trading
- High-Level Questions
  - Does it support my instruments?
  - Does it (or can it) connect to my required execution venue?
  - Can it run the system type(s) we want?
    - Multi-symbol, Factor filter & rank, Pairs, Portfolio-level logic, Multi-timeframe, Custom bar intervals, etc.
  - How much will it cost to deploy to our team?
    - For research
    - For live trading
  - How good is the support?
  - Is there significant vendor risk?
- Technical Questions
  - What software language do I wish to use?
    - Compiled: C/C++, Java, C#
    - Interpreted: MATLAB, Python, R
    - Platform-specific: EasyLanguage, PowerLanguage, MT4
  - What kind of IDE do I wish to use?
    - Premium: Visual Studio, MATLAB, JetBrains
    - Open Source: Eclipse, NetBeans, Spyder, etc.
    - Browser: Quantopian, Jupyter
    - Platform-specific: MultiCharts, TradeSignal, MetaTrader
  - Is there a stepping debugger? – show of hands
  - What libraries are available?
    - Technical analysis, Optimization, Statistics, Econometrics, Machine Learning, Sentiment, Text analytics, etc.
  - How open is the platform?
    - Backtesting: Import historical data, Custom fill models?
    - Easy to integrate 3rd party libraries?
    - Easy to integrate new datafeeds, execution venues?
  - How easy / productive / frustrating is it to use?

Evaluating Suitability

- This is an example checklist
- You will probably want to create your own

MultiCharts - Suitability Matrix

Research / Alpha Discovery	Built-in (Closed)	Built-in (customizable)	Available as Library	Build myself	Unavailable / Impractical
Stepping Debugger					x
Plot variables while in debug session					x
Backtesting Framework	x				
Charting	x				
Sample Trading Strategies		x			
Standard Indicators		x			
Optimization - Brute Force	x				
Optimization - Genetic	x				
Walk-Forward-Optimization - Brute Force	x				
Walk-Forward-Optimization - Genetic	x				
Historical Data - supports my requirements		x			
Tradeable Renko/Range/Tick Count/Volume bars	x				
Adequate Trade Cost / Slippage Models	x				
Fill on Bid/Ask					x
Custom Fill models					x
Statistical analysis					x
Machine Learning					x
Text Analytics					x

Live Execution	Built-in (Closed)	Built-in (customizable)	Available as Library	Build myself	Unavailable / Impractical
Live Market Datafeed	x				
Execution/Broker bridge	x				
Event-Driven programming	x				
Multiple strategies on same instrument	x				
Supports instruments as indicators	x				
Multiple instruments traded in single strategy					x
Multiple strategies on different instruments	x				
Strategy "health monitoring" while active				x	
Custom bars built from ticks or minutes					x
Supports limit/stop orders on multi-leg spreads					x
Good for high-volume high-frequency strategies					x

General

OS Support:	Windows
Strategy Code identical for Backtest vs. Live Trading?	YES
Time to implement:	Quick for Backtest = Live Trading
Vendor-related risk:	Low

Licensing	Unit	Per Team
Number of licenses	1	4
Initial cost per license:	$1,500	$6,000
Recurring cost:	$0	$0

Support

Vendor bugfixes:	Slow/Inconsistent
Vendor feature requests:	Slow/Inconsistent
Vendor documentation:	Adequate
Vendor training:	N/A
Vendor tech support:	Limited
Forums or online community:	Adequate
Ease of hiring third-party developer:	Excellent

MATLAB - Suitability Matrix

Research / Alpha Discovery	Built-in (Closed)	Built-in (customizable)	Available as Library	Build myself	Unavailable / Impractical
Stepping Debugger	x				
Plot variables while in debug session	x				
Backtesting Framework				x	
Charting	x			x	
Sample Trading Strategies				x	
Standard Indicators				x	
Optimization - Brute Force				x	
Optimization - Genetic		x	x	x	
Walk-Forward-Optimization - Brute Force				x	
Walk-Forward-Optimization - Genetic		x	x	x	
Historical Data - supports my requirements				x	
Tradeable Renko/Range/Tick Count/Volume bars				x	
Adequate Trade Cost / Slippage Models				x	
Fill on Bid/Ask				x	
Custom Fill models				x	
Statistical analysis		x	x	x	
Machine Learning		x	x	x	
Text Analytics		x	x	x	

Live Execution	Built-in (Closed)	Built-in (customizable)	Available as Library	Build myself	Unavailable / Impractical
Live Market Datafeed			x	x	
Execution/Broker bridge			x	x	
Event-Driven programming				x	
Multiple strategies on same instrument				x	
Supports instruments as indicators				x	
Multiple instruments traded in single strategy				x	
Multiple strategies on different instruments				x	
Strategy "health monitoring" while active				x	
Custom bars built from ticks or minutes				x	
Supports limit/stop orders on multi-leg spreads			?	?	?
Good for high-volume high-frequency strategies					x

General

OS Support:	Windows, Linux, Mac
Strategy Code identical for Backtest vs. Live Trading?	NO
Time to implement:	Quick for Backtest, Slow for Live trading
Vendor-related risk:	Low

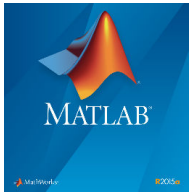
Licensing	Unit	Per Team
Number of licenses	1	4
Initial cost per license:	$7,000	$28,000
Recurring cost:	$2,000	$8,000

Support

Vendor bugfixes:	Excellent
Vendor feature requests:	None
Vendor documentation:	Excellent
Vendor training:	Excellent
Vendor tech support:	Limited
Forums or online community:	Excellent
Ease of hiring third-party developer:	Excellent

Platform Pros and Cons – 1 of 4

Subjective opinions – Feel free to disagree!

Platform	Strengths	Weaknesses	Notes
MATLAB 	> Excellent IDE with best debug/profile features > Powerful toolboxes for statistics, ML, optimization, econometrics, and many other kinds of analytics > Professoinal-quality connectivity to many data sources > Code can be protected using encrypted "pcode" > Strong community of researchers and developers > Almost as flexible as an open-source tool, but better maintained	> No built-in backtesting > Object oriented & event driven features are afterthoughts	> $2k-$10k per user, depending on which toolboxes are needed > Best for quant-heavy and non-standard types of strategies, unless data frequency is too fast > Ideal for research, OK for live trading in certain scenarios
Python (not Quantopian) 	> Open source, with strong and growing community > Attempt to duplicate MATLAB's quant-friendly IDE (but not quite there IMHO) > Jupyter notebooks, Commercial IDE's, etc > Powerful libraries like Pandas and scikit-learn > Good object-oriented and event-handling features > Code can be semi-protected by packaging as .exe	> DateTime hell > Stepping debugger is not as powerful as MATLAB, and more time-consuming to use > Charts are not as interactive as MATLAB	> A good open-source alternative to MATLAB for research > Free and open source, makes deployment cheaper but less secure > Many quant devs are converting to Python from other tools (MATLAB, R, Java)

Platform Pros and Cons – 2 of 4

Subjective opinions – Feel free to disagree!

Platform	Strengths	Weaknesses	Notes
MultiCharts	> Out-of-the-box connectivity to a wide range of data sources and brokers > Excellent charting and well-built UI > PowerLanguage is very easy and beginner-friendly > Can handle mutliple strategies on a single instrument, or multiple strategies on different instruments > Portfolio trader that's useful in certain scenarios > Same strategy code works in backtest and live trading	> No stepping debugger > No meaningful source-code control > Each strategy can trade only one instrument (but may use other instruments as indicators) > Managed orders constrain the types of strategies that can be implemented > Portfolio trader is a clunky afterthought, and suitable only for simple applications > Impractical for most advanced quant strategies (AI/ML, advanced stats, econometrics, etc.) > $15k plus for SDK to add a broker or datafeed > No custom bar types, custom fill models, impractical to backtest on bid/ask, etc. > In live trading, can bog down. Requires daily restart if running multiple tick-based strategies	> $1500 for perpetual license, or $10k to add Bloomberg + Thomson Reuters > Originally a retail product, it's become powerful enough to be used by many professionals
MultiCharts.NET	> Compared to standard MultiCharts, uses C# or VB.NET as the strategy language > Works with Visual Studio, including stepping debugger > Can use 3rd party libraries written in .NET	> C# strategy implementation is inelegant > Can't do unmanaged orders in a backtest (but can do in live trading)	> Most people won't choose this platform. Standard MultiCharts is much easier to use, and nearly equal in functionality

Platform Pros and Cons – 3 of 4

Subjective opinions – Feel free to disagree!

Platform	Strengths	Weaknesses	Notes
RightEdge 	> Very flexible object model allows you to override bar types, fill models, performance reports, create custom broker and data plugins, etc. > Entirely C#, works with Visual Studio, can use any .NET library > Can create elaborate event-driven strategies	> Not very user friendly > Requires significant developer time/skill to build a strategy > Does not have concept of multiple strategies. User must implement logic for sub-strategies herself > Forums/support not very active > Very limited charting	> Very powerful no-frills tool for very low price ($500 per user) > Better for execution than for research > Perpetual license mitigates vendor risk
SmartQuant 	> Similar to Deltix QuantOffice in capability (see below)		> Around $30k for a standard setup > I have limited experience with this one, thus the lack of detail
Deltix QuantOffice 	> Very powerful platform can implement almost anything given enough time and effort > C# strategy language, works with Visual Studio (including stepping debugger) > Can use any 3rd party libraries written in .NET > Same strategy code works in backtest and live trading > Comes with excellent support - bugfixes, phone help, code examples, even implementation	> TimeBase database takes lots of care and feeding, and is not easily extended to non-standard datatypes > Requires significant developer time/skill to build a strategy	> Starts around $100k per year for full product > A good option if you are building an algo-trading desk and have a significant budget

Platform Pros and Cons – 4 of 4

Subjective opinions – Feel free to disagree!

Platform	Strengths	Weaknesses	Notes
Deltix Cloud Services (DCS) 	> Same backtesting engine as QuantOffice, but without custom data (data is hosted in their cloud) > Backtesting is local, so Visual Studio and stepping debugger are supported	> Research only. Does not support live trading > No custom data. No tick data. Only 10 second, minute, and daily data on a limited set of instruments provided by them	> Starts at $100 per month > A way to test the waters with Deltix, but not a long-term solution
Quantopian (cloud) 	> Fully-integrated cloud-based approach makes it free and easy to get started > Good coverage of US Equities, including fundamentals from MorningStar	> Browser-based IDE is frustrating to use > Backtests are slow > Very limited broker connectivity > Does not support faster (tick based) strategies > Fetcher (for outside data) is limited	> Not an option for most professional applications > Investing heavily and improving quickly > May eventually become disruptive, or may go bankrupt
QuantConnect 	> Compared to Quantopian, supports more languages (including C#) and includes data for more asset classes (Equities + FX) and timeframes (tick, second, minute, daily) > Supports Visual Studio and local backtesting & debugging	> Very limited broker connectivity > May not work for non-standard strategies	> I don't have direct experience with this one, so discount my opinions accordingly

Event-Driven vs. Vectorized Implementation

- Show of hands – Are you familiar with the meaning of these terms?
- Tools for Vectorized backtesting
  - MATLAB
  - Python (using Numpy, Pandas, etc.)
  - AmiBroker (retail product)
- Tools for Event-Driven backtesting & trading
  - Deltix QuantOffice & DCS
  - SmartQuant
  - RightEdge
  - QuantConnect?
- Tools for managed Event-Driven backtesting
  - MultiCharts & MultiCharts.NET
  - Quantopian

Agenda

- **9:00 – 10:30 – Introduction to Algorithmic Trading**
  - Our backgrounds.
    - LPPL Project with Didier Sornette.
  - What is Algorithmic Trading
    - Some different trading styles and timeframes
  - Technology stack
    - Platform Pros & Cons
- **10:30 - 11:00 - Morning Break + Software Install**
- **SESSION 2 - 11:00 - 13:00**
  - Research and alpha discovery
    - Order Types
    - Preparing data
    - Back testing
    - Performance metrics
    - Optimisation
      - Exhaustive versus Genetic
    - Back testing – issues
    - Deltix demonstration – strategy session
- **13:00 – 14:00 – Lunch**
- **SESSION 3 - 14:00 - 16:00**
  - **PRACTICAL EXERCISE**
    - Trend Follower
    - Portfolio Trader
    - Experimentation
- **16:00 - 16:20 – Afternoon Break**
- **16:20 -17:00 - QUESTIONS**
  - Other challenges in Algorithmic Trading

Software Download & Installation

- If you have a Windows machine
  - You can install a 30 day free trial of MultiCharts
- If you don't have a Windows machine
  - Please find a neighbor with whom to share
- Please download the files at the following Dropbox folder
  - http://bit.ly/2f8GgLh
- It's LARGE – about 500 MB – So please share or stagger your downloads
  - multicharts64.9.1.12587.400_release_x64_r4.exe – Installer for 64 bit windows machines
  - multicharts.9.1.12586.400_release_r4.exe – Installer for 32 bit windows machines
  - MultiCharts Market Data 1M 1D.qmd – File containing 1M and 1D historical market data for 5 continuous futures contracts
  - MultiCharts CQF Example Signal Code.pla – Contains example code you can compile and run today
- If you can share with a neighbor using a USB drive, please feel free
- You only need to download the one version compatible with your machine

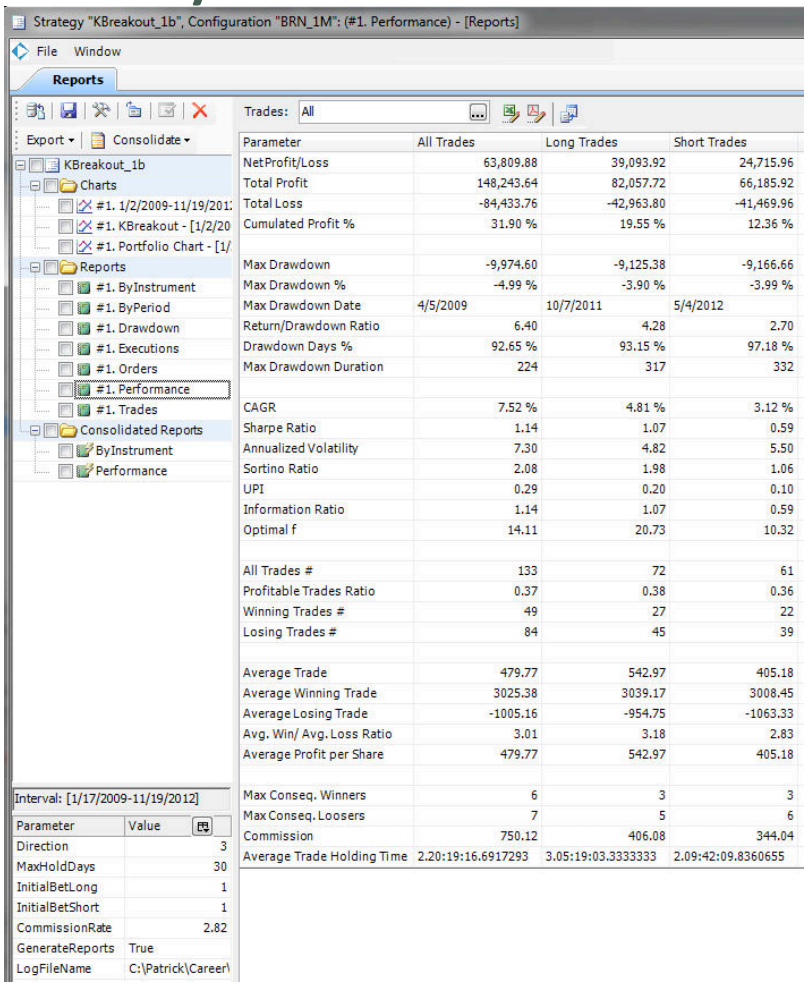
Research & Alpha Discovery

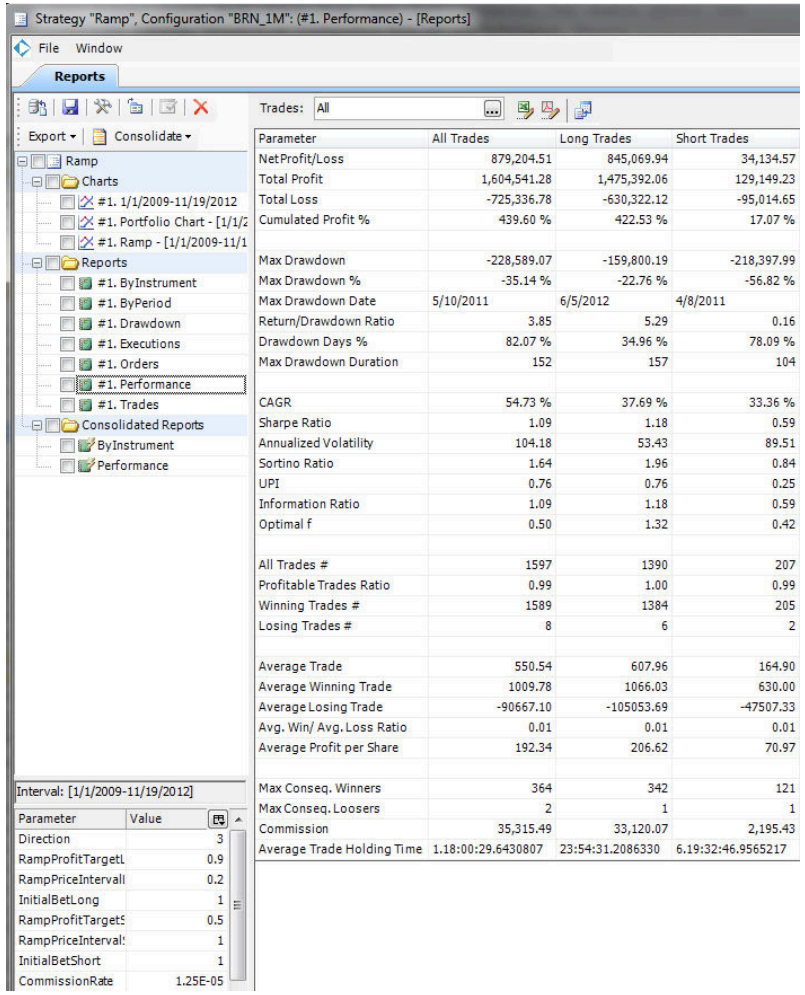
Workflow for Strategy Development

- Rather than present a tidy flowchart, we will instead provide some anecdotes and real-world examples

A Tale of Two Systems – Performance Metrics

- Which strategy would you rather trade?
  - Both trade the same instrument over the same dates
- What metrics are important to you?
- Are you missing any necessary information?





A Tale of Two Systems – Equity Curves

- Top chart is the price series (with trade flags), bottom is equity curve
- Which one looks better?
- Are there any deal-breakers?
- What additional information do you need?

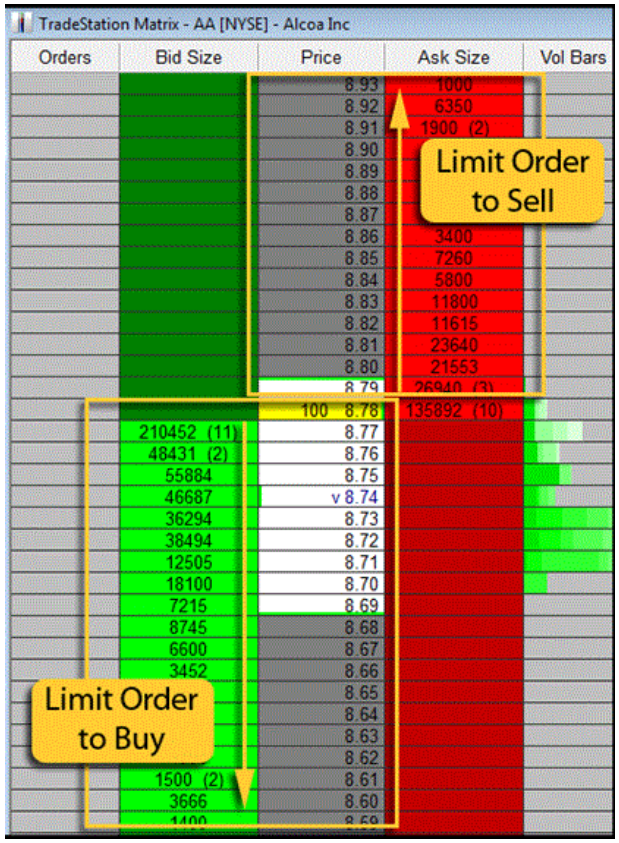
Performance Metrics – Sharpe and Sortino

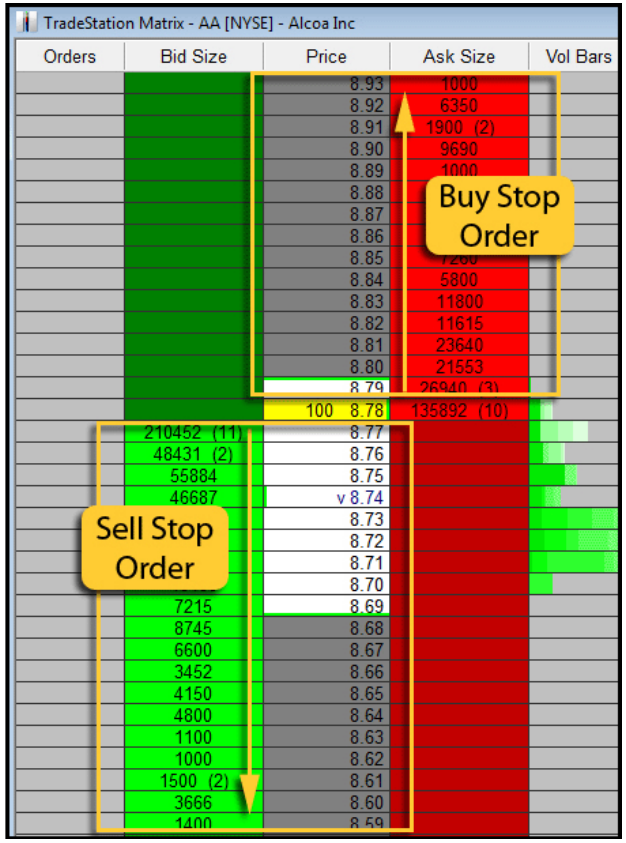
- Most interesting performance metrics reward returns while penalizing risk
  - But there are many ways of doing so
  - Two of the most common metrics are the Sharpe ratio and the Sortino ratio
- Sharpe Ratio
  - Mean of "excess" returns / Standard deviation of returns
    - Risk is defined as the standard deviation of returns
    - Upside surprise and downside surprise are equally penalized
- Sortino Ratio
  - Mean of "excess" returns / Downside deviation of returns
    - Only downside surprise is penalized. Large positive months are treated as a good thing
- Many other performance metrics are used
  - We encourage you to explore them and find the ones that fit your situation
  - However, not all metrics will work as an objective in an optimization

```
% Sharpe Ratio ============================================================
perf.MoAvgReturnPA  = mean(returnSeriesMo)  * 12;
perf.MoVolatilityPA  = std(returnSeriesMo)  * sqrt(12);

perf.RiskFreeRate = RFR;
perf.ExcessReturnPA = perf.MoAvgReturnPA - perf.RiskFreeRate;
if length(returnSeriesMo) > 1
   perf.SharpeRatio  = perf.ExcessReturnPA  / perf.MoVolatilityPA;
else
   perf.SharpeRatio = NaN;
end

% Sortino Ratio ===========================================================
marMo = MAR/12;
downsideReturnSeriesMo = zeros(size(returnSeriesMo));

ix = returnSeriesMo < marMo;
if sum(ix) > 0
    % downsidReturnSeriesMo has same number of elements, so upside returns
    % reduce the downside deviation.  Upside returns are considered to have an
    % "underperformance of 0"
    downsideReturnSeriesMo(ix) = returnSeriesMo(ix) - marMo;
    downsideDeviationMo = sqrt(sum(downsideReturnSeriesMo .^ 2) / ...
                          length(downsideReturnSeriesMo));
else
    % None of our returns underperformed the MAR.  Sortino will be Inf.
    downsideDeviationMo = 0;
end

perf.downsideReturnSeriesMo = downsideReturnSeriesMo;

perf.MAR = MAR;
perf.DownsideDeviation = downsideDeviationMo * sqrt(12);
perf.SortinoRatio = (perf.MoAvgReturnPA - MAR) / perf.DownsideDeviation;
```

Order Types

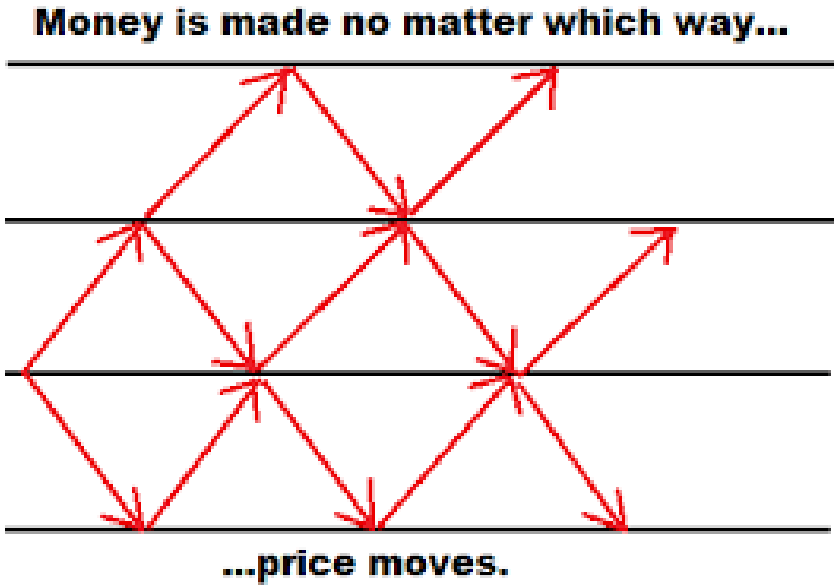
- Major order types
  - Market
  - Limit
  - Stop
  - Stop Limit
- Other useful types
  - Trailing stop
  - OCO
  - OSO
  - Bracket

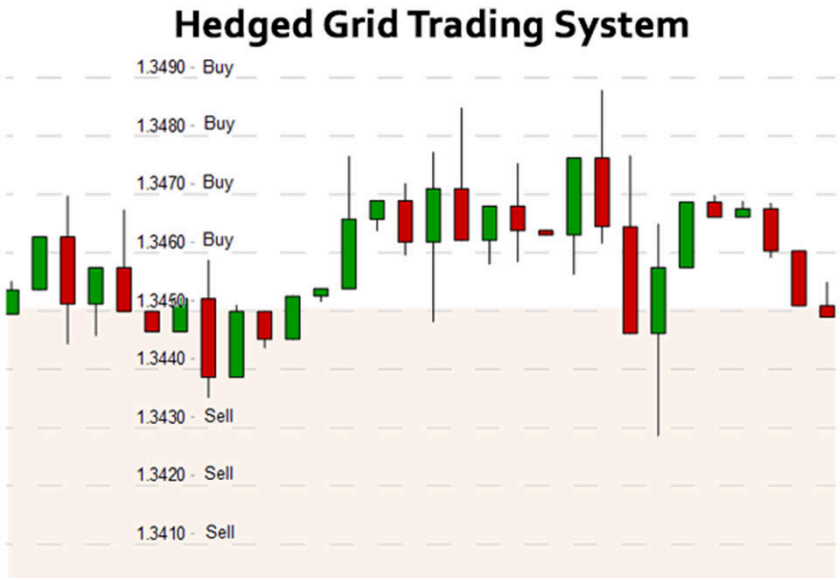




Grid Trader – Not recommended

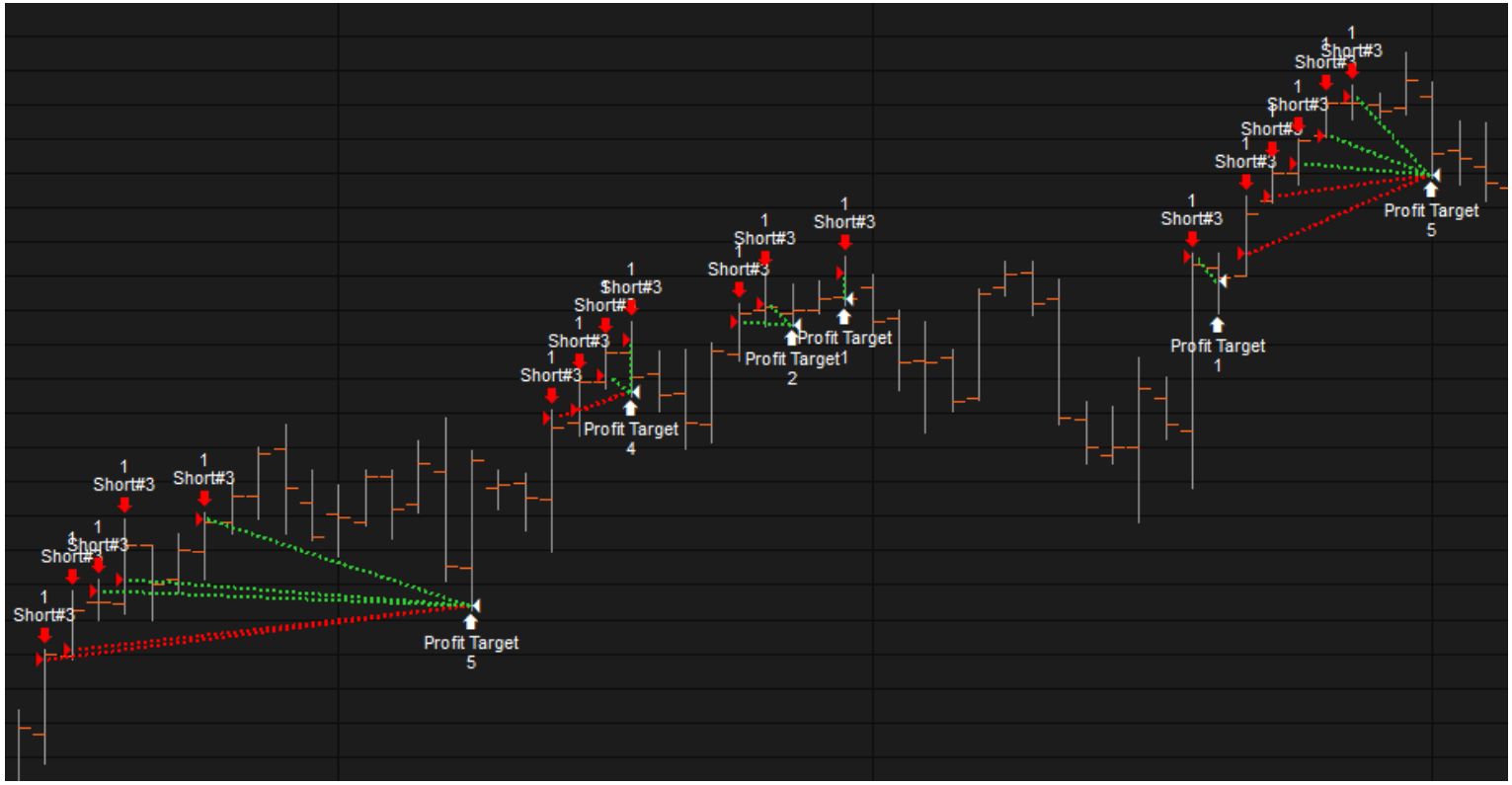
- Many people try some variant of this strategy
- It can show great results for a while
- It often ends badly





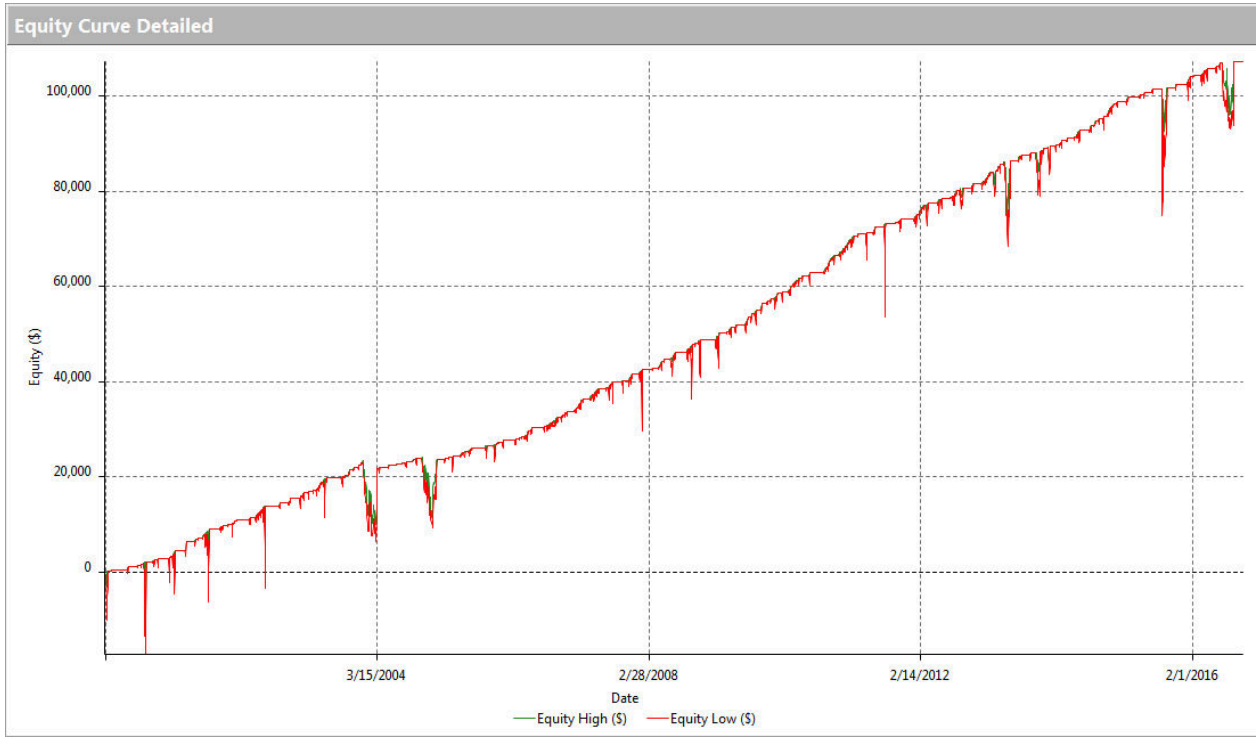
Grid Trader – Trade series

- Many versions have been used
- Common feature: "Average in" to a losing position
  - Resting limit orders are placed in a "grid"
  - Increases the size of your position as price moves against you
  - A small retracement gets you out with profit



Grid Trader – Equity curve

- Can be very seductive
  - Almost never loses
- Distinguishing features
  - Amazing win rate
  - Deep spikey drawdowns that end in small profits
- Risks
  - When drawdowns do occur, they can last a long time
  - Very high margin utilization when averaging into a losing position
    - Can lead to a margin call or risk violation, which ends in ruin



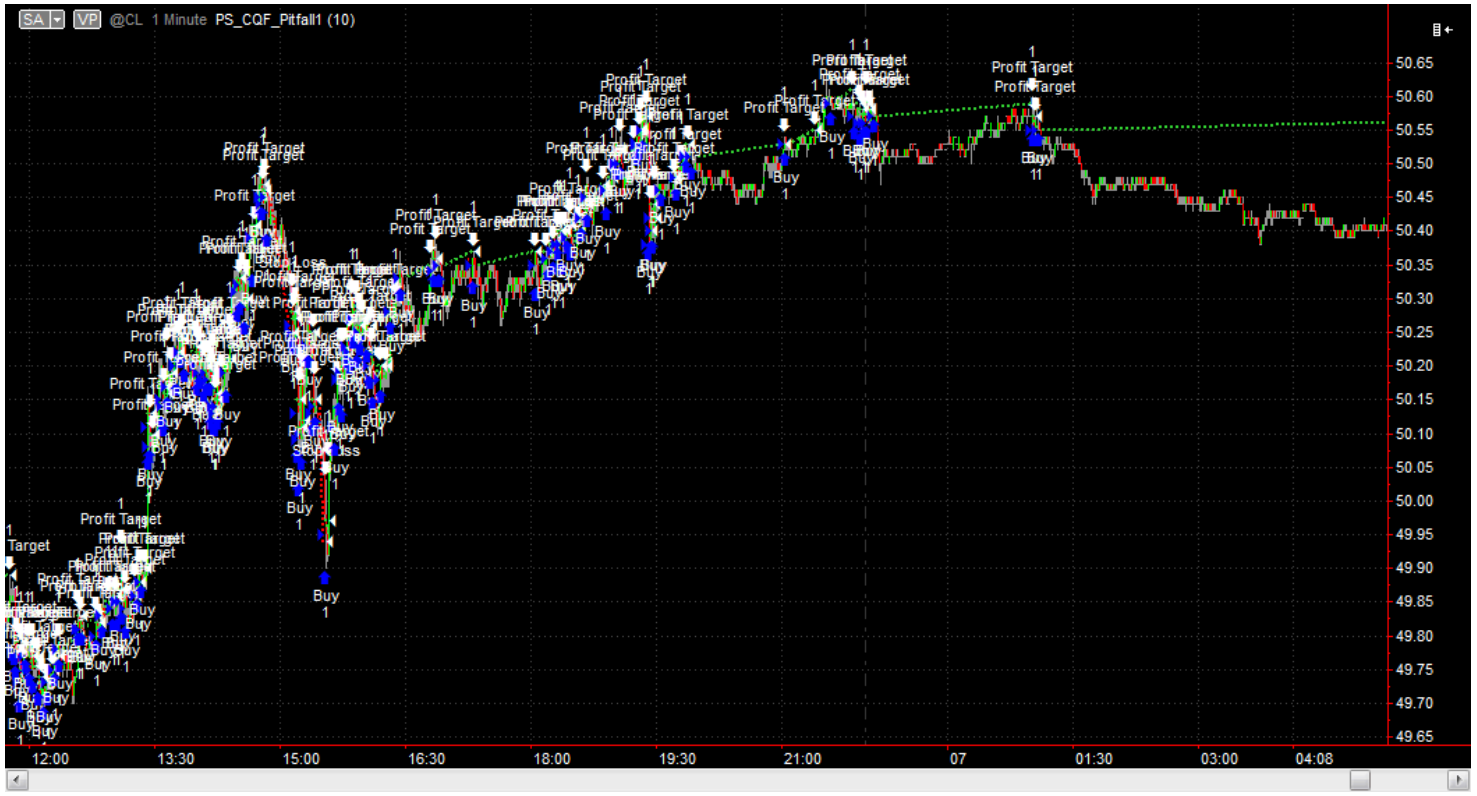
A Deceptively Simple Strategy – Code

- Only 20 lines of code, with very simple logic
- Who expects this strategy to be profitable?

```
Inputs:
   LenATR(10);

Variables:
   atr(0),
   pTarg(0),
   tickSize(MinMove / PriceScale),
   tickValue(MinMove / PriceScale * BigPointValue);

atr = averagetruerange(LenATR);

if Close < Open then begin
   buy this bar at close;
   pTarg= maxlist(atr * BigPointValue / 10, 2 * MinMove * PointValue);
   pTarg= tickValue * round(pTarg / tickValue, 0);
   //print (symbol, " pTarg=",  pTarg, " MinMove=", MinMove, " PointValue=", PointValue, " PriceScale=", Pr
end;

setprofittarget(pTarg);
setstoploss(10*pTarg)
```

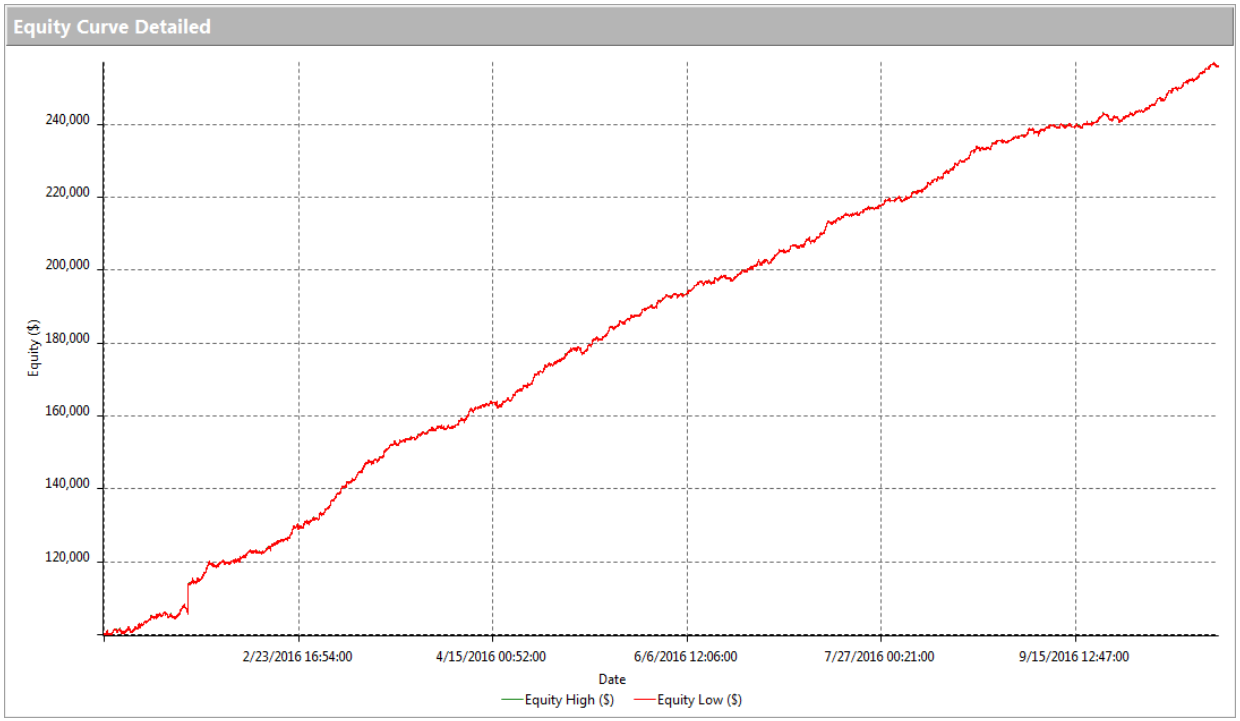
A Deceptively Simple Strategy – Trade series

- Applied to CL with 1M bar interval



A Deceptively Simple Strategy – Equity curve

- Results look amazing!
  - How is this possible?



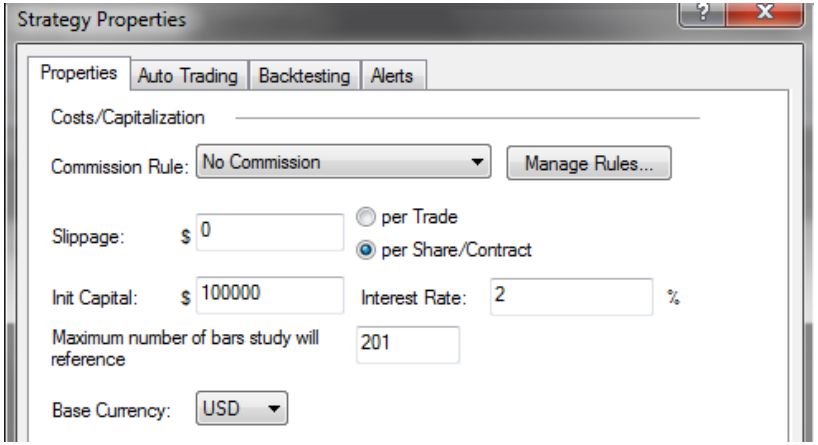
Total Trade Analysis

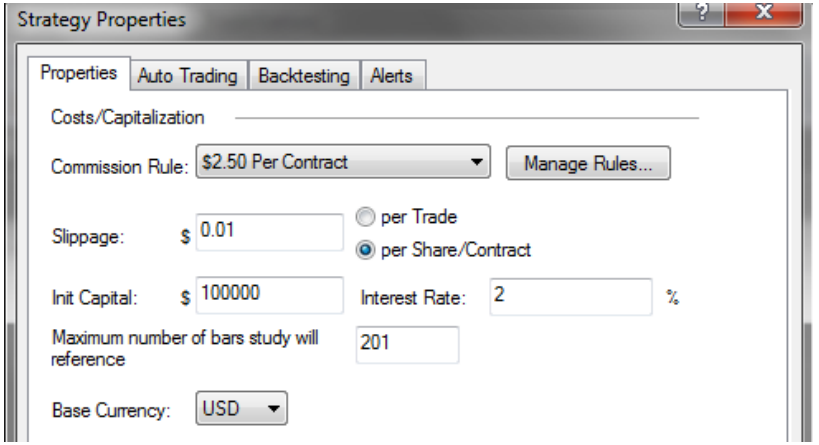
	All Trades	Long Trades	Short Trades
Total # of Trades	35369	35369	0
Total # of Open Trades	1	1	0
Number Winning Trades	32829	32829	0
Number Losing Trades	2540	2540	0
Percent Profitable	92.82%	92.82%	0%
Avg Trade (win loss)	$4.41	$4.41	n/a
Average Winning Trade	$20.30	$20.30	n/a
Average Losing Trade	($200.94)	($200.94)	n/a
Ratio Avg Win / Avg Loss	(0.1)	(0.1)	n/a
Largest Winning Trade	$8,060.00	$8,060.00	n/a
Largest Losing Trade	($1,940.00)	($1,940.00)	n/a
Avg # Bars in Trades	7.3	7.3	n/a
Avg # Bars in Winning Trades	5.7	5.7	n/a
Avg # Bars in Losing Trades	28.7	28.7	n/a
Avg # Bars Between Trades	n/a	n/a	n/a
Avg # Bars Between Winning Trades	3.1	3.1	n/a
Avg # Bars Between Losing Trades	84.3	84.3	n/a

A Deceptively Simple Strategy – Trade costs

- No slippage or commissions were applied

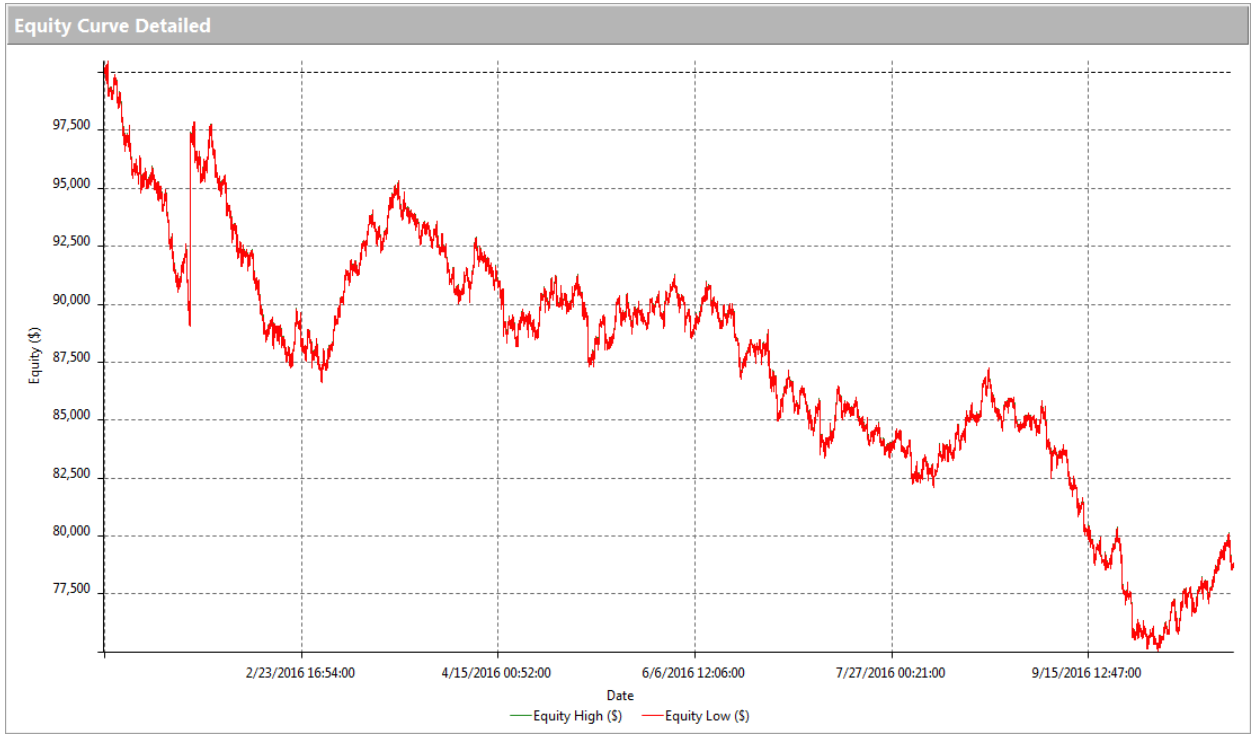
- Assume a very competitive commission
- Assume a modest cost of slippage
  - Applies to market orders and stops





A Deceptively Simple Strategy – Net of trade costs

- Not so amazing anymore
  - And still, this is too optimistic



A Deceptively Simple Strategy – Buy on close?

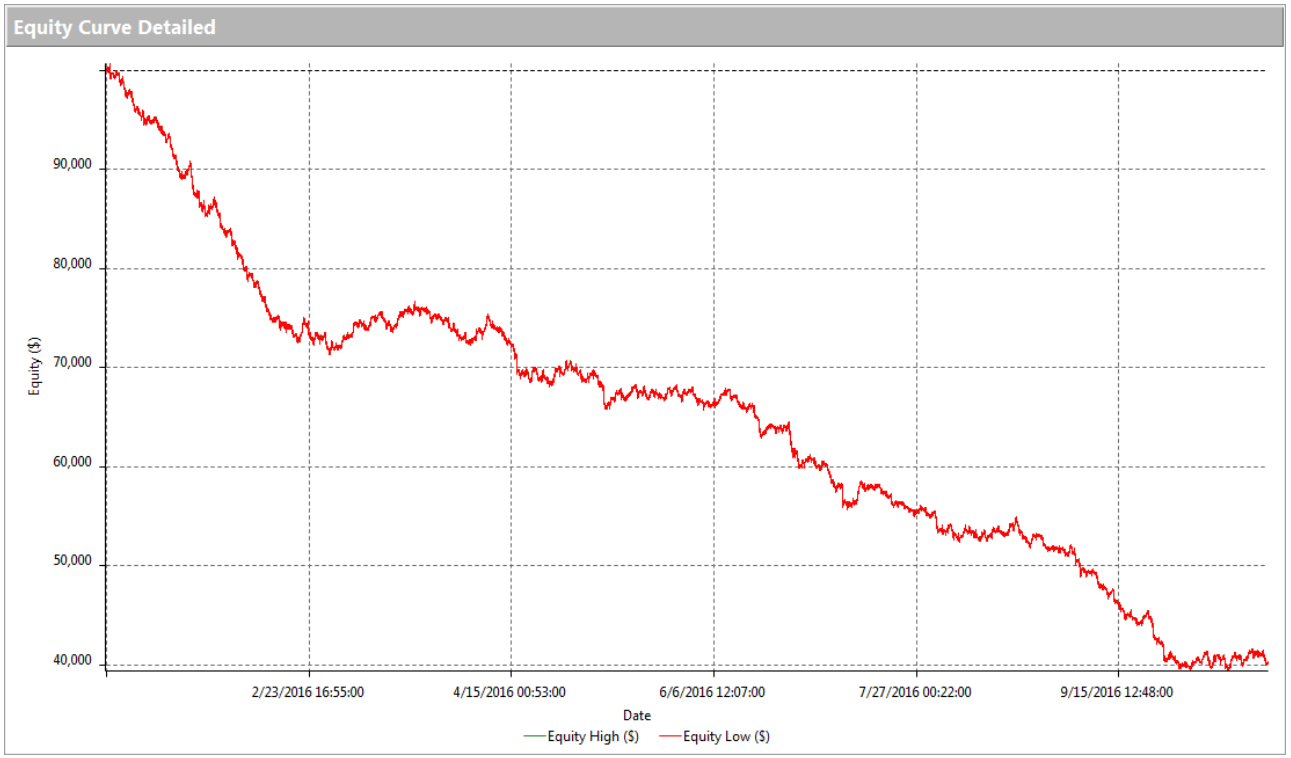
- If you can't do it in the real world, you shouldn't do it in a backtest

```
Inputs:
   LenATR(10);

Variables:
   atr(0),
   pTarg(0),
   tickSize(MinMove / PriceScale),
   tickValue(MinMove / PriceScale * BigPointValue);

atr = averagetruerange(LenATR);

if Close < Open then begin
   buy this bar at close;
   pTarg= maxlist(atr * BigPointValue / 10, 2 * Mir
   pTarg= tickValue * round(pTarg / tickValue, 0);
   //print (symbol, " pTarg=",  pTarg, " MinMove=",
end;

setprofittarget(pTarg);
setstoploss(10*pTarg)
```

```
Inputs:
   LenATR(10);

Variables:
   atr(0),
   pTarg(0),
   tickSize(MinMove / PriceScale),
   tickValue(MinMove / PriceScale * BigPointV

atr = averagetruerange(LenATR);

if Close < Open then begin
   buy next bar at open;
   pTarg= maxlist(atr * BigPointValue / 10, 2
   pTarg= tickValue * round(pTarg / tickValue
   //print (symbol, " pTarg=",  pTarg, " MinM
end;

setprofittarget(pTarg);
setstoploss(10*pTarg)
```

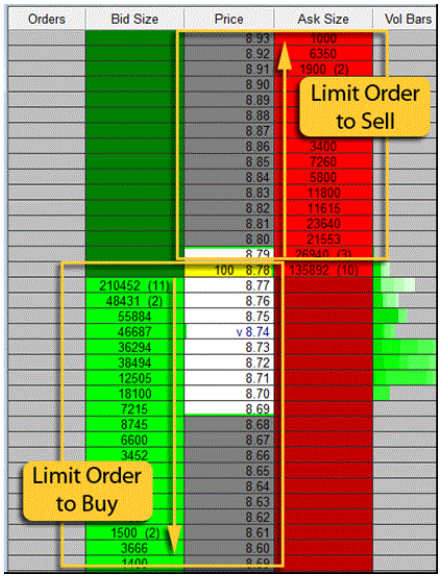
A Deceptively Simple Strategy – Buy next bar

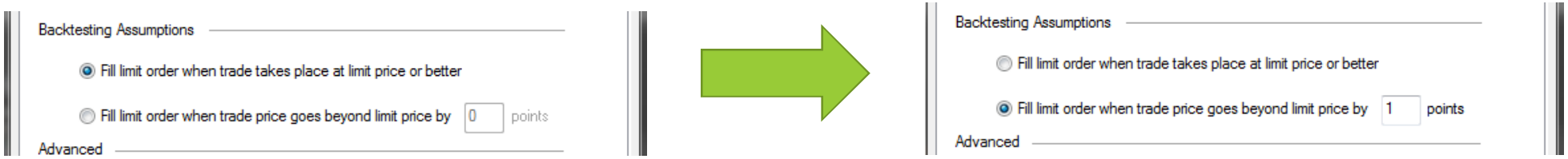
- This is getting painful



A Deceptively Simple Strategy – Conservative fill model for limit orders

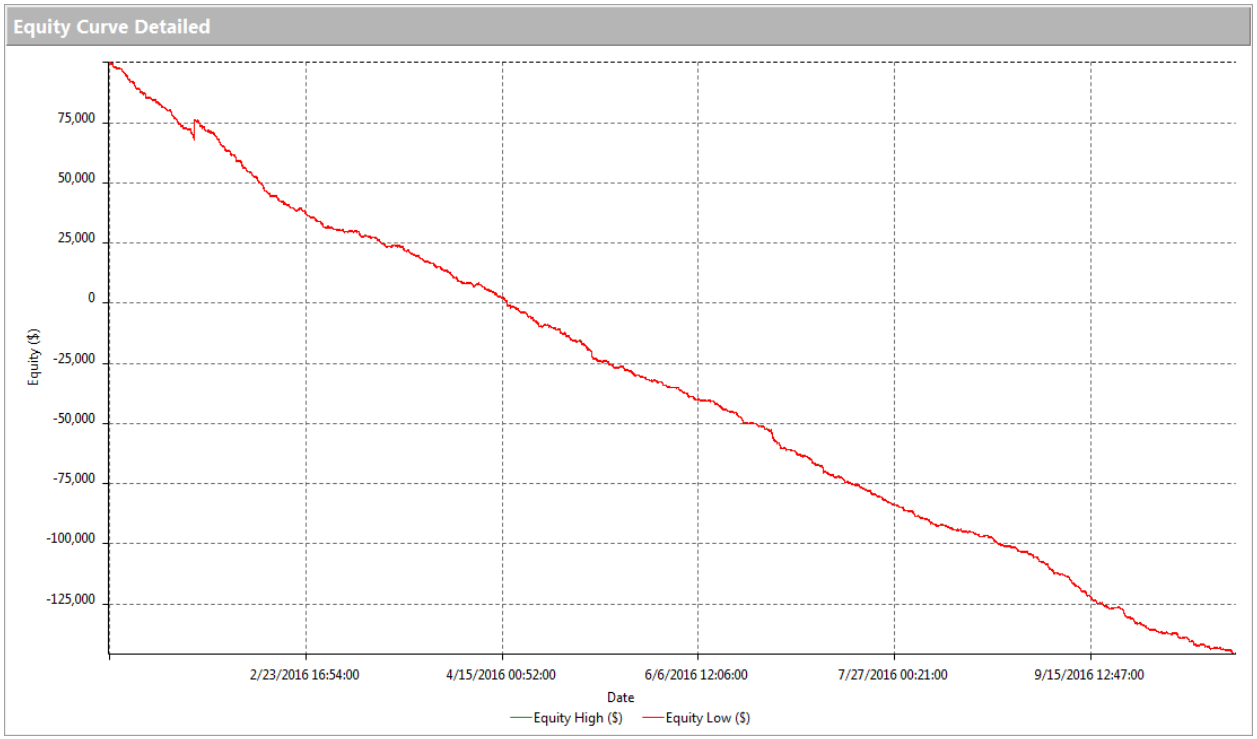
- When does the backtester fill our limit orders?
  - A limit-order book (aka Level2 or DOM or Matrix), is made up of resting limit orders
  - In live trading, our limit orders are among them
  - When a marketable order arrives at the exchange, fills are matched by price priority, then by time priority
  - There are probably other people with limit orders at the same price as ours, who are ahead of us in the queue
  - A conservative assumption is that the price must trade 1 tick past our limit price in order to fill
    - i.e. we are assuming that all other traders at our price placed their limit orders before we did





A Deceptively Simple Strategy – With conservative fill model for limit orders

- How far the mighty have fallen

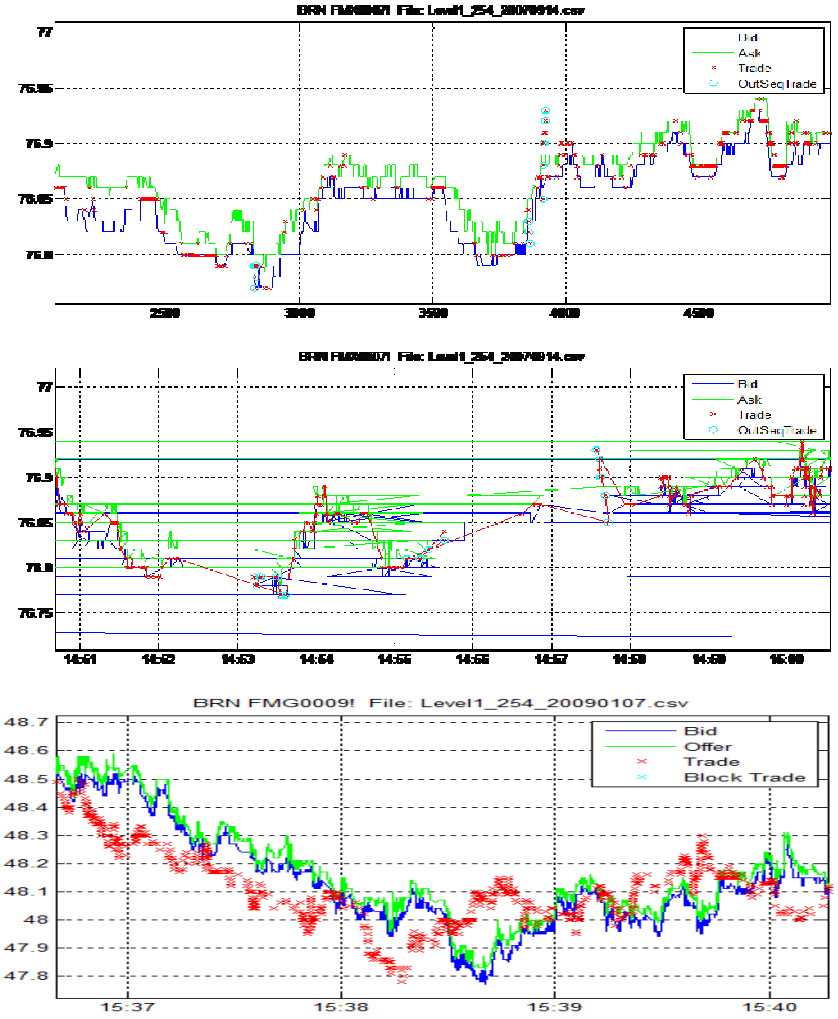


Backtesting pitfalls

- Methodology Issues
  - Fill models
  - Transaction costs
  - Market impact
  - Look-ahead errors
  - Overfitting
    - You should have many (many many) more trades than parameters
  - Data snooping
    - See Halbert White's reality check and its successors
  - Data errors
    - Single-instrument trade and quote sync
    - Multi-instrument time-zone and clock sync
    - Unrealistic continuous contracts (futures)
  - Survivorship bias
- Worldview Issues
  - Every statistical model depends on one crucial assumption
    - We expect the future to somehow resemble the past
  - Markets are not "stationary"
  - There can be sudden changes in the market that break your system, due to no fault of your own
    - But it still results in losses, so you'd better have a plan to deal with it

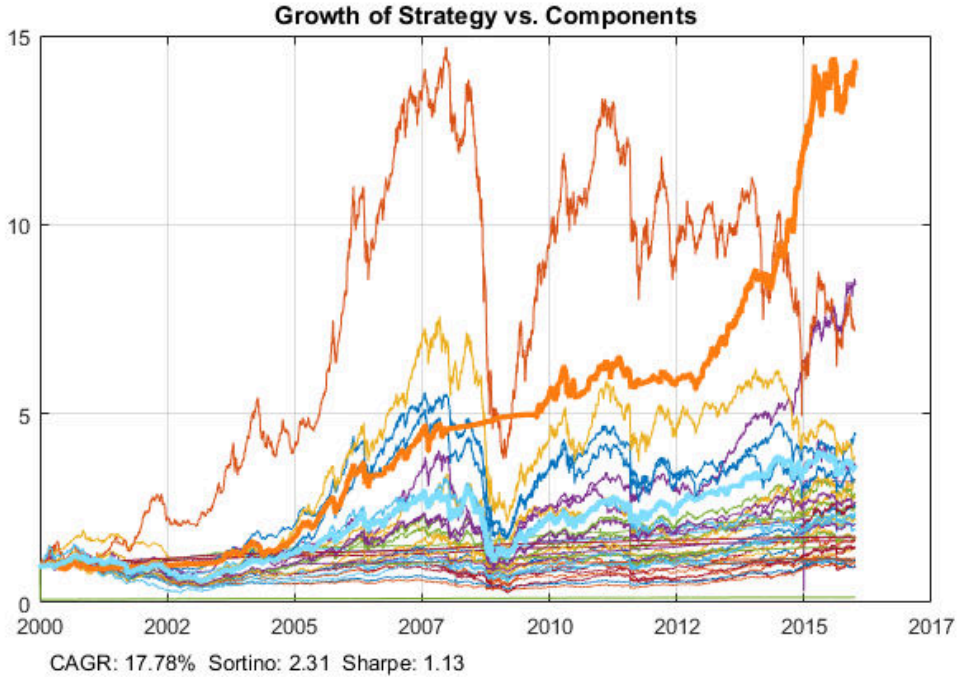
Data errors – TAQ Sync

- This is trade and quote data sourced directly from the exchange
- We found many anomalies
  - Bid & Ask apparently on a different clock than the Trade ticks
  - Occasionally trades arrived out of order during market congestion
- We developed procedures to clean the data before importing into Deltix TimeBase



Longer-Term Strategy

- This strategy re-allocates quarterly
- From a universe of stocks, ranks them and allocates to the best few
  - Long only, no leverage
  - Ranking criteria are the alpha driver here
  - Can handle a universe consisting of hundreds (or thousands) of instruments
- Custom backtester and live allocator written in MATLAB
- Data sourced from Bloomberg via API
- This strategy is easily implemented and scaled
  - Algorithmic allocation, manual execution



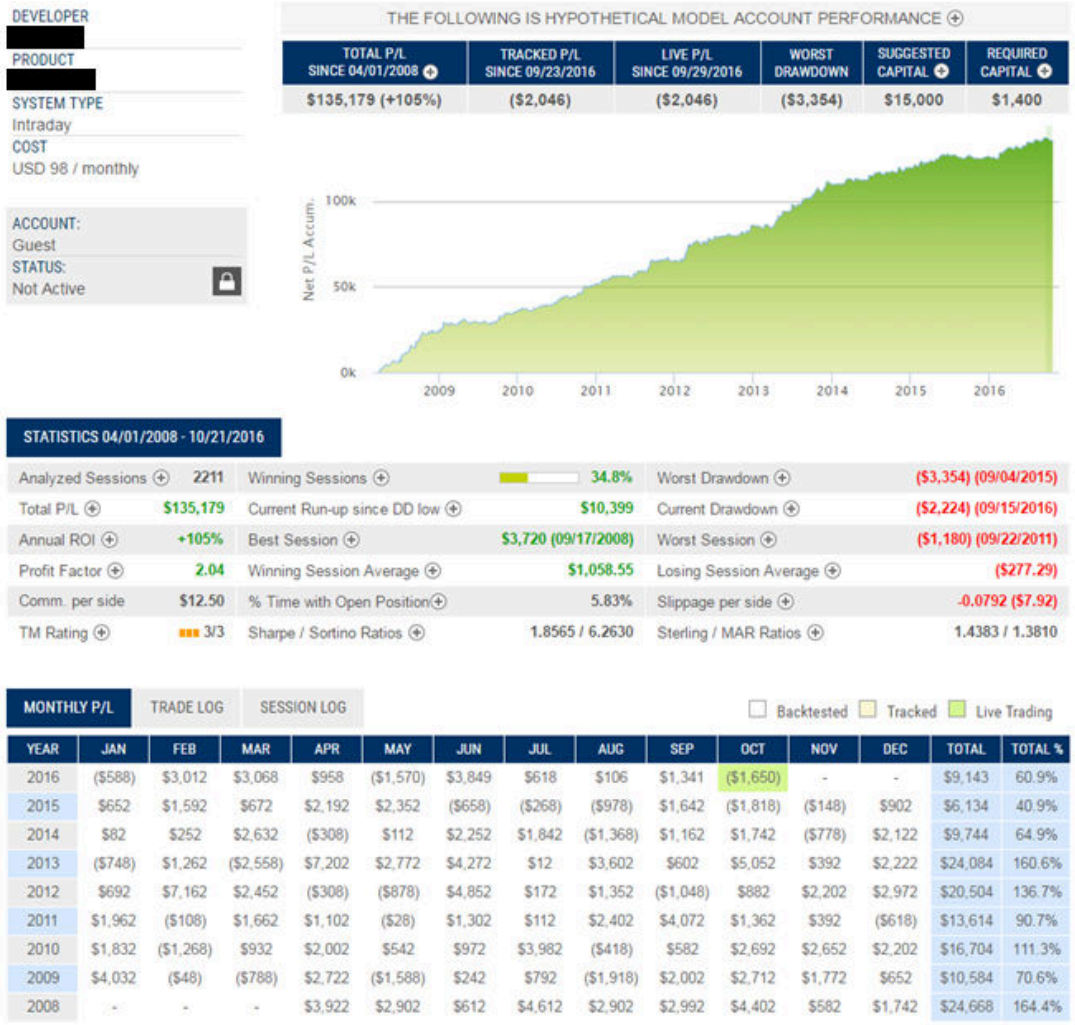
- Caveats
  - Bloomberg does not provide “graveyard” stocks
    - This causes survivorship bias
  - Many backtests and factors have been tried
    - This may cause a “data snooping” bias

Optimization – As Applied to Algo Trading Strategies

- Optimization is the process of searching a parameter space
  - Parameters are defined by the developer as inputs to a trading algo
  - An "exhaustive" or "brute-force" optimizer tries a set of values for each parameter, and runs a backtest
    - The curse of dimensionality
      - 2 parameters with 10 values each requires $10^2 = 100$ backtests
      - 10 parameters with 10 values each requires $10^{10} = 10{,}000{,}000{,}000$ backtests
      - If each one takes a second, that's over 300 years
  - A "Genetic" or "Evolutionary" optimization algorithm is a clever biologically-inspired way of searching the parameter space
    - Can be made to work with constraints and nonlinear objective functions
- Methodology concerns
  - Trying a large number of strategy ideas over the same set of data risks "data snooping"
    - Each backtest in an optimization can be considered an idea
  - Optimizing over a large number of parameters risks "overfitting"
    - Especially if you use the same set of historical data for your training and testing
  - It's important to separate your data into subsets
    - Never set your future expectations based on data used in training or model fitting
    - In some problem domains, "cross-validation" is appropriate for this purpose
    - In backtesting, a "walk forward" optimization is often appropriate
    - Other thoughtful ways of partitioning your data can also be employed

Caveat Emptor

- This is a system being offered today on a “marketplace for automated trading systems”
- It looks wonderful
- What could possibly go wrong?



DEVELOPER [redacted]
PRODUCT [redacted]
SYSTEM TYPE
Intraday
COST
USD 98 / monthly

ACCOUNT:
Guest
STATUS:
Not Active

THE FOLLOWING IS HYPOTHETICAL MODEL ACCOUNT PERFORMANCE

TOTAL P/L SINCE 04/01/2008	TRACKED P/L SINCE 09/23/2016	LIVE P/L SINCE 09/29/2016	WORST DRAWDOWN	SUGGESTED CAPITAL	REQUIRED CAPITAL
$135,179 (+105%)	($2,046)	($2,046)	($3,354)	$15,000	$1,400

STATISTICS 04/01/2008 - 10/21/2016

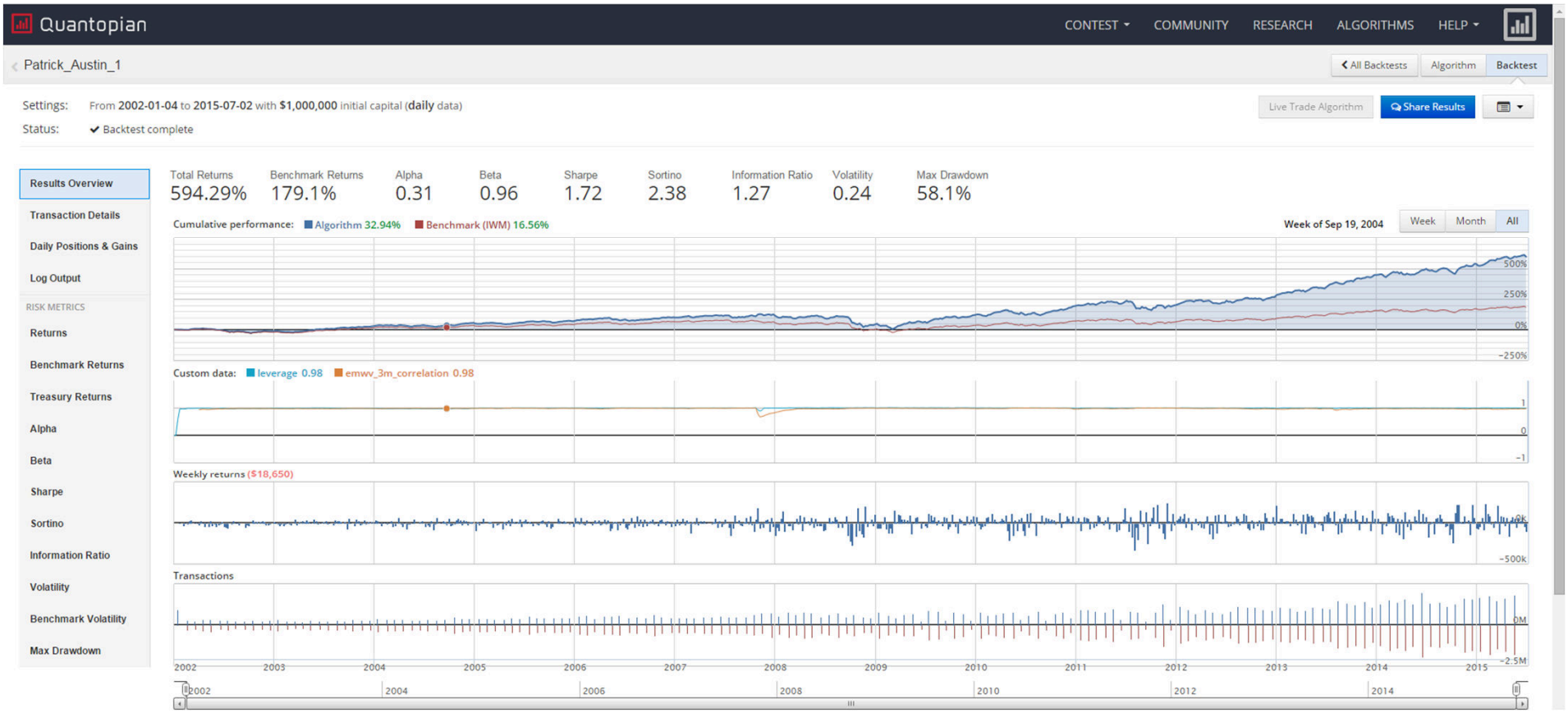
Analyzed Sessions	2211	Winning Sessions	34.8%	Worst Drawdown	($3,354) (09/04/2015)
Total P/L	$135,179	Current Run-up since DD low	$10,399	Current Drawdown	($2,224) (09/15/2016)
Annual ROI	+105%	Best Session	$3,720 (09/17/2008)	Worst Session	($1,180) (09/22/2011)
Profit Factor	2.04	Winning Session Average	$1,058.55	Losing Session Average	($277.29)
Comm. per side	$12.50	% Time with Open Position	5.83%	Slippage per side	-0.0792 ($7.92)
TM Rating	3/3	Sharpe / Sortino Ratios	1.8565 / 6.2630	Sterling / MAR Ratios	1.4383 / 1.3810

MONTHLY P/L | TRADE LOG | SESSION LOG

Backtested Tracked Live Trading

YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	TOTAL	TOTAL %
2016	($588)	$3,012	$3,068	$958	($1,570)	$3,849	$618	$106	$1,341	($1,650)	-	-	$9,143	60.9%
2015	$652	$1,592	$672	$2,192	$2,352	($658)	($268)	($978)	$1,642	($1,818)	($148)	$902	$6,134	40.9%
2014	$82	$252	$2,632	($308)	$112	$2,252	$1,842	($1,368)	$1,162	$1,742	($778)	$2,122	$9,744	64.9%
2013	($748)	$1,262	($2,558)	$7,202	$2,772	$4,272	$12	$3,602	$602	$5,052	$392	$2,222	$24,084	160.6%
2012	$692	$7,162	$2,452	($308)	($878)	$4,852	$172	$1,352	($1,048)	$882	$2,202	$2,972	$20,504	136.7%
2011	$1,962	($108)	$1,662	$1,102	($28)	$1,302	$112	$2,402	$4,072	$1,362	$392	($618)	$13,614	90.7%
2010	$1,832	($1,268)	$932	$2,002	$542	$972	$3,982	($418)	$582	$2,692	$2,652	$2,202	$16,704	111.3%
2009	$4,032	($48)	($788)	$2,722	($1,588)	$242	$792	($1,918)	$2,002	$2,712	$1,772	$652	$10,584	70.6%
2008	-	-	-	$3,922	$2,902	$612	$4,612	$2,902	$2,992	$4,402	$582	$1,742	$24,668	164.4%

A Quantopian Experience – Performance



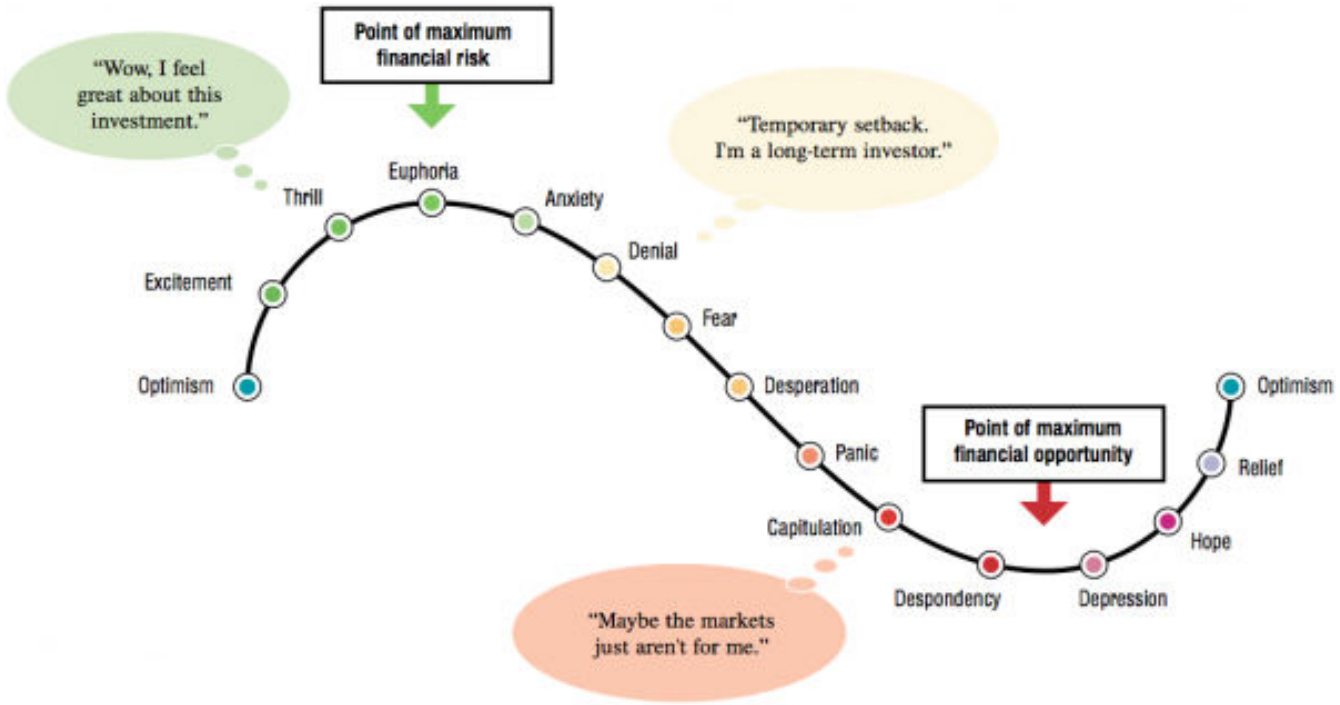
A Quantopian Experience – Explanation

- Developed a strategy to attempt to beat a well-known stock index
  - Within each sector, rank by a sector-specific value metric
    - If the metric isn't present in the DB, use price to book instead
    - Combine heterogeneous metrics using z-score
- We reconstructed the index using internal logic, to use as a performance baseline
- Problem: Our performance baseline significantly outperformed the index
  - We traced the cause to a bug in Quantopian's fill model
  - They fixed the bug, and the apparent alpha went away

Practical Exercise

- If you have a Windows machine
  - You can install a 30 day free trial of MultiCharts
- If you don't have a Windows machine
  - Please find a neighbor with whom to share
- Please download the files at the following Dropbox folder
  - http://bit.ly/2f8GgLh
- It's LARGE – about 500 MB
  - multicharts64.9.1.12587.400_release_x64_r4.exe – Installer for 64 bit windows machines
  - multicharts.9.1.12586.400_release_r4.exe – Installer for 32 bit windows machines
  - MultiCharts Market Data 1M 1D.qmd – File containing 1M and 1D historical market data for 5 continuous futures contracts
  - MultiCharts CQF Example Signal Code.pla – Contains example code you can compile and run today
- If you can share with a neighbor using a USB drive, please feel free
- You only need to download the one version compatible with your machine

Trading Psychology



Trading Psychology

- Split trading research from execution
- Execution incentivised with different KPI's
- Difficult if you're a one or two man desk!

Trading in the Zone

The problem with a novice trader:

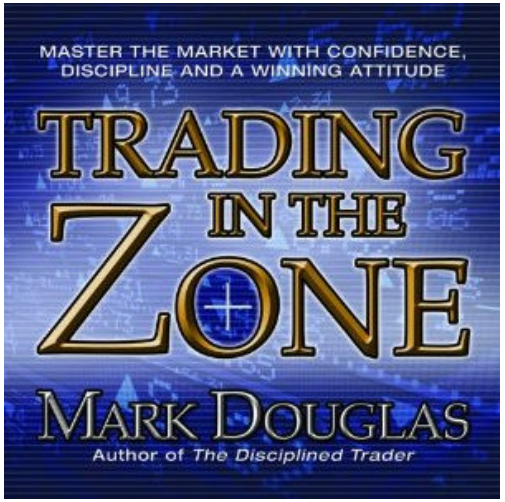
- They yet to realise that he will only truly be successful when he sees that no one can predict the next move in the market only the next PROBABLE move

- They has yet to fully experience greed and fear and the devastating consequences it will have on his bottom line

- They have yet to stop treating each trade as if it is a matter of life or death and finally he has no rule based structure to approach the market with.

"I will leave you with one final thought - I personally know some of the smartest guys on the planet when it comes to market analysis but they make less money than some of the dimmest guys I know - why? Simple the dimmer guys STICK TO THE PLAN."

- Neerav Vadera - comments on Amazon.co.uk



Key lessons

- Intelligence does not matter, just do not be stupid
- Determination
  - http://www.paulgraham.com/determination.html
- Patience
- It is not about reinventing the wheel

Conclusion

- Algorithmic trading is an intellectually interesting field
- It is also a psychologically demanding field
- We hope we have provided some useful information today

Thank you for your interest!