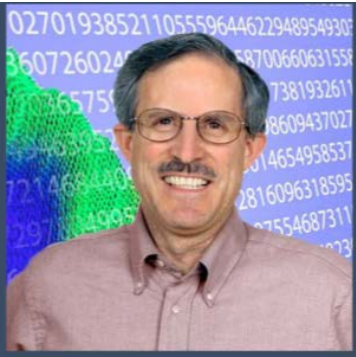


Big data computing: Science and pseudoscience

Speakers: Prof David H. Bailey (Lawrence Berkeley Lab (retired) and U.C. Davis, USA) and Prof Jonathan Borwein (University of Newcastle)

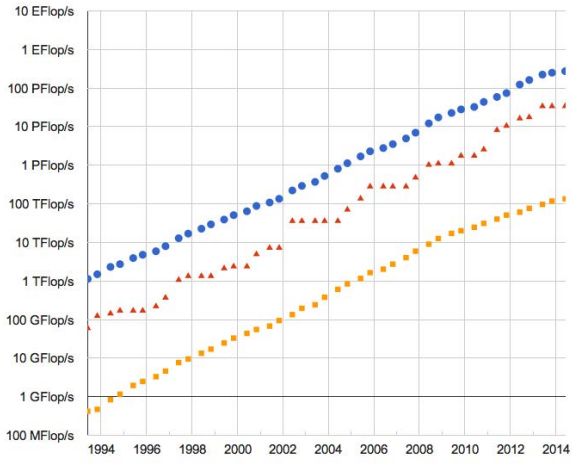
Present-day petascale supercomputers

The memory capacity of supercomputers is measured in "Pbytes" or "petabytes" (namely one quadrillion bytes), and their performance is measured in "Pflop/s" or "petaflops" (namely one quadrillion 64-bit floating-point operations per second).

It is amusing to consider for a moment how big a quadrillion is:

- The volume of Sydney Harbor is roughly 5×10^{11} litres, or 1/10,000 of one quadrillion litres.
- The volume of Lake Tahoe, California is roughly 1.51×10^{14} litres, or 1/7 of one quadrillion litres.
- The distance to Alpha Centauri is roughly 4.15×10^{13} kilometres, or 41.5 quadrillion meters.
- The number of minutes since the big bang is roughly 7.25×10^{15} minutes, i.e., about 7 quadrillion minutes.

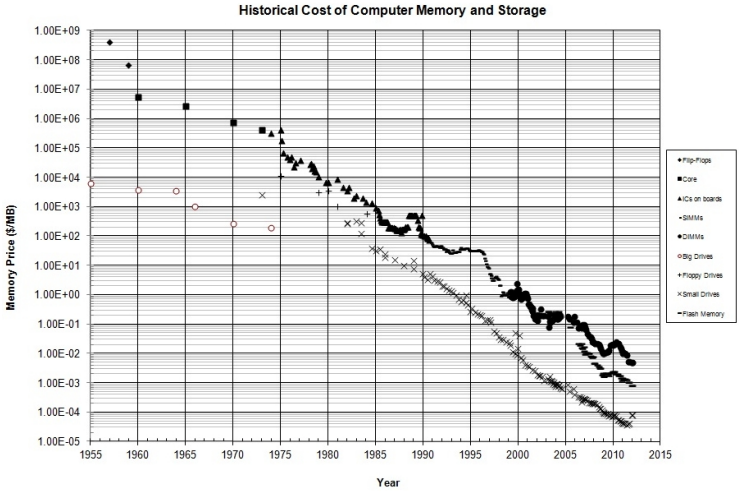
Increasing performance of the top 500 supercomputers (1994 – present)



Red = #1
Orange = #500
Blue = $\sum$ #1 thru #500

- "Performance development," Top500.org, available at `http://top500.org/statistics/perfdevel`.

Declining cost of data storage (1955 – 2012)



- John C. McCallum, "Disk drive storage price decreasing with time (1955-2012)," available at `http://www.jcmit.com/disk2012.htm`.

Systems at the U.S. National Energy Research Scientific Computing (NERSC) Center at the Lawrence Berkeley National Laboratory

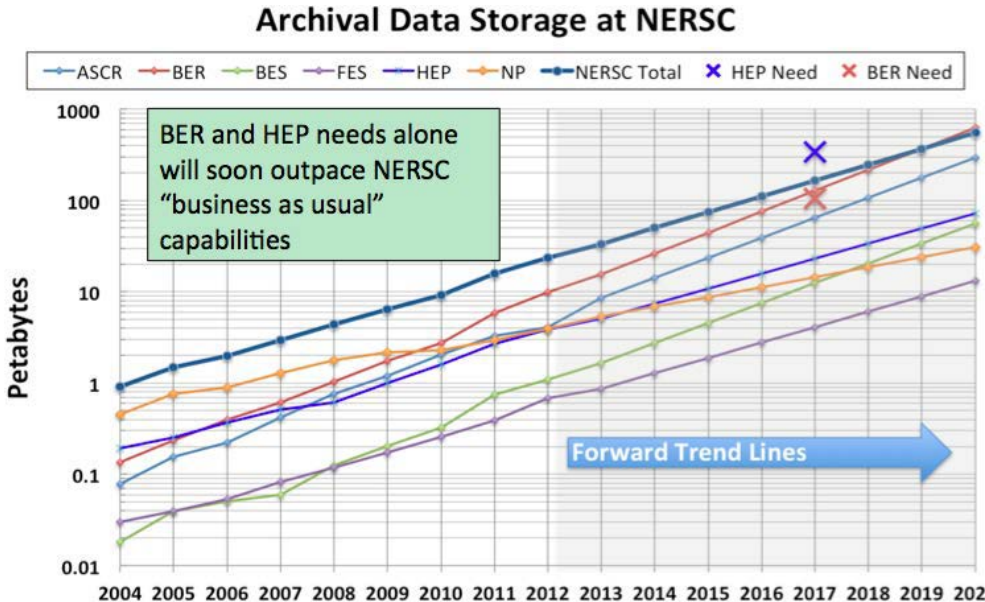
Current system ("Edison"):

- 5576 Intel "Ivy Bridge" nodes.
- Memory per node: 64 Gbyte.
- Performance per node: 460.8 Gflop/s.
- Total main memory: 357 Tbyte.
- Total performance: 2.57 Pflop/s.
- Total disk storage: 7.56 Pbyte.

To be installed in 2016 ("Cori"):

- 9300 Intel "Knight's Landing" nodes.
- Main memory per node: 64 Gbyte.
- Performance per node: 3 Tflop/s.
- Total main memory: 595 Tbyte.
- Total peak performance: 28 Pflop/s.
- Total disk capacity: 28 Pbyte.

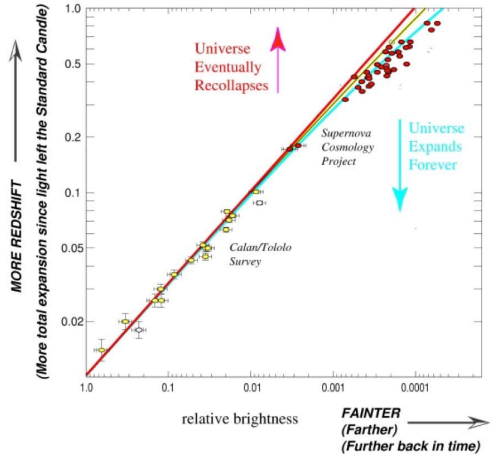
Future archival storage requirements at NERSC



Big data applications: The accelerating universe

In 1998, two teams of astronomers (one led by Saul Perlmutter of LBNL, and the other led by Brian P. Schmidt of ANU), came to the paradoxical conclusion that the expansion of the universe is accelerating, not slowing down.

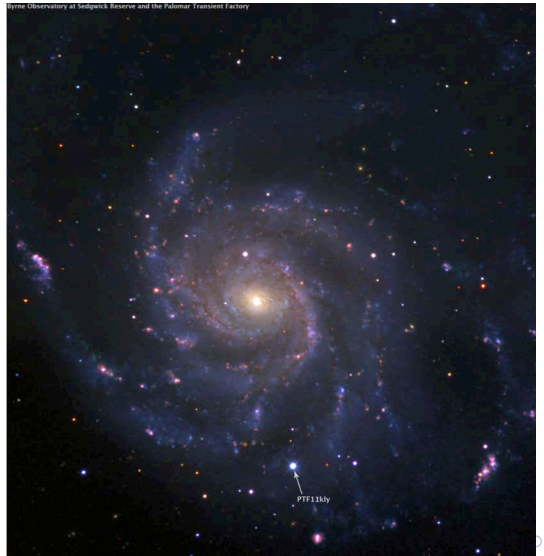
Both teams based their results on careful measurements of distant supernovas, which in turn were found by sifting through reams of digital telescope data. The U.S. team in particular relied heavily on large computer systems at the NERSC facility, coupled with a worldwide network of collaborating astronomers.



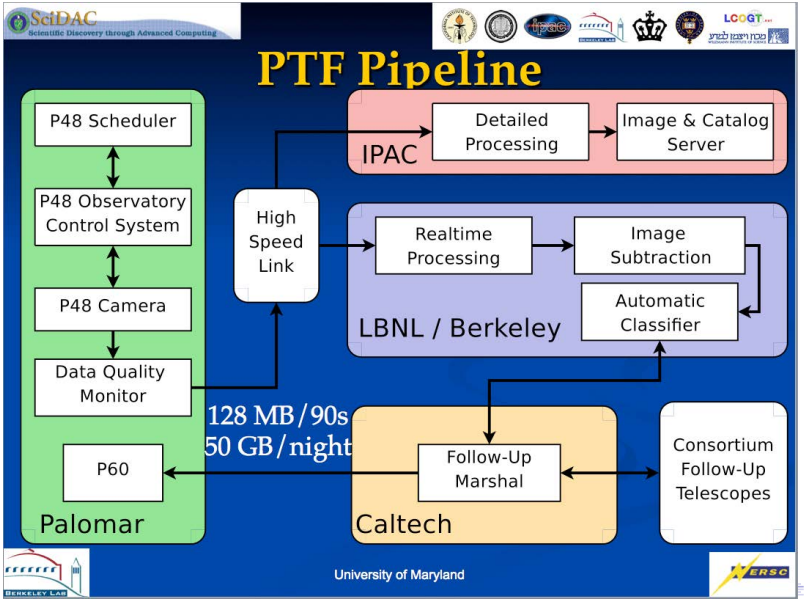
Big data applications: Discovery of supernova in the Pinwheel Galaxy

In 2011, Peter Nugent of LBNL, working within a worldwide consortium of fellow astronomers, discovered a Type Ia supernova in the Pinwheel Galaxy, which is "only" 21 million light-years from earth. It is the closest and brightest supernova of this type seen in the last 30 years.

Nugent and his team utilized the Palomar Transient Facility (PTF), a robot-controlled telescope that produces digital images, plus a worldwide network of data processing and storage facilities.



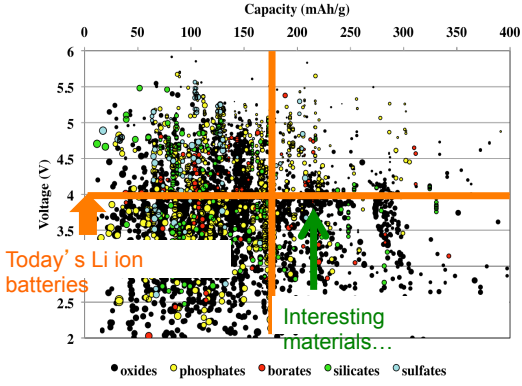
The Palomar Transient Facility (PDF) data pipeline



Big data applications: The Materials Project

Researchers at LBNL and Harvard University started the "Materials Project":

- Reduce lag time between materials science advances and real-world commercialization.
- Invert the conventional materials science paradigm: Ask "What properties do I want?," then "Which materials have them?"
- Method: Perform *ab initio* calculations of tens of thousands of potentially interesting compounds, then save in searchable database.
- Now in operation: `http://www.materialsproject.org`.



Voltage vs. capacity for over 20,000 Li-ion cathode compounds using high-throughput *ab initio* methods.

Implications of big data computing

1. Scientific computing is in the midst of a paradigm shift, from data-poor to data-rich computing.
2. Nowadays most data being processed on supercomputers is *experimental*, not simulation data.
3. Advanced visualization facilities are no longer optional — they are absolutely essential. A significant portion of a researcher's time is now spent analyzing and visualizing data.
4. Sophisticated statistical machine learning techniques are now being applied to classify data and to focus on "interesting" items in the data.

Machine learning for supernova research

Recently machine learning techniques were applied to help automate the supernova detection process at the PTF. Challenges include:

- Raw data must be processed in real time. Future, higher-resolution telescopes will exacerbate this problem.
- Observations must be classified as supernovas with very high reliability; false negatives or false positives both cause problems.

Preliminary results:

- Using a statistical rule ensemble method, we were above to find reliable classifications using only 21 of the 39 attributes now being used.
- Work is currently underway to develop a new preprocessing method that extracts even better features, and to decrease the false negative rate.

DANGER AHEAD

In spite of the successes of big data computing, danger lies ahead:
Supercomputers operating on big data can generate nonsense faster than ever before!

Key concerns:

- Are the algorithms, data sources and processing methods well documented?
- Are the results reproducible by other researchers, or even by the same team of researchers?
- Are the results statistically sound?
- Are the results numerically reliable?
- Have the results been validated using tests designed by the researchers or others?

Reproducibility in scientific computing

A December 2012 workshop on reproducibility in computing, held at Brown University in Rhode Island, USA, noted that

> *Science is built upon the foundations of theory and experiment validated and improved through open, transparent communication. With the increasingly central role of computation in scientific discovery, this means communicating all details of the computations needed for others to replicate the experiment. ... The "reproducible research" movement recognizes that traditional scientific research and publication practices now fall short of this ideal, and encourages all those involved in the production of computational science ... to facilitate and practice really reproducible research.*

- V. Stodden, D. H. Bailey, J. Borwein, R. J. LeVeque, W. Rider and W. Stein, "Setting the default to reproducible: Reproducibility in computational and experimental mathematics," `http://www.davidhbailey.com/dhbpapers/icerm-report.pdf`.

Reproducibility in scientific computing, continued

Issues identified in the ICERM report and other studies include:

- The need to carefully document the full context of computational experiments—system environment, input data, code used, computed results, etc.
- The need to save the code and output data in a permanent repository.
- The need for reviewers, research institutions and funding agencies to recognize the importance of computing and computing professionals, and to allocate funding for after-the-grant support and repositories.
- The increasing importance of numerical reproducibility, and the need for tools to ensure and enhance numerical reliability.
- The need to encourage publication of negative results—other researchers can often learn from them.
- The re-emergence of the need to ensure responsible reporting of performance.

Reproducibility in biomedicine

The biomedical field has been stung by numerous cases where pharma products look good based on clinical trials, but later disappoint in real-world usage, or the results cannot be reproduced in separate studies. Examples:

- In 2004, GlaxoSmithKline acknowledged that while some trials of Paxil found it effective for depression in children, other unpublished studies showed no benefit.
- In 2011, Bayer researchers reported that they were able to reproduce the results of only 17 of 67 published studies they examined.
- In 2012, Amgen researchers reported that they were able to reproduce the results of only 6 of 53 published cancer studies.
- In 2014, a review of Tamiflu found that while it made flu symptoms disappear a bit sooner, it did not stop serious complications or keep people out of the hospital.

These experiences have exposed a fundamental flaw in methodology:
Only publicizing the results of successful trials introduces a bias into the results.

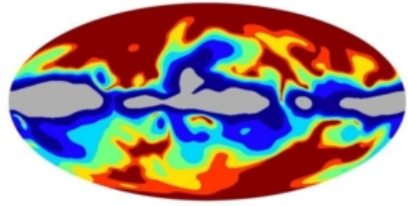
The AllTrials movement would require all results to be public: `http://www.alltrials.net`

Reproducibility in physics

In March 2014, a team of researchers from Harvard University made the dramatic announcement that they had discovered an interesting "twisting" pattern in cosmic microwave background data, measured using their BICEP2 experimental system.

This pattern fit very well with the hypothesized pattern of the most commonly assumed model of "inflation" in the first tiny fraction of a second after the big bang, and thus has been trumpeted as the first experimental evidence of the inflationary cosmology.

But other researchers had difficulty reconstructing the claimed results. Finally, two teams challenged the BICEP2 findings, saying that the results could more readily be explained by dust in the Milky Way.



- Ron Cowen, "Doubt grows about gravitational waves detection," *Scientific American*, 2 Jun 2014.

Reproducibility in social science

The "blank slate" paradigm (1920–1990):

- The human mind at birth is a *tabula rasa* ("blank slate").
- Heredity and biology play no significant role in human psychology; all personality and behavioral traits are socially constructed.
- Pre-modern societies were peaceful, devoid of psychological and social problems.

Current consensus, based on latest research:

- Humans at birth possess sophisticated facilities for social interaction, language acquisition, pattern recognition, navigation and counting.
- Heredity, evolution and biology are major factors in human personality.
- Some personality traits are more than 50% heritable.
- Pre-modern societies had more crime, war and social problems than today.

How did the 20th century social scientists get it so wrong?

- Sloppy experimental methodology and analysis.
- Pervasive wishful thinking and politically correct biases.
- Ignoring or dismissing data that runs counter to predisposition.

- S. Pinker, *The Blank Slate: The Modern Denial of Human Nature*, Penguin Books, 2003.

Reproducibility in finance

Finance, like the pharmaceutical world, has been stung with numerous instances of investment strategies that look great on paper, but fall flat in practice. A primary cause is *statistical overfitting of backtest (historical market) data*.

When a computer can analyze thousands or millions of variations of a given strategy, it is almost certain that the best such strategy, measured by backtests, will be overfit and thus of dubious value.

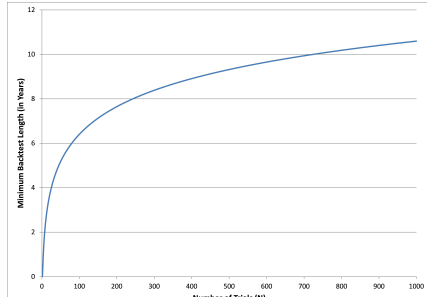
In two 2014 papers by DHB, JMB, Marcos Lopez de Prado and Jim Zhu, we derive (a) a formula relating the number of trials to the minimum backtest length, and (b) a formula for the probability of backtest overfitting. We also show that under the assumption of memory in markets, overfit strategies are actually prone to *lose* money.

- D. H. Bailey, J. M. Borwein, M. Lopez de Prado and Q. J. Zhu, "Pseudo-mathematics and financial charlatanism: The effects of backtest over fitting on out-of-sample performance," *Notices of the American Mathematical Society*, May 2014, pg. 458–471.
- D. H. Bailey, J. M. Borwein, M. Lopez de Prado and Q. J. Zhu, "The probability of backtest overfitting," manuscript, 12 Feb 2014, `http://ssrn.com/abstract=2326253`.

How easy is it to overfit a backtest? Answer: Very easy!

- If only 2 years of daily backtest data are available, then no more than 7 strategy variations should be tried.
- If only 5 years of daily backtest data are available, then no more than 45 strategy variations should be tried.
- *A backtest that does not report the number of trials N makes it impossible to assess the risk of overfitting.*
- Given any desired performance level, a financial researcher just needs to keep trying alternative parameters for that strategy!

$$MinBTL \approx \left(\frac{(1-\gamma)Z^{-1}\left[1-\frac{1}{N}\right] + \gamma Z^{-1}\left[1-\frac{1}{N}e^{-1}\right]}{E[\max_N]} \right)^2$$



An absurd investment strategy

- A financial advisor sends letters to $10,240 = 10 \times 2^{10}$ potential clients, with 5120 letters predicting a certain security will go up, and the other half predicting it will go down.
- One month later, the advisor sends letters only to the 5120 investors who were previously sent the correct prediction, with 2560 letters predicting a certain security will go up, and the other half predicting it will go down.
- The advisor continues this process for 10 months.
- The remaining ten investors, so impressed by the advisor's ten consecutive spot-on predictions, will entrust to him/her all of their assets!

This strategy is absurd; even fraudulent.

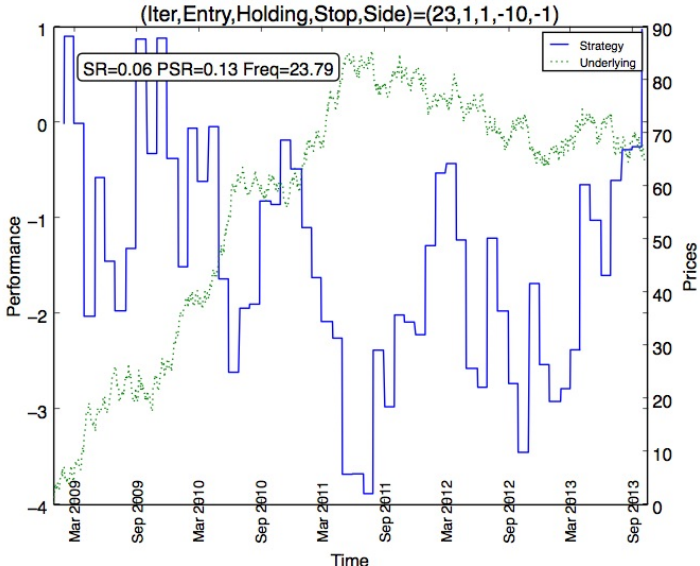
But why is marketing a statistically overfit strategy, where potential investors are not informed of the number of trials behind the strategy, any different?

Optimizing an investment strategy to fit pseudorandom time series

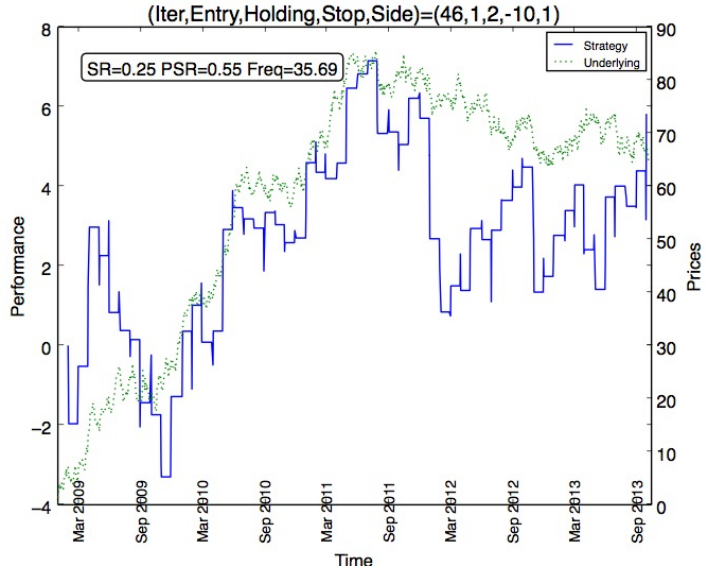
- The following 23 viewgraphs present the results of different steps in an attempt to find an "optimal" investment strategy, based on a fixed market price dataset.
- The underlying dataset was generated by a pseudorandom number generator!
- As you can see, by tweaking some very basic parameters (entry price, sell price, stop-loss, etc), we can fit and "predict" the underlying dataset quite well.
- The final (24th) viewgraph presents the results of implementing the resulting strategy on a continuation of the underlying (pseudorandom) dataset.

- Code due to Stephanie Ger, Harvard University, modified from earlier code by Marcos Lopez de Prado.

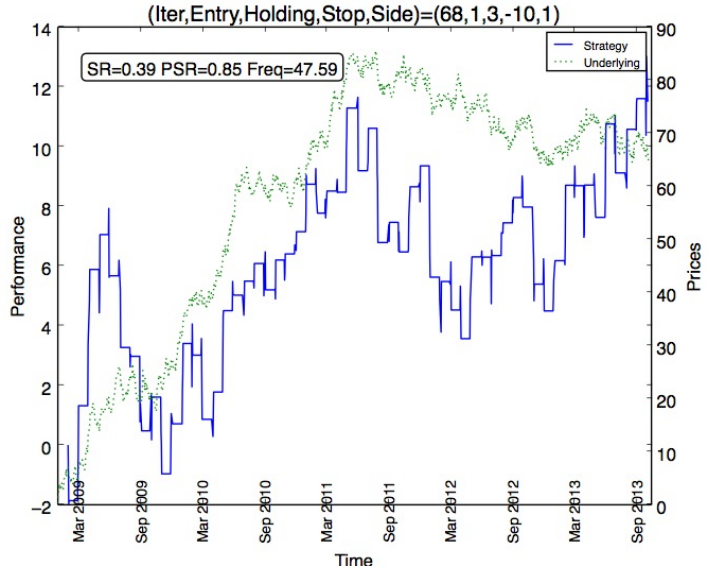
Optimizing an investment strategy to fit pseudorandom time series, pg. 01



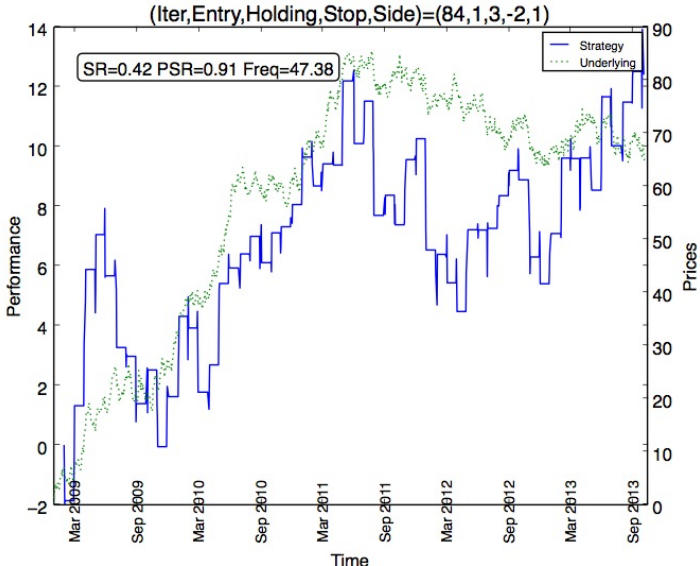
Optimizing an investment strategy to fit pseudorandom time series, pg. 02



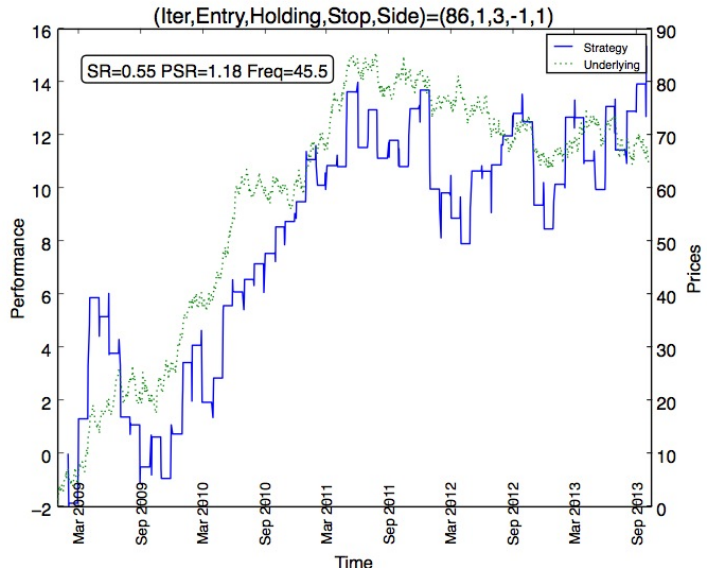
Optimizing an investment strategy to fit pseudorandom time series, pg. 03



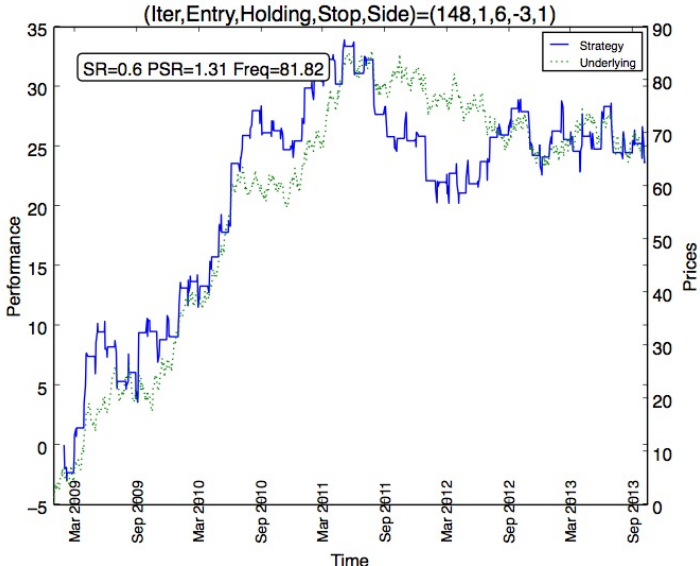
Optimizing an investment strategy to fit pseudorandom time series, pg. 04



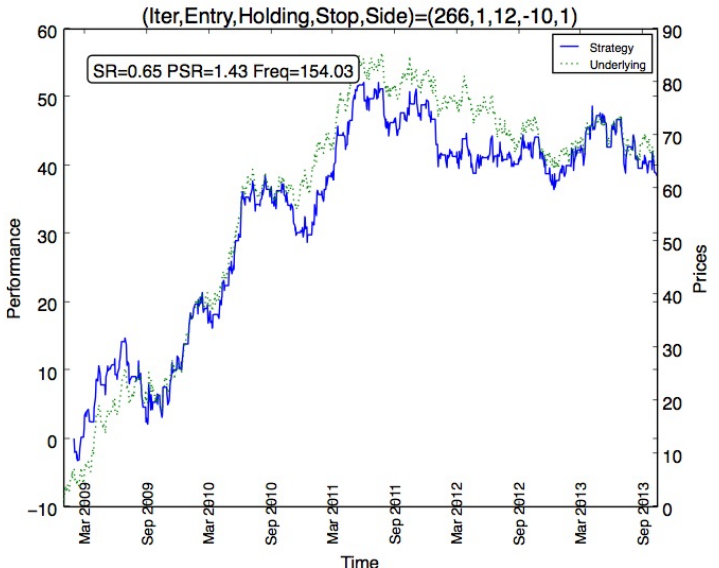
Optimizing an investment strategy to fit pseudorandom time series, pg. 05



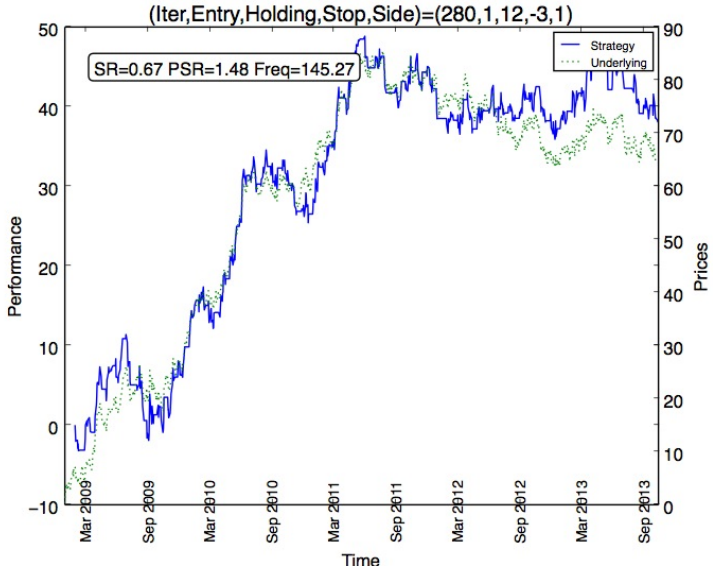
Optimizing an investment strategy to fit pseudorandom time series, pg. 06



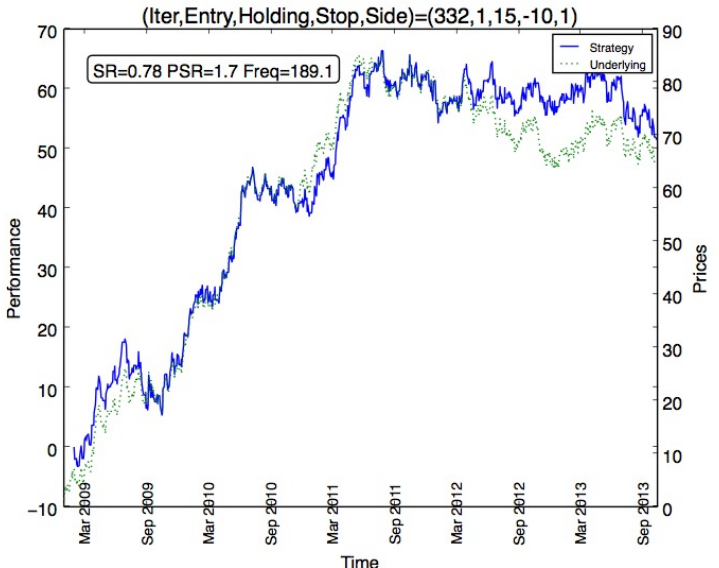
Optimizing an investment strategy to fit pseudorandom time series, pg. 07



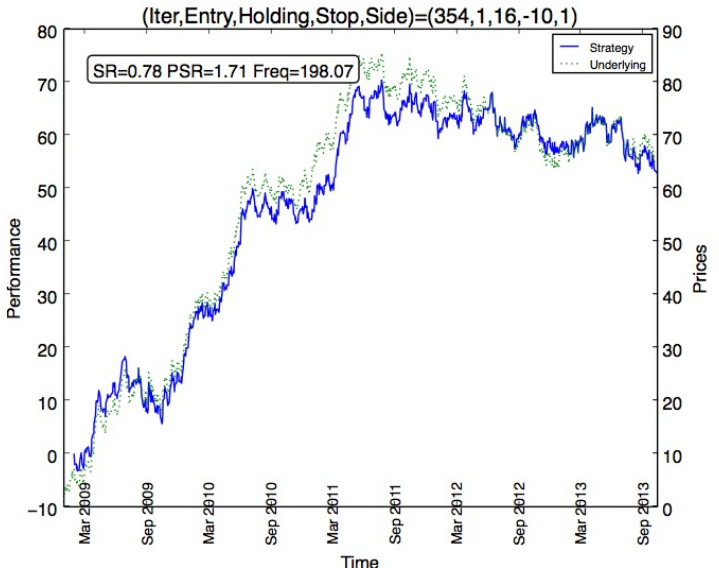
Optimizing an investment strategy to fit pseudorandom time series, pg. 08



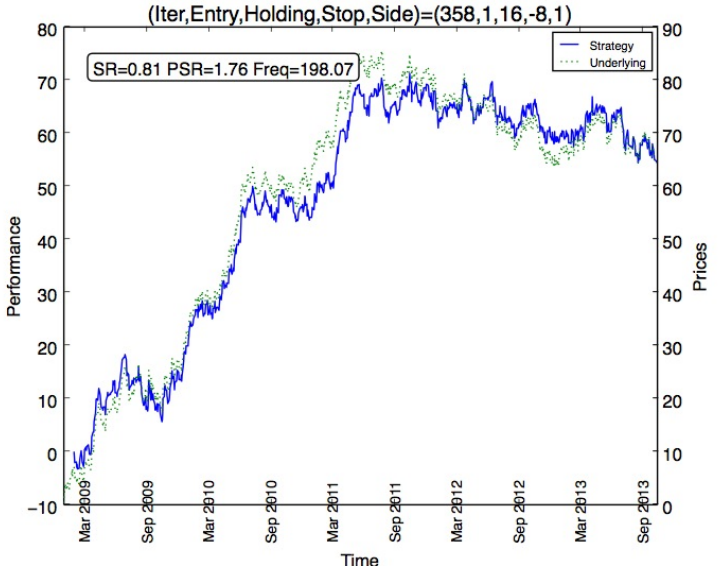
Optimizing an investment strategy to fit pseudorandom time series, pg. 09



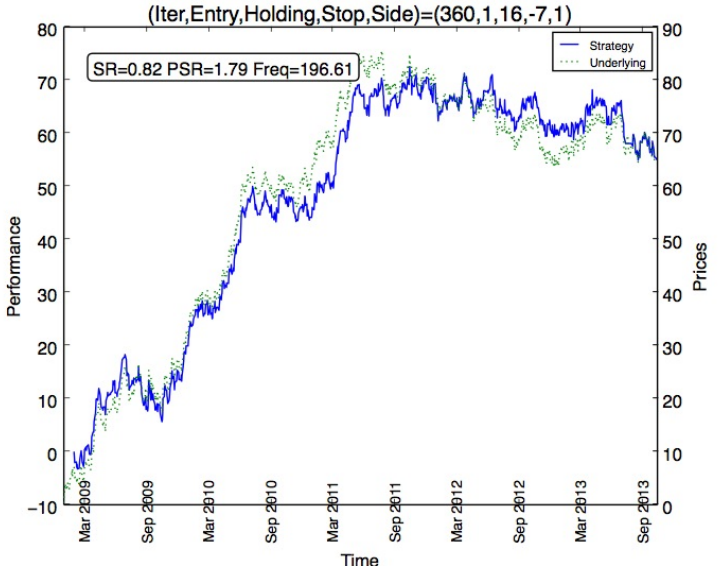
Optimizing an investment strategy to fit pseudorandom time series, pg. 10



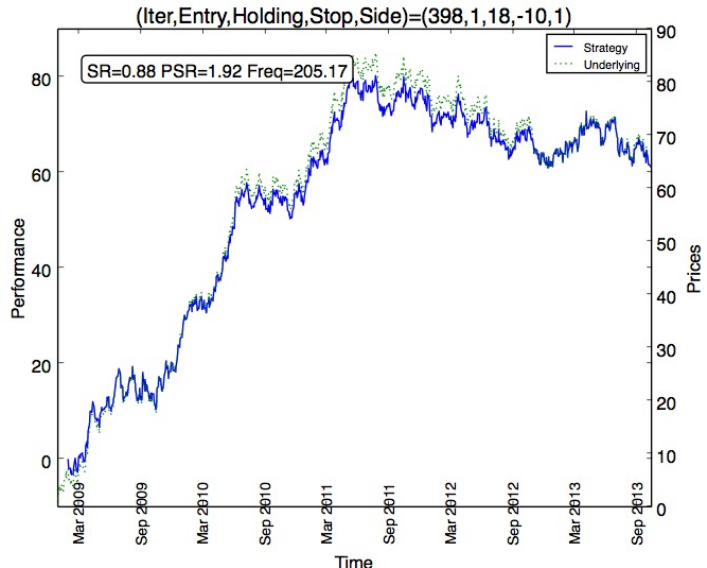
Optimizing an investment strategy to fit pseudorandom time series, pg. 11



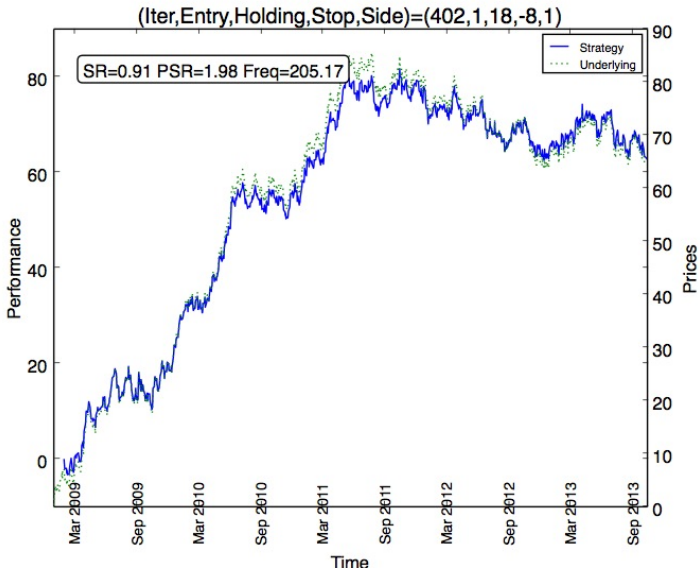
Optimizing an investment strategy to fit pseudorandom time series, pg. 12



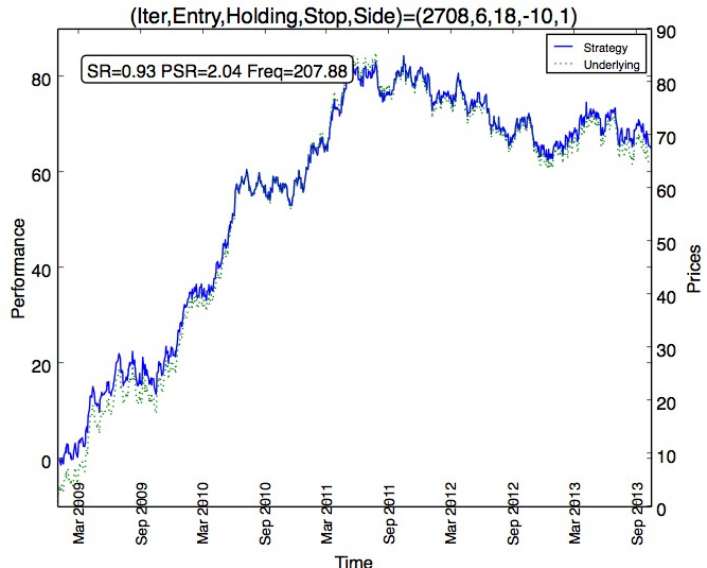
Optimizing an investment strategy to fit pseudorandom time series, pg. 13



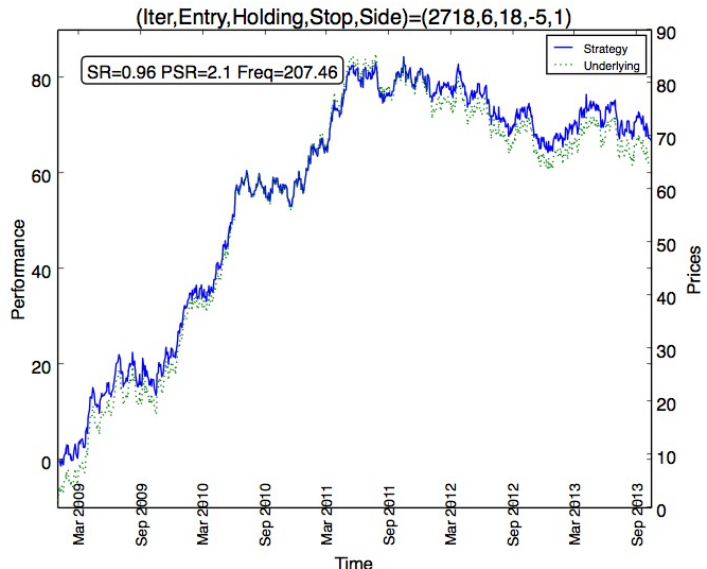
Optimizing an investment strategy to fit pseudorandom time series, pg. 14



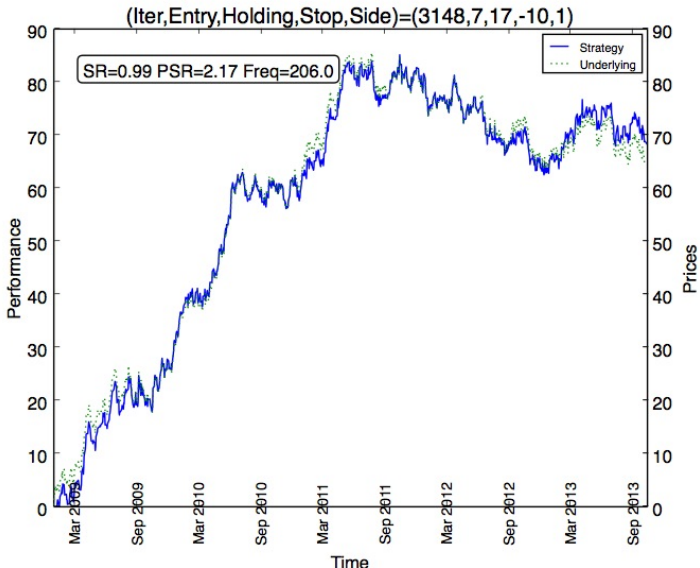
Optimizing an investment strategy to fit pseudorandom time series, pg. 15



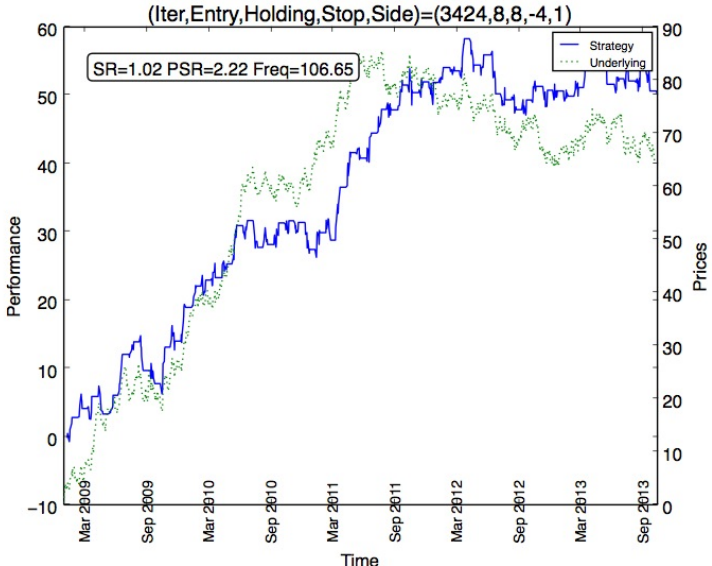
Optimizing an investment strategy to fit pseudorandom time series, pg. 16



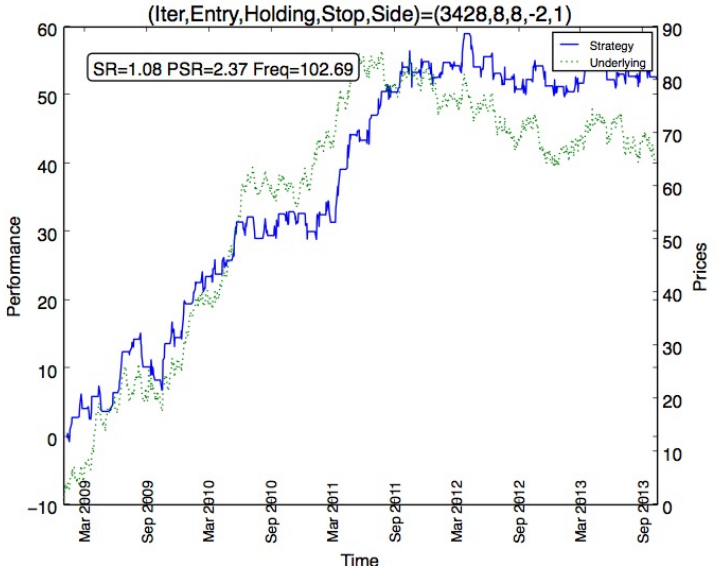
Optimizing an investment strategy to fit pseudorandom time series, pg. 17



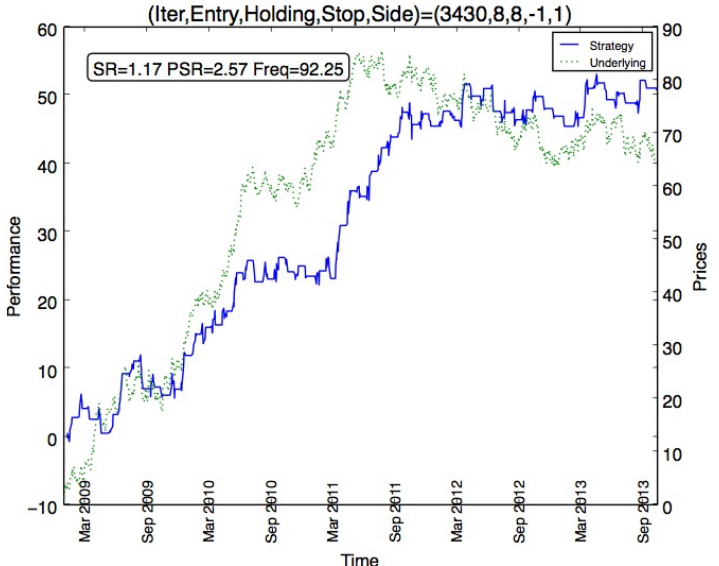
Optimizing an investment strategy to fit pseudorandom time series, pg. 18



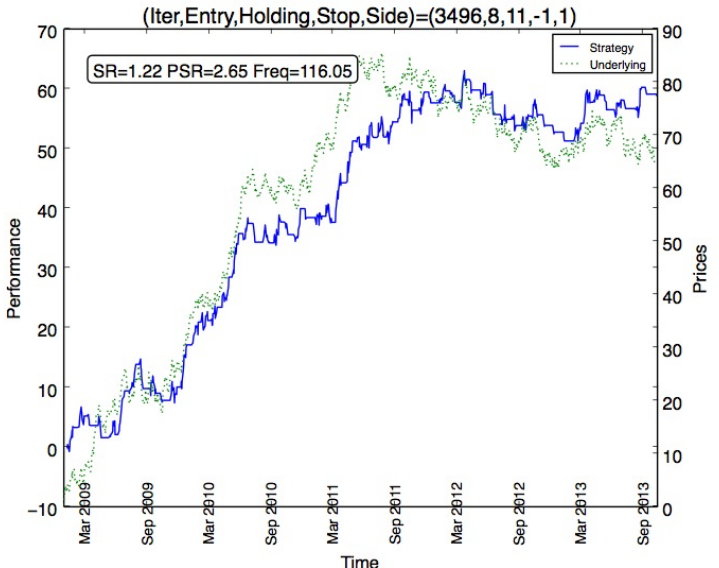
Optimizing an investment strategy to fit pseudorandom time series, pg. 19



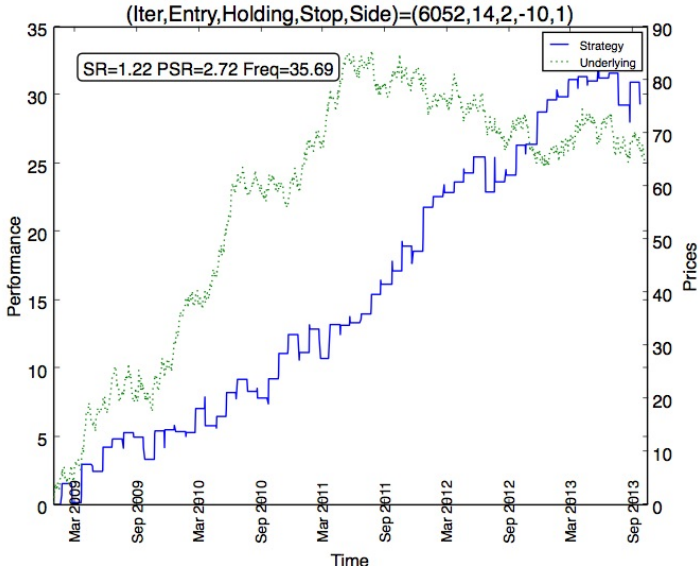
Optimizing an investment strategy to fit pseudorandom time series, pg. 20



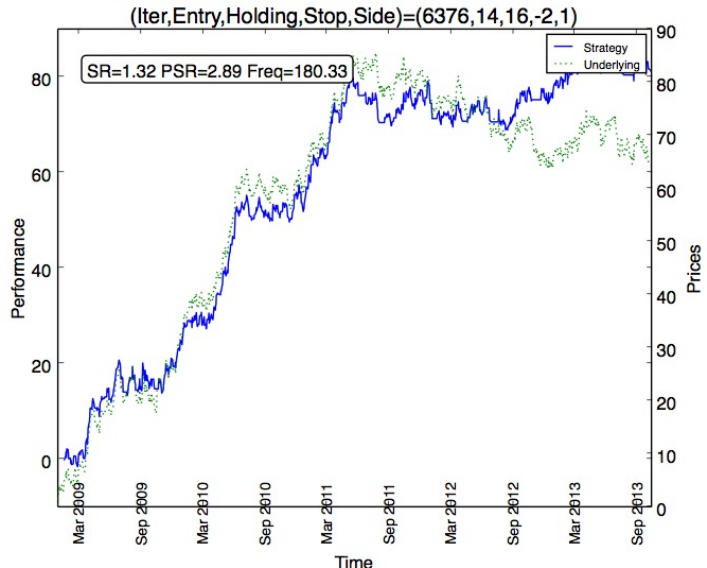
Optimizing an investment strategy to fit pseudorandom time series, pg. 21



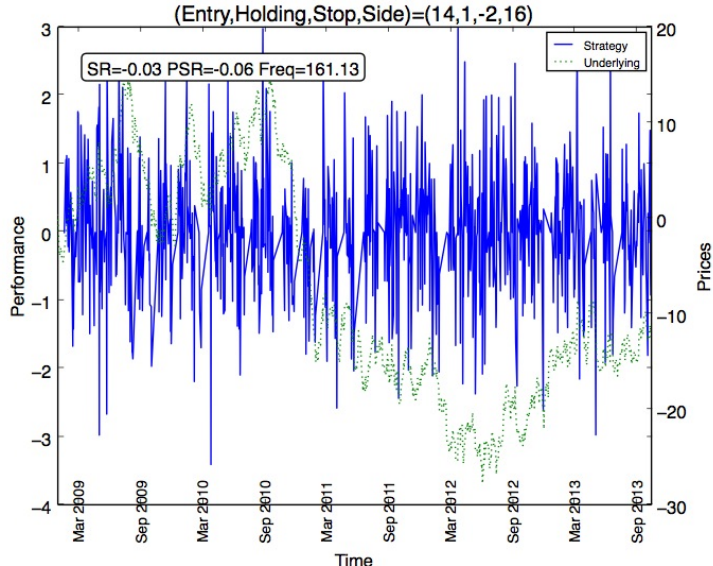
Optimizing an investment strategy to fit pseudorandom time series, pg. 22



Optimizing an investment strategy to fit pseudorandom time series, pg. 23



Deploying the resulting strategy on a continuation of the time series



John von Neumann on the danger of statistical overfitting

On the danger of statistical overfitting, Enrico Fermi recalled John von Neumann's warning:

> *I remember my friend Johnny von Neumann used to say, with four parameters I can fit an elephant, and with five I can make him wiggle his trunk.*

Why the silence in the mathematical finance community?

- Historically scientists have led the way in exposing those who utilize pseudoscience to extract a commercial benefit: e.g., in the 18th century, physicists exposed the nonsense of astrologers.
- Yet financial mathematicians in the 21st century have remained disappointingly silent with the regards to those in the community who, knowingly or not:
  1. Fail to disclose the number of models or variations that were used to develop an investment strategy.
  2. Make vague predictions that do not permit rigorous testing and falsification.
  3. Misuse probability theory, statistics and stochastic calculus.
  4. Use dubious technical jargon: "stochastic oscillators," "Fibonacci ratios," "cycles," "Elliot wave," "Golden ratio," "parabolic SAR," "pivot point," "momentum," etc.

As we recently wrote in our paper "Pseudo-Mathematics and Financial Charlatanism": **"Our silence is consent, making us accomplices in these abuses."**

- D. H. Bailey, J. M. Borwein, M. Lopez de Prado and Q. J. Zhu, "Pseudo-mathematics and financial charlatanism: The effects of backtest over fitting on out-of-sample performance," *Notices of the American Mathematical Society*, May 2014, pg. 458–471.

Numerical reproducibility in high-performance computing

The report mentioned above on reproducibility in high-performance computing noted:

> *Numerical round-off error and numerical differences are greatly magnified as computational simulations are scaled up to run on highly parallel systems. As a result, it is increasingly difficult to determine whether a code has been correctly ported to a new system, because computational results quickly diverge from standard benchmark cases. And it is doubly difficult for other researchers, using independently written codes and distinct computer systems, to reproduce published results.*

- V. Stodden, D. H. Bailey, J. Borwein, R. J. LeVeque, W. Rider and W. Stein, "Setting the default to reproducible: Reproducibility in computational and experimental mathematics," Jan 2013, available at `http://www.davidhbailey.com/dhbpapers/icerm-report.pdf`.

Numerical reliability

Many applications routinely use either 32-bit or 64-bit IEEE arithmetic, and employ fairly simple algorithms, assuming that all is well. But problems can arise:

1. Large-scale, highly parallel simulations, running on systems with hundreds of thousands or millions of processors — numerical sensitivities are greatly magnified.
2. Certain applications with highly ill-conditioned linear systems.
3. Large summations, especially those involving $+/-$ terms and cancellations.
4. Long-time, iterative simulations (such as molecular dynamics or climate models).
5. Computations to resolve small-scale phenomena.
6. Studies in computational physics or experimental mathematics often require huge precision levels.

- D. H. Bailey, R. Barrio, and J. M. Borwein, "High precision computation: Mathematical physics and dynamics," *Applied Mathematics and Computation*, vol. 218 (2012), pg. 10106–10121.

Analysis of collisions at the Large Hadron Collider

- The 2012 discovery of the Higgs boson at the ATLAS experiment in the LHC relied crucially on the ability to track charged particles with exquisite precision (10 microns over a 10m length) and high reliability (over 99% of roughly 1000 charged particles per collision correctly identified).
- Software: 5 millions line of C++ and python code, developed by roughly 2000 physicists and engineers over 15 years.
- Recently, in an attempt to speed up the calculation, researchers found that merely changing the underlying math library resulted in some collisions being missed or misidentified.

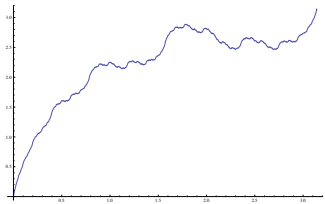
Questions:

- How serious are these numerical difficulties?
- How can they be tracked down?
- How can the library be maintained, producing numerically reliable results?

Enhancing reproducibility with selective usage of high-precision arithmetic

Problem: Find the arc length of the irregular function $g(x) = x + \sum_{0 \le k \le 10} 2^{-k} \sin(2^k x)$, over the interval $(0, \pi)$ (using 10^7 abscissa points).

- If this computation is done with ordinary double precision arithmetic, the calculation takes 2.59 seconds and yields the result 7.073157029008510.
- If it is done using all double-double arithmetic (31-digit accuracy), it takes 47.39 seconds seconds and yields the result 7.073157029007832.
- But if only the summation is changed to double-double, the result is identical to the double-double result (to 15 digits), yet the computation only takes 3.47 seconds.



Graph of $g(x) = x + \sum_{0 \le k \le 10} 2^{-k} \sin(2^k x)$, over $(0, \pi)$.

U.C. Berkeley's CORVETTE project and the "Precimonious" tool

Objective: Develop software facilities to find and ameliorate numerical anomalies in large-scale computations:

- Facilities to test the level of numerical accuracy required for an application.
- Facilities to delimit the portions of code that are inaccurate.
- Facilities to search the space of possible code modifications.
- Facilities to repair numerical difficulties, including usage of high-precision arithmetic.
- Facilities to navigate through a hierarchy of precision levels (32-bit, 64-bit, 80-bit or higher as needed).

The current version of this tool is known as "Precimonious."

- C. Rubio-Gonzalez, C. Nguyen, H. D. Nguyen, J. Demmel, W. Kahan, K. Sen, D. H. Bailey and C. Iancu, "Precimonious: Tuning assistant for floating-point precision," manuscript, May 2013.

Reproducibility in highly parallel computing: History from 1990-1994

Background:

- Many new parallel systems had been introduced; each claimed theirs was best.
- Many researchers were excited about the potential of highly parallel systems.
- Few standard benchmarks and testing methodologies had been established.
- It was hard to reproduce published performance results; much confusion reigned.
- Overall, the level of rigor and peer review in the field was rather low.

In response, in 1991 DHB published a humorous essay "Twelve Ways to Fool the Masses," poking fun at some of the abuses.

Since abuses continued, DHB presented a talk at Supercomputing 1992 and published a paper with specific examples.

- D. H. Bailey, "Misleading performance reporting in the supercomputing field," *Scientific Programming*, vol. 1., no. 2 (Winter 1992), pg. 141–151.

1991 paper: "Twelve ways to fool the masses in highly parallel computing"

1. Quote 32-bit performance results, not 64-bit results, but don't mention this in paper.
2. Present performance figures for an inner kernel, then represent these figures as the performance of the entire application.
3. Quietly employ assembly code and other low-level language constructs.
4. Scale up the problem size with the number of processors, but omit any mention of this.
5. Quote performance results projected to a full system.
6. Compare your results against scalar, unoptimized code on conventional systems.
7. When run times are compared, compare with an old code on an obsolete system.
8. Base Mflop/s rates on the operation count of the parallel implementation, instead of the best practical serial algorithm.
9. Quote performance as processor utilization, parallel speedups or Mflop/s per dollar.
10. Mutilate the algorithm used in the parallel implementation to match the architecture.
11. Measure parallel run times on a dedicated system, but measure conventional run times in a busy environment.
12. If all else fails, show pretty pictures and animated videos, and don't discuss performance.

1992 paper: Scaling performance results to full-sized system

In some published papers and conference presentations, performance results on small-sized parallel systems were linearly scaled to full-sized systems, without even clearly disclosing this fact.

Example: 8,192-CPU performance results were linearly scaled to 65,536-CPU results, simply by multiplying by eight.

Excuse: "We can't afford a full-sized system."

This and the other examples mentioned in the next few viewgraphs are presented in:

- D. H. Bailey, "Misleading performance reporting in the supercomputing field," *Scientific Programming*, vol. 1., no. 2 (Winter 1992), pg. 141–151.

1992 paper: Using inefficient algorithms on highly parallel systems

In many cases, inefficient algorithms were employed for the highly parallel implementation, requiring many more operations, thus producing artificially high Mflop/s rates:

- Numerous researchers cited parallel PDE performance based explicit schemes, where implicit schemes were known to be much better.
  Excuse: Explicit schemes "run better" on the researchers' parallel system.
- One paper cited performance for computing a 3D discrete Fourier transform by direct evaluation of the defining formula ($8n^2$ operations), rather than by using a fast Fourier transform ($5n \log_2 n$).
  Excuse: Direct computation was "more appropriate" for the architecture being analyzed.

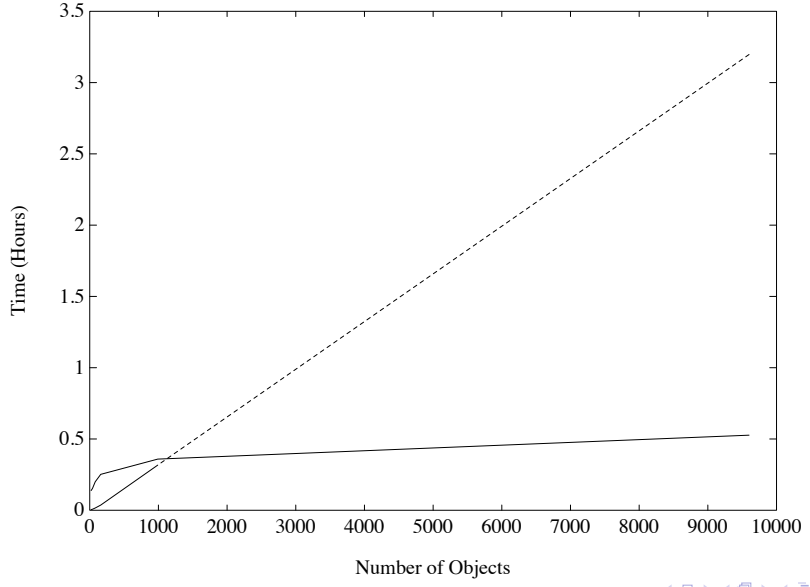
Both examples violate a rule of professional performance reporting, namely to base the operation count (when computing Mflop/s or Gflop/s rates) on the *best practical serial algorithm*, no matter what scheme was actually used on the parallel system.

1992 paper: Not actually performing a claimed computation

Abstract of published paper: "The current Connection Machine implementation runs at 300-800 Mflop/s on a full [64K] CM-2, or at the speed of a single processor of a Cray-2 on 1/4 of a CM-2."

- Excerpt from text: "This computation requires 568 iterations (taking 272 seconds) on a 16K Connection Machine."
  In other words, the computation was run on a 16K system, not on a 64K system; the figures cited in the Abstract were merely multiplied by four.
- Excerpt from text: "In contrast, a Convex C210 requires 909 seconds to compute this example. Experience indicates that for a wide range of problems, a C210 is about 1/4 the speed of a single processor Cray-2."
  In other words, the computation mentioned in the Abstract was not actually run on a Cray-2; instead, it was run on a Convex system, and a questionable rule-of-thumb scaling factor was used to produce the Cray-2 rate.

1992 paper: Performance plot [parallel (lower) vs vector (upper)]



1992 paper: Data for performance plot

Problem size (x axis)	Parallel system run time	Vector system run time
20	8:18	0:16
40	9:11	0:26
80	11:59	0:57
160	15:07	2:11
990	21:32	19:00
9600	31:36	3:11:50*

Details in text of paper:

- In last entry, the 3:11:50 figure is an "estimate."
- The vector system code is "not optimized."

Note that the parallel system is actually slower than the vector system for all cases, except for the last (estimated) entry. Also, except for the last entry, all real data in the graph is in the lower left corner.

Fast forward to 2014: New ways to fool the masses

- Cite performance rates for a run with only one processor core active in a shared-memory multi-core node, producing artificially inflated performance (since there is no shared memory interference) and wasting resources (since most cores are idle).
  - Example: Cite performance on "1024 cores," even though the code was run on 1024 multicore nodes, one core per node, with 15 out of 16 cores idle on each node.
- Claim that since one is using a graphics processing unit (GPU) system, that efficient parallel algorithms must be discarded in favor of more basic algorithms.
- Cite performance rates only for a core algorithm (such as FFT or linear system solution), even though full-scale applications have been run on the system.
- List only the best performance figure in the paper, even though the run was made numerous times (recall the experience of pharmaceutical tests).
- Employ special hardware, operating system or compiler settings that are not appropriate for real-world production usage.
- Define "scalability" as successful execution on a large number of CPUs, regardless of performance.

Summary

- Super-powerful, highly parallel computer systems, and correspondingly powerful software, are tools of unprecedented power for researchers and engineers.
- Many fields, long starved for reliable data, are now drowning in data.
- Dangers lie ahead, particularly in reproducibility and statistical reliability.
- Potential dangers are illustrated in mathematical finance, where almost any desired performance can be achieved by massaging a model long enough.
- Numerical reproducibility is of particular concern, especially for very large-scale computations. Do results have any numerical validity?
- The issue of questionable performance reporting practices is again emerging in scientific computing.

Science or pseudoscience? That is the question.

This talk is available at `http://www.davidhbailey.com/dhbtalks/dhb-pseudoscience.pdf`.