The Many Pitfalls Of Backtesting: Why Your Backtest Is Almost Certainly Wrong

Keith Glennan — February 2026 (v1.3)

Contents

1 Introduction

The barriers to entry for quantitative trading research have fallen dramatically over the past decade. Many trading platform products promise built-in backtesting capabilities, often touting their robustness credentials. And for those who are not using a trading platform, Python and open-source libraries such as Backtrader, Zipline, VectorBT, and various pandas-based frameworks have made it possible for individuals with modest programming skills to construct, test, and evaluate systematic trading strategies. This accessibility has been widely celebrated as a democratising force in financial markets.

But accessibility and rigour are not synonymous. The ease with which a backtested equity curve can be produced has led to a proliferation of strategy research that ranges from mildly optimistic to grossly unrealistic. Individual backtest errors are not the central issue; the ecosystem of tools, data sources, educational materials, and community conventions systematically biases users toward overestimating strategy performance. The typical retail back-tester does not know what they are not modelling. The tools do nothing to tell them.

This paper catalogues the principal failure modes of backtesting as commonly practised, with particular emphasis on areas where the gap between simulation and reality is largest and least understood. I give special attention to the under-examined problem of bar resolution (the information that is lost when price paths are compressed into OHLC bars, particularly at the daily level), which affects even strategies operating on longer timeframes, and to the pervasive data quality issues that contaminate results at the source level I also examine the limitations of the robustness testing methods that traders commonly rely upon to validate their results: Monte Carlo simulation, synthetic data generation, and walk-forward analysis. And I examine broader ecosystem factors, including the self-reinforcing dominance of Python, the monoculture of approaches encouraged by standard frameworks, and the tendency to treat signal generation in isolation from risk management and portfolio construction.

I should note at the outset that the criticisms presented here are not directed at Python as a programming language per se. Python is a capable general-purpose language with legitimate applications across many domains. My concern is with the ecosystem that has formed around it in the trading context: the libraries, the conventions, the educational materials, and the implicit assumptions absorbed by developers who adopt the dominant toolchain without sufficient critical examination.

2 Data Quality: The Contaminated Foundation

2.1 The Taxonomy of Data Errors

A backtest is only as reliable as the data on which it is built, and the quality of freely available and even commercially distributed historical market data is far worse than most traders realise. Data quality issues fall into several categories, each capable of producing misleading results.

- **Erroneous prints:** A single bad tick, a trade printed at a price far from the prevailing market, whether due to a fat-finger error, an exchange reporting glitch, or a data vendor processing failure, can generate an apparent trading opportunity with outsized returns that never actually existed. In tick or minute-level data, such errors may be individually identifiable, but in daily data they can distort the reported high, low, or settlement price without any obvious indication of error. A daily bar showing an anomalous low, for example, could trigger a limit-order entry in a mean-reversion system and subsequently show a profitable exit, when in reality the print that generated the entry signal was erroneous and no fill would have occurred.

- **Timestamp errors:** Markets operate across time zones with varying session times, and data vendors are not always consistent in how they handle session boundaries, daylight saving transitions, or the distinction between electronic and pit trading sessions. A strategy that relies on the timing of price action (for example, a breakout of the prior session high) can produce very different results depending on whether the data vendor defines "session" consistently. Compounding this problem, different approaches to timestamping can be found even within data sets from the same provider. One product may deliver timestamps in UTC, while another is localised to the exchange's native time zone, and a third uses an arbitrary reference time zone defined by the vendor. The researcher who loads data from multiple instruments or multiple products without verifying the timestamp convention of each is silently mixing time zones, a class of error that will not produce obviously wrong prices but will misalign bars across instruments and corrupt any cross-market signal. It would not be unreasonable to assume that, in 2026, the timestamping of financial data is a solved problem. Data providers seem almost to go out of their way to demonstrate that it is not.

- **Missing data:** Gaps in the record, whether from holidays, half-day sessions, exchange outages, or vendor processing failures, can introduce artefacts into any indicator that relies on a continuous time series. Moving averages and volatility

estimates, among others, are sensitive to missing observations.

- **Phantom data:** The inverse of missing data: bars or ticks reported for periods in which no trading actually occurred. This can arise from vendor systems that generate synthetic bars to fill gaps in the record, from incorrect session boundary definitions that attribute activity to the wrong date or session, or from exchange-reported indicative prices being treated as actual trades. A strategy backtested over phantom bars will generate signals and simulated fills at prices that were never available in the market. Unlike erroneous prints, which at least reflect a real (if incorrect) data point, phantom data fabricates trading activity where none existed, and the resulting backtest entries and exits are entirely fictitious.

These error categories are not rare edge cases. They are routine. Data vendor infrastructure and processing pipelines are themselves significant and demonstrably provable sources of error. Vendors aggregate data from multiple exchange feeds, apply their own session definitions, normalise timestamps across time zones, construct continuous contracts, and backfill historical records. Each step introduces opportunities for systematic errors that propagate silently into every downstream backtest. When these errors are identified and reported to vendors, the response is frequently inadequate: errors are acknowledged but not corrected, or corrections are applied only to newly delivered data while the historical record remains contaminated. This may seem implausible, but vendors serving a retail client base that rarely audits data quality have little incentive to invest in the engineering effort required to correct historical records. Traders end up developing and evaluating strategies on data that the vendor itself knows to be defective.[1]

I want to be blunt about this: the trader who assumes that commercially distributed data has been rigorously validated is making an assumption that is demonstrably false.

Cryptocurrency markets are a worse case again. Many crypto venues engage in wash trading: fabricated volume reported to inflate apparent liquidity. Credible analyses have estimated that a large fraction of reported crypto volume, even on major exchanges, is non-genuine. A backtest that uses reported volume for signal confirmation or liquidity estimation may be building on data that is, in a meaningful sense, fictional. This is different in kind from the erroneous prints and timestamp errors discussed above. The data is not wrong by accident. It is wrong by design.

The April 2020 WTI crude oil event, in which the May 2020 contract settled at $-37.63 per barrel, exposed a related class of data assumption that few traders had considered. Negative prices broke systems at every level: brokers could not process them, trading

[1] I have encountered vendors who, when confronted with documented errors in their data, pretend that they are serious about fixing it, but never do.

platforms rejected them, and data feeds that stored prices as unsigned values simply could not represent them. The event has since been incorporated into historical data sets, but vendors differ in how they handle it. Some report the negative settlement correctly; others clip it to zero or omit the session entirely. Any of these choices contaminates a backtest differently. What this exposed is that data schemas encode implicit assumptions about what prices can be, and those assumptions are occasionally wrong.

2.2 Vendor Divergence and Database Drift

The taxonomy above describes categories of error in the abstract. A more direct way to see the problem is to run an identical strategy against multiple vendors' data and compare the results. Concretum Group did exactly this: they ran the same Opening Range Breakout strategy, with the same code, the same parameters, and the same nominal time period, against intraday data from five providers (Polygon, Interactive Brokers, Databento, IQFeed, and Alpaca). The terminal portfolio value ranged from roughly $226,000 to $726,000 depending solely on whose data the engine consumed. Identical code; threefold variation in equity.

The mechanisms behind the divergence map onto the error categories described earlier, but several are worth singling out. Phantom highs and lows: isolated price spikes that appeared in one vendor's bars and nowhere else, reflecting differences in how each vendor filtered trades for inclusion. Stale bars: extended sequences of unchanged OHLC values, typically the result of dropped ticks or improperly aggregated data. Tick-to-bar assignment ambiguity: trades occurring at exact time boundaries (09:35:00.000 being the canonical case) were assigned to different one-minute bars by different vendors, and the resulting bars contained different OHLC values from vendor to vendor. Early-close day leakage: some vendors continued to emit intraday bars past the official close on half-day sessions while others stopped cleanly. Venue coverage divergence: providers that consolidated from SIP feeds saw a different trade population than providers using proprietary subsets, and the implied highs, lows, and closes of identical time intervals differed accordingly.

The amplification effect documented in the Concretum study is itself a useful diagnostic. Strategies whose stops were anchored to intraday highs and lows showed extreme dispersion across vendors. Strategies whose stops were anchored to ATR, a quantity derived from daily ranges with substantial smoothing, converged tightly across the same vendors. If a backtest's behaviour is sensitive to which vendor's intraday bars happened to be on the disk that day, the strategy is implicitly trading data-pipeline noise dressed up as a market signal. Running the same code against an independent source is one of the

cheapest sanity checks available, and almost nobody does it.

A related and equally damaging problem is temporal drift in a single vendor's data. The Concretum authors compared their own archive of Interactive Brokers minute bars from 2023 against a fresh download from the same vendor in 2026 for the same instrument and the same dates. The 2023 copy showed continuous price movement across the session. The 2026 copy contained more than 350 out of 390 one-minute bars with zero variation in the same session. Same vendor, same query, different data. Whatever pipeline change caused the divergence, the practical consequence is that any backtest result obtained against a third-party feed is reproducible only if the dataset is archived at the time the backtest is run. Re-running the same code against the same vendor a year later may produce different results because the data has been silently rewritten in between. The retail back-tester who treats the vendor's database as a stable historical record is making an assumption the vendor itself does not honour.

Snapshot the dataset at the time the backtest is run, and archive it alongside the code. If the strategy depends on intraday extremes, also run it against at least one independent source. None of this is technically demanding. It is, again, a discipline issue, and it is not done because the retail ecosystem assumes the data is fine.

2.3 Historical Trading Venue Transitions

An underappreciated data quality problem in futures markets arises from the historical transition in trading venues. Many futures contracts have passed through three distinct eras: a pit-only era in which all trading occurred on exchange floors via open outcry; a transitional era in which pit and electronic trading coexisted, often with different session hours, different liquidity profiles, and occasionally different price discovery characteristics; and the current electronic-only era in which the pit has been entirely eliminated.

Each era produces data with very different properties. Pit-only data typically reflects shorter trading sessions with different volatility patterns, wider effective spreads, and reporting latencies that can affect the accuracy of recorded timestamps. During the transitional period, the relationship between pit and electronic prices was not always straightforward; the pit session and the electronic session could trade at different prices, and the "official" settlement price might be derived from one venue while most executable liquidity resided in the other. The electronic-only era brought near-continuous trading hours and entirely different order book dynamics.

These eras are not interchangeable. A backtest that treats a twenty- or thirty-year futures history as a homogeneous data set is implicitly assuming that the microstructure of 1995

is comparable to that of 2025. In reality, a strategy that would have been executable in the electronic era may have been impractical in the pit era due to execution delays and wider spreads, quite apart from the impossibility of automated order placement. Conversely, strategies that exploited inefficiencies specific to open-outcry markets (such as the predictable patterns around pit opening and closing) ceased to function when the trading floor was eliminated. Failing to segment historical data by venue regime, or at minimum to acknowledge the structural breaks that venue transitions introduce, can produce backtest results that reflect no single coherent market environment.

2.4 Look-Ahead Bias and Data Asynchronicity

Look-ahead bias is among the most dangerous classes of data error: the implicit assumption that information was available at a time before it actually existed. The most obvious forms (using a future price to calculate a current signal) are easily avoided. But subtler forms pervade standard datasets and are invisible to the researcher who does not understand the provenance of the data.

Market closing times introduce a related form of asynchronicity. European equity markets close several hours before U.S. markets. Asian markets are closed before New York opens. Without relatively sophisticated data management, a strategy that uses daily closing prices across multiple regions may be implicitly assuming these prices are contemporaneous, when in fact they may be separated by half a day or more. A major U.S. market move in the final hours of the New York session will not be reflected in that day's European or Asian close, creating spurious correlations and false arbitrage signals in the historical record. The retail back-tester working with a simple table of daily closes, one column per market, will see none of this.

Even within a single market, the treatment of missing data introduces silent errors. Most database systems assign a value of zero to a missing data point. A price of zero and an unknown price are not the same thing. The first implies total loss; the second means no data were received. (Most spreadsheet software, incidentally, makes the same conflation.) A system that fails to distinguish between these two states can generate catastrophic false signals: a moving average calculated over a series that includes a spurious zero will produce a wildly distorted output, potentially triggering large trades based on data that never existed. Institutional data operations invest substantial effort in distinguishing between "zero" and "blank" across every field and every timestamp in their databases. The retail trader downloading a CSV file has no such safeguard.

2.5 Survivorship and Selection Bias

Survivorship bias is well-documented in the academic literature on equity backtesting, but its effects extend beyond equities and beyond the commonly discussed problem of universe selection. In futures markets, contracts are periodically delisted or their specifications significantly altered. Commodity markets that no longer trade, or that have been restructured (such as the transition from open-outcry to electronic trading, which transformed market microstructure), are often absent from freely available data sets. A universe of instruments constructed from those currently trading implicitly excludes instruments that failed or were delisted, biasing results toward markets that happened to persist.

However, the more damaging form of survivorship bias operates not at the level of instrument selection but at the level of interpreting backtest output. When a strategy is tested across a large universe of instruments, the temptation to focus on the subset that performed well is almost impossible to resist. A researcher who tests a mean-reversion strategy on fifty futures contracts and finds that it is profitable on twelve of them faces a choice: report the aggregate results across all fifty (which may be mediocre or negative), or present the twelve "validating" instruments as the strategy's target universe, implicitly discarding the thirty-eight that failed to confirm the hypothesis. The latter approach produces a survivorship-biased result that overstates expected performance, even if each individual backtest is technically correct.

This problem is amplified considerably by what might be called the algorithmic generator approach, the trading equivalent of the infinite monkey theorem. Given sufficient computational resources, it is easy to generate thousands or even millions of strategy variants by systematically permuting parameters, indicator combinations, entry and exit rules, and filter conditions. Among a sufficiently large population of random strategies, some will inevitably show impressive backtested performance purely by chance. The probability of finding at least one strategy with apparently excellent performance metrics approaches certainty as the number of variants tested increases, regardless of whether any genuine edge exists in the underlying logic.

The survivorship bias in this context is severe: out of a million generated strategies, the few hundred that survived the performance filter are presented as discoveries, when they are in fact the expected tail outcomes of a large random sample. The strategies that emerge from such processes are frequently characterised by opaque combinations of indicators, filters, conditions, and timing rules that have no coherent theoretical basis, a

phenomenon that could be referred to as "indicator soup."[2] The resulting rules may appear sophisticated, but their complexity is an artefact of overfitting rather than evidence of genuine market insight. Without rigorous correction for the number of strategies tested (a correction that is almost never applied in practice), such results are meaningless. This is worse than it sounds, because the researcher is often not fully aware of the number of implicit comparisons made, as is often the case when iterative manual refinement substitutes for explicit combinatorial search. Even a researcher who does not deliberately generate thousands of variants may, through repeated adjustment and re-testing, effectively sample a large strategy space while believing they have tested only a handful of ideas.

A common belief is that the problems of overfitting and survivorship bias in strategy generation can be adequately addressed by reserving a separate out-of-sample period or by employing walk-forward analysis. While these techniques are valuable and represent an improvement over naive in-sample-only evaluation, they do not eliminate the problem. When a large number of strategy variants are subjected to an out-of-sample filter, the variants that pass may simply be those that happened to fit the noise in the out-of-sample period as well as the in-sample period. The out-of-sample test, in effect, becomes another selection criterion in a multi-stage filtering process, and the strategies that survive all stages are not necessarily those with genuine predictive power but rather those whose particular pattern of overfitting happened to generalise to the specific out-of-sample window chosen. Walk-forward analysis mitigates this to some degree by using multiple out-of-sample windows, but it remains vulnerable when the number of candidate strategies is large relative to the effective degrees of freedom in the data. The reassurance that "it passed out-of-sample testing" is less meaningful than it appears.

A subtler failure mode occurs when the researcher iterates between in-sample and out-of-sample periods. A model is fitted in-sample, tested out-of-sample, found wanting, revised based on what was learned from the out-of-sample failure, re-fitted, and re-tested. Each iteration effectively contaminates the out-of-sample data, converting it into a second in-sample period. The researcher believes the final model was validated on unseen data, but the data were not truly unseen: they were examined and used to inform model revisions.

This iterative contamination is nearly universal in practice and is rarely acknowledged.

The problem runs deeper still. An experienced researcher who has tested many strategies over many years develops familiarity with the major events and regime shifts in the historical record: the 1987 crash, the internet bubble, the 2008 financial crisis, the 2020

[2]The metaphor may be too generous. A soup at least has a recipe. These strategies have the complexity of a recipe with none of the intentionality.

pandemic sell-off. This familiarity means that even a nominally "fresh" out-of-sample test is compromised: the researcher already knows, at least broadly, what happened during the test period and will unconsciously or consciously design models and select parameters that accommodate those known events. Institutional quantitative firms recognise this problem and take countermeasures that range from separating the strategy research function from the strategy selection function to physically withholding portions of the database from researchers, so that the researcher cannot know what data will be used for out-of-sample validation. Some firms go further, randomising which portions of the data are used for fitting versus testing, or having independent teams conduct the validation. The retail back-tester, working alone with a single dataset and full knowledge of the historical record, has no access to any of these institutional safeguards and is therefore maximally exposed to this form of look-ahead bias.

The underlying methodological failure is an inversion of the scientific method. Properly conducted research begins with a theory derived from observation, deduces testable consequences, and then seeks to falsify those consequences to find evidence that the theory is wrong. A theory that survives repeated attempts at falsification gains credibility, but it is never "proved." The retail backtesting workflow inverts this process entirely: the researcher searches for parameters that produce attractive historical performance. That is, the researcher seeks confirmation rather than falsification.

The distinction is critical. Seeking confirmation is trivially easy in a noisy dataset with many degrees of freedom; any sufficiently flexible model can be made to fit historical data. Seeking falsification is hard, and it is the hardness that gives the surviving theories their value. A backtest that begins with the question "does this combination of parameters make money?" rather than "under what conditions would this theory fail, and does the evidence show those conditions?" is not applying the scientific method; it is engaging in a sophisticated form of confirmation bias.

A concrete and timely illustration of this failure mode can be seen in the recent proliferation of gold trading strategies among retail systematic traders. Gold reached a succession of all-time highs in 2024 and 2025, and the result has been a predictable surge of interest in developing gold-specific strategies.[3] The aspiring trader, having watched gold's ascent in real time, decides to build a trend-following or breakout system for the gold futures market. They obtain twenty years of historical data, reserve the most recent two or three years as an out-of-sample holdout, fit their model to the earlier period, and then validate it against the withheld data. The out-of-sample period, which happens to coincide with

[3]The same pattern has played out previously with Bitcoin, natural gas, and crude oil. The instrument changes; the methodological error does not.

one of the strongest gold rallies in history, produces impressive results, and the trader concludes that the strategy has been rigorously validated on unseen data.

But the entire exercise is contaminated by a form of look-ahead bias that no amount of out-of-sample discipline can correct. The decision to build a gold strategy in the first place was made because gold had recently performed spectacularly well. The trader did not, in 2015 or 2018, survey the universe of tradeable futures markets and select gold on theoretical grounds; the trader selected gold in 2025, with full knowledge that it had reached record highs, and then tested a strategy designed to capture trending behaviour against a holdout period that they already knew (perhaps just subconsciously) contained a powerful trend. The out-of-sample test does not test whether the strategy can identify trends it has never seen. It tests whether a trend-following strategy makes money during a period the trader already knows was dominated by a trend. The answer is almost guaranteed to be yes, and it tells the trader almost nothing about the strategy's forward viability.

The self-deception is compounded by a failure of counterfactual honesty. The trader must ask: had they been trading this strategy live over the full historical period, would they have persisted through the years when gold went essentially nowhere? Gold spent much of 2013 through 2019 in a broad range, delivering the kind of choppy, mean-reverting price action that is maximally punishing for trend-following systems. A live trader enduring several years of whipsaws and negligible returns would face enormous psychological pressure to abandon the strategy.

Most would.

The backtest, however, glides serenely through this period because the trader viewing the historical equity curve already knows that the payoff is coming. The backtested return includes the years of frustration as though they were costless to endure; in practice, they are anything but. The strategy that "works" over twenty years of history is, for most human traders, untradeable over the difficult middle years that make the long-term return possible.

This pattern (selecting an instrument after observing its recent success, fitting a strategy to its historical data, "validating" the strategy on a holdout period that the trader already knows was favourable, and then treating the result as evidence of a robust edge) is among the most common and most damaging mistakes made by novice systematic traders, and among the hardest to detect, because every individual step in the process appears methodologically sound. The data were split properly; the out-of-sample period was genuinely withheld; the strategy was not re-fitted after seeing the holdout results. But

the fatal contamination occurred before any of these steps, at the moment the trader chose the market.

No statistical technique can correct for an instrument selection decision that was itself driven by knowledge of the outcome.

One partial countermeasure, available in some backtesting frameworks but rarely employed by retail traders, is logarithmic detrending of the price series before evaluation. Detrending mathematically removes the dominant directional trend from the historical data, leaving only the deviations around that trend for the strategy to exploit. A trend-following strategy tested on detrended gold data cannot profit simply from the secular upward move; it must demonstrate an ability to capture shorter-term directional movements that would exist regardless of the long-term trajectory. If the strategy's performance collapses on detrended data, this is strong evidence that the apparent edge was nothing more than the underlying trend, the hindsight artefact described above. Detrending does not eliminate all forms of look-ahead bias (the instrument selection problem remains), but it removes the most obvious one: the strategy that appears to work only because the chosen market happened to go up. Despite the simplicity and diagnostic power of this technique, it is almost unknown among retail systematic traders, in part because few of the popular backtesting platforms (including legacy platforms such as TradeStation, which has no built-in detrending capability) offer it as a standard feature, and in part because the trader who has already convinced themselves of their strategy's robustness has little motivation to apply a test that might disprove it.

2.6 Continuous Contract Construction

For futures-based strategies, the method of constructing continuous price series from individual contract months is a source of substantial variation in backtest results that is rarely discussed in retail contexts. The back-adjusted (or "Panama") method preserves point-to-point price changes but produces artificial price levels that may become negative for some instruments over long histories. Ratio-adjusted (proportional) methods preserve percentage returns but alter the apparent magnitude of price moves. Unadjusted series preserve actual prices but introduce discontinuities at roll dates.

Each method produces a different equity curve for the same underlying strategy. Same trades, different numbers. More importantly, strategies that depend on absolute price *levels* (such as those using support and resistance concepts) may produce entirely spurious signals on back-adjusted data, since the price levels in the adjusted series never actually existed in the market. The qualification matters: it is the levels that are fictitious, not the point-to-point *changes*. A difference-based rule (a moving-average crossover, a breakout,

a momentum signal computed on changes) reads the same series correctly, because it keys off increments that the adjustment preserves rather than off an absolute level the adjustment has shifted. A level-based rule does not, and the shift is not even stable: re-anchoring the series on the next roll moves every historical level again, so the same fixed-threshold rule can fire on different bars depending on when the continuous series was built. This anchor-dependence is the more useful diagnostic than the blanket claim that adjusted data is "unreal" — difference-based rules survive it, level-based rules do not. The choice of roll date, roll method (calendar-based versus volume-based versus open interest-based), and adjustment methodology collectively represent a set of implicit assumptions that many back-testers never examine.

A related and frequently overlooked issue is the change in contract specifications and effective notional value over time. Many futures contracts have undergone substantial changes in contract size, tick value, or margin requirements during their history. Even where specifications have remained nominally stable, the notional value of a single contract can change dramatically with price level. A backtest that trades "one contract" of the E-mini S&P 500 throughout a twenty-year history is implicitly treating a position that represented a modest notional exposure at early-2000s price levels as equivalent to one representing roughly three to four times that exposure at current levels. The risk characteristics of the strategy are therefore not constant across the sample period, and position sizing rules calibrated to current contract values will produce misleading results when applied retrospectively to periods when the same contract represented a very different risk exposure.

The S&P 500 futures complex provides a particularly instructive example of this distortion. When the E-mini contract (ES) was introduced in September 1997, the S&P 500 index stood at roughly 950. The full-size S&P 500 contract (SP), with its $250 multiplier, had a notional value of roughly $237,500.[4] The E-mini, at one-fifth the size with its $50 multiplier, represented approximately $47,500 of notional exposure. As the index has appreciated over the subsequent decades—recently touching 7,000—the E-mini's notional value has grown to more than $300,000, comfortably exceeding what the full-size contract represented when the E-mini was introduced. The CME officially delisted the full-size SP contract in 2021 because the E-mini had completely supplanted it. CME Group subsequently launched the Micro E-mini (MES) in 2019 at one-tenth the E-mini's size, intended to make the S&P 500 accessible to smaller accounts. Yet even the Micro E-

[4]When the E-mini launched in September 1997, the full-size SP contract actually carried a $500 multiplier, giving it a notional value closer to $475,000 at prevailing index levels. The CME halved the multiplier to $250 just two months later, in November 1997, precisely because the contract had grown too large for many participants. The $237,500 figure cited here reflects the post-adjustment value.

mini, with its $5 multiplier, now carries a notional value of more than $30,000 at current index levels—smaller than the original E-mini's $47,500 at launch, but a far cry from the sub-$10,000 position size that many retail traders assume they are taking on when they trade "the small contract."

For the back-tester, this means that a strategy tested over twenty years of E-mini data is not operating on a consistent instrument: the contract at the start of the sample is, in economic terms, a wholly different proposition from the contract at the end.

The full-size SP contract itself illustrates a further wrinkle. When the E-mini launched in September 1997, the SP contract carried a $500 multiplier, giving it a notional value near $475,000 at the prevailing index level. Just two months later, in November 1997, the CME halved the multiplier to $250 because the contract had simply grown too large for many participants. Any backtest spanning this period must account for the fact that "one contract" of the SP before and after November 1997 represents a fundamentally different economic exposure, since the post-change contract was half the size of its predecessor. Whether a given data source correctly reflects this multiplier change or silently treats the pre- and post-change contracts as identical, is yet another data nuance that may or may not be handled correctly and that the back-tester must verify independently.

This issue is especially dangerous for strategies that use fixed-dollar stop-losses, a practice that is common among retail traders who calibrate their risk in terms of "the most I am prepared to lose on this trade." A trader who backtests with a $3,000 stop over a twenty-year history is implicitly assuming that $3,000 represents the same risk tolerance throughout the sample. It does not, for two compounding reasons. First, inflation alone has substantially eroded the purchasing power of $3,000 over two decades: in real terms, $3,000 today buys considerably less than $3,000 did in 2005. Second, and more importantly for strategy mechanics, the contract's notional value and typical daily range have grown dramatically. A $3,000 stop on an E-mini S&P position twenty years ago, when the contract's notional value was roughly $60,000 and a typical daily range might have been 15 points ($750), afforded the strategy approximately four days' worth of adverse movement before triggering the stop. The same $3,000 stop today, when the notional value exceeds $300,000 and a typical daily range may be 60 points barely provides a single day's cushion. The backtest will show the strategy surviving adverse excursions in the early years that would trigger the stop almost immediately under current conditions. The historical equity curve is therefore constructed not from a single strategy but, effectively, from a series of evolving strategies: the early portion of the equity curve is built from a strategy that has generous room to move and absorb volatility, while the latter portion is effectively scalping with a tight stop. Yet both appear as a single continuous track

record. Stops based on volatility measures such as ATR partially mitigate this problem by adapting to the prevailing range, but even ATR-based stops are distorted over long histories if the underlying relationship between range and notional value has shifted, as it has for any contract whose price level has changed a great deal. Continuous contract adjustments compound the problem: back-adjusted series accumulate roll adjustments that grow larger the further back one looks, progressively distorting the relationship between the adjusted price level and the actual trading ranges that prevailed at the time. An ATR calculated on back-adjusted data from twenty years ago may bear little resemblance to the ranges a trader would have actually experienced. Extreme price events create further complications. When WTI crude oil traded at negative prices in April 2020 (the May 2020 contract settling at $-37.63), several brokers were unable to process negative prices in their systems, and the same is true of backtesting platforms. Any framework that stores prices as unsigned values, computes percentage returns, or uses logarithmic transformations will fail on a negative price. Back-adjusted continuous series that pass through this event can produce negative adjusted prices for earlier contracts even if the unadjusted prices were positive, propagating the problem backward through the entire history. A backtest of any strategy spanning this period that does not specifically handle negative prices is unreliable, and the trader who has not checked whether their platform can handle this case is carrying a risk of which they may not be aware.

2.6.1 The drift is real money, not a measurement artefact

There is a subtler point here that is easy to get backwards, and getting it backwards leads to the wrong fix. The cumulative adjustment that back-adjustment piles onto historical data is often described as a distortion to be cleaned away. For absolute price *levels* that description is correct. For the price *changes*, and therefore for profit and loss, it is not. The total change of an additively back-adjusted series from the start of the history to the end equals exactly the profit and loss of holding one continuously-rolled position over that period. The roll gaps that the adjustment strips out are not noise; they are the actual cashflows of the roll. At each roll the expiring contract is closed at its price and the next is opened at its price, and the difference between the two is money that genuinely changed hands. Additive back-adjustment is, by construction, faithful to that profit and loss in points. This is precisely why it is the default method, and why "the adjusted prices never existed" — true of the levels — does not imply "the adjusted returns are fictitious." They are not.

This matters because the drift is not random. In a market that sits persistently in contango, the roll gap carries a consistent sign, and the cumulative sum of same-signed

gaps imparts a deterministic, low-noise, highly autocorrelated drift to the level series: persistent contango drives the back-adjusted series steadily *downward* as one looks further back, persistent backwardation drives it *upward*. That drift is the roll yield, and it is real return that a position holder earns or pays. The contracts most exposed to driving an adjusted series through zero — low-priced instruments with long histories and a persistent curve sign, such as short-term interest-rate futures and some energy contracts — are exposed precisely because the roll yield is large and one-directional relative to their price, not because of any arithmetic accident.

The practical consequence is a denominator problem that is easy to introduce and hard to see. Any quantity computed as a percentage of the adjusted price — return volatility for position sizing, the entries of a correlation matrix, a log return — is wrong if the adjusted close is used as the denominator, because that denominator is an anchor-dependent fiction that drifts further from the real contract price the further back one looks. The fix is not to abandon additive adjustment but to never divide by the adjusted level: compute each period's return as the change in the adjusted series divided by the *actual* contract price that was trading at the time, not by the back-adjusted close.[5] In practice this means carrying the unadjusted contract price alongside the adjusted series at the resolution the strategy trades on, so the real denominator is always available. A volatility-scaling or correlation routine that silently divides by the adjusted close is carrying a bug that grows with the length of the history, and on a large multi-instrument book it corrupts both the per-instrument position sizes and the cross-instrument risk model at once.

It is also worth mentioning that continuous contract construction in back-testing should be handled in the same way that the trader will trade the strategy in real life. In many cases, strategy developers test strategies on contracts without understanding the specific rules used to construct that construct. Care should be taken to ensure that the contract roll heuristics used in real life are also modelled in the backtest. Furthermore, most backtests don't factor in roll costs; this may be a factor for strategies that regularly hold positions across contract roll dates.

2.7 Equity Price Adjustments: Dividends, Splits, and Consolidations

The continuous contract construction problem described above has a direct analogue in equities, though it manifests differently and is, if anything, less well understood by the retail back-tester. Equity data vendors routinely supply "adjusted" historical prices that

[5]Carver (2023) makes this point repeatedly in the context of his own systematic futures work, and his `pysystemtrade` framework implements it directly: percentage returns are computed against a `daily_denominator_price` that holds the price of the contract actually being traded, never the back-adjusted level. The same logic applies to any volatility estimate or correlation used for risk scaling.

have been retroactively modified to account for corporate actions: stock splits, reverse splits (consolidations), and dividend payments. The adjusted series is intended to produce a continuous return stream, so that a chart or backtest can treat the instrument's history as though these events had not occurred. The intention is reasonable. The execution is a source of widespread and frequently unrecognised distortion.

Consider a stock that has undergone a 2-for-1 split. On the split date, the share price is halved and the number of outstanding shares is doubled. To prevent the split from appearing as a 50% loss in the historical record, the data vendor retroactively halves all pre-split prices. This preserves the percentage return series: a move from \$100 to \$110 pre-split becomes a move from \$50 to \$55 in the adjusted data, and the 10% return is correctly maintained. But the adjusted prices are now fictitious. No one ever traded this stock at \$50 before the split; the actual market price was \$100. Any strategy that depends on absolute price levels, whether through fixed-dollar stop-losses, support and resistance levels, or round-number effects, will generate signals on the adjusted series that bear no relation to the conditions that actually prevailed. This is the same class of error described above for back-adjusted futures, and it is just as damaging.

Reverse splits (consolidations) introduce the mirror-image problem. A company executing a 1-for-10 reverse split is typically doing so because its share price has fallen to very low levels, often to avoid delisting requirements. The adjusted series retroactively multiplies all pre-consolidation prices by ten, transforming what was in reality a low-priced, wide-spread, difficult-to-trade instrument into one that appears to have always traded at a respectable price level. A backtest run on the adjusted data will simulate entries and exits at these inflated historical prices, entirely concealing the fact that the actual trading environment featured penny-stock spreads, thin liquidity, and the particular microstructure pathologies that afflict very low-priced equities. The strategy's apparent historical performance includes a period that was, in practice, untradeable at the costs and fill quality the backtest assumes.

Dividend adjustments are subtler but arguably more consequential, because they are universal rather than occasional. When a stock pays a dividend, the standard adjustment methodology reduces all pre-dividend historical prices by the dividend amount (for point-adjusted series) or by the dividend yield (for proportionally adjusted series). Over a long history, for a stock that has paid regular dividends for decades, the cumulative effect of these adjustments is enormous. Historical prices in the adjusted series can be reduced to a small fraction of their actual traded values. A stock that traded at \$40 twenty years ago may appear in the adjusted series at \$15 or less, once the cumulative effect of two decades of quarterly dividends has been subtracted from the historical record.

This creates several concrete problems for the back-tester. First, as with splits, the adjusted prices are fictional, and any strategy logic that references absolute price levels is operating on numbers that never existed in the market. Second, the adjustment methodology itself varies between vendors. Some vendors adjust for regular dividends but not special dividends. Some adjust for all cash distributions. Some adjust only the closing price; others adjust the full OHLC bar. The same stock, over the same period, can produce noticeably different adjusted price histories depending on the vendor and adjustment method, and consequently different backtest results. The researcher who downloads adjusted data from one source, develops a strategy, and then attempts to validate it against data from another source may find discrepancies that are not errors in either dataset but artefacts of differing adjustment methodologies.

Third, dividend-adjusted data distorts volatility measures in ways that are easy to miss. A stock whose actual price was \$40 with a daily range of \$1 (2.5% of price) may appear in the adjusted series at \$15, but the daily range is also adjusted to approximately \$0.38. In absolute terms, the range has been compressed; in percentage terms, it is preserved. This means that any volatility metric calculated in percentage or logarithmic terms (such as standard deviation of returns) will be correct, but any metric calculated in absolute terms (such as ATR in dollars, or a fixed-point stop-loss) will be distorted by the cumulative adjustment. A strategy that uses a dollar-denominated ATR to set position sizes will systematically oversize positions in the early part of the history, where the adjusted prices are artificially low, and the resulting equity curve will overstate both returns and risk in that period.

Fourth, and most damaging, the adjustment methodology embeds future information into past prices. The closing price for a stock on 1 January 2010, in a dividend-adjusted series downloaded today, has been reduced by the cumulative effect of every dividend declared between January 2010 and the day of the download. None of those dividends had been declared at the time the historical bar was created. A strategy that places a stop at a round number, or that triggers an entry on a breach of a fixed price threshold, is responding to a price level that was itself computed using knowledge of dividends that did not yet exist. This is look-ahead bias hidden inside the data. The contamination is structural, embedded in the historical record by future corporate actions, and it does not show up as an implementation error in the strategy code.[6]

[6]There is a long-running thread on r/algotrading that captures the practical confusion this causes for beginning back-testers: when to use adjusted versus unadjusted data, and what the choice implies for the strategy. The answer depends on whether the strategy reasons in returns or in absolute levels. The fact that the question is asked again every few months suggests the typical retail tutorial fails to address it at all.

The fix is to keep two clean series side by side: an unadjusted series for any logic that depends on absolute levels, and a total-return series for return calculations. Most retail backtesting infrastructure stores only one of the two and forces the researcher to choose. The choice is rarely an informed one.

A further complication arises from the interaction between dividend adjustments and total return calculations. A backtest on price-only (unadjusted) data will understate the returns of a dividend-paying stock, because the dividends are not captured. A backtest on dividend-adjusted data will correctly capture the total return, but only if the researcher understands that the "price" series they are using is not a price series at all; it is a total-return index expressed in price-like units. Mixing adjusted and unadjusted data within the same strategy, or applying logic designed for one to the other, produces results that are internally inconsistent. A cross-sectional strategy that ranks stocks by price, for example, will produce different rankings depending on whether the prices are adjusted or unadjusted, and the adjusted rankings will reflect cumulative dividend history as much as current market valuation, which is almost certainly not the researcher's intention.

The researcher who downloads an adjusted price series and treats it as though it were a record of actual traded prices is making an error that is conceptually identical to the futures trader who treats a back-adjusted continuous contract as a record of actual futures prices. In both cases, the data have been transformed to serve a specific analytical purpose (preserving return continuity), and using them for any other purpose (absolute price comparisons, dollar-denominated risk calculations, cross-instrument rankings) produces artefacts that the backtest will silently incorporate into its results.

3 Execution Simulation Fidelity

3.1 The Fill Assumption Problem

The most fundamental question in any backtest is deceptively simple: would this trade have been filled, and at what price? The vast majority of retail backtesting engines answer this question with naive assumptions that bear little resemblance to real market microstructure.

The most common assumption is that a trade is filled at the closing price of the bar on which the signal is generated, or at the opening price of the subsequent bar. In either case, the implicit model is one of infinite liquidity at a single price point with zero market impact. This is problematic for several reasons. First, the closing price of a daily bar is a single print that may reflect a momentary condition rather than an achievable execution level. Second, the opening price of the next bar (particularly in futures markets where overnight gaps are common) may be materially different from any price at which a resting order could reasonably have been filled. Third, for any strategy trading meaningful size, the act of execution itself moves the market, creating slippage that increases non-linearly with order size relative to available liquidity.

These execution problems are worse again in cryptocurrency markets. Crypto exchanges permit leverage of 50x to 125x, and forced liquidations are visible via public APIs, creating a reflexive dynamic in which liquidations trigger further liquidations. During these cascades, order book depth can collapse entirely, producing price dislocations that no smooth slippage model will capture. A crypto backtest using the same volatility-scaled slippage assumptions that work tolerably well for regulated futures is likely to be dangerously optimistic about fills during the episodes that matter most.

For limit orders, the situation is more problematic still. A backtest that assumes a limit order is filled whenever price touches the limit level is ignoring queue position entirely. In reality, a limit order resting at a popular price level may never be filled even as the market trades through that level, because orders ahead in the queue absorb the available liquidity. This distinction between "price touched" and "order filled" is one of the largest sources of phantom profitability in backtested strategies, particularly for mean-reversion systems that rely on passive entry.

A related and often overlooked phenomenon is adverse selection. Limit orders that do get filled are disproportionately filled in situations where the market is moving aggressively against the order, precisely because it is the aggressive flow that consumes resting liquidity. A limit buy order, for example, is most likely to be filled when selling pressure is intense enough to sweep through the order book to that price level. The fill

itself is therefore actually a negative signal about future price direction. Getting filled is the bad news. Backtests that model limit order fills as occurring at the limit price, without accounting for the conditional probability that a fill implies adverse momentum, systematically overstate the profitability of passive entry strategies. Empirical studies of limit order book dynamics (Cont, Stoikov, and Talreja, 2010) confirm that the expected post-fill price movement conditional on execution is, on average, unfavourable to the limit order placer.

Stop orders are mishandled in a closely related way. When a backtest detects that a stop level was breached inside a bar, it almost always books the fill at the stop price itself. In live trading the stop becomes a market order at the moment of the breach, and it fills at the next available traded price, which in any fast or thin market is some distance through the stop level rather than at it. Stops gap through; backtests do not. Combined with the OHLC sequence ambiguity discussed in Section 4.2, the result is that the simulator is simultaneously over-counting target hits and under-pricing stop hits, and the two errors compound in the same direction.

These naive fill assumptions (fills at the close, fills at the open, fills at the limit price without queue consideration, stops at the stop price) should be avoided at all costs. And they can be, without heroic effort. It is useful to distinguish two tiers of execution realism.

The first tier is achievable with modest engineering effort and represents an enormous improvement over platform defaults: requiring price to trade through a limit level before assuming a fill, applying empirical spread data rather than fixed estimates, incorporating volatility-dependent slippage, modelling delayed fills, and treating partial fills pessimistically. These adjustments are well understood and address the most egregious sources of phantom profitability in retail backtests.

The second tier (queue position modelling from market-by-order or Level 2 depth data, calibrated market impact curves (Almgren & Chriss, 1999), latency simulation, adverse selection modelling conditional on fill, and auction-print execution dynamics) is a genuine research problem. Each component requires its own data and calibration, and each introduces assumptions that are themselves subject to uncertainty. Production desks invest heavily in this infrastructure; for the retail trader, the second tier is aspirational but not essential. What *is* essential is the first tier, and the fact that most retail backtesting engines continue to ship without even these basic corrections is a failure of the tooling, not a reflection of any inherent difficulty in the problem.

3.2 Your Platform May Be Working Against You

In some cases, the unrealistic fill model is not a default that the researcher has failed to override; rather, it is a constraint imposed by the platform itself. In some platforms it is possible to turn on settings to improve the fill model, but it's not always obvious how to do this or that is sufficiently important to make sure that it is considered in the backtest.

Many platforms do not provide a mechanism for placing exit orders in response to an entry fill event. A protective stop or profit target can only be placed after the strategy detects the filled position on the next bar, introducing a mandatory one-bar delay between entry and the activation of any risk management order. On daily bars, this means an entire trading day of unprotected exposure after every entry.[7] In live trading, the trader may have had a stop order linked to the initial entry order, and this is what should be modelled. Rather, we tend to see the common problem of the "tail wagging the dog", whereby instead of traders backtesting strategies that fit with their real-life usage, they model strategies according to the limitations of their chosen platform.

A workaround exists: the trader can place a pre-emptive stop order speculatively on the same bar as the entry, before knowing whether the entry will fill. But this workaround is itself problematic. Even with this approach, it is not possible to place a stop on the same bar as the entry with knowledge of the actual fill price. The stop level cannot be dynamically computed from the fill (for example, a fixed dollar amount or volatility multiple below the actual execution price, adjusted for any slippage that occurred) because the fill has not yet happened when the stop must be specified. The trader must either use a pre-determined price level that may not reflect the actual entry or accept the delay.

This is more than a theoretical nuisance. In live trading, the trader would almost certainly place a protective stop immediately upon receiving a fill confirmation, with the stop level computed from the actual execution price. The platform's inability to simulate this behaviour means the backtest cannot model the way the trader would actually operate in practice. The resulting simulation diverges from live trading not because the researcher made a poor modelling choice, but because the platform makes the correct model architecturally impossible. The backtest shows positions surviving periods of unprotected exposure that the live trader would never have tolerated, producing exactly the kind of systematic positive bias described throughout this paper.

[7] TradeStation deals with this by allowing strategies to stipulate that they should be evaluated every tick (real or guessed) but this introduces logic differences in the strategy code that inadvertently lead to other problems. In practice, this is not something most TradeStation developers seem to consider.

3.3 Transaction Cost Modelling

Commission costs are the most visible component of transaction expenses and, accordingly, are the component most frequently included in backtests. However, commissions are often the smallest portion of true execution costs. The bid-ask spread represents a cost incurred on every round-trip trade, and its magnitude varies considerably with market conditions, time of day, and instrument liquidity. A backtest that uses a fixed spread assumption averaged over the entire sample period systematically underestimates costs during periods of stress, the very periods when many strategies are most active.

Market impact (the price movement caused by the act of trading itself) is almost universally ignored in retail backtests. For small accounts, this may be defensible. For anything larger, it is not. Any strategy intended to scale, or trading instruments with limited depth of book, needs an impact model; without one, capacity estimates are meaningless.

Several additional categories of transaction friction are routinely omitted from retail backtests yet can dominate execution costs for common strategy types. For short equity strategies, borrow availability and borrow fees are material: shares that appear freely shortable in historical data may have been hard-to-borrow or unavailable during the periods the strategy would have needed them, and borrow rates can spike from basis points to annualised double digits during short squeezes or high-demand periods. Forced buy-ins, where the lender recalls shares, can close positions at at the worst possible time. For strategies trading adjusted-price equity series, the handling of dividends and corporate actions (splits, mergers, spin-offs) is a common source of error; the distinction between price-return and total-return series can significantly alter apparent performance, particularly for dividend-rich universes over long sample periods. For high-turnover strategies using limit orders on maker-taker exchanges, the net effect of exchange rebates and fees on profitability can be substantial: a rebate of a few tenths of a cent per share, compounded across thousands of round-trip trades per year, may represent the difference between a profitable strategy and an unprofitable one. Finally, for strategies that employ leveraged products, including contracts for difference (CFDs), perpetual swaps, and margin loans, the financing cost of maintaining positions is a continuous drag that must be modelled explicitly, particularly in regimes of elevated interest rates where overnight funding charges can erode returns that appear attractive on an unfinanced basis.

Crypto derivatives introduce a financing cost with no close analogue in traditional futures. The dominant crypto instrument is the perpetual swap, which has no expiry date and instead uses a periodic funding rate to tether its price to spot. This funding rate is endogenous to market positioning: it can swing violently when leverage is crowded, and during short squeezes it frequently spikes to annualised rates of several hundred percent.

A crypto backtest that ignores funding, or treats it as a flat cost, can misstate strategy returns by margins that dwarf the strategy's apparent edge. Funding in crypto is a variable tax on positioning whose magnitude depends on what everyone else is doing.

A related and widely ignored friction is the behaviour of margin requirements themselves during periods of extreme volatility. Exchanges do not maintain fixed margin rates; they raise them, sometimes sharply and with little warning, when market conditions deteriorate. During the onset of the COVID-19 pandemic in March 2020, for example, CME Group increased initial margin requirements on several major futures contracts by 30% to 50% or more within the space of days. Other exchanges and clearinghouses took similar action. For the trader who was fully deployed at pre-crisis margin levels, these increases had immediate and painful consequences: existing positions that had been within margin limits were suddenly in deficit, requiring either the deposit of additional capital or the forced liquidation of positions at the worst possible time. A backtest run over the same period will show none of this. Standard backtesting engines treat margin requirements as fixed throughout the simulation, if they model them at all. The strategy that the backtest shows holding calmly through the March 2020 sell-off may, in practice, have been forcibly liquidated by a margin call triggered not by a trading loss but by the exchange's decision to raise collateral requirements in the middle of a crisis. The backtest records a position that survived and eventually recovered; the live trader's position was closed at the lows.

This is not a one-off occurrence. Exchanges routinely raise margin during volatility spikes, geopolitical shocks, and periods of unusual market stress. The pattern is predictable in its general form (margins go up when volatility goes up) even if the specific timing and magnitude are not. Any backtest that holds margin requirements constant is implicitly assuming that the trader had unlimited additional capital to meet margin calls during precisely the periods when capital is hardest to raise and most painful to deploy. For leveraged strategies in particular, the gap between backtested and live performance during crisis periods often has more to do with margin mechanics than with the strategy's signals.

A small discipline that catches a surprising fraction of bad strategies is to report gross and net returns side by side at every reporting stage of the development pipeline. The gap between the two is a direct measure of how much of the apparent edge is being eaten by friction. If the gross-to-net ratio is small (the strategy survives realistic costs comfortably) the strategy is genuinely robust to its execution assumptions. If the ratio is large, the backtest is one realistic cost assumption away from being unprofitable, and the rest of the analysis should be conducted with that in mind. Many retail backtests report only gross returns, or only nominally-net returns based on a commission assumption alone.

The full delta from gross to fully-modelled net is usually larger than the trader expects, and the larger it is, the closer the strategy sits to the edge of viability.

3.4 Futures Rolling Costs

In futures markets, the costs of rolling positions at contract expiry deserve particular attention. They are frequently overlooked or underestimated in backtests. The roll involves closing a position in the expiring contract and simultaneously opening an equivalent position in the next contract month. This operation incurs several distinct costs.

First, there are the direct execution costs of two trades: commissions and exchange fees on both legs, plus the bid-ask spread. For strategies that maintain continuous exposure across multiple instruments, these costs are incurred regularly (monthly, quarterly, or at whatever interval the contract cycle dictates) and accumulate over the life of a backtest. Second, the spread between the expiring and next contract (the calendar spread) may be wider than typical bid-ask spreads in either contract individually, particularly in markets with pronounced contango or backwardation structures. The cost of crossing this spread is a real friction that does not appear in a continuous price series.

Third, and more subtly, the roll introduces basis risk. The price relationship between the expiring and next contract is not fixed; it moves during the roll window as supply and demand for each contract shift. A strategy that assumes rolling occurs at a single price ratio (as most continuous contract construction methods implicitly do) is ignoring the execution uncertainty inherent in the roll. In less liquid markets or during periods of stress, the realised roll cost may differ sharply from the theoretical cost implied by the continuous price adjustment.

Fourth, concentrated roll activity around standard dates (such as the days surrounding first notice day or contract expiry) can itself move the calendar spread, creating adverse market impact specifically around the roll event. For strategies with positions across many futures contracts, the aggregate drag from rolling costs over a multi-year backtest can be substantial, yet most retail backtesting frameworks treat the continuous price series as if it were a single instrument with no roll friction whatsoever.

3.5 Exchange Rate Risk

For any trader whose account is denominated in a currency different from the currency in which an instrument is quoted, exchange rate movements represent a constant and often entirely unmodelled source of profit-and-loss variation. Consider an Australian trader holding funds in USD to trade US futures, a sterling-based trader accessing yen-

denominated contracts, or a euro-based portfolio with exposure to any non-European market. By way of example, over the last 25 years, the AUD-USD exchange rate has fluctuated between 0.50 and 1.10, which is clearly a significant source of variation in the strategy's returns.

The most direct effect is on trade-level profit and loss. A futures trade that generates a profit of one thousand U.S. dollars has a different value to the Australian-dollar trader depending on the AUD/USD exchange rate at the time the profit is realised. If the Australian dollar has strengthened against the U.S. dollar between entry and exit, the profit converted back to the trader's home currency is smaller than the nominal gain in dollar terms. If the Australian dollar has weakened, the converted profit is larger. Over a long backtest spanning periods of significant exchange rate movement, the cumulative effect of ignoring this conversion can alter both the magnitude and the trajectory of the equity curve. A strategy that appears to deliver steady returns in the instrument's native currency may, when properly converted to the trader's home currency, exhibit markedly different return and drawdown characteristics. The error is directional. Exchange rate trends can persist for years, meaning that the currency effect can systematically inflate or deflate apparent performance across extended portions of the backtest.

The second manifestation involves margin requirements and the cash held to support them. Futures positions require margin deposits, and for instruments traded on foreign exchanges or denominated in foreign currencies, these deposits are typically held in the instrument's native currency. The value of this margin collateral, expressed in the trader's home currency, fluctuates with the exchange rate. A backtest that tracks available capital and margin utilisation in purely nominal terms (ignoring the currency exposure inherent in foreign-denominated margin balances) will misstate the trader's true buying power and risk of margin shortfall. During periods of adverse exchange rate movement, the effective margin buffer can shrink even if no trading losses have occurred, and this erosion is invisible to a backtest that does not model the currency overlay.

Third, and perhaps most subtly, cash balances held in a foreign currency earn interest at that currency's prevailing rate, not the trader's domestic rate. For strategies that maintain substantial uninvested cash, either as margin collateral or as a risk management buffer, the interest differential between the domestic and foreign currency can meaningfully affect long-term returns. In periods where there is a significant interest rate differential between currencies, the carry effect on idle cash compounds over time and can represent a material drag or tailwind that a currency-naive backtest will not capture. The effect is particularly pronounced for strategies that allocate only a small fraction of capital to active positions and hold the remainder as cash, which describes a large

proportion of volatility-targeted and risk-parity approaches.

The aggregate effect of these three mechanisms (trade P&L conversion, margin collateral revaluation, and interest rate differentials on foreign cash) is that any backtest of a cross-currency strategy that does not incorporate an exchange rate model is reporting results in a currency that the trader does not actually hold. Every number is wrong: the equity curve, the drawdown statistics, the Sharpe ratio, all of it. For strategies that trade exclusively in the trader's home currency, this issue does not arise. But for the many retail traders who access global futures markets from a non-US-dollar-denominated account (and this includes a large proportion of traders based in Australia, Europe, the United Kingdom, and Asia) the omission of exchange rate effects is a structural error that pervades every calculation the backtest produces, not a minor refinement to be addressed later. Matters are worse still because some widely used platforms provide no mechanism for currency conversion at all: all profits and losses are reported in the instrument's native currency, with no facility for translating results into the trader's actual account currency. On such platforms, the exchange rate failures described above are not optional omissions by the researcher. They are structural impossibilities imposed by the tool.[8]

A subtler error is to model the currency overlay as a single end-of-sample conversion: take the final P&L in the instrument's native currency, multiply by the exchange rate on the day the backtest ends, and report the result. This is the cleanest implementation choice, and the implicit one in many tutorials. It is also wrong. Every trade settles in the instrument's currency at the time it is taken, and the value of that cash flow in the trader's home currency depends on the rate prevailing at that moment. Every open position carries an embedded FX exposure that varies with the home-currency mark-to-market of the instrument. Every cash balance accrues interest at its native currency's rate. Collapsing all of this into a single rate at the end of the sample destroys the path dependency of the FX overlay, and the resulting equity curve can differ from the dynamic version by more than the strategy's apparent alpha over multi-year periods of significant currency drift.

The mechanical fix is simple in principle. Run an FX series alongside the price series, convert each trade-level P&L, margin balance, and interest accrual at the prevailing rate, and report performance in the trader's actual currency throughout. The fix is rarely implemented in practice because the average retail platform makes it inconvenient, and because the trader has often never been prompted to think about it.

[8]Since a lot of trading tools have historically come from US-based companies, and platforms such as TradeStation naturally favour US-based markets, many US-based traders have never had to think about these issues!

3.6 Counterparty Risk

A backtest treats the broker, the exchange, and the clearinghouse as silent infrastructure. They settle in the background, they remain solvent, and they keep their part of the bargain. Live trading does not always cooperate.

In regulated futures and equities, client assets are typically segregated from broker working capital, and clearinghouse failure is a remote contingency. Remote, but not impossible. MF Global collapsed in October 2011 with a roughly $1.6 billion shortfall in segregated customer accounts. Refco failed in 2005, two months after its IPO, and customer assets that should have been ringfenced turned out to be entangled with the parent's collapse. The Lehman Brothers bankruptcy in September 2008 left the firm's prime brokerage clients (including a large number of hedge funds) unable to access positions and balances for an extended period. These events are rare. They have happened before. A backtest that quietly assumes the broker is always there is making a probabilistic assumption it has never stated explicitly.

Crypto markets dispense with the rare-tail-event framing altogether. The exchange, broker, and custodian are typically the same entity, client assets may be commingled with the firm's own balance sheet, and rehypothecation of customer collateral has been documented at multiple major venues. FTX, Mt. Gox, QuadrigaCX, Celsius, Voyager, BlockFi: the list of crypto venues that have either failed outright or frozen withdrawals during the past decade is too long to credibly characterise as a series of one-off accidents. Any crypto backtest spanning more than a few years implicitly assumes that the chosen exchange survived, that assets were never frozen, and that withdrawals were possible at the times the backtest needed them. Those assumptions are demonstrably false for a substantial fraction of the venues that have ever existed.

Stablecoin exposure sits underneath all of this as an additional credit overlay. Strategies denominated in USDT or USDC implicitly assume the peg holds, yet USDT traded as low as 0.88 in 2018, USDC depegged during the Silicon Valley Bank episode in 2023, and several smaller stablecoins (UST being the most notorious) have collapsed entirely. The peg is a credit-and-reserve assumption dressed up as a currency. A backtest that quotes returns "in USDT" is quietly assuming the issuer's reserves are good throughout the sample period.

The honest treatment of counterparty risk is uncomfortable. No clean simulation captures the discontinuity of an exchange suddenly closing its doors. A trader can apply a probability-weighted haircut to long-horizon returns to reflect the historical base rate of venue failure. They can restrict the strategy universe to venues with adequate safeguards

(segregated accounts, audited reserves, genuine regulatory oversight). They can also flag the residual exposure as an explicit caveat on the reported performance, rather than pretending it isn't there. Most retail crypto backtests do none of these. They report returns as though every venue in the sample is permanent and solvent. Both assumptions have failed within the lifetime of most active traders.

3.7 Margin Fluctuations

Return on capital is one of the most consequential measures of a trading strategy, and the capital required to deploy a position in any leveraged instrument is set by the broker's margin requirements. Those requirements move. They move in two different ways at once, and both effects compound.

The first is the steady drift in the notional value of a single contract as the price of the underlying changes. The E-mini S&P 500's percentage margin may sit at a stable fraction of notional for years, but the dollar amount required to hold one contract has grown by roughly 5x over the past two decades because the index itself has appreciated by that much. A strategy that backtests with a constant $5,000 margin assumption is silently sliding between very different leverage regimes across the sample period.

The second is the sharp, discretionary increase in percentage margin imposed by exchanges and clearinghouses during periods of stress. CME, ICE, LME, and Eurex all reserve the right to raise margins on short notice, and they exercise that right. In the GFC of 2008, in March 2020 at the onset of COVID, in February 2022 after the Russian invasion of Ukraine, and at numerous smaller flashpoints between, initial margin on flagship contracts has doubled or tripled within days. The trader who was fully deployed at pre-crisis margin levels suddenly found themselves in deficit having taken no trading loss whatsoever. The options were two: deposit additional capital on a timetable dictated by the broker, or liquidate at exactly the wrong moment.

Multiply the two effects together. If the contract's notional value doubles over the sample period and the exchange doubles the percentage margin during a crisis, the dollar capital required to hold one contract at that moment is four times what the naive backtest assumed. Position sizing rules calibrated against the average historical margin will be wrong by a factor approaching that during precisely the periods when being wrong is most painful.

Reconstructing the historical margin schedule is also surprisingly difficult. Exchanges publish current margin requirements but seldom maintain a clean public archive of historical changes. Data vendors typically do not carry this series at all. Trying to back

out the actual schedule from press releases and SPAN files is painful, and the result is incomplete even after substantial effort. Most retail backtesters never attempt it.

The practical choice is between three uncomfortable options. A single fixed margin assumption for the entire backtest is tractable but obviously wrong, and most commonly overstates available leverage during the periods that matter most. Modelling margin as a fixed percentage of contemporaneous notional captures the slow drift but misses the discontinuous jumps that cause forced liquidations. Reconstructing the actual schedule produces a defensible result at the cost of substantial work. None of these is ideal.

The LME nickel episode in March 2022 is worth flagging as a worst case. The London Metal Exchange suspended trading and retroactively cancelled trades after a short squeeze drove the price from around $30,000 to over $100,000 per tonne in a matter of hours. Positions on either side of those cancelled trades were extinguished by exchange fiat. No backtest captures this kind of event, because no backtest contains the concept of trades being unwound by the venue after the fact. The trader who held the surviving leg of one of those positions discovered that an exchange's unilateral right to act on a contract is a risk dimension no margin model contains.

Margin assumptions matter. They can convert a strategy that the backtest shows surviving a crisis into one that, in live trading, would have been forcibly liquidated at the lows. The backtest records a position that recovered; the live trader's position was closed at zero. These are different equity curves describing nominally the same strategy.

3.8 Limit Halts and Continuous Trading Assumptions

Most retail backtests assume the market was continuously tradable for the duration of the sample period. They assume this implicitly, by simulating fills at the printed close price or the next open without checking whether the venue was actually accepting orders at that moment. The assumption is wrong more often than the backtest reveals.

Futures exchanges impose daily price limits on a long list of contracts. Soybeans, corn, wheat, lean hogs, live cattle, lumber, natural gas, several of the interest-rate complexes, and most energy contracts have published limit moves that, when reached, either halt trading entirely or expand to a wider band. When the market hits limit, no trades print at prices beyond the band, but the underlying value of the position has clearly moved. A backtest that uses the settlement price on a limit-locked day implicitly assumes the trader could have exited at that price. They could not. They could only have exited if the market reopened within the band, which sometimes happens within the same session and sometimes takes days.

The ICE Brent and NYMEX WTI futures hit expanded daily limits multiple times during the spring of 2022 as the Russian invasion of Ukraine disrupted energy markets. The wheat complex hit limit-up several times in the same period for the same reason. A backtest of a trend-following or breakout strategy run across that window, assuming daily settlement fills, is silently assuming a trader could have exited at the printed settle on every one of those days, when in practice the position would have been carried to the next session with no opportunity to close.

Equity markets have analogous mechanisms. The SPX market-wide circuit breakers halt trading at -7%, -13%, and -20% intraday moves. The first two trigger 15-minute halts; the third closes the market for the day. Single-stock LULD (Limit Up Limit Down) bands halt individual securities for five minutes when prices move outside their volatility-derived bands. Stock-specific trading halts happen routinely for pending news, regulatory action, or extreme volatility. Each of these events represents a period in which the backtest's assumed liquidity simply did not exist.

The LME nickel halt in March 2022 mentioned earlier is a different and more aggressive case: the venue halted trading and then cancelled completed trades retroactively. Most exchanges will not do this. The point is that they can, and that no backtest's model of fills survives this kind of intervention.

Crypto exchanges are nominally always open, but the same effect plays out through liquidity collapse rather than formal halt. During the cascading liquidations of May 2021, the chain of failures in mid-2022, and the November 2022 FTX collapse, order book depth on major venues evaporated to a fraction of normal levels for periods of hours. The backtest that prices fills at the printed last trade on those days is assuming the trader was the only person who wanted to transact at those prices. They were not, and the spread the trader actually faced was orders of magnitude worse than the printed mid.

The practical guidance is to flag and special-case any bar where price moved by more than the contract's published limit, or where the session was halted. Vendors typically do not annotate these events in their data, so the trader has to maintain their own list of known limit-locked days and halt sessions. Strategies that assume the trader could have exited at any printed price are systematically overstating achievable performance, and the overstatement is concentrated in exactly the events the strategy is meant to profit from.

3.9 Tax and Holding-Period Effects

Backtests universally report pre-tax returns. Live results are post-tax. The gap between the two is large, and the size of the gap depends on choices the strategy designer rarely thinks about during development.

Most jurisdictions tax short-term and long-term gains at different rates, and the boundary between the two is defined by the holding period. In the United States, short-term gains on positions held less than a year are taxed as ordinary income, at marginal rates that can exceed 40% once federal and state taxes are combined; long-term gains receive preferential rates of 15% or 20% for most filers. In the United Kingdom, capital gains tax applies a separate schedule from income tax, with allowances and rates that depend on the taxpayer's overall bracket. In Australia, capital gains on positions held for more than 12 months receive a 50% discount, effectively halving the marginal tax rate on long-term gains. Other jurisdictions have their own variants, but the pattern is universal: short holding periods are penalised, long holding periods are rewarded.

The implication for backtesting is direct. A strategy with an average holding period of three days, generating a pre-tax return of 12% per year, may yield around 7% after tax for a US investor in a high marginal bracket. A buy-and-hold strategy with the same pre-tax return yields close to 10% after tax. The two strategies look identical on the backtest equity curve. They are not, in any economically meaningful sense, identical. The ranking between strategies of different turnover can flip entirely once the tax overlay is applied.

Wash sale rules add a further layer of complication. In the United States, losses realised on a security cannot be deducted if a substantially identical position is re-established within 30 days. For high-turnover strategies that frequently rotate in and out of the same instruments, the wash sale rule can disallow a meaningful fraction of realised losses, distorting the after-tax return relative to a naive calculation.[9] Other jurisdictions have their own anti-avoidance rules that achieve similar effects.

The honest approach is to report pre-tax and post-tax returns side by side, with the trader's actual marginal rates plugged in, and to compare strategies on the after-tax basis when ranking. Tax-loss harvesting, the timing of realisations across tax years, and account-type effects (retirement accounts versus taxable accounts) all change the picture in ways the backtest never sees. None of this appears in standard backtest output. The trader who selects a high-turnover strategy on the basis of pre-tax returns may be selecting

[9]The wash sale rule applies to securities but not to futures or section 1256 contracts, which receive their own preferential treatment in the US. Strategy designers should know which regime their instruments fall under, and most do not.

a strategy that, after tax, underperforms a passive benchmark.

4 The Bar Resolution Problem

4.1 Why Every Bar Conceals Information

Every OHLC bar, regardless of its timeframe, is a lossy compression of the true price path. A bar records the open, the high, the low, and the close, but nothing about the trajectory between them. Any order that could have been triggered at some point within the bar (a stop-loss, a limit entry, a profit target) must be evaluated against a price path that the bar does not fully describe. The backtesting engine must therefore make assumptions about what happened inside the bar, and those assumptions may or may not correspond to reality.

This problem exists at every level of aggregation, but its severity is a direct function of bar size. At fine granularity (one-minute or five-minute bars) the price range within each bar is typically small, the number of plausible intra-bar paths is tightly constrained, and the assumptions required to evaluate orders introduce only modest error. For the vast majority of retail systematic trading, backtesting with bars of this granularity provides results that sufficiently approximate reality. The exception is high-frequency scalping, where even sub-minute price dynamics and order book position matter, but this is generally outside the domain of the retail trader.

As bar size increases, so does the divergence between what the backtest assumes and what would have occurred in live trading. An hourly bar conceals more than twelve five-minute bars; a daily bar conceals far more than all its hourly bars. The high-to-low range of a daily bar in a liquid futures contract can easily span two or three percent, a range within which stops and profit targets may both have been triggered, but the bar provides no information about the sequence in which prices were visited. Over thousands of trades across a multi-year backtest, these guesses compound into a systematic bias. The intuition that these errors should cancel out (that random guesses about intra-bar paths would overstate some trades and understate others in roughly equal measure) is understandable but wrong. The net effect is an optimistic overstatement of performance. The mechanism is survivorship within the backtest itself: when the engine guesses that a stop was not triggered (because it cannot see the intraday breach), the position survives and participates in subsequent price recovery. The symmetric error (guessing that a profitable target was not reached) is less consequential, because the position simply remains open and is eventually closed by another signal. In net, errors on the loss side tend to be eliminated from the record while errors on the profit side tend to be deferred rather than eliminated, producing a directional bias that systematically flatters the equity curve. This error is most visible on systems that can have a profitable exit or stop loss triggered

on the same bar.

This failure is most acute, and most widespread, at the daily bar level, because a large number of retail traders use end-of-day (EOD) daily bars data as the foundation for their backtesting. Many do so in the belief that, because they are developing longer-term strategies with weekly or monthly signals, the intraday price path is irrelevant.

This belief is fundamentally incorrect.

A strategy's signal generation timeframe and its order evaluation timeframe are separate concerns. A trader may generate signals from daily or weekly bars, but the orders those signals produce (stops, limits, targets) interact with the market continuously, not at daily intervals. Any strategy that places a stop-loss is implicitly asking the market a question about the intraday path: "did price reach this level at any point during the session?" End-of-day data cannot answer that question reliably.

Consider a long-term trend-following system that uses a trailing stop. The daily close may show a price comfortably above the stop level, but the intraday low may have breached it. If the strategy is implemented live with a resting stop order, the stop will be triggered by the intraday breach; if the backtest only sees the daily close, it will show the position as still open. Over a multi-year test, this discrepancy accumulates. The backtest will show a strategy that holds through intraday volatility and captures the subsequent recovery, while the live strategy will have been stopped out. The result is a systematic positive bias in the backtested equity curve relative to achievable performance. The specific failure mode this creates (the OHLC sequence ambiguity, in which multiple orders could all have been triggered within a single bar but in an unknowable order) is examined in detail in Section 4.2.

The stop-concealment problem is compounded on some platforms by the complete absence of out-of-session order evaluation. In TradeStation, for example, a stop order placed at the session close is simply not evaluated until the next session opens. For futures contracts that trade nearly twenty-three hours per day (the remaining hour being a maintenance window that most traders forget exists until it catches them), this means that overnight price movement through a stop level is invisible to the backtesting engine: the stop is never triggered, the position survives, and the backtest records a different outcome than would have been produced in live trading.

This limitation would not arise if the backtest were configured to use the full trading session, the entire period during which the exchange accepts orders. But many traders do not want to execute entries outside of regular trading hours (RTH), preferring to restrict their algorithms to a subset of the full trading day, such as the equity regular session

from 9:30am to 4:00pm Eastern time. This is a legitimate design choice: RTH sessions typically offer deeper liquidity and more representative price discovery. Restricting entries to the RTH session while still evaluating protective exits across the full session is, in fact, the way most traders would operate in live markets. But implementing this distinction (entries only during RTH, exits active at all times) requires the platform to treat entry and exit orders differently with respect to session boundaries, and platforms that lack this capability force the trader to choose between two unsatisfactory alternatives: use the full session for everything (accepting entries at potentially illiquid overnight prices) or restrict the entire strategy to RTH (leaving stops unevaluated outside those hours).

The trader who chooses the latter, restricting the session to RTH, eliminates the need for complex time-of-day management in the strategy code. Without this restriction, the code must determine whether the current bar falls within the desired trading window, whether the next bar will be the last bar before the window closes, and so on. This is already non-trivial for a single instrument, but it becomes especially difficult when trading symbols across multiple time zones, where the RTH windows in the trader's local time shift with daylight saving transitions in both the trader's location and the exchange's location.

Platforms like TradeStation make this difficulty worse still by requiring all time references in strategy code to be expressed in the user's local computer time, regardless of where the traded instrument's exchange is located. This means the researcher must manually convert every session boundary and time-based condition from the exchange's native time zone to their own, and must track the daylight saving schedules of both zones to ensure the conversion remains correct throughout the year. The variations are not negligible: depending on whether the local user's time zone and the exchange's time zone are independently entering or leaving daylight saving time, the apparent start time of a session can shift by one hour or, in some cases, two hours relative to the user's clock. This can be the difference between evaluating a bar at 9:00am and evaluating one at 11:00am, a distinction with material consequences for any strategy that references time of day. The issue can be worked around with careful coding, but it represents an extra source of logic errors and needless overhead, one that is entirely a platform artefact rather than an inherent difficulty of the problem. In practice, most traders either give up on strategies that require this kind of cross-timezone session management, or proceed without realising that the underlying time-conversion issues exist at all, silently corrupting their backtest results in ways that may never be detected.

4.2 The OHLC Sequence Ambiguity

The information loss described above produces a specific and well-defined failure mode when a strategy generates multiple price-contingent orders that could both trigger within a single bar. A bar compresses an entire continuous price path into four numbers: open, high, low, close. The true tick-by-tick sequence between the open and the close is destroyed, but during the bar price visited *some* ordered path that touched both the high and the low. At least two paths are consistent with any given OHLC quadruple:

- **Path A:** open → high → low → close.
- **Path B:** open → low → high → close.

These are not equivalent for a strategy that has both a profit target and a stop loss live inside the bar. On Path A the target fills first; on Path B the stop fills first. The realised profit and loss from those two paths can have opposite signs, yet the bar looks identical in the data file.

Consider a concrete example: a system enters long at the open at 100, with a stop at 98 and a target at 104. The day's OHLC is 100/105/97/103. Both the stop and the target were reached during the session. The system's result depends entirely on which was hit first. If the market rallied to 105 before declining to 97, the target was achieved and the trade is a winner. If the market dropped to 97 first, the stop was triggered and the trade is a loser. The daily bar does not contain this information.

4.2.1 Why engines have to guess, and why they guess optimistically

Without tick or sub-bar data, a backtester cannot know in which order the high and the low occurred. It has to adopt a convention. Four conventions are common, and each is unsatisfactory in its own way:

1. **Assume high-before-low on up bars (close ≥ open) and low-before-high on down bars.** Superficially reasonable but completely unjustified at intra-bar resolution. Plenty of up bars spike down first; plenty of down bars spike up first. The convention encodes a folk belief about intra-bar dynamics that the data themselves cannot support.
2. **Assume the favourable level is hit first** (target before stop on a long trade, stop before target where the stop is the upside). Wishful thinking, and the one that inflates equity curves most dramatically.
3. **Assume the unfavourable level is hit first.** Conservative, and the only safe default. It under-reports profitability rather than over-reporting it, which is the right direction for the error to point.

4. **Fill both orders on the same bar in entry order, or net them out.** Incoherent: both orders cannot have executed at their respective trigger prices on the same realised path unless the path order is known.

Most retail engines end up at something close to convention 1 or convention 2 in their default configuration. Backtrader, for example, processes intra-bar orders in queue submission order, with the consequence that an entry, target, and stop submitted together cannot all be honoured correctly within the entry bar; the limitation is discussed further in Section 7. The practical effect of the default behaviour is the same regardless of how it is rationalised: when a bar's range straddles both the target and the stop, the engine tends to book the winner.

A small number of better-designed engines surface the choice as an explicit setting on the strategy, with four typical values that correspond directly to the four conventions above. A "best-guess" mode uses the close-versus-open heuristic of convention 1: if the bar closed above its open the low is assumed to have come first, and vice versa. A "stop-first" mode hard-wires convention 3: whenever a bar's range contains both levels, the stop is assumed to have been hit, regardless of where the bar closed. A "target-first" mode hard-wires convention 2 in the opposite direction, assuming the favourable level was hit first. A "neither" mode declines to resolve the ambiguity at all: if the sequence cannot be inferred without guessing, the bar simply produces no exit, and the position carries forward to be resolved by the next bar or by some other exit rule. Exposing the choice as a strategy parameter is a marked improvement over silently encoding it as a default, because it forces the researcher to think about which assumption is being made and to live with the consequences. The "stop-first" and "neither" modes are the only defensible choices; the "best-guess" mode is a folk heuristic dressed up as a default; the "target-first" mode is an active misrepresentation of the strategy's performance and should never be used.

4.2.2 Why this overstates profitability rather than just adding noise

If the convention were unbiased – equivalent to flipping a fair coin per bar to decide which level was hit first – the resulting error would be noise around the true expectancy and would shrink with sample size. The trap is that the bias is systematically in the strategy's favour, for three independent and compounding reasons.

The first is **conditional sampling**. The path assumption only matters on bars whose range contains both the target and the stop. On every other bar the outcome is unambiguous and the convention is irrelevant. The engine's guess is therefore not being averaged over all bars but over the subset of bars where reality is itself a coin flip be-

tween $+kR$ and $-R$, where R is the risk per unit and k is the reward-to-risk ratio. If the engine resolves all of those ambiguous bars as wins, the error per ambiguous bar is not "$+kR$ half the time, $-R$ half the time, mean zero". It is "$+kR$ every time, when truth is $+kR$ half the time and $-R$ half the time", and the expected error per ambiguous bar is:

$$\mathbb{E}[\text{error}] = \tfrac{1}{2}(+kR - (+kR)) + \tfrac{1}{2}(+kR - (-R)) = \tfrac{1}{2}(k+1)R.$$

That is strictly one-signed (positive) and scales with the strategy's reward-to-risk ratio. It does not cancel across bars because every draw is biased in the same direction. Running more bars tightens the estimate of a positive number, not of zero. Worse, the cleaner the strategy looks, the more acute the problem: a tight stop and a tight target relative to typical bar range increase the fraction of bars that are ambiguous, and therefore the fraction of trades that are subject to the bias. Scalping systems on hourly bars and mean-reversion systems on daily bars are the worst offenders, because the daily range routinely engulfs both bracket levels.

The second reason is **asymmetric, discontinuous payoffs**. Even if the engine's path guess were genuinely unbiased – a fair coin per ambiguous bar – the payoffs themselves are not symmetric around the entry. A bracket trade pays $+kR$ on a target hit and $-R$ on a stop hit, where k is generally not 1 and the distribution of which level is closer to entry is itself skewed by how the strategy was designed. Strategies are almost never built with target and stop equidistant from entry in tick terms; they are built around an asymmetric reward-to-risk ratio that the trader believes carries positive expectancy. The engine's guess interacts with that asymmetry, and the interaction has a non-zero mean unless the path distribution is exactly the inverse of the payoff asymmetry, which would require knowing the true intra-bar process. Even an unbiased coin produces biased profit and loss, because the expectation of a product is not the product of expectations when the terms are correlated with the strategy's structure.

The third reason is **survivorship through the optimisation loop**, and it is the one that turns a small per-bar bias into a catastrophic equity-curve illusion. A strategy is not backtested once. It is backtested across hundreds of parameter combinations, of which the best is kept. The path-assumption bias is largest for parameter sets with tight stops and tight targets relative to bar range – exactly the parameter sets that look most attractive on the inflated metric. The optimiser therefore preferentially selects strategies whose apparent edge is the most contaminated. The bias is not just non-zero on average; it is concentrated in the region of parameter space the search is going to pick. This is a close cousin of the Bailey and López de Prado backtest-overfitting argument discussed

in Section 5, but with a sharper edge: it is not only that the search is data-mining noise, it is that the search is data-mining a *systematically positive* artefact. Even an honest, non-overfit search procedure will be pulled toward the contaminated region because that region has the highest reported Sharpe ratio by construction.

The symmetry intuition that "the errors should cancel" implicitly assumes the error is additive Gaussian noise on a fixed underlying signal: engine profit and loss equals true profit and loss plus an independent zero-mean disturbance. Under that model the disturbance averages out and its standard error shrinks like $1/\sqrt{N}$. The actual error model is multiplicative and conditional: the error is non-zero only on ambiguous bars, the indicator for "bar is ambiguous" is not independent of the strategy's parameters, the bracketed payoff difference has strictly non-negative mean under any optimistic convention, and the whole expression is then maximised over a parameter grid. None of those three properties is present in the additive-noise mental model that makes "it averages out" feel obvious. Stated formally, the path assumption is a biased estimator of a path-dependent functional, and bias does not vanish with sample size. Variance vanishes; bias persists. A long backtest of a path-sensitive strategy on OHLC data converges, with high confidence, to the wrong answer.

4.2.3 What it looks like in equity

A strategy that is genuinely break-even after costs can show a smooth, monotonic equity curve in a naive event-driven backtest with intra-bar exits enabled. Sharpe ratios in the 2 to 4 range on daily bars for a mean-reversion system with tight stops are almost always this artefact. When the same logic is re-run against tick data, or against minute data honestly resampled, the curve typically collapses to a random walk or worse, because the ambiguous bars redistribute roughly 50/50 and the cost structure (commissions, the real fill price on stops) accounts for the rest. The diagnostic in Section 4.5, which compares a daily-bar backtest of a strategy to the same strategy run with orders evaluated against the underlying minute data, is the direct empirical manifestation of this bias.

4.2.4 Honest mitigations

The trader who acknowledges the ambiguity has several options, ordered roughly from cheapest to most rigorous:

- **Pessimistic resolution.** When a bar contains both bracket levels, always assume the stop was hit first. This is the "stop-first" mode described above. It produces a lower bound on strategy profit and loss and is the only convention that does not lie to the researcher. Where the engine also offers a "neither" mode that declines

to resolve the ambiguity at all, that is more conservative still: it produces no exit from the ambiguous bar and defers resolution to a subsequent bar or exit rule, at the cost of holding the position through an interval the strategy may have intended to close.

- **Sub-bar data.** Drop to a granularity at which the ambiguity is rare. If the target and the stop are 50 basis points apart, the backtest needs bars whose typical range is much smaller than 50 basis points, not larger.
- **Tick or quote replay on ambiguous bars.** A hybrid approach that keeps the bulk of the backtest cheap by running at bar resolution, while resolving the ambiguous bars honestly against the recorded intra-bar path.
- **Explicit bracket-order semantics.** Model the bracket as a single atomic construct rather than as two independent orders that the engine arbitrates between, and acknowledge that on the entry bar itself most retail engines cannot honour an intra-bar exit at all (the Backtrader IBOG limitation discussed in Section 7).
- **Report the spread.** Run the backtest twice, once with the optimistic convention and once with the pessimistic one. If the two equity curves diverge meaningfully, the strategy's edge is inside the simulator's measurement error, and no edge has actually been demonstrated.

The deeper point is that this is not a bug in any particular backtesting engine. It is an information-theoretic limit. OHLC is a lossy compression of the price process, and any strategy whose payoff is sensitive to intra-bar ordering is computing a function the input data cannot support. The engine is forced to fabricate the missing information, and the fabrication is not neutral. For the trader constrained to coarse bar data, the operational conclusion is straightforward: either move to intra-bar order evaluation along the lines discussed in Section 4.5, or design strategies whose order logic does not generate multiple price-contingent orders that could both trigger within a single bar.

4.3 Settlement Price versus Last Trade

In futures markets, a further subtlety arises from the distinction between the settlement price and the last traded price. The daily settlement price is determined by the exchange using a methodology that may incorporate trades, bids, offers, and spread relationships during a defined settlement window. It is not simply the last trade of the day. In illiquid contracts or far-dated maturities, the settlement price may differ significantly from any price at which a trade actually occurred. A backtest that treats the settlement price as an achievable execution level is implicitly assuming liquidity that may not have existed.

This distinction is particularly important for calendar spread strategies and relative-value

approaches, where the settlement prices of individual legs are used to calculate spread values. Because settlement prices are determined by a formulaic process, the implied spread may never have been directly tradable at the reported level.

An analogous problem exists in equity markets, where the "close" price in a daily bar is typically the result of a closing auction, a discrete price-setting mechanism with different fill dynamics than continuous trading. Closing auctions concentrate substantial volume into a narrow window, and participation requires submitting orders that compete for priority in the auction's matching algorithm. A backtest that assumes the strategy can execute at the official closing price is implicitly assuming the strategy has priority in an auction that may be dominated by index funds, ETF creation/redemption flows, and institutional rebalancing activity. For strategies that generate signals from the daily close and assume execution at or near that price, the gap between the auction print and the price achievable by a retail participant submitting a market order in the final minutes of continuous trading can be material, particularly on days with large index reconstitution or options expiry flows. This is one of the most common institutional-versus-retail backtest discrepancies, and it is almost never modelled in retail backtesting engines.

4.4 Volatility and Risk Estimation

Position sizing in many systematic strategies is based on volatility estimates, commonly derived from daily close-to-close returns. But close-to-close volatility understates true intraday risk exposure. A market that opens sharply against a position, trades at extreme levels during the session, and then recovers to close near the prior day's level will appear benign in a close-to-close volatility calculation. The position holder, however, experienced substantial mark-to-market drawdown and, if leveraged, may have faced margin calls that forced liquidation at the worst possible time. Strategies sized on close-to-close volatility will systematically underestimate risk and overstate risk-adjusted returns.

Parkinson (1980) and Garman and Klass (1980) proposed range-based volatility estimators that incorporate the high and low of each bar, offering a more accurate picture of true volatility. However, these estimators are rarely used in standard backtesting practice, and even they cannot capture the gap risk between sessions or the path-dependent risk experienced during the session. For strategies with leverage or nonlinear payoff profiles, the underestimation of intraday volatility can lead to dangerously oversized positions.

The problem extends beyond position sizing to the way that strategy performance is reported and evaluated. The vast majority of backtested performance metrics, including drawdown statistics and risk-adjusted return ratios, are computed on a close-to-close or monthly-close basis. These figures describe the strategy's behaviour as observed through

the narrow lens of periodic snapshots. But the trader's lived experience is a continuous exposure to intraday price movement, not a series of snapshots. A strategy that reports a maximum drawdown of 15% on a close-to-close basis may have experienced intraday mark-to-market drawdowns of 25% or more, drawdowns that triggered real margin calls, real psychological distress, real risk of forced liquidation, and potentially real capitulation, none of which appear in the published performance summary. The intraday volatility at the smallest observable resolution is the actual experience of the position holder, and any performance reporting that smooths over this reality by measuring only at periodic intervals is presenting an incomplete and systematically flattering picture of the strategy's risk profile.

4.5 The Rationalisation of Coarse Data

The technical problems described above (concealed stop triggers, OHLC sequence ambiguity, settlement price artefacts, and understated volatility) grow more severe as bar size increases, and are at their worst with end-of-day data. Yet many retail systematic traders continue to develop and evaluate strategies using exclusively daily bars. This persistence owes less to ignorance than to motivated reasoning. The trader who lacks access to intraday data, or who does not wish to pay for it, or who lacks the storage and computational resources to process it, faces a choice: acknowledge a material limitation in their research infrastructure, or construct a narrative in which the limitation does not matter. Most choose the latter.

The narrative takes a characteristic form. The trader argues that their strategy operates on a sufficiently long timeframe (weekly signals, monthly rebalancing) that intraday price movements are irrelevant noise. Decisions are made at the daily close; orders are placed at or near the next open; the intraday path between one close and the next is, on this account, immaterial. The argument sounds reasonable. It is also delivered with considerable confidence, sometimes with a note of indignation at the suggestion that more granular data might be necessary.

But, the argument is almost never tested empirically by those who make it. The trader who asserts that daily data is sufficient for their medium-term strategy has, in most cases, never run that same strategy on minute-level data to verify the claim.

This is the critical omission.

When the comparison is performed, the results frequently differ not in degree but in kind. The equity curves diverge, sometimes dramatically, because the guesswork required to interpolate prices within daily bars compounds across thousands of simulated trades over

a multi-year backtest. Every time the backtesting engine must resolve an ambiguous fill (did the stop or the target get hit first? was the limit order reached before or after the reversal?), it is making an assumption that may or may not correspond to what actually happened. With intraday data, many of these assumptions become unnecessary: the price path is observed rather than inferred, and fills can be evaluated against actual traded prices at actual times. The accumulated difference can alter the fundamental shape of the equity curve, turning apparent winners into losers, and causing trades that would have been taken to be missed, transforming smooth upward trajectories into sequences interrupted by drawdowns that the daily-bar version never revealed.

A concrete diagnostic is available to any trader with access to both daily and minute-level data: take any strategy that uses stop-loss or profit-target orders, run it once with signals generated from daily bars and orders evaluated against daily OHLC using the platform's default bar-resolution convention, then run the identical strategy with signals still generated from daily bars but orders evaluated against the underlying minute-level data within each bar. Compare the trade-by-trade outcomes. The distribution of mismatches (trades classified as winners in the daily-bar version that are losers in the minute-resolution version, and vice versa) and the net direction of the aggregate performance difference provide a direct, reproducible measure of the bias introduced by coarse-bar order evaluation. In practice, this comparison almost invariably reveals a systematic optimistic bias in the daily-bar version, consistent with the stop-concealment mechanism described above.[10] Any trader who has not performed this comparison on their own strategies is operating on an untested assumption about the fidelity of their simulation.

The technology to eliminate this problem already exists and is simple in design. A properly architected backtesting engine stores (at least) minute-level data as its foundational layer and aggregates upward to whatever bar interval the strategy requests. When the trader specifies daily bars, the engine builds each daily bar from the underlying minute records, presents the aggregated bar to the strategy logic for signal generation, and then evaluates all resulting orders against the original minute-level data within that bar. The trader writes and thinks in terms of daily bars (daily closes, daily ranges, and daily indicators), but the execution simulation operates at minute resolution transparently, without any additional effort from the user. This architecture means that even a strategy designed around weekly signals and monthly rebalancing benefits from intra-bar order resolution: stops are evaluated against the actual intraday path, limit orders are filled

[10] In one test across a diversified futures portfolio, the daily-bar version showed a Sharpe ratio roughly 40% higher than the minute-resolution version of the identical strategy. The trade count was the same; only the fill prices and stop triggers differed.

at the price and time they would have been reached in the real market, and the OHLC sequence ambiguity described earlier is resolved by observation rather than convention. The computational overhead of this approach is modest for the typical retail use case: a decade of one-minute bars for a single liquid futures contract occupies a few hundred megabytes, and even a diversified portfolio of twenty to thirty instruments remains well within consumer-grade storage and processing capacity.[11] (Tick-level or Level 2 depth-of-book data, and large multi-asset universes, can become operationally heavier, but these are not required for the order-evaluation improvements described here.) The accuracy gain is substantial. That many traders continue to resist this approach owes more to motivated reasoning than to any genuine technical constraint. Havin

The question that daily-data advocates ask is "why would I need anything more granular?" But this formulation inverts the burden of proof. The correct question is the opposite: if more granular data is available, why would you *not* use it? The downside cost of using minute-level data (somewhat larger storage requirements and modestly longer processing times) is trivial relative to the capital at risk. The downside cost of not using it (a backtest whose results may diverge badly from achievable live performance) is potentially catastrophic. In an era when minute-bar data for most liquid futures and equities is readily available at modest cost, and when even consumer-grade hardware can store and process decades of minute-level history without difficulty, the case for relying exclusively on daily data is far weaker than most traders assume. Many sound approaches to systematic trading can be designed around the constraints of daily data, provided the researcher understands what those constraints are and designs their strategy accordingly. But the trader who uses daily data without acknowledging its limitations, who does not verify their results against more granular data when it is available, and who rationalises the omission as a principled choice rather than a convenience, is accepting an unnecessary and unquantified source of error.

[11] Having said the overhead is modest, it is not so modest that TradeStation can deal with it. Turning on look-inside-bar-backtesting with 1-minute granularity quickly renders TradeStation inoperable.

5 Statistical Methodology Failures

5.1 The Multiple Comparisons Problem

Harvey, Liu, and Zhu (2016) demonstrated that the threshold for statistical significance in backtested strategies must be adjusted for the number of strategies tested. Their work, which examined the factor zoo in academic finance, showed that a t-statistic of 2.0 (the traditional threshold for significance at the 5% level) is statistically inadequate when hundreds or thousands of strategy variants have been evaluated. They proposed a minimum t-statistic of approximately 3.0 for newly discovered factors, accounting for the implicit multiple testing that pervades the field.

This problem is even more acute in retail backtesting, where the number of variations tested is typically far greater than in academic research and almost never reported. A researcher who tests a moving average crossover strategy with twenty combinations of fast and slow periods, across ten instruments, with three different filters, has evaluated six hundred strategy variants. Presenting the best performer as "the strategy" without adjustment for multiple testing is meaningless. Yet this is standard practice.

Bailey, Borwein, Lopez de Prado, and Zhu (2014) formalised this intuition with what they called the false strategy theorem. The expected maximum Sharpe ratio among N independent zero-skill strategies grows as approximately $\sqrt{2 \ln N}$, which means that trying just ten configurations of a worthless strategy is expected to produce one with a Sharpe ratio above 1.5 in-sample, while the true out-of-sample expectation remains zero. They derived a corresponding minimum backtest length (MinBTL): with five years of daily data, no more than forty-five independent configurations should be tested before the expected maximum Sharpe ratio from pure noise reaches 1.0. With only two years of data, the budget drops to seven. Most retail traders blow past these thresholds before lunch on the first day of development.

The false strategy theorem describes a search across configurations of a zero-skill process: the inflated Sharpe ratio is pure noise, and the mitigation is the multiple-testing correction the theorem prescribes. The position is materially worse for any strategy whose payoff is sensitive to intra-bar order sequencing. As Section 4.2 sets out, the OHLC path assumption generates a per-trade error that is structurally one-signed in the strategy's favour, and the per-trade bias grows with the tightness of the bracket relative to typical bar range. A parameter search then preferentially selects configurations with tight stops and tight targets, which is exactly the region of parameter space where the bias is largest. The search is no longer data-mining unbiased noise; it is data-mining a systematically positive artefact. The Bailey and López de Prado correction adjusts for the search; it does

not remove a bias that was already present before the search began. For path-sensitive strategies on coarse bar data, the two effects multiply, and no multiple-testing correction alone can recover honest performance metrics from the contaminated simulator.

Wiecki, Campbell, Lent, and Stauth (2016) provided striking empirical confirmation of this problem. Using a dataset of 888 algorithmic trading strategies developed on the Quantopian platform, each with at least six months of out-of-sample performance, they found that commonly reported backtest metrics such as the Sharpe ratio offered almost no predictive value for out-of-sample results ($R^2 < 0.025$). More pointedly, they found a statistically significant positive relationship between the amount of backtesting a researcher performed on a strategy and the magnitude of the discrepancy between in-sample and out-of-sample performance: the more a strategy was tested and refined, the worse it performed in live trading relative to its backtest.

This is the multiple comparisons problem made empirically visible at scale. Each iteration of parameter adjustment constitutes an implicit additional test, inflating in-sample performance while degrading the strategy's generalisability to unseen data.

The natural response to this evidence is to ask: what can be done? The academic literature offers a family of methods specifically designed to correct for multiple testing in strategy evaluation, yet these tools are almost unknown among retail back-testers. White's Reality Check (2000) uses bootstrap resampling to test whether the best-performing strategy from a set of candidates is genuinely superior to a benchmark after accounting for the number of alternatives tested. Hansen (2005) refined this into the Superior Predictive Ability (SPA) test, which is more powerful against specific alternatives and has been extended by Romano and Wolf into stepwise procedures that identify *which* strategies in a set retain significance after correction, not merely whether any of them do. Bailey and Lopez de Prado (2017) proposed the Probability of Backtest Overfitting (PBO), estimated via Combinatorially Symmetric Cross-Validation (CSCV): the data is partitioned into multiple subsets, strategy parameters are fitted on every possible training combination, and the frequency with which the best in-sample configuration underperforms out-of-sample provides a direct estimate of the probability that the backtest is overfit. Their related work on the Deflated Sharpe Ratio (2014) adjusts the reported Sharpe ratio for the number of trials conducted, non-normality of returns, and sample length, producing a statistic that is far more informative than the raw Sharpe about whether the observed performance is distinguishable from chance. The common thread is that *the number of strategies tested must be treated as a parameter of the evaluation*, not an incidental detail to be omitted from the report. Any researcher who has tested more than a handful of variants and does not apply some form of multiple-testing correction is, in

effect, reporting the expected maximum of a set of random draws (exactly the quantity the false strategy theorem estimates) and presenting it as an expected value.

For traders who will not invest the effort to implement the formal corrections above, a useful back-of-envelope heuristic has been proposed in the practitioner literature: discount the reported expected return by a factor of $1 - 0.95^N$, where N is the number of strategy variants tested.[12] The heuristic is crude. It is also vastly better than the standard retail practice of testing 50 variants and reporting the equity curve of the best.

5.2 Insufficient Independent Observations

A ten-year daily backtest of a monthly rebalancing strategy produces only one hundred and twenty observations, and fewer if the strategy is not always in the market. The number of truly independent observations may be smaller still if signals are serially correlated, as they often are in trend-following and momentum strategies where a single trend can generate a cluster of correlated trades.

The statistical power of a test with one hundred and twenty observations to detect a real but modest edge (say, a Sharpe ratio of 0.5) is discouragingly low. The confidence intervals around estimated performance metrics are wide, and the probability of a truly profitable strategy appearing unprofitable (or vice versa) in a single sample is substantial. Yet backtests are routinely presented with precision ("Sharpe ratio: 1.47") that implies a level of certainty the data cannot support. Two decimal places do not make a number reliable. Bailey and Lopez de Prado (2014) have argued persuasively that the Sharpe ratio, as commonly estimated from backtest data, is a deeply unreliable measure of forward-looking performance.

5.3 Regime Dependence and Non-Stationarity

Financial markets are non-stationary systems. The statistical properties of returns (their mean, variance, autocorrelation structure, and tail behaviour) change over time as market structure, participant composition, regulatory frameworks, and macroeconomic conditions evolve. A strategy optimised on a period dominated by a particular regime (low-volatility trending markets, for example) may perform entirely differently in an alternative regime (choppy, mean-reverting markets with elevated volatility). A backtest that spans multiple regimes may show acceptable aggregate performance while concealing extended periods of severe underperformance. The aggregate hides the pain.

[12]The heuristic is from a Quantreo newsletter post on the multiple-testing problem (see references). It falls well short of the formal corrections such as Reality Check or PBO in statistical power. It does at least force the researcher to acknowledge that testing 50 variants of an idea is a different statistical exercise from testing one. At $N = 50$, the heuristic discounts the reported return by roughly 92%.

The failure to account for regime dependence is related to the broader problem of overfitting. A strategy with sufficient free parameters can be fitted to any historical data set, including one containing multiple regimes, but the fitted model captures the specific sequence of regimes in the sample rather than any stable underlying relationship. Walk-forward analysis and out-of-sample testing can mitigate this problem but do not eliminate it, particularly when the out-of-sample period is short relative to the regime cycle. Worse, these techniques can only validate a strategy against conditions that have already occurred in the historical record; they offer no protection against structural changes or unprecedented market events. The limitations of these widely trusted validation methods are examined in the following section.

Cryptocurrency markets illustrate regime dependence in an especially stark form. Regulatory changes in crypto can be abrupt and binary: instruments declared securities retroactively, exchanges banned from entire jurisdictions overnight, leverage limits imposed with little warning. A crypto backtest spanning 2019 to 2025 assumes a degree of regulatory continuity that simply did not exist during that period. More broadly, crypto amplifies virtually every failure mode discussed in this section. Regime shifts are faster, leverage is higher, liquidity cliffs are more extreme, and the institutional structure of the market itself is less stable than in any traditional asset class. If the arguments in this paper apply to regulated futures and equities, they apply to crypto with considerably greater force.

5.4 The Parameter Plateau Illusion

A widely taught principle of strategy optimisation is that robust strategies should exhibit "stable regions" or "plateaux" in their parameter space: zones where performance degrades only gradually as parameters are varied, in contrast to narrow spikes where a single parameter value produces good results but neighbouring values do not. The intuition is sound: a strategy whose performance is insensitive to small parameter changes is more likely to generalise than one whose performance depends on a precise setting. However, the standard method of identifying these plateaux contains a subtle mathematical artefact that is almost never discussed.

Consider a moving average crossover strategy optimised over a lookback range of 10 to 200 bars in steps of 10. On the surface, each step represents an equal increment: 10 additional bars of data. But the *proportional* change in the information available to the indicator varies enormously across the range. Stepping from 10 bars to 20 bars doubles the lookback window, a 100% increase in the data the indicator considers. Stepping from 190 bars to 200 bars similarly adds 10 bars, but this represents only a 5.3% increase in

the data available to the indicator. The indicator's output changes far less in response to a 5% perturbation than a 100% perturbation, and this is true regardless of whether the strategy has captured a genuine signal or has merely fitted to noise.

The consequence is that parameter plateaux are *mathematically expected* at the upper end of any lookback range, even in the complete absence of a real edge. An indicator fitted to noise at a lookback of 190 bars will produce nearly identical output at 200 bars, because the two windows share approximately 95% of their data. The apparent stability is not evidence of robustness; it is an artefact of the diminishing marginal information content of each additional bar. Conversely, the lower end of the range, where each step represents a large proportional change, will naturally exhibit greater variability, which may be misinterpreted as fragility even if the strategy does capture a genuine short-lookback effect.

This artefact has practical consequences for strategy selection. A researcher who scans a broad parameter range and selects the plateau region is systematically biased toward longer lookback values, which may correspond to slower, more smoothed versions of the strategy that appear robust in optimisation but are simply insensitive to parameter perturbation because each step changes so little. The correct approach is to evaluate parameter sensitivity on a proportional basis (for example, testing lookbacks of 10, 15, 22, 33, 50, 75, 112, 168, each approximately 50% larger than the last) so that each step represents a comparable change in the information available to the indicator. Very few retail traders or backtesting tutorials employ this technique; the standard linear parameter sweep, with its built-in bias toward apparent stability at longer lookbacks, remains the default.

6 The Robustness Testing Illusion

Traders who are aware of the overfitting problem described in the preceding section often turn to a family of validation techniques (Monte Carlo simulation, synthetic data generation, and walk-forward analysis) in the belief that passing these tests constitutes evidence of genuine robustness. Each of these methods has legitimate applications, but each also has fundamental limitations that are poorly understood and rarely disclosed. What looks like methodological rigour can, in practice, provide false confidence rather than genuine validation.

The tests feel rigorous. That is precisely what makes them dangerous.

6.1 Monte Carlo Trade Shuffling

The most common form of Monte Carlo analysis in retail backtesting involves randomly shuffling the sequence of trades produced by a backtest and re-computing the equity curve across thousands of permutations. The resulting distribution of outcomes is used to estimate confidence intervals around metrics such as maximum drawdown and the probability of ruin. The technique is widely recommended in trading education and is built into several commercial platforms.

The method rests on an assumption that is rarely examined: that the individual trades are independent and identically distributed, such that any ordering of the trade sequence is equally plausible. For a narrow class of strategies (those with fixed position sizing, no portfolio-level risk filters, and no dependence on recent trade outcomes) this assumption may be approximately valid. But for the majority of strategies that traders actually deploy, it is not.

Consider first the problem of path dependency in portfolio-level risk management. A strategy that manages exposure to a margin budget, or that reduces position size during drawdowns, or that filters new entries when portfolio heat exceeds a threshold, produces a trade sequence in which each trade's existence and size depend on the outcomes of preceding trades. Shuffling the sequence destroys this dependency. A permuted sequence may place a cluster of large losses early, triggering a drawdown-based position reduction that would have prevented several of the subsequent trades from being taken at all. Conversely, it may front-load winners, creating equity and margin headroom that would have permitted larger positions than the strategy's rules would actually have allowed at that point. The shuffled paths are not alternative histories of the same strategy; they are histories of a strategy that could not have existed.

The distortion grows worse when the strategy employs any form of dynamic position siz-

ing. Systems that scale position size based on recent win rate, current equity, volatility regime, or streak length produce trade records in which the dollar magnitude of each trade is a function of the trades that preceded it. Shuffling the sequence while preserving the original dollar amounts produces paths in which large positions appear at points where the sizing algorithm would have mandated small ones, and vice versa. Shuffling the sequence and recalculating sizes for each permutation is more defensible but computationally expensive and still fails to account for the path-dependent decision of whether to take the trade at all.

The deeper problem is that Monte Carlo trade shuffling is entirely agnostic to market microstructure. The shuffled sequences imply no relationship between trade timing and market conditions. A strategy that trades based on specific price patterns, bar sequences, or structural setups produces trades that are inherently tied to the market context in which they occurred. A mean-reversion trade entered after a three-day decline followed by a hammer candle at support cannot meaningfully be relocated to an arbitrary point in the timeline; the market conditions that generated the entry signal would not have existed at that point, and the subsequent price behaviour that determined the trade's outcome would have been entirely different.

The shuffled paths are not improbable. They are impossible.

Confidence intervals derived from impossible paths are not conservative estimates; they are meaningless.

Monte Carlo simulation has genuine value when applied to well-understood stochastic processes with clearly defined assumptions (such as modelling the distribution of portfolio returns under parametric assumptions about return distributions). But the trade-shuffling variant as commonly applied in retail backtesting tools provides a veneer of statistical sophistication over an analytically unsound procedure. The researcher who reports that their strategy "survived 10,000 Monte Carlo simulations" has demonstrated only that a set of impossible trade sequences produced a range of outcomes, a finding with limited bearing on the strategy's actual robustness.

6.2 Synthetic Data and Noise Injection

A related class of validation techniques involves perturbing the input data rather than the trade sequence. Common approaches include adding random noise to price series, generating synthetic price paths from fitted statistical models, bootstrapping returns to create alternative histories, and shifting entry or exit prices by random amounts to simulate execution uncertainty.

The appeal is intuitive: if a strategy remains profitable when the underlying data is perturbed, it is presumably not dependent on the precise historical path and is therefore more likely to generalise. In practice, however, the value of this approach depends entirely on how the perturbations are constructed, and the most common methods introduce distortions that undermine the validity of the test.

Adding Gaussian noise to a price series destroys the autocorrelation structure, volatility clustering, and fat-tailed behaviour that characterise real market data. A strategy that exploits mean reversion after volatility spikes, or that depends on the serial correlation of daily returns during trending regimes, will perform differently on noise-corrupted data not because it is fragile but because the noise has destroyed the statistical properties the strategy was designed to exploit. Demonstrating that a strategy fails when its edge is removed from the data is not evidence of fragility; it is a tautology. In any case, strategies rarely fail in the real world because the market suddenly exhibits more random noise. They fail because the underlying structural changes: dominant participants enter or leave, volatility regimes shift, central banks intervene, or liquidity conditions deteriorate. Testing a strategy against artificially jittered data demonstrates that the algorithm is not hyper-sensitive to a few ticks of slippage, a useful but narrow finding, but it does nothing to establish that the strategy relies on a genuine and persistent market inefficiency, or that it will survive a structural shift in the conditions that generated the apparent edge.

Synthetic price generation from fitted models (geometric Brownian motion, GARCH processes, regime-switching models) suffers from a different problem: the generated paths reflect the assumptions of the generative model, not the properties of real markets. If the model fails to capture the specific microstructure features that the strategy exploits (and it almost certainly will not, since no standard generative model reproduces the full complexity of real market dynamics) then poor performance on synthetic data is uninformative. Conversely, strong performance on synthetic data that shares the broad statistical properties of the training set provides weak evidence of robustness, since the synthetic paths are, by construction, drawn from the same distribution as the original data and therefore test in-distribution generalisation rather than resilience to genuinely novel conditions.

A more extreme variant of this approach, implemented in some commercially available tools,[13] constructs entirely new "out-of-sample" price series by randomly extracting individual bars from the historical record and stitching them together, typically using logarithmic returns to ensure the resulting series looks visually plausible. The output

[13]These features are typically marketed as "stress testing" or "robustness validation." The marketing copy rarely mentions that the generated data has no serial structure, which is rather the point of the entire exercise.

may pass a casual visual inspection: the price path meanders in a generally realistic fashion and the overall series resembles a real market. But the construction is a catastrophic misuse of time-series data. Randomly extracting days from a continuous financial time series destroys the autocorrelation, volatility clustering, momentum persistence, and path-dependency that define the behaviour of real markets. The resulting series is a sequence of unrelated daily snapshots arranged in an arbitrary order: a Tuesday from 2017 may be immediately followed by a Friday from 2010, which is followed by a Monday from 2023. No trend can develop across such a series because trends are, by definition, serial phenomena requiring consecutive bars to move in a correlated direction. No volatility regime can persist because the regime information is encoded in the sequence, which has been destroyed. A trend-following or momentum strategy tested on such data is not being tested at all in any meaningful sense. It is being asked to find serial structure in a series that has been explicitly constructed to contain none. That the strategy fails is uninformative: the failure tells us nothing about whether the strategy would fail on genuinely new but structurally intact market data. The technique has the appearance of scientific rigour (randomisation, out-of-sample construction, large sample generation) but it is testing an impossibility and interpreting the inevitable failure as evidence about the strategy rather than about the test.

The most defensible form of data perturbation is the systematic variation of execution assumptions: testing the strategy across a range of slippage multipliers and timing offsets to establish how sensitive the results are to execution quality. This is not, strictly speaking, a robustness test of the strategy's edge (it is a sensitivity analysis of the execution model) but it addresses a genuine and quantifiable source of uncertainty that directly affects achievable performance.

6.3 Walk-Forward Analysis

Walk-forward analysis (WFA) and its optimisation-oriented variant, walk-forward optimisation (WFO), represent another approach available to the retail systematic trader. The basic procedure (optimising strategy parameters on an in-sample window, testing the optimised parameters on an immediately subsequent out-of-sample window, and repeating this process across the full historical period) directly addresses the overfitting problem by separating the data used for fitting from the data used for evaluation. When executed correctly, WFA produces a synthetic out-of-sample track record that provides stronger evidence of generalisability than a simple backtest.

However, passing WFA is a necessary but not sufficient condition for genuine robustness, and the method has several fundamental limitations that are frequently overlooked.

Window selection bias. The choice of in-sample and out-of-sample window sizes shapes WFA results, and there is no objectively correct window size. Changing the window configuration can dramatically alter whether a strategy passes or fails. The walk-forward matrix technique (running WFA across multiple in-sample and out-of-sample window combinations and looking for clusters of positive results) mitigates this problem but does not eliminate it, since the choice of which window combinations to test is itself a degree of freedom.

Meta-overfitting. The most damaging limitation of WFA is that the validation process itself can become a source of overfitting. A researcher who tests a strategy with multiple fitness functions, multiple window configurations, multiple parameter ranges, and multiple filter combinations, selecting the configuration that produces the best walk-forward results, has effectively optimised the validation procedure to the historical data. This meta-overfitting defeats the entire purpose of out-of-sample testing but is extremely difficult to detect from the outside, because the reported results show a clean walk-forward pass. And, when pressed, most researchers cannot even state how many configurations they tested. The number of WFA configurations tested is almost never disclosed, yet it is subject to exactly the same multiple comparisons problem that WFA is intended to solve.

Regime change lag. WFA responds to regime changes only after they have occurred. When market conditions shift (from trending to mean-reverting, from low volatility to high volatility, from accommodative to restrictive monetary policy) the strategy's performance deteriorates before the walk-forward procedure can adapt by re-optimising on the new regime's data. For slow-moving regime changes, this lag may be manageable. For abrupt structural breaks (a central bank policy reversal, a liquidity crisis, a geopolitical shock) the strategy may suffer catastrophic losses before WFA has any opportunity to respond.

The stationarity assumption. WFA implicitly assumes that the data-generating process is sufficiently stationary that patterns observed in one window will persist into the next. This assumption is violated when market participant composition changes (the rise of algorithmic market-making, the growth of passive investing), when regulatory frameworks shift (decimalisation, MiFID II, Dodd-Frank), when information propagation speeds change, or when macroeconomic paradigms shift. WFA tested on data spanning a single monetary policy regime provides no evidence of robustness across regime transitions, yet such transitions are exactly the conditions most likely to produce large losses.

In-distribution only. WFA can only validate a strategy against conditions that appear in the historical record. It cannot anticipate flash crashes (2010, 2015), pandemic market

dislocations (March 2020), currency peg breaks (Swiss franc, January 2015), or any other event without historical precedent. The trader who reports that a strategy "passed walk-forward analysis across twenty years of data" has demonstrated robustness within the distribution of those twenty years, a useful finding, but one that provides no guarantee against out-of-distribution events. This limitation is inherent to any validation method that relies on historical data, not a defect specific to WFA.

A strategy that fails walk-forward analysis is almost certainly over-fitted and should be discarded. But the reverse is not necessarily true: a strategy that passes walk-forward analysis is not necessarily robust. A strategy that passes has merely cleared the minimum bar for further consideration. It has not been shown to be resilient under conditions outside the historical distribution. It is trivial to demonstrate that testing with a large enough parameter set can produce apparently robust strategies that are not robust in the real world.

Notwithstanding the merits of WFA, it will not correct for many of the problems discussed in this paper. The pitfalls of survivorship and selection bias discussed in Survivorship and Selection Bias and the comparison problems discussed in The Multiple Comparisons Problem should always be front of mind when evaluating strategies that have been developed using WFA (or with any strategy development process).

7 The Python Ecosystem: Network Effects and Technical Constraints

7.1 The "Coding as Edge" Fallacy

Python's dominance in retail algorithmic trading is a sociological phenomenon, not the result of a deliberate technical evaluation. What we observe is the self-reinforcing network effect of tutorials, libraries, and community conventions. This dominance feeds a misconception among novice traders: the belief that learning to program in Python is a skill to be learned because it confers a trading edge. In fact, it may do just the opposite because of the increased risk of implementing a substandard or subtly-wrong evaluation environment (plagued with many of the problems discussed earlier).

In reality, systematic trading is already populated by elite software engineers; technical proficiency is a prerequisite, not an edge. A genuine edge stems from market insight, risk management, realistic cost modelling, and the statistical literacy to distinguish signal from noise. Furthermore, while AI code generation tools have lowered the barrier to writing backtesting scripts, they do not replace the domain expertise required to spot subtle look-ahead biases, concurrency failures, or unrealistic fill assumptions hidden within generated code. The experienced developer who adopts AI-assisted tools extracts far more from them than the novice: they know what to ask for and where the subtle bugs are likely to hide. The novice, lacking this evaluation framework, is prone to accepting generated code at face value, and AI-generated backtesting code is subject to all the same pitfalls discussed throughout this paper.

Underlying this is a more fundamental confusion: the conflation of programming with trading.

Programming is a tool; trading is the objective.

The vast majority of programmer-traders fail not because their code is insufficiently sophisticated, but because trading requires competencies that programming does not develop: market intuition, business judgement, risk awareness that extends beyond mathematical models, and the discipline to adhere to a process during periods when the process appears to be not working. The Python ecosystem obscures this reality by centring the narrative on code rather than on the far harder work of developing market judgement.

This confusion is amplified by the proliferation of backtested strategy showcases across YouTube, Substack, and similar platforms where the barrier to publication is zero: no track record requirement, no methodology review, and no accountability when the showcased strategy fails in live markets. The aspiring trader who consumes this content

absorbs not just strategy ideas but an implicit standard of evidence that is far too low. At best, such content serves as idea generation; the key to progress lies in developing one's own rigorous evaluation process, not in replicating the unverifiable published work of others.

I have spoken to many aspiring algorithmic traders who have faithfully followed the guidance of industry "experts" (who often charge substantial course fees to impart their wisdom), only to find that what they have been taught does not survive first contact with the enemy. Unfortunately, it seems that this is often one of the stepping stones along path to success in algorithmic trading. I'd suggest that may aspiring traders would be shocked to learn that many esteemed authors and teachers are just that: authors and teachers. They are not traders.

A personal anecdote illustrates the point. I was approached by a relatively high-profile industry figure who required my help because of the data issues they were having, specifically with TradeStation, but this was a much more general problem of managing a large amount of data. The task required quite a bit of programming to achieve. By his own admission, this person had spent months trying to solve the problem. I committed to helping him in return for the output of the research that would be done using the data I prepared. Disappointingly, that same figure continues to tell his many acolytes that programming is a skill that is not required to be a successful trader, even though he was clearly prepared to use my help and, as far as I know, is using the help of other people to achieve his objectives. This level of disingenuity is unfortunately very common in the trading industry (or at least in the trading education and course-selling industry).

7.2 Technical Limitations and Production Realities

Python is an exceptional tool for exploratory research and rapid prototyping. However, it presents significant challenges as the foundation for full-lifecycle production trading engines. The default CPython runtime has historically limited CPU-bound multithreading via the Global Interpreter Lock (GIL); while free-threaded builds have been available since Python 3.13, they are not yet the default installation and ecosystem support remains uneven. In practice, achieving the predictable latency and throughput required by production systems typically involves multi-process designs, native extensions, or offloading compute-intensive work to non-Python cores. These workarounds are effective but add architectural complexity that undermines one of Python's core selling points. Dynamic typing, meanwhile, permits silent runtime errors (type mismatches, unexpected `None` values, shape misalignment in array operations) that would be caught at compile time in statically typed languages and that can corrupt backtest results in ways that are

difficult to detect after the fact.

Worse, the Python backtesting ecosystem encourages a mode of development in which the backtest is a standalone artefact, separate from the live trading system. This separation means the transition to production frequently involves a complete reimplementation. Every reimplementation introduces bugs. And bugs in a live trading system cost money.

Many traders justify remaining in Python due to its library ecosystem, but this is a false economy. Standard technical indicators and data manipulation routines are mathematically simple and trivial to implement in any language. The edge in systematic trading does not come from easy access to a moving average function; it comes from the fidelity of the execution model and the robustness of the portfolio simulation.

These limitations are not unique to Python. Legacy compiled platforms can suffer from constraints that are, in some respects, even more severe. TradeStation's EasyLanguage runs in a 32-bit environment with hard memory limits, is restricted to single-threaded execution, is confined to Microsoft Windows, and is widely regarded by its own user base as unstable.[14] The 32-bit memory constraint directly prevents loading the minute-level datasets that I argue are necessary for credible order evaluation (Section 4). The single-threaded limitation precludes the exhaustive walk-forward and sensitivity analysis that rigorous strategy evaluation demands (Section 5 and Section 9). The broader point is that platform limitations shape research quality regardless of whether the platform is interpreted or compiled, open-source or commercial.

7.3 The Case for Alternatives

Languages such as Go and Rust offer significant architectural advantages for production trading systems: compile-time type safety against data corruption and silent errors, true concurrency to handle parallel workloads, and, in CPU-bound simulation workloads that cannot be pushed into vectorised native libraries, order-of-magnitude speedups over interpreted Python. The point is not speed for its own sake; it can be the difference between running a handful of walk-forward configurations and running a genuinely exhaustive sensitivity and stability programme. A backtest that takes hours in Python may complete in minutes in a compiled language, making thorough parameter exploration feasible within practical time constraints. Go's goroutine model provides lightweight concurrency primitives that map naturally to the parallel demands of a trading system, while Rust's

[14]The TradeStation user forums (now not publicly accessible, which I suspect is in part to hide the blunt feedback from users, and in part to hide the fact that the user base is dwindling fast) contain years of threads documenting crashes, memory exhaustion, and unexplained backtest discrepancies. The platform's defenders generally do not dispute these problems; they argue that the platform's other strengths compensate for them.

ownership model prevents data races at compile time. Both produce statically linked binaries that eliminate the dependency management complexity that plagues Python deployments.

None of this should be read as a dismissal of Python. The practical recommendation is to use Python for what it does best (data science, hypothesis testing, preliminary analysis) while relying on robust, compiled systems for the unforgiving realities of execution and risk simulation. Nor should it be read as an argument that superior tooling is a substitute for superior judgment. The history of systematic trading contains numerous examples of traders achieving sustained success with relatively simple tools because they understood what those tools could and could not tell them. A trader who knows their platform intimately (its fill assumptions, its cost model, the specific ways its simulations diverge from reality) can work productively within those constraints in a way that a more technically advanced programmer-trader, lacking that self-awareness, cannot. Many of the most durable systematic strategies succeed not because they are computationally demanding, but because the trader's understanding of markets and risk is deep enough to compensate for the simplicity of the tooling.

8 Framework Monoculture and Methodological Homogeneity

Off-the-shelf backtesting frameworks are not neutral tools. Each framework embodies a set of assumptions about how strategies should be structured, what data they consume, how signals are generated, and how execution is modelled. These assumptions are absorbed by users through documentation and example code, and they produce a characteristic sameness in the way that problems are framed and solutions developed.

Most popular frameworks, for example, encourage a pattern in which a strategy is expressed as a series of indicator calculations followed by threshold-based entry and exit rules, evaluated bar-by-bar on a fixed-frequency time series. This pattern is natural for a certain class of strategies but is poorly suited to others: event-driven strategies, strategies that operate across multiple timeframes simultaneously, strategies that incorporate non-price data, or strategies whose execution logic is itself a source of edge. When the framework's implicit recipe does not accommodate these approaches, developers tend to adapt their ideas to fit the framework rather than the reverse, leading to a narrowing of the strategy space that is explored.

The more experienced users of a given framework develop conventions and "best practices" that are passed on to newcomers through tutorials and forum discussions. These conventions often reflect the limitations of the framework as much as any principled methodological choice. The community that forms around a given framework ends up approaching strategy development in remarkably similar ways: using the same indicators, the same entry patterns, the same default parameters, and the same evaluation metrics. Strategies that emerge from such an environment are correlated with one another. This is a problem for both individual traders (whose strategies may be crowded) and for the field as a whole (whose collective output represents a narrower exploration of the strategy space than its volume would suggest). And, since crowded strategies tend to unwind at the same time, the correlation creates systemic risk that no individual backtest can detect.

A concrete illustration of this phenomenon can be found in platforms such as TradeStation, which has for decades been one of the most widely used retail algorithmic trading environments. TradeStation's EasyLanguage scripting system and its built-in strategy development workflow are, by design, optimised for a specific class of strategies: indicator-driven, single-instrument, bar-by-bar systems evaluated on fixed-frequency data. The platform makes this class of strategy extraordinarily accessible: a user with modest programming experience can have a moving-average crossover system running within hours of first contact with the software. But the platform's strengths are also its constraints.

The ease of building indicator-based systems within TradeStation's framework implicitly discourages approaches that do not fit its paradigm: multi-instrument portfolio strategies, strategies with complex position-sizing logic, strategies that incorporate non-price data, or strategies whose execution model departs from the platform's built-in assumptions. The tool shapes the work, and when thousands of users adopt the same tool, the collective output converges toward the same narrow set of approaches, evaluated under the same set of assumptions, producing strategies that are correlated not because the underlying market relationships demand it but because the platform's architecture channels all users toward the same destination. This is not a criticism unique to TradeStation (it applies, in varying degrees, to every framework that makes certain approaches easy and others difficult) but TradeStation's longevity and large user base make it a particularly visible example of how platform constraints become community conventions.

It should be noted that many people over the years, including myself, have spent a considerable amount of time trying to add band-aid solutions (resorting to DLL callouts, implementing a client-server architecture to access TradeStation from outside its walled garden, mis-using the TradeStation Optimizer API to improve the user experience, building cut-and-paste code fragments to export data for processing outside TradeStation, and much more). The degree to which users of the TradeStation platform have used their ingenuity to try to work around its many shortfalls is a testament to their tenacity, but also an indicator of a dangerous blind spot: traders who become proficient on a particular platform cannot overcome the inertia of moving to a new platform, even if it is obviously better. In the brave new world of AI, it is even more imperative for traders to critically assess the relevance and competency of their platform to ensure that they are not left behind. My working assumption is that whoever takes the opposite side of any trade I place is probably smarter than I am, so I don't want to make it any harder than it already is by bringing the trading equivalent of the knife to a gunfight. To be fair to TradeStation, every platform has its limitations. The software development experience on most (all?) of the widely-available platforms is extremely poor. Some (including me in this paper!) will argue that it's not all about the platform. That's true, but I see no reason to operate on inferior tools. In my case, I have built my own backtesting platform. (For almost all traders, I would strongly recommend against going down that path!)

Other platforms have recognised specific instances of these problems and introduced partial mitigations. TradingView's "Bar Magnifier" mode exists because the OHLC sequence ambiguity described in Section 4.2 is a real and widely encountered failure: the feature uses finer-resolution data to resolve intra-bar order fills during backtesting, an implicit acknowledgement that standard bar-by-bar evaluation is insufficient. TradeStation and

MultiCharts offer an "Intra-Bar Order Generation" (IOG) mode that evaluates orders within each bar rather than only at bar close, addressing the same class of problems through a different mechanism. QuantConnect's LEAN engine takes a more explicit architectural stance, pushing users toward finer-resolution data for realistic fills and exposing custom slippage and margin models as first-class configuration options. That these features exist at all confirms the severity of the underlying problems; that they are options to be configured rather than default behaviour confirms that the retail backtesting ecosystem continues to prioritise ease-of-use over simulation fidelity.

The constraints in TradeStation's case go beyond conceptual preferences encoded in documentation and examples; many are hard limitations enforced by the platform's parser and runtime environment. EasyLanguage does not permit orders to be placed inside loops, eliminating the possibility of dynamic pyramiding or algorithmic scaling into positions. Named orders must be string literals rather than dynamically constructed expressions, preventing programmatic management of multiple concurrent entries. The platform cannot query how many data series have been provided to the strategy, making it impossible to write strategies that adapt to an arbitrary number of instruments. Variable back-references are subject to hard-coded limits that constrain lookback calculations. Complex data structures (maps, trees, graphs, and the arbitrary nesting of collections that modern languages provide as primitives) are effectively non-existent. Portfolio-level support is extremely limited, meaning that strategies are evaluated in isolation from one another with no mechanism for cross-strategy risk management or capital allocation, a limitation that directly reinforces the signal isolation problem discussed in Section 9. And there is no facility for custom optimisation metrics, so the researcher is constrained to the platform's built-in fitness measures when searching the parameter space.

Each of these limitations individually forces a design compromise; collectively, they make entire classes of strategy design impossible within the platform. A trader who has spent years developing within these constraints may not recognise what they have never been able to try. The strategies they build are shaped, to a degree that is difficult to appreciate from inside the ecosystem, by what the tool permits as much as by their ideas about markets. This is the mechanism by which platform constraints propagate into methodological homogeneity: not through explicit instruction, but through the gradual, invisible narrowing of the design space to the subset that the tool can express.

9 The Isolation Fallacy: Signal Generation Without Context

9.1 The Signal Isolation Blindspot

Novice and retail trading system developers suffer from what might be termed the signal isolation blindspot: an overwhelming focus on a single dimension of the problem, generating a profitable entry and exit signal (the alpha component), to the near-total exclusion of everything else that constitutes a complete trading system. This focus is understandable: the signal is the most intellectually engaging component, and it is the component that backtesting frameworks are designed to evaluate. However, a trading signal is only one element of a viable trading system. A complete system requires equally serious attention to transaction costs, risk management, portfolio construction, and execution. And deficiency in any one of these disciplines can render even a genuinely profitable signal unprofitable or catastrophic in practice.

The treatment of transaction costs is a particularly revealing symptom of this blindspot. Most retail traders regard transaction costs as an afterthought, a minor friction to be estimated and plugged into the model when the algorithm is otherwise near completion. The reality would shock most of them. The true quantum of transaction costs, encompassing not just commissions but bid-ask spreads (which vary with volatility and time of day), market impact (which scales non-linearly with order size), roll costs in futures (including calendar spread crossing costs and roll-window impact), financing costs for leveraged positions, and the adverse selection costs inherent in limit order execution, can easily consume the entirety of a strategy's apparent edge. A strategy that shows an annualised return of 8% before costs may show 2% or less after realistic cost modelling, and may be outright unprofitable once market impact at any meaningful scale is considered.

Treating costs as an afterthought reflects a fundamental misunderstanding of the problem. Transaction costs are a structural constraint that should shape strategy design from the outset. A strategy that is not designed with its cost structure in mind may have no reason to exist.

Backtesting engines and programming knowledge do nothing to address this blindspot. A programmer-trader can be highly proficient with Python, fluent in the API of their chosen backtesting framework, and capable of generating sophisticated equity curves, while remaining entirely ignorant of the disciplines that determine whether those curves are achievable in practice. The tools make the signal generation problem easy and the remaining problems invisible.

This is the core asymmetry, and I believe it is the single largest contributor to retail trading failure.

9.2 The Contaminated Signal

The isolation blindspot has a second, subtler form. So far the argument has been that the signal is only one component of a system and that the others get ignored. But the signal itself can be contaminated by a return that the researcher has not isolated from it, so that what looks like a validated edge is actually two distinct return sources blended together, only one of which the strategy is designed to capture. Futures trend-following on back-adjusted data is the cleanest example, and it is worth spelling out because it is so easy to walk into.

As discussed in Continuous Contract Construction, additive back-adjustment imparts a deterministic drift to the level series whenever a market sits persistently in contango or backwardation, and that drift is real roll yield rather than an artefact. The drift is also smooth, low in noise, and highly autocorrelated — which is to say it has exactly the statistical signature that a trend or breakout rule is built to detect. A momentum system run over such a series will lock onto the roll-induced drift and report it as trend. The backtest's measured trendiness (autocorrelation, Hurst exponent) and its Sharpe ratio are both inflated by a carry component the researcher never intended to trade and, more importantly, never measured. The system has not validated a trend edge; it has validated a trend-and-carry blend.

The danger is not that the profit is fake — the roll yield is real money, and a position that harvests it is genuinely paid. The danger is misattribution. The carry component persists only while the term-structure regime persists, and a curve that flips from contango to backwardation reverses the sign of the drift the trend rule was leaning on. A strategy whose backtested edge was partly carry, credited entirely to trend, can therefore degrade or invert in live trading for reasons that never appear in the historical record, because the historical record bundled the two returns into a single equity curve. The discipline that prevents this is the same one that defines the rest of this chapter: do not evaluate a signal in isolation. Decompose the backtested return into a roll-yield component and a residual price-trend component, and confirm that the part of the edge being credited to the trend rule is actually trend. A system tested across many contracts is more exposed to this, not less: the more instruments carry a persistent curve sign, the more of the aggregate Sharpe can be roll yield wearing the costume of trend.

9.3 The Risk Management Blindspot

Risk management deserves particular attention as a backtesting blindspot because of how widely it is misunderstood by the majority of retail traders. Most traders, if they address risk management at all, treat it as a backtesting step that begins and ends with

the selection of a basket of markets to trade. Having chosen a diversified-looking set of instruments (perhaps a mix of equity indices, bonds, commodities, and currencies) the trader considers the risk management problem solved and returns to the more engaging task of signal optimisation. And, in fairness, signal optimisation *is* more interesting than calibrating margin requirements or modelling roll costs. That is part of the problem.

This view of risk management is grossly inadequate. Selecting a basket of markets is, at best, the first layer of a multi-layered discipline, and even that first layer is often executed poorly. Naive instrument-level diversification (holding positions across many markets) provides far less risk reduction than most traders assume because correlations between markets are not stable. Markets that appear uncorrelated during benign conditions frequently become highly correlated during periods of stress, just when diversification is most needed. The trader who has "diversified" across equity indices and industrial commodities may discover during a risk-off event that all of their positions are, in effect, a single correlated bet on global growth.

Beyond this first layer, genuine risk management encompasses a multi-layered discipline:

- **Position sizing:** Calibrated to realistic, range-based volatility estimates rather than close-to-close volatility that understates true intraday risk (as discussed in Volatility and Risk Estimation).
- **Exposure limits:** Hard constraints applied at the instrument, sector, and portfolio levels.
- **Correlation monitoring:**Accounting for regime-dependent co-movement, recognising that markets that appear to be uncorrelated frequently become highly correlated during stress events.
- **Drawdown controls:** Predefined rules for position reduction during adverse equity excursions.
- **Tail risk analysis:** Statistical stress-testing of the strategy under adverse conditions that have not yet occurred but plausibly could.

Each of these disciplines requires a working understanding of probability distributions, the non-stationarity of financial return series, and the limitations of historical data as a guide to future risk. The trader who stops at market selection is exposed to all manner of risks they have not identified (concentration risk masquerading as diversification, leverage risk arising from volatility underestimation, liquidity risk during stress events, and correlation risk from regime changes, among others.

Portfolio construction, the discipline of combining multiple strategies or positions to

achieve a desired risk-return profile, requires an understanding of diversification that goes beyond naive instrument-level diversification to consider strategy-level correlation, regime-dependent co-movement, and the interaction between position sizing and portfolio volatility. This discipline is almost entirely absent from standard backtesting workflows and educational materials. Most traders have never heard of it.

The standard Python backtesting workflow does little to develop any of these competencies. A typical tutorial progresses from data acquisition to indicator calculation to signal generation to equity curve plotting, with risk management reduced to a fixed fractional position size and portfolio construction addressed not at all. The trader who follows this path emerges with a tool that can generate backtested equity curves but without the framework to evaluate whether those curves represent a tradable system or a statistical artefact.

A separate but related failure stems from the availability of ready-built strategies and strategy pipelines within popular algorithmic trading platforms. These offerings are attractive to inexperienced traders because they appear to eliminate the need for deep domain knowledge: the strategy is already backtested and already showing an impressive equity curve. The novice is therefore tempted to overlook the fundamental question of why the strategy might work, what market inefficiency or structural feature it exploits, under what conditions that feature is likely to persist, and what would cause it to disappear. Without answers to these questions, the trader has no basis for distinguishing a genuine edge from a statistical artefact, and no framework for deciding when to continue trading through a drawdown versus when to acknowledge that the strategy has ceased to function. The more experienced developer, by contrast, is far more likely to insist on understanding the causal mechanism behind a strategy before committing capital, recognising that a backtested equity curve without a plausible explanation is not evidence of an edge but merely evidence that a pattern existed in historical data.

9.4 The Drawdown Misconception

One of the most consequential forms of self-deception among novice system developers is the overestimation of one's ability to tolerate drawdowns. A backtested equity curve that shows a 30% peak-to-trough drawdown followed by a recovery to new highs appears manageable in retrospect: the viewer knows how the story ends. Living through that same drawdown in real time, with real capital, with no certainty of recovery, is an entirely different psychological experience.

Research in behavioural finance consistently demonstrates that losses are experienced approximately twice as intensely as equivalent gains (Kahneman and Tversky, 1979). A

drawdown that appears tolerable on a historical chart is, in practice, far more difficult to endure than the chart suggests. The temptation to abandon a strategy mid-drawdown (the worst possible time to do so, if the strategy retains its edge) is overwhelming for most traders, and the probability of strategy abandonment increases non-linearly with drawdown depth and duration.[15]

Moreover, backtested drawdowns understate the drawdowns that will be experienced in live trading. Every bias I have discussed pushes in the same direction. Optimistic fill assumptions, understated transaction costs, concealed intraday stops, overfitted parameters: all contribute to equity curves that are smoother and shallower than reality. The trader who sizes their account to tolerate the backtested maximum drawdown is, in effect, sizing for a best-case scenario. When live drawdowns inevitably exceed backtested drawdowns, the psychological and financial pressure to abandon the strategy can become irresistible.

The situation is worse than mere understatement. Bailey, Borwein, Lopez de Prado, and Zhu (2014) proved that when the return series exhibits serial dependence (as it does for most trend-following and mean-reversion strategies), there is a provably negative linear relationship between in-sample and out-of-sample Sharpe ratios: the more aggressively a researcher optimises in-sample, the worse the expected out-of-sample performance becomes. Not worse than the backtest. Worse than not optimising at all. Their result implies that the standard disclaimer "past performance is not an indicator of future results" is, in this context, too optimistic. When backtest overfitting is not controlled for, good backtested performance is an indicator of *negative* future results. For retail traders developing mean-reversion or trend-following strategies on serially correlated return series, which is to say nearly all of them, this is a direct prediction about their money.

Learning to code a backtest in Python and plugging in a few libraries is a necessary starting point, but it leaves a prospective systematic trader far short of what is required. I estimate that technical signal generation is perhaps 20% of the problem; risk management, portfolio construction, execution infrastructure, statistical literacy, and psychological preparedness constitute the remaining 80%. The backtesting ecosystem, by making the 20% easy and the 80% invisible, actively contributes to the failure rate among aspiring systematic traders.

If signal generation is indeed only 20% of the problem, it follows that the programming effort most retail traders pour into backtesting may itself be misallocated. Building a

[15] A useful exercise: ask any trader who claims they can tolerate a 30% drawdown whether they have ever actually experienced one with real money. In my experience, very few have, and those who have are notably more conservative in their claims the second time around.

backtesting engine that faithfully models execution, transaction costs, portfolio-level risk, and the full lifecycle of order management is an enormously demanding engineering task. It requires not only a high degree of programming skill but also deep domain knowledge of market microstructure, broker execution mechanics, and the statistical properties of financial return series, the very knowledge that most aspiring algorithmic traders have not yet developed. The sheer scope of a proper backtesting engine puts it well beyond the reach of many programmers, let alone those who are simultaneously learning both programming and trading. The result is that the novice trader who sets out to build their own backtesting infrastructure is likely to produce a system riddled with the very deficiencies catalogued throughout this discussion (unrealistic fill assumptions, understated costs, inadequate risk modelling) while consuming months or years of effort that could have been directed elsewhere.

For the yet-to-be-profitable trader, then, the question is not "how do I build a better backtest?" but "is building a backtest the highest-value use of my time at this stage of my development?"

For many, it is not.

The disciplines that constitute the other 80% of the problem (understanding risk management at a level that goes beyond selecting a basket of instruments, developing the statistical literacy to evaluate whether a result is genuine or an artefact, learning enough about execution to understand what a backtest cannot tell you, and cultivating the psychological resilience to trade through inevitable drawdowns) are arguably more important and more scarce than the ability to code a backtesting loop. The trader who has a deep understanding of risk, a realistic model of costs, and a genuine theoretical basis for their strategy but who uses a simple, even crude, simulation tool is likely to outperform the trader who has built an elaborate backtesting engine but lacks these foundational competencies. The tooling matters, but the judgment that governs its use matters more.

10 Proposed Minimum Standards for Credible Backtesting

Based on the foregoing analysis, I propose the following minimum standards for a backtest to be considered credible evidence of a potentially viable trading strategy. These standards are not sufficient to guarantee forward profitability, but their absence should be treated as a strong signal that reported results are unreliable.

Standard	Requirement Summary
Data quality	Documented source; multi-vendor cross-check on intraday extremes; archived dataset snapshots; specified continuous contract methodology; equity adjustment treatment (price-only versus total-return) declared explicitly
Execution model	Empirical spreads; volatility-dependent slippage; limit order queue modelling; explicit handling of limit halts, circuit breakers, and session halts
Transaction costs	All material costs: commissions, spreads, roll costs, financing, market impact; gross and net returns reported side by side
Currency effects	FX on P&L, margin collateral, and interest differentials; FX applied dynamically across the sample (not only at exit); report in base currency
Margin dynamics	Time-varying margin schedule modelled where possible; sensitivity to percentage-margin spikes during stress periods; combined effect of percentage and notional changes acknowledged
Counterparty risk	Venue failure base rate acknowledged; haircut applied to long-horizon crypto returns; stablecoin peg exposure flagged
Tax reporting	Pre-tax and after-tax returns reported side by side; holding-period regime declared; wash-sale equivalents accounted for
Intraday resolution	Intraday data for any strategy using path-dependent orders

Standard	Requirement Summary
Statistical rigour	Confidence intervals; multiple-comparison correction (Reality Check, SPA, DSR, PBO); pre-defined OOS period
Regime analysis	Performance by regime; drawdown depth and duration contextualised
Risk management	Position sizing; risk limits; portfolio interaction analysis
Robustness validation	Walk-forward analysis with disclosed window configurations; limitations acknowledged
Capacity assessment	Estimated maximum notional before edge degradation; impact model at scale
Process discipline	Defined pipeline; robustness tests aligned to strategy class

{.nowrap-col1}

Each standard is expanded below.

- **Data quality:** The data source must be documented and its known limitations disclosed. Adjustments for erroneous prints, phantom bars, missing sessions, and corporate actions must be described. Commercially sourced data must not be assumed to be clean; independent spot-checks against exchange records or alternative sources should be performed, particularly for older historical periods. For strategies sensitive to intraday extremes, the same strategy should be re-run against at least one independent vendor's data to detect pipeline-specific artefacts. Datasets should be archived at the time the backtest is run, because vendors silently rewrite their historical records and re-running months later can produce different results from the same code. For equity strategies, the treatment of dividend and split adjustments (price-only versus total-return) must be stated explicitly, and absolute-level logic must use unadjusted prices. For futures data, the continuous contract construction methodology must be specified, and the historical trading venue regime (pit, mixed, electronic) must be identified and its implications for strategy applicability discussed.
- **Execution model:** The backtest must employ a fill model that accounts for bid-ask spread (using empirical spread data where available, not a fixed assumption), slippage as a function of order size and market volatility, realistic order sequencing,

and the adverse selection inherent in limit order fills. Strategies using limit orders must model queue position or, at minimum, require price to trade through the limit level by a specified buffer before assuming a fill. The model must also handle limit-locked sessions, circuit breakers, and single-stock trading halts, rather than assuming the trader could have executed at the printed price during any such period.

- **Transaction costs:** All material costs must be included: commissions, exchange fees, spread costs, roll costs for futures (including calendar spread crossing costs and roll-window market impact), financing costs for leveraged positions, and an estimate of market impact for strategies intended to operate at scale. Gross and net returns should be reported side by side at every stage of the development pipeline; the gap between them is a direct measure of how much of the apparent edge depends on optimistic cost assumptions.
- **Currency effects:** For any strategy trading instruments denominated in a currency other than the account's base currency, the backtest must model exchange rate effects on trade-level profit and loss, on the value of margin collateral held in foreign currency, and on interest accrual differentials between the domestic and foreign currency. The FX overlay must be applied dynamically across the sample (at every trade, every margin revaluation, and every interest accrual), not as a single conversion at the end. Performance metrics must be reported in the trader's actual base currency, not in the instrument's native currency.
- **Margin dynamics:** For any leveraged strategy, the backtest should model the time-varying behaviour of margin requirements rather than assume a fixed schedule. The combined effect of percentage-margin changes and notional drift must be acknowledged. Where the actual historical margin schedule is not available, sensitivity to percentage-margin spikes (doubling or tripling during stress periods) should be tested and reported.
- **Counterparty risk:** For long-horizon backtests, the implicit assumption that the broker and exchange remained solvent and accessible should be made explicit. For crypto in particular, a haircut reflecting the historical base rate of venue failure should be applied, or the strategy universe should be restricted to venues with genuine segregation and audited reserves. Stablecoin peg exposure should be flagged as a credit and reserve assumption.
- **Tax reporting:** Backtests must report pre-tax and after-tax returns side by side, with the holding-period regime declared and the trader's actual marginal rates plugged in. Wash-sale equivalents and the differential treatment of futures versus

securities must be accounted for where relevant. Strategies of very different turnover cannot be compared on pre-tax returns alone.

- **Intraday resolution for path-dependent orders:** Any strategy that uses stop-losses, profit targets, trailing stops, or other intraday-sensitive order types must be tested with intraday data of sufficient resolution to determine order fill sequence. Daily OHLC data is insufficient for this purpose regardless of the strategy's rebalancing frequency.
- **Statistical rigour:** Results must be reported with confidence intervals. The number of strategy variants tested must be disclosed, and significance thresholds must be adjusted for multiple comparisons using established methods such as White's Reality Check, Hansen's Superior Predictive Ability test, or the Deflated Sharpe Ratio. Where feasible, the Probability of Backtest Overfitting should be estimated via Combinatorially Symmetric Cross-Validation. Out-of-sample results must be reported alongside in-sample results, with the out-of-sample period defined prior to optimisation.
- **Regime analysis:** Performance must be decomposed by market regime, with explicit identification of the conditions under which the strategy is expected to underperform. Maximum drawdown and drawdown duration must be reported and contextualised.
- **Risk management and portfolio context:** The strategy must be presented with a defined position sizing methodology, explicit risk limits, and an analysis of how it interacts with other portfolio components. Standalone signal evaluation without risk management context is insufficient.
- **Robustness validation:** The strategy must be validated using walk-forward analysis with fully disclosed window configurations, fitness functions, and the number of configurations tested. The walk-forward results must be presented as a necessary filter rather than sufficient proof of robustness. If Monte Carlo analysis or synthetic data testing is employed, the assumptions underlying the method must be stated and their applicability to the specific strategy class must be justified. Claims of robustness derived from trade-shuffling Monte Carlo applied to strategies with dynamic position sizing, portfolio-level risk filters, or pattern-dependent entries should be treated with particular scepticism, as described in Section 6.
- **Capacity assessment:** The backtest must include an estimate of the maximum notional the strategy can deploy before its edge is degraded by market impact. This requires, at minimum, comparing typical order sizes to historical volume and

depth-of-book data for the traded instruments. A strategy that is profitable at one contract but whose returns collapse at ten contracts has a capacity constraint that must be disclosed, and that is, for practical purposes, the binding constraint on the strategy's economic value. Market impact is both temporary (price recovers after the order is absorbed) and permanent (the information content of the order shifts the equilibrium price), and strategies with high turnover or concentrated execution windows are disproportionately affected. The absence of any capacity estimate renders a backtest incomplete: the strategy may be valid but untradeable at the scale required to justify the infrastructure investment.

- **Process discipline:** The strategy creation and testing process should follow a defined, repeatable pipeline that enforces a scientific and statistically valid methodology. This pipeline should specify the sequence of steps from hypothesis formulation through data preparation, in-sample fitting, out-of-sample validation, and robustness testing, with each step's criteria defined in advance rather than adjusted after the fact. The robustness tests and evaluation metrics employed must be appropriate to the specific class of strategy being tested and aligned with what the researcher is attempting to prove or disprove. A trend-following strategy, for example, demands different robustness criteria than a mean-reversion strategy, and a strategy intended for a single instrument requires different validation than one designed for a diversified portfolio. The absence of a structured evaluation process is itself a red flag: if the researcher cannot describe the pipeline that produced their results, the results should be treated with corresponding scepticism.

11 Conclusion

The current state of retail backtesting practice is characterised by a significant gap between the apparent sophistication of the tools and the actual rigour of the analysis they produce. The Python ecosystem has made it easy to generate a backtested equity curve, but the resulting curves are misleading.

The errors are not random. They are systematically biased toward overstating performance. Naive fill assumptions that ignore adverse selection and limit halts, understated transaction costs including roll friction and the dynamic behaviour of margin, contaminated data that diverges between vendors and is silently rewritten over time, the failure to model exchange rate effects on cross-currency positions, the implicit assumption that the broker and exchange are always there, and the fundamental inability of coarse bar data to resolve intraday order execution collectively produce results that are more optimistic than achievable reality.

There is also a class of failure that operates on the reporting side of the backtest rather than the simulation side. Pre-tax returns can overstate the after-tax outcome by 30% or more for high-turnover strategies in a high marginal bracket. A nominal-currency equity curve says little about the home-currency balance the trader actually holds. Counterparty risk is silently assumed away by every backtest that does not apply a haircut for the historical base rate of venue failure. These gaps do not require a more sophisticated simulation engine to address. They require the researcher to acknowledge that the backtest output is a different object from the live trader's experience, and to translate from one to the other.

The bar resolution problem deserves particular emphasis because it is so widely underestimated. The intuition that longer-term strategies are immune to intraday data requirements is not entirely wrong (strategies can be designed to work within the constraints of daily data) but it is seriously incomplete. Any strategy that interacts with the market through price-contingent orders, which includes virtually all strategies that employ risk management, is subject to the OHLC sequence ambiguity and the stop-loss concealment problem described in Section 4. The severity of these problems scales directly with bar size: negligible at one-minute resolution, modest at hourly resolution, and potentially disqualifying at the daily level. Daily data for such strategies does not merely introduce noise; it introduces a directional bias that systematically flatters performance. The trader who uses daily data responsibly must design their strategy to avoid dependence on intrabar order resolution, and must acknowledge the resulting limitations in their performance claims.

Layered atop these mechanical problems are statistical methodology failures that afflict even technically correct backtests: insufficient sample sizes, uncontrolled multiple testing, and the failure to account for regime dependence. The survivorship bias inherent in selecting apparently successful strategies from a large universe of tested variants (whether that universe is generated deliberately through combinatorial search or implicitly through iterative refinement) further inflates reported performance. The combination of mechanical and statistical errors means that a backtest must clear a very high bar before it constitutes meaningful evidence of a tradable edge.

The robustness testing methods commonly applied to address these concerns (Monte Carlo trade shuffling, synthetic data generation, and walk-forward analysis) provide less protection than is generally assumed. Monte Carlo trade shuffling violates the independence assumption for any strategy with path-dependent position sizing or portfolio-level risk filters, producing confidence intervals derived from trade sequences that could never have occurred. Synthetic data methods either destroy the statistical properties the strategy was designed to exploit or test generalisation within the same distribution rather than across genuinely novel conditions. Walk-forward analysis, while the most methodologically sound of the three and an essential minimum standard, is vulnerable to meta-overfitting and can only validate against conditions present in the historical record. The trader who treats a successful walk-forward pass as conclusive evidence of robustness has confused a necessary condition with a sufficient one.

Beyond the technical deficiencies, the ecosystem surrounding retail backtesting contributes to poor outcomes in ways that are less obvious but equally consequential. The self-reinforcing dominance of Python channels traders toward a particular set of tools and conventions that, while accessible, encourage methodological homogeneity and obscure the disciplines of risk management and portfolio construction that are essential to converting a trading signal into a viable system. The novice trader who emerges from this ecosystem is equipped with the ability to generate impressive-looking equity curves but is critically underprepared for the realities of live trading, including the psychological challenge of enduring drawdowns that are almost certainly deeper than those shown in the backtest.

I am not arguing that Python (or any other language) is without value for backtesting. Python remains an excellent tool for exploratory research and the preliminary stages of strategy development. Nor am I arguing that daily bar data is inherently unusable; strategies can be designed around the constraints of coarse bars, provided those constraints are understood and respected. But I do believe that easy access to a programming language and its trading libraries is sending many aspiring traders down a disappointing path,

because it encourages them to believe that the programming is the hard part and the trading will follow naturally.

The reality is the reverse.

Programming is the accessible part. The trading competencies that determine success (risk management, statistical literacy, cost awareness, psychological discipline, and the judgment to distinguish a genuine edge from a statistical artefact) are far harder to acquire.

A carefully constructed simulation, built on high-quality data with realistic execution modelling and rigorous statistical methodology, remains an essential tool for strategy development.

But, the distance between a credible backtest and what is commonly produced is enormous. The ease with which the latter is generated has created a false sense of confidence that is, in aggregate, likely destroying more capital than it creates.

References

1. Almgren, R., & Chriss, N. (1999). Optimal execution of portfolio transactions. *Journal of Risk*, *3*(2), 5–39.

2. Bailey, D. H., & Lopez de Prado, M. (2014). The deflated Sharpe ratio: Correcting for selection bias, backtest overfitting, and non-normality. *The Journal of Portfolio Management*, *40*(5), 94–107.

3. Bailey, D. H., Borwein, J. M., Lopez de Prado, M., & Zhu, Q. J. (2017). The probability of backtest overfitting. *Journal of Computational Finance*, *20*(4), 39–69.

4. Bailey, D. H., Borwein, J. M., Lopez de Prado, M., & Zhu, Q. J. (2014). Pseudo-mathematics and financial charlatanism: The effects of backtest overfitting on out-of-sample performance. *Notices of the American Mathematical Society*, *61*(5), 458–471.

5. Concretum Group. (2026). Can you trust your intraday database? *Concretum Group Substack*. https://concretumgroup.substack.com/p/can-you-trust-your-intraday-database

6. Cont, R., Stoikov, S., & Talreja, R. (2010). A stochastic model for order book dynamics. *Operations Research*, *58*(3), 549–563.

7. Garman, M. B., & Klass, M. J. (1980). On the estimation of security price volatilities from historical data. *The Journal of Business*, *53*(1), 67–78.

8. Harvey, C. R., Liu, Y., & Zhu, H. (2016). …and the cross-section of expected returns. *The Review of Financial Studies*, *29*(1), 5–68.

9. Hansen, P. R. (2005). A test for superior predictive ability. *Journal of Business & Economic Statistics*, *23*(4), 365–380.

10. Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, *47*(2), 263–291.

11. Lopez de Prado, M. (2018). *Advances in financial machine learning*. John Wiley & Sons.

12. Parkinson, M. (1980). The extreme value method for estimating the variance of the rate of return. *The Journal of Business*, *53*(1), 61–65.

13. Quantreo. (2025). Stop trusting your backtests until… *Quantreo Newsletter*. https://www.newsletter.quantreo.com/p/stop-trusting-your-backtests-until

14. White, H. (2000). A reality check for data snooping. *Econometrica*, *68*(5), 1097–1126.

15. Wiecki, T., Campbell, A., Lent, J., & Stauth, J. (2016). All that glitters is not gold: Comparing backtest and out-of-sample performance on a large cohort of trading algorithms. *The Journal of Investing*, *25*(3), 69–80.

16. Carver, R. (2023). *Advanced futures trading strategies: 30 fully tested strategies for multiple trading styles and time frames*. Harriman House. See also the `pysystemtrade` framework and Carver's blog, https://qoppac.blogspot.com, on back-adjusted prices and the denominator used for percentage returns.