Digital Finance v4: Automation & Intelligence

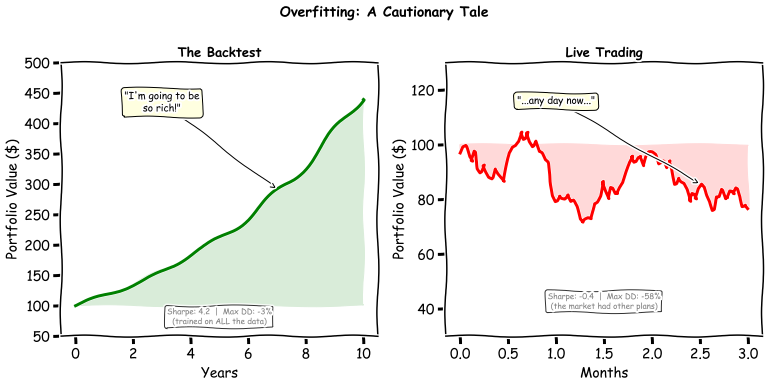
Lesson 5.3: The Limits of Prediction – Time Series, NLP, and Market Efficiency

FHGR

April 14, 2026

If ML is so powerful, why can't we predict stock prices? Understanding the limits matters as much as understanding the tools.

My Backtest Made Millions!



The gap between backtest performance and live trading is the most expensive lesson in quantitative finance.

Learning Objectives

By the end of this lesson, you will be able to:

1. Explain why most ML trading strategies fail out-of-sample [Understand]
2. Identify common pitfalls in financial ML (data snooping, look-ahead bias, survivorship bias) [Understand]
3. Apply the Efficient Market Hypothesis to evaluate prediction claims [Apply]
4. Design a sentiment analysis pipeline for financial text using NLP techniques [Create]
5. Evaluate the practical limitations of reinforcement learning for trading [Evaluate]

Bloom's levels: Understand (1,2), Apply (3), Evaluate (5), Create (4). Critical thinking about prediction claims protects you from costly overconfidence.

From LLMs to Market Prediction

Bridge from Lesson 5.2:

- LLMs are powerful for understanding and generating text
- NLP can extract sentiment, entities, and relationships from financial documents
- **But can any model – no matter how advanced – reliably predict markets?**

The central tension of this lesson:

- ML excels at pattern recognition in structured, stationary data
- Financial markets are *non-stationary*, *adversarial*, and *reflexive*
- Understanding **why prediction is hard** is as valuable as any algorithm

---

Markets are not image classifiers – the data generating process changes because participants learn.

Key Definition: Stationarity (NEW)

[colback=mllavender!10, colframe=mlpurple, title=**Stationarity**] A time series $\{y_t\}$ is **stationary** if its statistical properties — mean, variance, and autocorrelation — do not change over time. Formally, the joint distribution of $(y_t, y_{t+1}, \ldots, y_{t+k})$ is identical to $(y_{t+\tau}, y_{t+\tau+1}, \ldots, y_{t+\tau+k})$ for any time shift τ.

Three requirements:

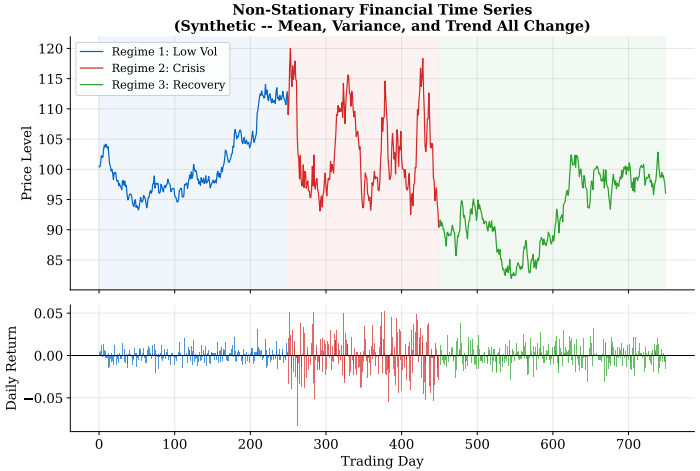
1. **Constant mean:** $E[y_t] = \mu$ for all t
2. **Constant variance:** $\text{Var}(y_t) = \sigma^2$ for all t
3. **Constant autocorrelation:** $\text{Cov}(y_t, y_{t-k})$ depends only on lag k, not on t

Why ML models fail on non-stationary data:

- Most ML algorithms assume training and test data are independent and identically distributed (i.i.d.)
- Financial prices violate this: trends, volatility clusters, regime shifts
- A model trained on Regime 1 has never seen Regime 2 patterns — it fails catastrophically

Stationarity is the foundational assumption most financial ML models quietly violate. Returns are closer to stationary but still exhibit structural breaks.

Non-Stationary Financial Time Series



- **What you see:** Three distinct regimes with different means, volatilities, and trends—Regime 1 (stable growth), Regime 2 (crisis), Regime 3 (recovery)

What Is Autocorrelation?

Autocorrelation measures how much today's value depends on yesterday's.

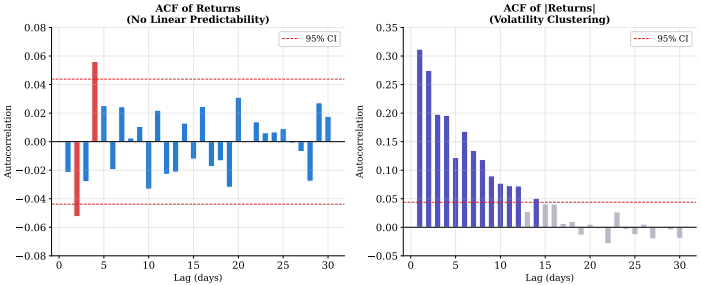
- **ACF (Autocorrelation Function):** correlation between r_t and r_{t-k} at lag k
- **Financial returns:** near-zero autocorrelation (consistent with weak-form EMH)
- **Absolute returns:** significant positive autocorrelation (**volatility clustering**)

Implication for ML:

- Linear prediction of next-day returns from past returns is nearly impossible
- Volatility *is* predictable – Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models exploit this
- ML models that claim to predict return *direction* should be treated with skepticism
- Predicting *volatility* (risk) is more achievable than predicting *direction* (alpha)

Returns are nearly unpredictable; volatility is clustered and forecastable – this asymmetry defines what ML can and cannot do.

Autocorrelation: Returns vs Absolute Returns



- **What you see:** Left plot shows return ACF near zero (within blue confidence band); right plot shows —return— ACF persistently high (outside band)
- **Key pattern:** Returns are unpredictable (consistent with weak-form EMH), but volatility is forecastable (volatility clustering)
- **Takeaway:** Predicting return *direction* is nearly impossible; predicting *volatility* (risk) is achievable—GARCH models exploit this asymmetry

Left: return ACF is within the confidence band (no linear predictability). Right: —return— ACF is highly significant (volatility clustering).

What Is a Regime Change?

A regime change is an abrupt shift in the statistical behavior of markets.

- **Volatility regime:** calm (VIX $\sim$12) $\rightarrow$ crisis (VIX $\sim$80)
- **Correlation regime:** diversified $\rightarrow$ "everything sells off together"
- **Monetary regime:** zero rates (2009–2021) $\rightarrow$ tightening (2022–2024)
- **Structural break:** new regulation, technology, or market microstructure change

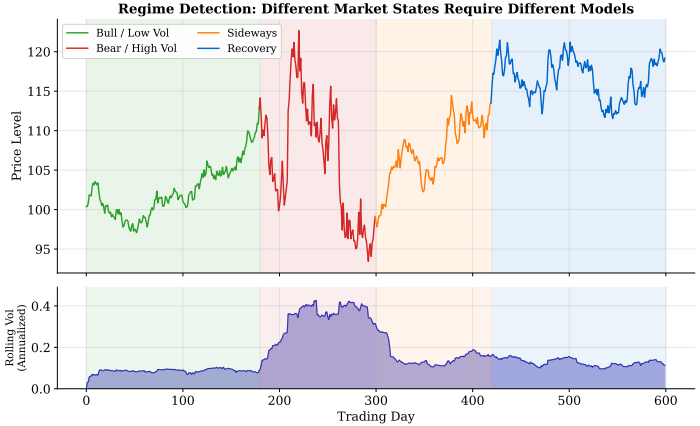
Why regime changes break models:

- Training data comes from one regime; deployment hits another
- Correlations, volatilities, and factor loadings all shift simultaneously
- No amount of in-sample testing protects against out-of-distribution events

---

Models trained on 2015–2019 (low vol, low rates) were blindsided by both COVID (2020) and the rate shock (2022).

Regime Detection: Different Markets Need Different Models



A single model cannot learn all regimes; regime-aware ensembles or online adaptation are needed.

What Is Data Snooping?

Data snooping is testing many strategies on the same data and reporting only the winners.

- Test 1,000 trading rules on 20 years of S&P 500 data
- By chance, $\sim$50 will be "significant" at $p < 0.05$
- Reporting the best one without correcting for multiple testing is **data snooping**

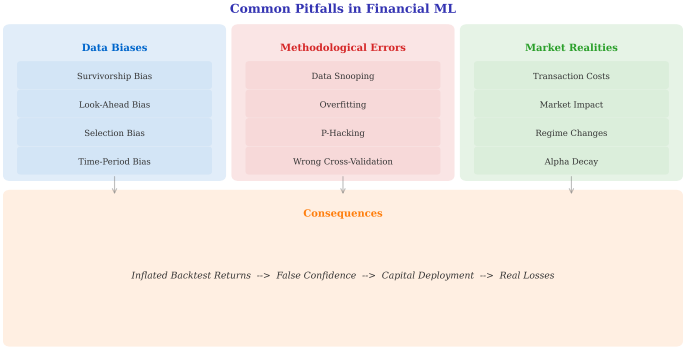
Related pitfalls:

- **Look-ahead bias:** using future information unavailable at decision time (e.g., restated earnings)
- **Survivorship bias:** backtesting only on companies that still exist (excludes bankruptcies)
- **P-hacking:** tweaking model until $p < 0.05$, then calling it "discovery"

Harvey et al. (2016) estimate that 50%+ of published finance factors are false discoveries.

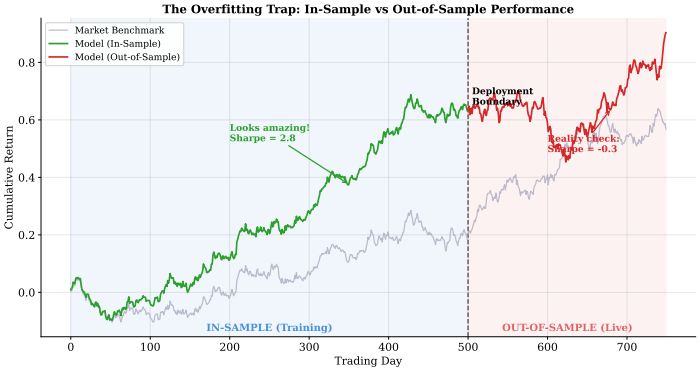
If you torture the data long enough, it will confess to anything – but the confession is meaningless out-of-sample.

Taxonomy of ML Trading Pitfalls



Pitfalls span data collection, methodology, and market realities – all lead to the same outcome: real losses.

The Overfitting Trap



In-sample Sharpe of 2.8 collapsed to -0.3 out-of-sample – the model memorized noise, not signal.

What Is the Efficient Market Hypothesis (EMH)?

EMH (Fama, 1970): prices fully reflect available information.

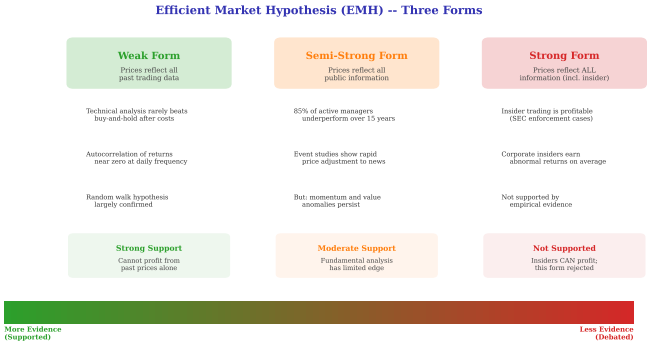
- **Weak form:** prices reflect all past price and volume data
  - Implication: technical analysis cannot systematically profit
- **Semi-strong form:** prices reflect all publicly available information
  - Implication: fundamental analysis and news trading have limited edge
- **Strong form:** prices reflect all information, including insider knowledge
  - Implication: even insiders cannot profit (empirically rejected)

Modern view – Adaptive Market Hypothesis (Lo, 2004):

- Markets are *adaptively* efficient – inefficiencies exist but are competed away
- Alpha is scarce, temporary, and capacity-constrained

85% of active managers underperform their benchmark over 15 years (S&P SPIVA 2023). EMH is approximately right.

EMH: Evidence Spectrum



Weak form is well-supported; semi-strong is debated; strong form is rejected by insider trading evidence.

What Is Sentiment Analysis in Finance?

Sentiment analysis uses NLP to extract subjective opinion from text.

- **Sources:** news articles, SEC filings (10-K, 10-Q), earnings call transcripts, social media
- **Methods:** dictionary-based (Loughran-McDonald), ML-based (FinBERT, LLM prompting)
- **Output:** positive / negative / neutral score per document or entity

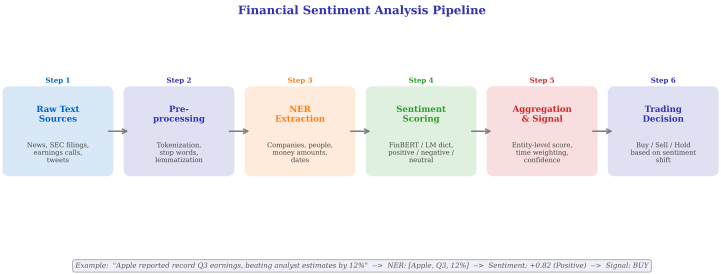
Key NLP components:

- **Named Entity Recognition (NER):** identify companies, people, amounts, dates
- **Earnings surprise:** compare actual vs. consensus estimate – sentiment shift matters
- **Temporal aggregation:** recent sentiment weighted more than old sentiment
- **Sentiment dictionaries:** Loughran-McDonald (a sentiment dictionary designed for financial documents) captures finance-specific tone better than general dictionaries

Limitation: Sentiment signals are noisy, crowded, and decay rapidly once widely traded.

FinBERT achieves ∼85% accuracy on financial sentiment classification, but accuracy $\neq$ profitable trading signal.

Financial Sentiment Analysis Pipeline



Each step introduces potential errors: NER misidentifies entities, sentiment scores are context-dependent, aggregation loses nuance.

What Is Alpha Decay?

Alpha decay: a profitable trading signal loses its edge over time.

- **Discovery phase:** signal is proprietary, generates excess returns
- **Publication phase:** academic paper or word-of-mouth spreads the idea
- **Crowding phase:** many funds trade the same signal, compressing returns
- **Death phase:** signal no longer covers transaction costs after crowding

Alpha half-life estimates:

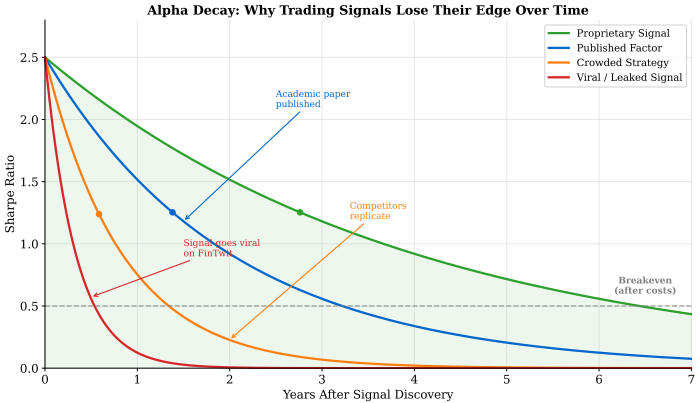
- Proprietary HFT signal: weeks to months
- Published academic factor (e.g., momentum): 3–7 years
- Viral social media signal: days to weeks

"Alpha is not a stock; it's a decaying asset." – Marcos López de Prado

---

Once a signal is known, rational actors arbitrage it away – the Adaptive Market Hypothesis in action.

Alpha Decay: How Signals Lose Their Edge



Proprietary signals decay slowly; published or crowded signals decay rapidly. All eventually approach the breakeven line.

What Is Reinforcement Learning for Trading?

RL trains an agent to maximize cumulative reward through trial and error.

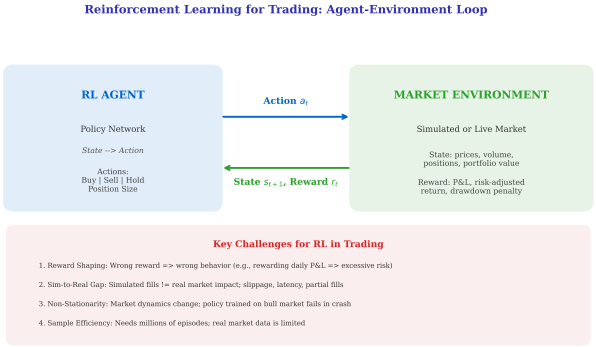
- **State:** current prices, portfolio positions, market indicators
- **Action:** buy, sell, hold; position sizing
- **Reward:** profit/loss, risk-adjusted return, drawdown penalty
- **Policy:** neural network mapping states to actions

Key challenges for RL in finance:

- **Reward shaping:** wrong reward $\Rightarrow$ wrong behavior (e.g., excessive risk-taking)
- **Simulation-to-reality gap:** simulated market $\neq$ real market (slippage, partial fills)
- **Non-stationarity:** policy learned in one regime fails in the next
- **Sample efficiency:** RL needs millions of episodes; real financial data is scarce

RL has succeeded in games (AlphaGo, Atari) but markets are adversarial, non-stationary, and partially observable.

RL Agent–Environment Loop for Trading



The sim-to-real gap is the Achilles heel of RL in trading: what works in simulation rarely transfers to live markets.

Proper Validation: Walk-Forward Testing

Standard cross-validation (random splits) is wrong for time series.

- Random splits create look-ahead bias (future data leaks into training)
- **Walk-forward validation:** train on past, test on next period, slide window forward

Walk-forward protocol:

1. Train on months 1–12, test on month 13
2. Train on months 1–13, test on month 14
3. Repeat until all data is used
4. Report average *out-of-sample* performance across all test periods

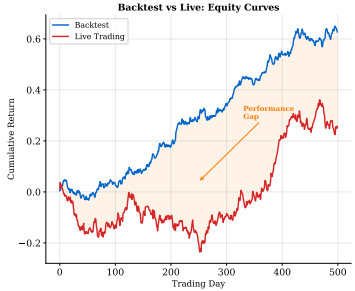
Additional safeguards:

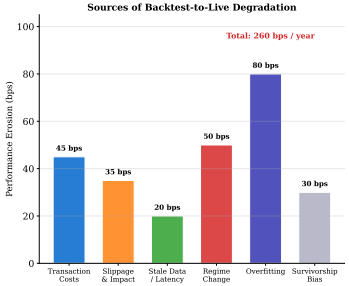
- **Purging:** remove training samples near the test boundary (avoid label leakage)
- **Embargo:** leave a gap between training and test periods
- **Purged cross-validation (a method that prevents data leakage in time series):** López de Prado (2018) – multiple test paths, remove overlapping samples

---

Walk-forward validation is the gold standard for time series ML – never use random train/test splits on financial data.

Backtest vs Live Trading





Transaction costs, slippage, regime change, and overfitting together erode 260 bps/year of backtest alpha.

Case Study: Long-Term Capital Management (LTCM)

Nobel-Prize-winning models, $4.6B loss (1998):

- **Team:** Myron Scholes & Robert Merton (Nobel 1997), top Wall Street traders
- **Strategy:** convergence trades – long undervalued bonds, short overvalued
- **Leverage:** 25:1 amplified small statistical edges into massive positions
- **What went wrong:** Russian default triggered global flight-to-quality
- **Model failure:** assumed normal distributions; ignored tail risk and liquidity
- **Rescue:** Fed-orchestrated $3.6B bailout by 14 banks to prevent systemic contagion

Lesson: Even the most sophisticated quantitative models fail when:

- Tail events exceed historical experience
- Leverage amplifies losses beyond recovery
- Liquidity vanishes when you need it most

"When Genius Failed" (Lowenstein, 2000) – the definitive account of model overconfidence in practice.

Realistic Performance Expectations

What is achievable in practice?

Metric	Good	Suspicious
Sharpe Ratio	0.8–2.0	> 3.0
Annual Return	8–25%	$> 50\%$
Max Drawdown	-15% to -30%	$< -5\%$
Win Rate	50–55%	$> 70\%$

Red flags in backtest results:

- Sharpe > 3.0 – likely overfitting or look-ahead bias
- Zero losing months – model memorized the data
- No drawdowns – unrealistic or missing transaction costs
- "100% accuracy" – data leakage or survivorship bias

The median quant hedge fund Sharpe ratio is 0.7 (2010–2023, BarclayHedge). Humility is a competitive advantage.

Survivorship Bias: The Invisible Error

What is survivorship bias?

- Analyzing only entities that "survived" (still exist) and ignoring those that failed
- Backtest on current S&P 500 members? You exclude companies that went bankrupt
- Hedge fund database? Only funds that chose to report (losers stop reporting)

Impact:

- S&P 500 backtest with survivorship bias overstates returns by $\sim$1–2% per year
- Hedge fund database returns are inflated by $\sim$3–5% per year (instant history + backfill)
- Mutual fund "track records" omit merged or closed funds (poor performers disappear)

Mitigation:

- Use point-in-time databases (constituents as of each date)
- Include delisted securities with their full return history
- Report results with and without survivorship correction

Abraham Wald's WW2 bomber analysis is the classic survivorship bias example: reinforce where the missing planes were hit.

Named Entity Recognition and Earnings Surprise

What is Named Entity Recognition (NER)?

- NLP technique that identifies and classifies entities in text:
  - Organizations: "Apple Inc.", "Deutsche Bank"
  - People: "Jamie Dimon", "Jerome Powell"
  - Money: "$4.2 billion", "12% growth"
  - Dates: "Q3 2025", "fiscal year"
- Critical for mapping sentiment to the *right* company or event

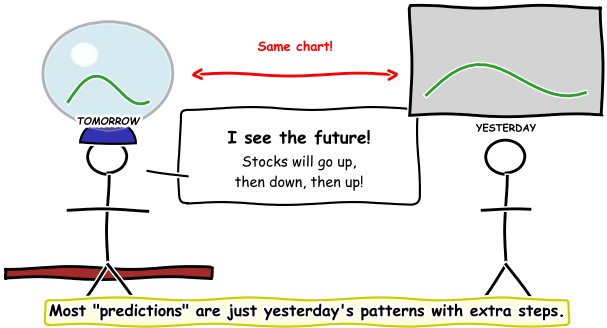
What is earnings surprise?

- Difference between actual earnings and analyst consensus estimate
- **Positive surprise:** actual $>$ estimate $\Rightarrow$ typically bullish price reaction
- **Post-earnings announcement drift (PEAD):** prices continue drifting for weeks
- One of the most robust anomalies – but increasingly traded and decaying

---

NER + sentiment + earnings surprise = a complete event-driven NLP pipeline for financial text.

One Last Thought...



Sometimes the best way to remember a concept is to laugh about it.

Key Takeaways

Core principles:

1. Financial time series are **non-stationary** – stationarity assumptions break down
2. Returns have near-zero autocorrelation; **volatility** is predictable, direction is not
3. **Data snooping**, look-ahead bias, and survivorship bias inflate backtest performance
4. The EMH is approximately right: **alpha is scarce, temporary, and capacity-constrained**
5. Sentiment analysis with NLP is powerful but noisy and subject to **alpha decay**
6. RL for trading faces the **sim-to-real gap** – simulated markets $\neq$ live markets

Practical advice:

- Use walk-forward validation, never random splits
- Include transaction costs, slippage, and market impact in every backtest
- If results look too good to be true, they almost certainly are

---

Understanding why prediction is hard is the first step toward building models that actually work.

Summary: The Limits of Prediction

What ML Can Do:

- Forecast volatility and risk
- Extract sentiment from text
- Detect anomalies and fraud
- Automate data processing
- Identify regime changes (after the fact)

What ML Cannot Do:

- Reliably predict price direction
- Avoid regime changes
- Replace risk management
- Guarantee alpha at scale
- Eliminate the need for human judgment

Next Lesson: 5.4 – Automation in Practice (putting all tools together in production)

The most valuable skill in financial ML is knowing when not to trust the model.

Appendix: Survivorship Bias in the Sharpe Numbers You Just Read

This lecture cited Sharpe ratios, fund returns, and quant-hedge-fund medians. Every one of those numbers is itself survivorship-biased. Find the bias before you trust the number.

The base rates the survivor numbers hide

- **Active US equity funds, 15-year horizon (2009–2024):** only 11.6% beat the S&P 500 (*S&P Dow Jones SPIVA US Scorecard Year-End 2024, Exhibit 3*, 2024) — i.e., **88% underperformed, many were liquidated or merged away**
- **Fund mortality:** ~55% of US equity funds active in 2004 were liquidated or merged by 2024 (*S&P SPIVA Persistence Scorecard 2024, Exhibit 2*, 2024); the "15-year average return" chart you see excludes them
- **Hedge funds:** median lifespan ~5 years (*Preqin Global Hedge Fund Report 2023; HFR Industry Report Q4 2023*, 2023); funds below $100M AUM shut before their track record hits a database
- **Backfill bias:** hedge-fund databases add a fund only *after* it opts in, and let it back-fill past returns $\Rightarrow$ reported indices overstate returns by 2–5% per year (*Bhardwaj, Gorton & Rouwenhorst, Rev. Financial Studies, 2011*, 2011)

The ironic examples in this very lecture:

- *"Median quant hedge fund Sharpe is 0.7"* — that median is over funds still *reporting* in 2023. The funds that blew up (LTCM 1998, Amaranth 2006, Long-Term Asset 2020) exit the sample.
- *"85% of active managers underperform SPIVA 15-yr"* — SPIVA itself corrects for survivorship; the often-quoted *industry-brochure* 10-year figure does not.
- *"Renaissance Medallion 66% pre-fee"* — this is the archetypal survivor. For every Medallion, there are hundreds of funds with identical stated strategies that closed silently.

Classroom exercise: take any chart in this lecture that shows an "average quant return." Ask: which funds had to exist in both the start year and end year to be counted? What happened to the ones that didn't? The answer is almost always "liquidation, no press release."

Wald's bomber lesson applies to quant finance: the strategies you can study are the ones that didn't get shot down. The strategies that got shot down are

Common Misconceptions About Predicting Markets

Misconception	Reality
"A 60% win rate means the strategy is profitable"	Win rate is meaningless without average win / average loss. A 60% win rate with 1:5 payoff is a losing strategy.
"Backtest Sharpe 2.5 = deployable edge"	Most reported backtest Sharpes reflect: (1) in-sample tuning, (2) survivorship bias, (3) look-ahead leaks. De Prado's "Deflated Sharpe" adjusts for trial count.
"Reinforcement learning can learn to trade"	RL requires millions of episodes. Real market history has $\sim$100 years of daily data. RL in trading typically overfits to simulator artifacts.
"LLMs extract novel alpha from news"	Any signal that arrives in structured news is priced in within seconds for liquid names. LLMs help with *synthesis* (compliance, research summaries), not alpha.
"The Efficient Market Hypothesis is refuted by Buffett"	EMH is a statement about the marginal investor. Buffett's alpha comes from leverage + low-volatility anomaly + permanent capital — consistent with behavioural-finance extensions of EMH.

The hardest part of quant finance is not building the model — it is not fooling yourself about how good the model is.

Self-Check: Limits of Prediction

Attempt these before turning the page.

1. [Understand] Define look-ahead bias, survivorship bias, and data-snooping bias. Give an example of each in an equities backtest.
2. [Apply] You test 100 strategies on the same data. Under the null (no edge), how many are expected to show t-stat > 2.0 by chance? Apply Bonferroni correction — what t-stat threshold is needed for 5% family-wise error?
3. [Evaluate] A startup claims Sharpe 3.0 on 2020–2023 crypto backtest using their ML model. You are the institutional allocator. Describe the two tests you would run before committing capital.

Solutions hidden unless `\solutionstrue` is set before compiling.