**Backtesting Data Stability:**

What Is Backtesting?

Backtesting is the process of running your trading strategy on historical data to see how it would've performed.

What Is "Data Stability" in Backtesting?

Data stability means the historical data you use is accurate, complete, and consistent. If it isn't, your strategy might:

- Look profitable in backtest but fail in real life
- Overfit to noise or faulty signals
- Miss real-world trading costs/slippage

## 🛑 Common Data Stability Issues

| Issue | Description | Consequence |
|---|---|---|
| **Missing data** | Gaps in time series (e.g., skipped candles) | False signals, missed trades |
| **Incorrect timestamps** | Out-of-order or misaligned timestamps | Misleading signal timing |
| **Survivorship bias** | Only including stocks that exist today | Overestimation of past returns |
| **Look-ahead bias** | Using data not available at that time | Unrealistic performance |
| **Inconsistent formats** | Price, volume, or fields are misaligned | Errors in indicators or execution logic |
| **No adjustments** | Not accounting for splits/dividends | Wrong entry/exit points |

## 🔐 How to Ensure Data Stability

### ✅ 1. Use Reliable Data Sources

- **Premium**: Polygon.io, Quandl (Nasdaq Data Link), TickData, Bloomberg
- **Free**: Yahoo Finance (via `yfinance`), Alpha Vantage, IEX Cloud (limited)

### ✅ 2. Verify Data Consistency

- Check for **null values**, **duplicate timestamps**, or **price anomalies**
- Use data validation routines before running backtests

### ✅ 3. Account for Stock-Specific Adjustments

- Use **adjusted close** (for splits/dividends)
- Filter out stocks that didn't exist during your backtest window (to avoid survivorship bias)

### ✅ 4. Log Trade Execution Time

- Simulate real-world delays
- Ensure you're not using **future candles** to make current decisions

### ✅ 5. Use Realistic Assumptions

- Include **commission, slippage, latency**
- Test under different market conditions (bull, bear, sideways)