Evaluating Basic Trading Strategies through Simulation and Historical Backtesting

By: Luca de Dios

Abstract

This project compares three simple trading strategies—coin flipping, momentum signals, and moving average crossovers—using both simulated and historical market data from 2022 to 2024. Simulations modeled different market conditions with a random walk, while real data tests covered TSLA, AAPL, BTC, and SPY. Results showed the 1-day momentum strategy performed best overall, while moving averages underperformed with low win rates. The analysis highlights the limitations of simplified models and suggests areas for improvement like including trading costs and dynamic sizing.

Strategy Introduction:

I chose three strategies to implement and compare through backtesting, using both a paper trading simulation and real historical data. The coin-flipping trading strategy served as a control: one side of the coin signaled a buy, while the other signaled a sell, with profit and loss calculated at each trade execution.

The next strategy was a momentum-based approach, where I measured momentum over various intervals to determine if there was a positive trend. If there were consecutive positive increases from the previous close, the strategy would buy; otherwise, it would sell. I tested 1-day, 2-day, and 3-day signals, and also compared the results within a bullish market context.

The final strategy was a moving average crossover strategy. I compared a short-term (5-day) average with a long-term (20-day) average. A buy signal was triggered when the short-term average crossed above the long-term average, and a sell signal when it crossed below.

Simulation Framework:

The price simulation used for the paper trading strategy was based on a discrete-time multiplicative random walk over a 100-day period. The update rule for each time step was:

$$P_{t+1} = P_t \cdot (1 + r_t), \qquad where\ r_t \sim U(-0.01, 0.03)$$

By default, r_t was sampled from a uniform distribution between -1% and +3% (i.e., U(−0.01, 0.03). This randomness — or stochasticity — introduces day-to-day variability that mimics the uncertainty and fluctuation seen in real markets.

To explore different market environments (e.g., more volatile, bullish, or bearish conditions), I varied the bounds of the uniform distribution:

- U(−0.05, 0.05) simulates higher volatility with zero expected drift
- U(−0.01, 0.03) introduces a bullish bias due to a positive average return
- U(−0.03, 0.01) simulates a bearish market with negative drift

The simulation begins with a base price of $100, and each day's price is calculated recursively using the sampled return. Over 100 days, this generates a full synthetic price path.

Using a multiplicative model (instead of an additive one) better reflects how financial assets behave:

- Returns are relative to price levels, not fixed in dollar amounts.
- Volatility scales with price, preserving realism over time.
- Percentage-based movement allows for easy adjustment across different simulations and asset types.

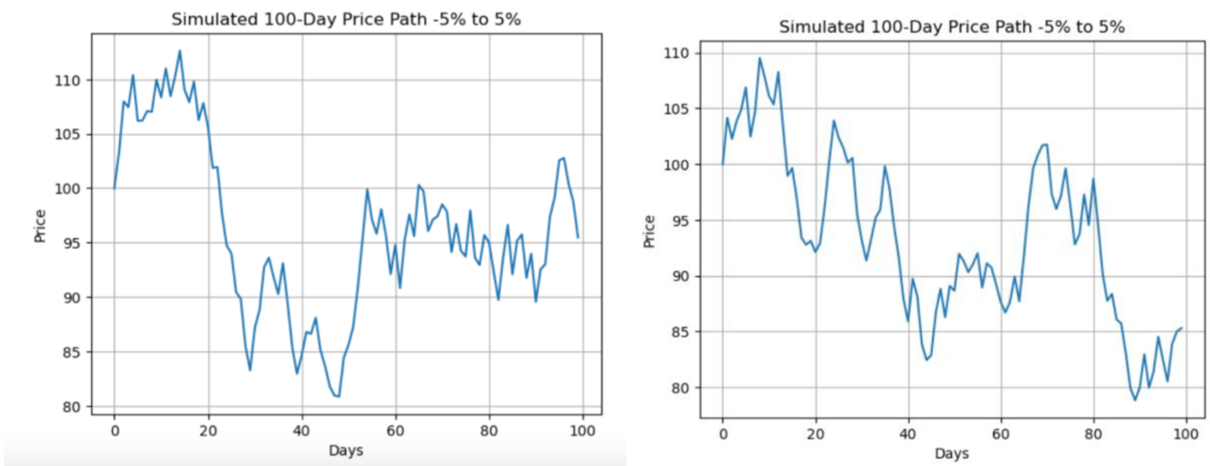

Figure 1: Simulated price path with returns drawn from U(−5%,+5%) **-** high volatility, no drift.

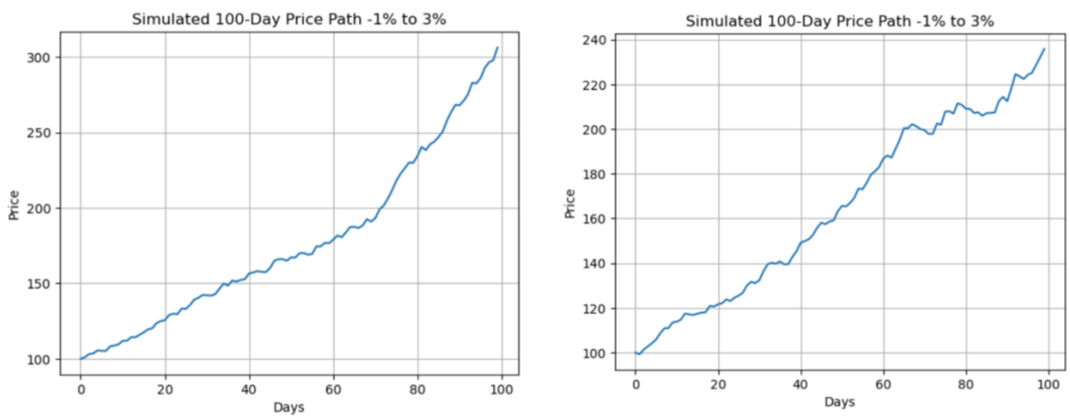

Figure 2: Simulated price path with returns from U(−1%,+3%) - upward drift with low volatility

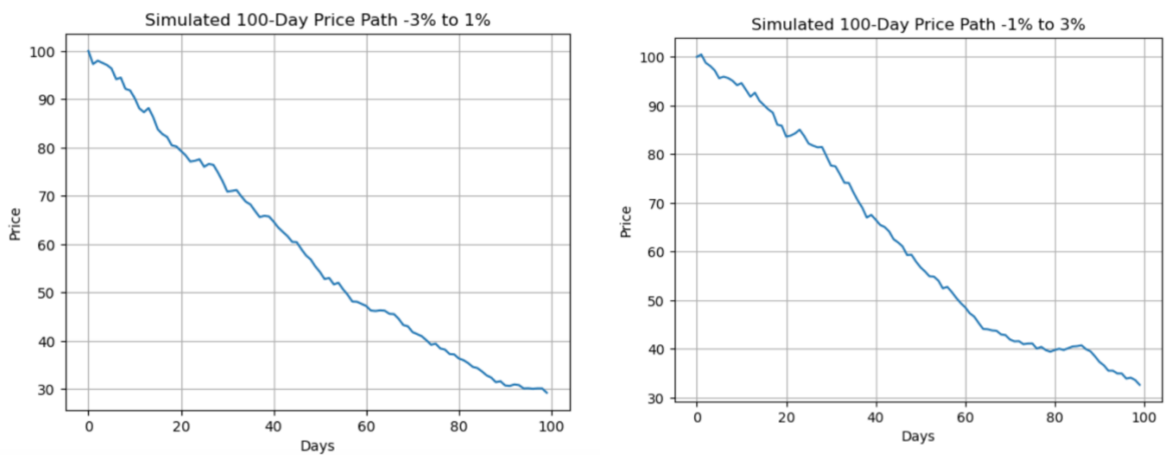

Figure 3: Simulated price path with returns from U(−3%,+1%) - downward drift with moderate volatility.

Strategy Logic:

Moving Average Crossover Strategy

Description:
This strategy looks for crossover points between a short-term moving average (5-day) and a long-term moving average (20-day). A buy signal is generated when the short-term average crosses above the long-term average, suggesting upward momentum. A sell signal is triggered when it crosses below.

Logic Explanation:

- Compare short-term and long-term averages at two time points: "before" and "now"
- If the short-term crosses above, buy → profit = next day's price increase
- If the short-term crosses below, sell → profit = next day's price drop
- Else, do nothing (no trade)

Pseudocode

```
short_prev = average(prices[i+15 : i+20])
long_prev  = average(prices[i : i+20])

short_now  = average(prices[i+16 : i+21])
long_now   = average(prices[i+1 : i+21])

if short_prev <= long_prev and short_now > long_now:
    pnl.append(prices[i+21] - prices[i+20])  # buy low, sell high
elif short_prev >= long_prev and short_now < long_now:
    pnl.append(prices[i+20] - prices[i+21])  # sell high, buy low
else:
    pnl.append(0.0)
```

Momentum Strategy (1-Day, 2-Day, and 3-Day Signals)

Description:
The momentum strategy aims to capture short-term trends by observing consecutive days of price increases. It assumes that upward momentum is likely to continue and responds accordingly. The strategy has three variations based on how many consecutive "up" days are required before entering a trade:

- 1-Day Signal: Buy if the last day showed a price increase.
- 2-Day Signal: Buy if the past two days both had price increases.
- 3-Day Signal: Buy only if the past three days were all up.

If the condition isn't met for a given signal strength, the strategy takes the opposite (sell) position instead.

Logic Explanation:

- Buy if the number of required consecutive upward moves is met
- Otherwise, sell
- Profit/loss is calculated based on next-day price movement

Pseudocode

```
# prices: list of floats
# n: number of consecutive "up" days to trigger a buy

for i in range(n, len(prices) - 1):
    if all(prices[i - j] > prices[i - j - 1] for j in range(n)):
        pnl.append(prices[i + 1] - prices[i])  # Buy
    else:
        pnl.append(prices[i] - prices[i + 1])  # Sell
```

Coin Flipping Strategy (Random)

Description:
This is a random baseline strategy. Each day, it randomly chooses to buy or sell with equal probability, with no relation to market behavior.

Logic Explanation:

- Flip a coin
  - Heads → buy → profit = price[i+1] - price[i]
  - Tails → sell → profit = price[i] - price[i+1]

<u>Pseudocode</u>

```
action = random.choice(["buy", "sell"])

if action == "buy":

    pnl.append(prices[i+1] - prices[i])

else:

    pnl.append(prices[i] - prices[i+1])
```

Profit Calculation

Description:
Each strategy executes trades based on buy or sell signals. Profits are calculated based on the change in price from the day of the signal to the next day. Every trade assumes a fixed size of 1 unit. The model ignores market frictions for simplicity.

Model:
Let P_t be the price at time t. Then the profit per trade is:

$$Profit = \begin{cases} P_{t+1} - P_t, & if\ Buy \\ P_t - P_{t+1}, & if\ Sell \end{cases}$$

Assumptions:

- Trades are executed at market close on day t, exited at market close on day t+1
- Fixed trade size of 1 unit (no leverage or compounding)
- No transaction costs
- No slippage
- Full execution (no partial fills)

Results & Visuals:

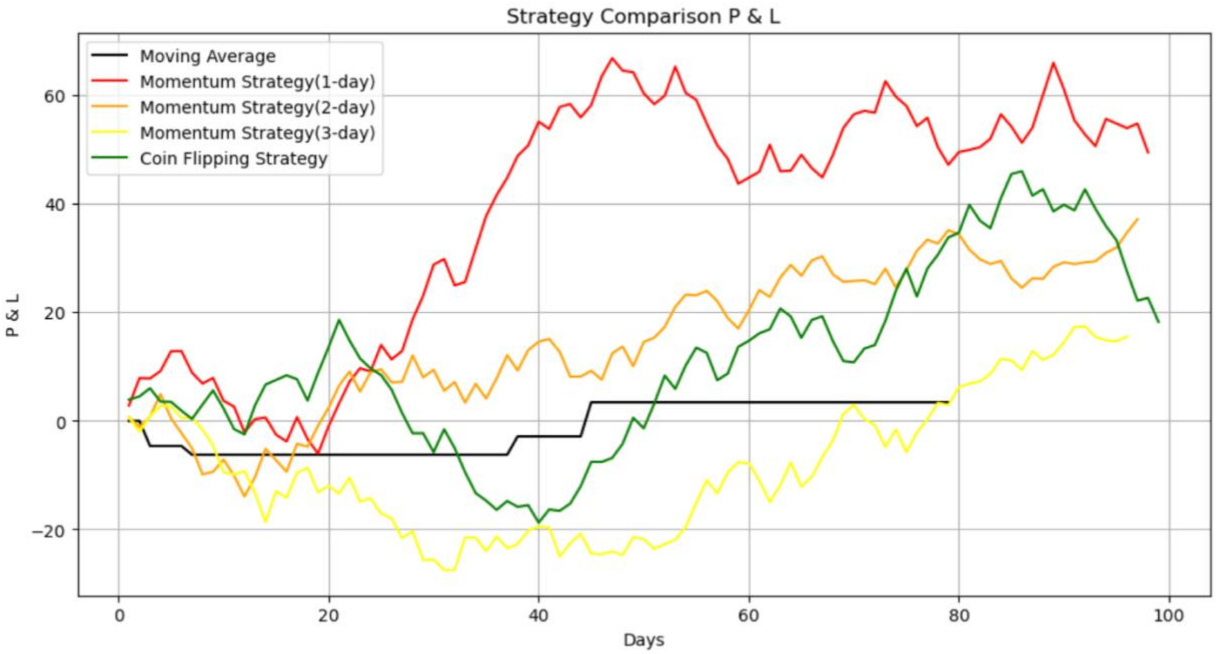

Figure 4: Example cumulative P&L trajectories for each strategy based on a single simulated price path. Results vary with each run due to randomness in the price simulation.

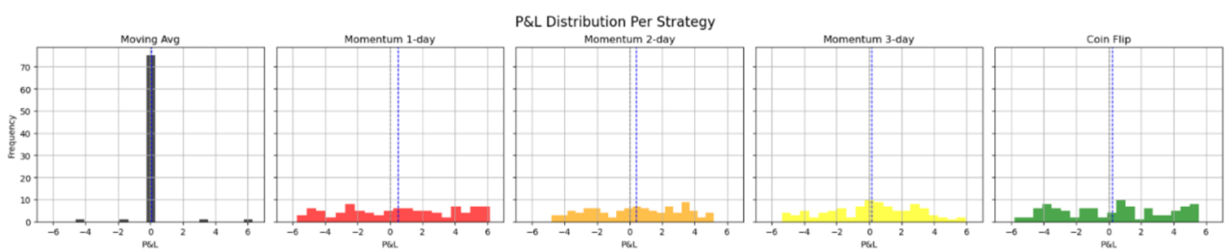

Figure 5: Histogram of per-trade P&L for each strategy from a single simulation run, showing differences in trade frequency, variance, and distribution shape across strategies.

Statistical Evaluation:

Used Monte Carlo simulation to model the probability and distribution of different price paths across five trading strategies over a 100-day period. Each strategy was run 10,000 times for statistical stability, and key metrics (Mean, Standard Deviation, Sharpe Ratio, Number of Trades, and Win Rate) were averaged and compiled into the summary table below.

Strategy	Mean	Std. Deviation	Sharpe Ratio	Trades	Win Rate (%)
Moving Average	0.0079	0.06949	0.0106	79.0	3.3804
Momentum 1-day	0.0365	2.8464	0.0154	98.0	50.8262
Momentum 2-day	-0.0076	2.8413	0.0050	97.0	49.6188
Momentum 3-day	-0.0057	2.8431	0.0100	96.0	49.6042
Coin Flip	-0.0008	2.8452	0.0005	99.0	50.1516

Back testing on Historical Data:

The strategies were tested on real historical data from January 2022 to December 2024 across four assets: TSLA, AAPL, BTC, and SPY. All data was sourced using the yfinance module.

TSLA Stock

Strategy	Mean	Std. Deviation	Sharpe Ratio	Trades	Win Rate (%)
Moving Avg	0.0075	1.9207	0.0039	731.0	3.1500
Momentum 1-day	0.4300	9.3793	0.0458	750.0	51.7300
Momentum 2-day	0.1923	9.3635	0.0205	749.0	50.8700
Momentum 3-day	0.0770	9.3671	0.0082	748.0	50.4000
Coin Flip	-0.0193	13.1365	-0.0016	99.0	50.4045

- Momentum 1-day had the highest mean P&L and a positive Sharpe ratio, which suggests some predictive power — though its volatility is very high (std dev ≈ 9.38).
- Momentum 2-day and 3-day decay in effectiveness → lower mean and Sharpe.
- Moving Average had very low win rate (3.15%), meaning most trades are losing, even if the mean P&L is slightly positive.
- Coin Flip is nearly random, but performs similarly to 2-day and 3-day momentum, which might indicate overfitting or lack of signal in those strategies for TSLA.

AAPL Stock

Strategy	Mean	Std. Deviation	Sharpe Ratio	Trades	Win Rate (%)
Moving Avg	0.0098	0.5038	0.0195	731.0	2.1900
Momentum 1-day	0.0041	2.8388	0.0014	750.0	51.0700
Momentum 2-day	-0.1329	2.8368	-0.0469	749.0	46.8600
Momentum 3-day	-0.1317	2.8368	-0.0464	748.0	45.9900
Coin Flip	-0.0354	3.4625	-0.0103	99.0	49.3024

- All strategies performed poorly, with low or negative Sharpe ratios, and Moving Avg has an extremely low win rate (2.19%).
- Momentum 2-day and 3-day had negative mean and Sharpe — they underperformed a Coin Flip, which is already near neutral.
- Momentum 1-day is barely positive (mean = 0.0041), with a near-zero Sharpe (0.0014).

SPY Stock

Strategy	Mean	Std. Deviation	Sharpe Ratio	Trades	Win Rate (%)
Moving Avg	0.0457	1.0190	0.0448	731.0	2.8700
Momentum 1-day	0.0711	4.6497	0.0153	750.0	51.3300
Momentum 2-day	-0.1693	4.6466	-0.0364	749.0	49.0000
Momentum 3-day	-0.1191	4.6512	-0.0256	748.0	48.8000
Coin Flip	0.0647	6.2468	0.0105	99.0	50.4243

- Moving Avg has a decent Sharpe (0.0448), considering the low volatility. It’s simple but effective.
- Momentum 1-day is slightly better than 2-day and 3-day, which deteriorate quickly.
- Coin Flip performs about the same as 1-day momentum in mean and Sharpe — weak signal overall.

BTC Stock

Strategy	Mean	Std. Deviation	Sharpe Ratio	Trades	Win Rate (%)
Moving Avg	0.0196	0.2853	0.0689	85.0	3.530
Momentum 1-day	-0.0758	1.1275	- 0.0672	104.0	46.150
Momentum 2-day	-0.2415	1.2088	- 0.1998	103.0	40.780
Momentum 3-day	-0.2429	1.1465	- 0.2119	102.0	42.160
Coin Flip	0.0047	1.2065	0.0040	99.0	48.858

- Moving Avg has the best Sharpe (0.0689) and low volatility — surprisingly stable.
- Momentum strategies (especially 2-day and 3-day) are strongly negative, meaning they're probably picking up noise or reversals.
- Coin Flip performs better than all Momentum strategies.

- **TSLA:** High volatility, occasional short-term trends (1-day momentum okay).
- **AAPL:** Stable, low edge — most strategies ineffective.
- **SPY:** Hard to beat — modest performance at best.
- **BTC:** Trendy, but momentum strategies misfire — MA better.

Limitations

While this was an exploratory analysis of simple trading strategies, several limitations were present to keep the simulation focused. The market behavior was modeled as a random walk, which does not capture the full complexity or noise of real financial markets. Important factors such as transaction costs, execution delays, and asset-specific trading volume were not included, which can significantly impact real-world performance. Features like leverage, stop-loss mechanisms, and dynamic position sizing were also left out but could be explored in future work to improve realism and robustness.

Future work could involve incorporating more realistic market dynamics, such as slippage, transaction fees, and order execution latency. Adding mechanisms like stop-losses, dynamic position sizing, or volume-based filters could improve strategy robustness and risk management. Exploring different types of market models beyond random walks—such as mean-reverting or regime-switching processes—could also provide a deeper understanding of strategy performance under varying conditions.

Conclusions

Overall, while the strategies were tested on historical data and offer some insight, they should be interpreted cautiously due to the simplified assumptions and lack of real-time adaptability. The moving average strategy consistently underperformed, with very low win rates and limited responsiveness, suggesting it may be too rigid for dynamic markets without further refinement. On the other hand, the 1-day momentum strategy showed the most promise, producing the highest average returns across most assets, though it also came with higher volatility. Longer momentum signals (2-day and 3-day) tended to decay in effectiveness, reinforcing that short-term signals might capture trends more efficiently. While these results are limited by the simulation's assumptions, they offer a useful starting point for evaluating and improving basic trading strategies.