Algorithmic Trading
Endterm Report

Harsh Mehta
Mentor: Sahil Barbade

July 2024

Contents

1 Introduction

In trading, algorithms automate strategies from basic to advanced levels, utilizing mathematical and statistical principles to analyze and execute trades. Pioneered further in the 20th century, algorithmic trading revolutionized finance by leveraging computational power and mathematical models to gain competitive edges, spawning the rise of quant finance and hedge funds like those founded by Alfred Winslow Jones in 1949. Algorithms in trading are systematic procedures that automate decision-making processes based on predefined rules. They start with inputs such as real-time market data and execute specific actions to achieve desired outcomes, like buying or selling securities. Parameters set by traders influence algorithmic behavior, crucially impacting trading success. These algorithms can range from simple, single-action rules to complex systems that continuously analyze data and adjust strategies dynamically. Implementing algorithms typically involves real-time data feeds, calculation triggers, and often manual order execution, emphasizing the importance of parameter optimization and risk management, including setting protective stop-loss orders.

2 Popular Algos

The following is a very sparse list of mainstream algos these algos is to 'get' the trade without market impact, anonymously, rapidly, without being 'front run.' Making an immediate profit on the trade is not of first importance to Tier 1 companies as most of these trades are of longer duration. Only the so-called 'high frequency' traders who use the technology to their main advantage by minimizing the ping time to the Exchanges and are happy with a couple of basis points per trade are 'immediate profit oriented.'. Algos for the use of individual traders are designed to provide immediate returns.

2.1 VWAP – Volume Weighted Average Price

VWAP (Volume Weighted Average Price) is an algorithm in trading, used predominantly by institutional investors to execute large orders while minimizing market impact. It calculates an average price based on volume data, slicing orders into smaller trades spread over time intervals to match market liquidity. VWAP serves as a benchmark for block trades, optimizing execution by reducing price disturbance and maintaining trade anonymity. Variations in VWAP include adjusting lookback periods and execution constraints to suit specific trading conditions. Despite its effectiveness, predicting volume patterns, especially for thinly traded stocks, remains challenging. Proprietary adaptations continually refine VWAP strategies for optimal performance and market adaptation.

$$P_v wap = \frac{\sum(P * V)}{\sum(V)}$$

2.2 TWAP – Time Weighted Average Price

This algo strategy simply divides the order more or less evenly over a user specified time frame. Usually the order is sliced up equally into a specified number of discrete time intervals, or waves to minimize market impact while risking detection by algorithmic traders. This is often combated by leaving out a wave or using a 'fuzzy' time interval spacing or even a 'fuzzy' number of shares per wave.

2.3 POV – Percentage of Volume

POV executes trades proportionally to current trading volume, minimizing visibility and adverse price movements for larger orders.

2.4 Black Lance – Search for Liquidity

Black Lance seeks liquidity in Dark Pools.This is accomplished by 'pinging' the many different venues and analyzing the responses to determine the level of liquidity available in the issue of interest.

2.5 PEG – Pegged Order

PEG adjusts limit orders relative to market movements, facilitating efficient execution while maintaining price sensitivity.

2.6 Iceberg – Hidden Large Order

Iceberg conceals large orders by splitting them into smaller, undisclosed segments, minimizing market impact and front-running risks.

2.7 Recursive Algos

These algorithms recursively call themselves until a specified condition is met.They are powerful but must be carefully managed due to potential infinite loops.

2.8 Serial Algos

Sequential algorithms execute one after another in a predefined order, often with conditional branching, making them suitable for structured processes but potentially slower for complex tasks.

2.9 Parallel Algos

These utilize multi-core processors to execute multiple tasks simultaneously, enhancing performance for tasks that can be divided into independent parts.

2.10 Iterative Algos

These algorithms use loops (e.g., 'if...then,' 'Do while,' 'for...Next') to repetitively execute instructions, useful for tasks requiring iterative refinement or decision-making based on changing conditions.

2.11 Pair trading

Pair trading strategies, initially pivotal to algorithmic trading, involve trading two correlated stocks to create a market-neutral position. This strategy, which doesn't rely on market direction, capitalizes on the price co-movement of two stocks, typically in the same sector.The strategy profits when diverging correlated stocks revert to their mean price relationship. The approach involves shorting the outperforming stock and buying the underperforming one. This method proved successful in exploiting price movements of correlated securities. Tools like correlation charts and algorithms help identify suitable stock pairs.

3 How to Optimize Individual Trader Algos

Strategies for individual traders according to the book by Edward Leshik and Jane Cralle should include:
1. $Volume > than 1 millionsharesoversession$.
2. $Tradeprice > than$ \$35. We generally prefer the higher priced stocks. LC volatility index >n.
3. Parameter optimization can be carried out by backtesting; we use five consecutive sessions as our lookback default in most cases. For very high volume stocks (>15m) increase to 10 trading sessions.
Optimizing individual trading algorithms involves careful stock selection, parameter tuning, and matching algorithms to stocks. Key criteria for stock selection include high trading volume and price. Parameter optimization is primarily achieved through backtesting, typically using a five-session lookback period. For high-volume stocks, a ten-session lookback may be used. Optimization focuses on maximizing profit per second of trade while minimizing market exposure. The process involves iterative adjustments to moving averages, trigger parameters, and ALPHA constants. Maintaining detailed logs of variations and results is essential. Consistent practice, methodical approaches, and small, frequent gains are prioritized over high-risk trades. the given code implements these strategies

```python
import yfinance as yf
import pandas as pd

NSE_VOLUME_THRESHOLD = 100000
PRICE_THRESHOLD = 500
VOLATILITY_INDEX_THRESHOLD = 0.02

nse_symbols = pd.read_csv("EQUITY_L.csv")
```

```
nse_symbols = nse_symbols['SYMBOL'].tolist()

def calculate_volatility(prices):
    return max(prices) - min(prices)

def filter_stocks(symbols):
    selected_stocks = []
    for symbol in symbols:
        symbol = f'{symbol}.NS'
        try:
            data = yf.download(symbol, start='2022-01-01', end='2023-01-01')
            if data.empty:
                continue

            volumes = data['Volume']
            closing_prices = data['Close']

            avg_volume = volumes.mean()
            avg_price = closing_prices.mean()
            volatility = calculate_volatility(closing_prices) / avg_price

            if avg_volume > NSE_VOLUME_THRESHOLD and avg_price > PRICE_THRESHOLD and volatility > VOLATILITY_INDEX_THRESHOLD:
                selected_stocks.append(symbol)
        except Exception as e:
            print(f"Error fetching data for {symbol}: {e}")
            continue

    return selected_stocks

if __name__ == "__main__":
    selected_stocks = filter_stocks(nse_symbols)
    print("Selected stocks and ETFs:", selected_stocks)
```

4 Math Toolkit

Most algo construction can be achieved with the simplest of tools.
Long moving averages are 'low-pass' filters whereas short moving averages are 'high-pass' filters. A 'low-pass' filter attenuates (damps, reduces) the high frequencies while letting the low frequencies pass through unimpeded. A 'high-pass' filter attenuates the low frequencies and lets the high frequencies pass through.
Metrics:
errors = forecast - actual
mse(mean squared error) = np.square(errors).mean()
rmse(root mean square error) = np.sqrt(mse)
mae(mean absolute error) = np.abs(error).mean()
mape(mean absolute percentage error) = np.abs(error/ x).mean()

Exponential Moving Average:

$$EMA = EMA_{t-1} + \frac{2}{n+1} \cdot (S_t - EMA_{t-1})$$

where n=the length of the lookback S=trade price.

5 Statistics Toolkit

The average price $\bar{S}$ is calculated as:

$$\bar{S} = \frac{\sum_{i=1}^{n} s_i}{n}$$

The mean deviation (MD) is given by:

$$MD = \text{mean deviation} = \frac{\sum_{i=1}^{n} |s_i - S|}{n}$$

Here, MD represents the mean deviation, s_i are individual data points, and S is the mean of the data points.

The z-score is given by:

$$\text{z-score} = \frac{x - \bar{x}}{\sigma}$$

This is the distance that a value lies above or below the mean of the data measured in units of standard deviation, sigma. Here,

- z-score represents the standardized score,
- x is the individual value,
- $\bar{x}$ is the mean of the dataset,
- σ is the standard deviation.

Outliers may be found using z-scores and Tchebysheff's Theorem tells us that almost all observations in the data will have z-scores of 3 or less. Thus a much higher z-score would indicate that the value is an outlier and should be ignored in the main deliberation on the dataset.

The standard deviation σ is given by:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

Here,

- σ represents the sample standard deviation,
- x_i are the individual data points,

- $\bar{x}$ is the sample mean,
- n is the sample size.

In a Gaussian distribution, we can typically find:

- 95% of the values within $\pm 2\sigma$
- approaching 100% of the values within $\pm 3\sigma$.

6 Benchmarks and Performance Measures

6.1 THE VIX

The VIX is an exchange-traded indicator that reflects the implied volatility in Standard & Poor's 500 options trading. High values of the VIX typically coincide with periods of market uncertainty, fear of loss, and overall nervous sentiment. Conversely, a low VIX reading often suggests a potential market rise, low volatility, and a general calmness among market participants.
The VIX is most insightful when interpreted at extreme high or low readings, as it tends to accurately predict future market events.

6.2 Beta (β)

Beta, symbolized by the Greek letter β, measures the relative volatility of a stock compared to a benchmark, often the S&P 500 index, which represents the broader market. A beta of 1 indicates that the stock moves in line with the benchmark. A beta greater than 1 suggests higher volatility than the benchmark, while a beta less than 1 indicates lower volatility.

For example, a stock with a beta of 2.2 would be expected to move 2.2 times more than the index in the same direction. Conversely, a stock with a beta of 1.24 would be 24% more volatile than the benchmark.

Beta is calculated using a simple linear regression of the stock's daily returns against the S&P 500 over a specified period, typically one year. It is used in portfolio analysis to diversify stock-specific volatility, aiming for a beta close to 1.

6.3 Alpha (α)

Alpha, represented by the Greek letter α, is a measure of volatility-adjusted performance. It indicates the excess return of a stock relative to its expected return based on its beta when the benchmark return (typically the S&P 500) is 0.

The formula for calculating Alpha is:

$$\alpha = \text{RET}_{\text{stock}} - \beta \times \text{RET}_{\text{benchmark}}$$

where:

- α is the Alpha,
- $\mathrm{RET}_{\mathrm{stock}}$ is the return of the stock,
- β is the beta of the stock

6.4 The Sharpe Ratio

The Sharpe Ratio, developed by Nobel Prize winner William Sharpe with Harry Markowitz, measures the risk-adjusted performance of investments. It quantifies whether the returns of a trading strategy exceed those of a risk-free asset per unit of volatility. This ratio helps determine if a trading strategy's success stems from smart decisions or excessive risk-taking. It's particularly sensitive to trade frequency, as noted by Irene Aldridge. A higher Sharpe Ratio suggests a more efficient strategy, with comparisons most relevant for similar return periods, especially for long-term investors.

6.5 The Basis Points per Second and Basis Points per Trade

Basis Points per Second (Bp/sec) is an absolute cumulative metric used to evaluate the profitability of specific ALPHA ALGO trading strategies over time. It measures the average profitability in basis points that an algorithm achieves per second of operation. This metric provides insights into the consistent profitability of the algorithm over a duration, despite potential fluctuations in performance.

On the other hand, Basis Points per Trade measures the cumulative profitability of an algorithm over the entire duration of a trade. It offers a different perspective by capturing the overall profitability achieved from start to finish of a trading activity. This metric helps in assessing the effectiveness of the algorithm in generating returns across complete trading cycles.

7 Volatility

Volatility is a multifaceted concept in finance that captures the degree of variation in the price of a stock over time. Understanding volatility is crucial for various trading strategies, particularly those involved in algorithmic trading and stock selection.

Fundamental Definition of Volatility

At its core, volatility represents the price change of a stock, expressed per unit time. It quantifies the fluctuation in stock prices, typically in terms of basis points per second or basis points per nT ticks (where n is often set to 100).

Key Properties of Volatility

Mean Reversion: Volatility tends to revert to a long-term mean. This means that periods of high volatility are usually followed by periods of low volatility, and vice versa.

Link to Returns: Volatility and returns are closely related. Volatility often provides insights into the risk associated with returns.

Timescales: Volatility can be measured over various timescales, ranging from very short intervals (such as seconds or ticks) to daily, monthly, or even annual periods.

Market Conditions: Volatility is present in all market conditions—whether the market is trending up, trending down, or moving sideways. It manifests as ripples of varying sizes over the underlying trend.

Estimating Volatility The estimation of volatility depends on the intended use and can vary based on the lookback period and the sampling frequency. Common approaches include:

High-Frequency Data: Using fine sampling resolutions and sequential lookback periods to observe volatility behavior over time, typically over three to ten trading sessions.

End-of-Day (EOD) Data: Tracking volatility using daily data to assess trends over longer periods.

Realized Volatility

With the availability of high-frequency tick data, realized volatility has become a common term. It refers to estimates using intraday squared returns at short intervals, such as every five minutes. In the discussed methods, even shorter intervals (around 60 seconds or less) are often used.

Volatility Clustering

A notable feature of volatility is clustering, where periods of high volatility are often followed by further high volatility, and quiet periods are followed by tranquility. This clustering behavior indicates that volatility changes are not random but exhibit patterns over time.

Calculation Methods

Volatility can be calculated in various ways, including:

Standard Deviation of Returns

This traditional method involves computing the standard deviation of period returns, which is calculated as:

$$\sigma = \sqrt{\frac{1}{P-1}\sum(r_p - \mu)^2}$$

where:

- P is the number of periods,
- r_p is the period return, and
- μ is the average return over the lookback period.

8 Teachnical Analysis

8.1 CROSSING SIMPLE MOVING AVERAGES

Crossing Simple Moving Averages (SMAs) is a fundamental technique in Technical Analysis (TA). It involves using short and long SMAs to generate buy and sell signals.

The "Black Cross" occurs when the short SMA crosses below the long SMA, signaling a sell or closure of a long position. Conversely, the "Golden Cross" happens when the short SMA crosses above the long SMA, indicating a buy signal.

The "ribbon" method uses multiple Simple Moving Averages (SMAs).

Multiple SMAs: Instead of using just two SMAs (short and long), the ribbon method involves plotting multiple SMAs of varying lengths simultaneously on a price chart. For example, traders might use 10 SMAs with increasing lengths, such as 10-day SMA, 20-day SMA, 30-day SMA, and so on up to 100-day SMA.

Crosses: Similar to the basic SMAs, traders look for crosses between the SMAs in the ribbon. A short-term SMA crossing above a longer-term SMA could indicate a bullish signal (buy), while a cross below could signal bearish sentiment (sell).

Spacing and Slope: The spacing between SMAs and their slopes can provide additional clues. Narrow spacing indicates consolidation or uncertainty, while widening spacing suggests increasing trend strength.

Market Environment: Traders interpret the ribbon based on the prevailing market conditions. For instance, during a strong uptrend, SMAs may maintain a well-defined order with shorter SMAs consistently above longer SMAs

The given code uses mutiple sma to decide buy and sell signals

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv("TATAMOTORS.NS2.csv")

df = df.dropna(subset=['Open', 'Close'])

df['Date'] = pd.to_datetime(df['Date'])
df.set_index('Date', inplace=True)
data = df[['Open', 'Close']]
data=data[6000:]

sma_periods = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
for period in sma_periods:
    data[f'SMA_{period}'] = data['Close'].rolling(window=period).
    mean()

buy_signals = []
sell_signals = []

for i in range(1, len(data)):
    for j in range(len(sma_periods)-1):
        short_sma = f'SMA_{sma_periods[j]}'
        long_sma = f'SMA_{sma_periods[j+1]}'

        if data[short_sma].iloc[i] > data[long_sma].iloc[i] and
    data[short_sma].iloc[i-1] <= data[long_sma].iloc[i-1]:
            buy_signals.append((data.index[i], data['Close'].iloc[i
    ]))
        elif data[short_sma].iloc[i] < data[long_sma].iloc[i] and
    data[short_sma].iloc[i-1] >= data[long_sma].iloc[i-1]:
            sell_signals.append((data.index[i], data['Close'].iloc[
    i]))

if len(buy_signals) > len(sell_signals):
    buy_signals = buy_signals[:len(sell_signals)]
elif len(sell_signals) > len(buy_signals):
    sell_signals = sell_signals[:len(buy_signals)]

returns = []
for i in range(len(buy_signals)):
    buy_price = buy_signals[i][1]
    sell_price = sell_signals[i][1]
    returns.append((sell_price - buy_price) / buy_price)

total_return = sum(returns)

plt.figure(figsize=(14, 7))
plt.plot(data['Close'], label='Close Price', color='black')

for period in sma_periods:
    plt.plot(data[f'SMA_{period}'], label=f'SMA {period}')

buy_dates, buy_prices = zip(*buy_signals)
sell_dates, sell_prices = zip(*sell_signals)
```

```
plt.scatter(buy_dates, buy_prices, marker='^', color='green', label
    ='Buy Signal', alpha=1)
plt.scatter(sell_dates, sell_prices, marker='v', color='red', label
    ='Sell Signal', alpha=1)

plt.title('Tata Motors - SMAs and Buy/Sell Signals')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid()
plt.show()

print(f'Total Return: {total_return * 100:.2f}%')
```

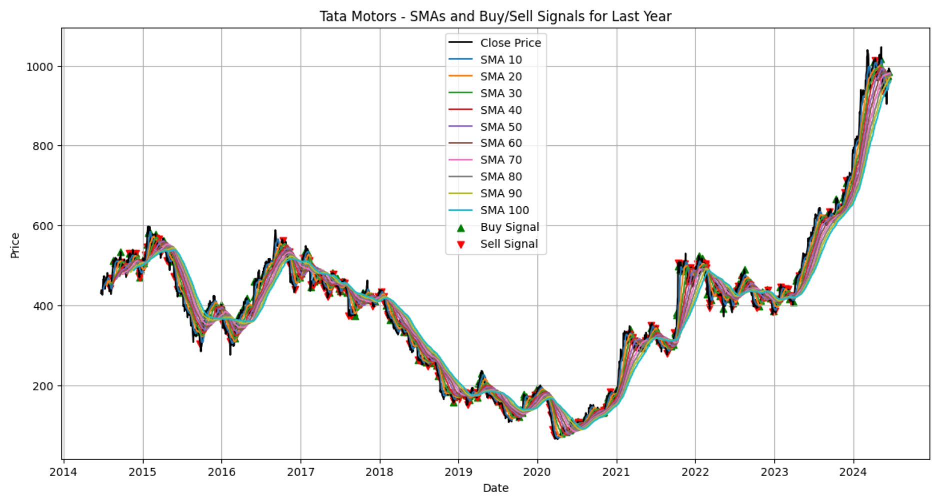

Figure 1: Tata Motors - SMAs and Buy/Sell Signals for Last Years

the given code uses two sma to decide buy and sell signals

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


df = pd.read_csv("TATAMOTORS.NS2.csv")


df = df.dropna(subset=['Open', 'Close'])


df['Date'] = pd.to_datetime(df['Date'])
df.set_index('Date', inplace=True)
data = df[['Open', 'Close']]
data =data[6000:]


short_window = 50  # Short-term SMA period
long_window = 200  # Long-term SMA period
data['SMA_short'] = data['Close'].rolling(window=short_window).mean
    ()
data['SMA_long'] = data['Close'].rolling(window=long_window).mean()


signals = pd.DataFrame(index=data.index)
signals['signal'] = 0


signals['signal'][short_window:] = np.where(
    data['SMA_short'][short_window:] > data['SMA_long'][
    short_window:], 1, 0)
signals['positions'] = signals['signal'].diff()
```

```python
initial_capital = 100000  # Initial capital in currency (e.g., INR)
capital = initial_capital
positions = 0  # Initial number of shares
long_positions = []
short_positions = []
total_profit = 0

for i in range(1, len(signals)):
    if signals['positions'].iloc[i] == 1:  # Buy signal
        buy_price = data['Close'].iloc[i]
        shares = capital // buy_price
        capital -= shares * buy_price
        positions += shares
        long_positions.append((data.index[i], buy_price))

    elif signals['positions'].iloc[i] == -1 and positions > 0:  #
    Sell signal
        sell_price = data['Close'].iloc[i]
        capital += positions * sell_price
        profit = positions * (sell_price - buy_price)
        total_profit += profit
        positions = 0
        short_positions.append((data.index[i], sell_price))


final_value = capital + positions * data['Close'].iloc[-1]
total_return = (final_value - initial_capital) / initial_capital
total_profit = final_value - initial_capital


plt.figure(figsize=(14, 7))
plt.plot(data['Close'], label='Close Price', color='black')
plt.plot(data['SMA_short'], label=f'SMA {short_window}', color='
    blue')
plt.plot(data['SMA_long'], label=f'SMA {long_window}', color='red')


buy_dates, buy_prices = zip(*long_positions) if long_positions else
     ([], [])
sell_dates, sell_prices = zip(*short_positions) if short_positions
    else ([], [])

plt.scatter(buy_dates, buy_prices, marker='^', color='green', label
    ='Buy Signal', alpha=1)
plt.scatter(sell_dates, sell_prices, marker='v', color='red', label
    ='Sell Signal', alpha=1)

plt.title('Tata Motors - SMAs and Buy/Sell Signals')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid()
plt.show()


print(f'Initial Capital: {initial_capital}')
```

```
print(f'Final Value: {final_value}')
print(f'Total Return: {total_return * 100:.2f}%')
print(f'Total Profit: {total_profit:.2f}')
```

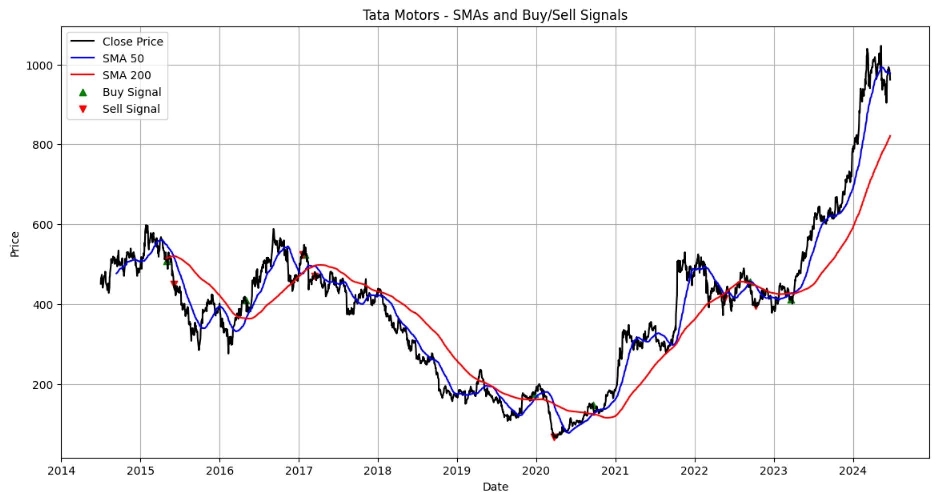

Figure 2: Tata Motors - SMAs and Buy/Sell Signals for Last Years

8.2 EXPONENTIAL MOVING AVERAGES

Buy Signals: Occur when a shorter EMA crosses above a longer one, particularly effective with longer EMAs in a two-line crossover system.
Single EMA Success: Using single EMAs over extended tick lookbacks (e.g., 200 ticks) has shown accurate identification of valleys and peaks in certain stocks, typically within 3 to 5 ticks.
Trading Strategy: Buy signals trigger when the EMA hits a valley, and exit signals are sought when the EMA crosses the zero line and reaches an upward peak. However, the reliability of sell signals is considered inconsistent, prompting the recommendation to use a $-Stop to close long trades.
$-Stop Strategy: Involves setting a predefined profit target (e.g., 50 basis points), which, when reached, automatically triggers a market order to sell, ensuring profitability is captured.
The code give buy and sell signals based on ema

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv("TATAMOTORS.NS2.csv")

df = df.dropna(subset=['Open', 'Close'])

df['Date'] = pd.to_datetime(df['Date'])
```

```
df.set_index('Date', inplace=True)
data = df[['Open', 'Close']]
data = data[:]

short_window = 5  # Short-term EMA period
long_window = 20  # Long-term EMA period
data['EMA_short'] = data['Close'].ewm(span=short_window, adjust=
    False).mean()
data['EMA_long'] = data['Close'].ewm(span=long_window, adjust=False
    ).mean()

signals = pd.DataFrame(index=data.index)
signals['signal'] = 0

signals['signal'][short_window:] = np.where(
    data['EMA_short'][short_window:] > data['EMA_long'][
    short_window:], 1, 0)
signals['positions'] = signals['signal'].diff()

initial_capital = 100000  # Initial capital in currency (e.g., INR)
capital = initial_capital
positions = 0  # Initial number of shares
long_positions = []
short_positions = []
total_profit = 0

for i in range(1, len(signals)):
    if signals['positions'].iloc[i] == 1:  # Buy signal
        buy_price = data['Close'].iloc[i]
        shares = capital // buy_price
        capital -= shares * buy_price
        positions += shares
        long_positions.append((data.index[i], buy_price))

    elif signals['positions'].iloc[i] == -1 and positions > 0:  #
    Sell signal
        sell_price = data['Close'].iloc[i]
        capital += positions * sell_price
        profit = positions * (sell_price - buy_price)
        total_profit += profit
        positions = 0
        short_positions.append((data.index[i], sell_price))

final_value = capital + positions * data['Close'].iloc[-1]
total_return = (final_value - initial_capital) / initial_capital
total_profit = final_value - initial_capital

plt.figure(figsize=(14, 7))
plt.plot(data['Close'], label='Close Price', color='black')
plt.plot(data['EMA_short'], label=f'EMA {short_window}', color='
    blue')
plt.plot(data['EMA_long'], label=f'EMA {long_window}', color='red')

buy_dates, buy_prices = zip(*long_positions) if long_positions else
     ([], [])
sell_dates, sell_prices = zip(*short_positions) if short_positions
    else ([], [])
```

```
plt.scatter(buy_dates, buy_prices, marker='^', color='green', label
    ='Buy Signal', alpha=1)
plt.scatter(sell_dates, sell_prices, marker='v', color='red', label
    ='Sell Signal', alpha=1)

plt.title('Tata Motors - EMAs and Buy/Sell Signals')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid()
plt.show()

print(f'Initial Capital: {initial_capital}')
print(f'Final Value: {final_value}')
print(f'Total Return: {total_return * 100:.2f}%')
print(f'Total Profit: {total_profit:.2f}')
```

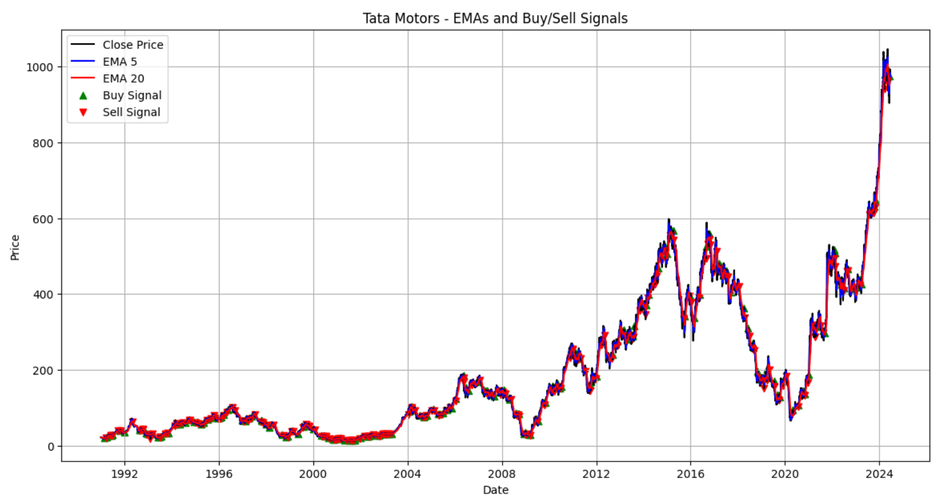

Figure 3: Tata Motors - EMAs and Buy/Sell Signals for Last Years

8.3 MOMENTUM and % MOMENTUM

Momentum in trading refers to the difference between the current price and the price from a defined interval ago. Specifically, n-day momentum is calculated as:

$$\text{Momentum} = \text{Price(now)} - \text{Price}(n)$$

where n typically represents a number of days ago. This indicator is also known as the Rate of Change (ROC). A lower ROC compared to the current price suggests the stock may be losing momentum or running out of steam.

To enhance the usability of Momentum, traders often convert it into a percentage form.

$$\%\text{Momentum} = \left(\frac{\text{Price(now)} - \text{Price}(n)}{\text{Price}(n)}\right) \times 100$$

8.4 RSI

The RSI is calculated using the formula:

$$\text{RSI} = 100 - \left(\frac{100}{1 + RS}\right)$$

where RS (Relative Strength) is the ratio of average closing prices over a 14-day period:

$$RS = \frac{\text{Average of days where prices closed up}}{\text{Average of days where prices closed down}}$$

The RSI is an oscillator that ranges between 0 and 100. Values above 70 indicate overbought conditions, suggesting potential market tops, while values below 30 indicate oversold conditions, suggesting potential market bottoms. This makes it a valuable tool for identifying potential reversal points in the market.

8.5 TRIX OSCILLATOR

This is a triple exponential average indicator for oversold and overbought markets oscillating about a center zero line. It may also be used for assessing momentum (positive values mean momentum is increasing). TRIX is calculated as the triple exponential moving average of the log of the price sequence. Long and short entries are signaled by crossing of the zero line.

8.6 THE STOCHASTIC

The formula for %K is given by:

$$\%K = 100 \times \left(\frac{\text{today's close} - \text{12-day low}}{\text{12-day high low range}}\right)$$

The Stochastic Oscillator always lies within the 0 to 100 range. 0 means the stock is trading at the lowest price of the 12-day period; 100 means it was trading at the peak of the prices set in the 12 days.

8.7 Bollinger Bands

Setup: Bollinger Bands use a 20-day Exponential Moving Average (EMA) with two bands plotted above and below the moving average. These bands are typically spaced 2 standard deviations apart. However, using a slightly smaller separation of 1.65 standard deviations (which corresponds to a 90% confidence level) has been successful in providing more trading opportunities compared to the traditional 2 sigma setup.

Volatility Adjustment: The width of the bands varies with market volatility. During highly volatile periods, the bands widen, whereas during calm periods, they narrow.

Trading Signals: Traders typically buy when the price touches or penetrates the upper Bollinger Band. John Bollinger, the creator of this method, observes that narrow bands often precede sharp price changes, indicating a potential reduction in volatility and a possible reversal.

8.8 Williams %R

Anticipatory Nature: One unique characteristic of Williams %R is its ability to anticipate stock price reversals. It often peaks and begins to decline well before the stock price reaches its peak, and similarly, it turns upward from a low before the stock price does the same.
Calculation Formula: The oscillator Williams %R is calculated using the formula:

$$\%R = \left(\frac{\text{MAX}_n - \text{Today's close}}{\text{MAX}_n - \text{Min}_n} \right) \times 100$$

where MAX_n is the highest high over a specified period (typically 14 days), and Min_n is the lowest low over the same period.
Range and Interpretation: The %R oscillator swings between 0 (highest reading) and -100 (lowest reading). Readings from 0 to -20 are considered overbought, suggesting a potential for market reversals or corrections, while readings from -80 to -100 are considered oversold, indicating potential buying opportunities.

8.9 The Arms Index or TRIN

Calculation: The TRIN is calculated using the formula:

$$\text{TRIN} = \frac{\text{Advancing Stocks/Declining Stocks}}{\text{Increasing Volume/Decreasing Volume}}$$

It measures the ratio of advancing to declining stocks relative to the ratio of increasing to decreasing volume.
Market Sentiment Indicator: The TRIN indicates whether volume is flowing into advancing or declining stocks. A TRIN below 1 suggests increased volume flowing into rising stocks, indicating bullish sentiment. Conversely, a TRIN above 1 suggests more volume going into declining stocks, reflecting bearish sentiment.
Usage: It is commonly used as a real-time market sentiment indicator, particularly useful in trading individual stocks to gauge market breadth and sentiment.

9 Backtesting And Automated Execution

Backtesting is a crucial process in developing and validating trading strategies. It involves testing a trading strategy using historical data to see how it would have performed in the past.

The more round trip trades there are in the backtest, the higher will be the statistical significance. But even if a backtest is done correctly without pitfalls and with high statistical significance, it doesn't necessarily mean that it is predictive of future returns. Regime shifts can spoil everything, and a few important historical examples will be highlighted.

9.1 Common Pitfalls of Backtesting

9.1.1 Look-ahead Bias

look-ahead bias means that your backtest program is using tomorrow's prices to determine today's trading signals. Or, more generally, it is using future information to make a "prediction" at the current time. A common example of look-ahead bias is to use a day's high or low price to determine the entry signal during the same day during backtesting. (Before the close of a trading day, we can't know what the high and low price of the day are.)

9.1.2 Data-Snooping Bias and the Beauty of Linearity

Data-Snooping Bias:
Definition: This bias occurs when models are over-fitted to historical data with too many parameters, making them less predictive for future data.

Minimizing Data-Snooping Bias:
Simplicity: The model should be as simple as possible with few parameters. Complicated models with many rules are more prone to data-snooping bias.
Linearity: Linear models are preferred over nonlinear ones. Nonlinear models, though potentially fitting historical data better, are not guaranteed to predict future values effectively.

Linearity in Models:
Taylor Series Approximation: Nonlinear models can often be approximated by simpler linear models. Unless a strong case is made for nonlinearity, a linear model should be used.
Gaussian Distribution: Despite its limitations in capturing extreme market events, it is often used due to its simplicity and the lack of clear alternatives

9.1.3 Stock Splits and Reverse Splits

Stock Splits:
Definition: A stock split increases the number of shares outstanding by issuing

more shares to existing shareholders. An N-to-1 split means that for every share previously held, the shareholder now has N shares.
Price Adjustment: To keep the price series consistent, historical prices before the ex-date should be divided by N. This ensures that the apparent price drop on the ex-date does not trigger erroneous trading signals.

Reverse Splits:
Definition: A reverse split reduces the number of shares outstanding, increasing the share price proportionally. A 1-to-N reverse split means that for every N shares previously held, the shareholder now has 1 share.
Price Adjustment: Historical prices before the ex-date should be multiplied by N to maintain consistency.

Dividends
Cash and Stock Dividends:
Definition: Dividends are distributions of a portion of a company's earnings to shareholders. Cash dividends are paid in cash, while stock dividends are paid in additional shares.
Price Adjustment: On the ex-date, the stock price typically drops by the dividend amount ($d per share) because the value of the dividend is no longer included in the stock price.
Adjustment for Historical Prices: To avoid erroneous trading signals, historical prices before the ex-date should be adjusted downward by the dividend amount.

9.1.4 Survivorship Bias

Survivorship Bias:
Definition: Survivorship bias occurs when a dataset only includes entities that have "survived" to the end of the observation period, excluding those that have failed or been delisted.
Impact: This bias can skew the results of a backtest, making a trading strategy appear more successful than it would have been in reality.

9.1.5 short-sale constraint

Locating Shares:
Brokers need to locate shares to borrow for shorting, often from other customers or institutions. If there's high demand for borrowing or limited stock float, certain stocks become "hard to borrow." Costs and Availability:
Short sellers may have to pay interest to the stock lender, especially for hard-to-borrow stocks. In extreme cases, stocks may be impossible to borrow in desired quantities or at all.
Short-sale constraints pose significant challenges to the accuracy of backtesting trading models. To mitigate their effects, traders must account for regulatory restrictions, availability, and costs associated with shorting stocks. By using accurate historical data and conducting thorough sensitivity analysis, traders

can better understand and manage the impact of short-sale constraints on their strategies' performance.

9.1.6 Future Continues Contracts

Continuous contracts play a vital role in futures trading strategy development and backtesting. However, their handling requires careful consideration of rollover adjustments and back-adjustment methods to ensure accurate performance measurement and signal generation.

9.1.7 Futures Close versus Settlement Prices

In futures trading, the price you often see at the end of the trading day is called the settlement price. It's not necessarily the same as the last price at which a contract was traded. Even if there were no trades during the day, the exchange will still assign a settlement price.

Now, when you're backtesting trading strategies, which price should you use? Generally, it's better to use the settlement price because it's the closest thing to what your actual transaction price would have been if you traded live. Using the last traded price could give you a misleading picture because it might have happened hours earlier and might not reflect the price you would have gotten at the end of the day.

9.2 Statistical Significance of Backtesting: Hypothesis Testing

Calculate a Test Statistic:
First, you run your trading strategy on historical data and calculate a key number. Let's say this number is the average daily return your strategy made.

Set Up the Null Hypothesis:
The null hypothesis is your starting assumption. In this case, it's that your strategy doesn't actually make any real profit. That is, if you ran your strategy over an infinite amount of data, the average return would be zero.

Determine the Probability Distribution:
You assume a certain probability distribution for daily returns. This distribution has a zero mean (as per the null hypothesis). Essentially, you figure out the typical range of returns if your strategy had no real edge and returns were just random.

Compute the p-value:
The p-value helps you determine how likely it is to observe your test statistic (average daily return) if the null hypothesis is true.
You calculate the probability (p-value) of seeing a return as high (or low) as

what you observed in your backtest, assuming the null hypothesis is true. If this probability is very low (e.g., less than 1%), you can confidently say your strategy's results are statistically significant and not just due to chance.

Z-Score Calculation

The z-score measures how many standard deviations away your test statistic is from the mean under the null hypothesis. It is calculated using the formula:

$$z = \frac{\text{Observed Average Daily Return} - \text{Null Hypothesis Mean}}{\frac{\text{Standard Deviation of Daily Returns}}{\sqrt{n}}}$$

where:
- Observed Average Daily Return is the average return observed over a given period.
- Null Hypothesis Mean is the mean return under the null hypothesis.
- Standard Deviation of Daily Returns is the standard deviation of the daily returns.
- n is the number of observations.

The p-value is the probability of observing a z-score as extreme as this one in a standard normal distribution. You can use a z-table to find p

Gaussian Method:
Calculate the p-value using the standard normal distribution. If the Sharpe ratio is high, you'll likely reject the null hypothesis.

Monte Carlo Method:
Simulate many price series with similar statistical properties to your actual data. Run your strategy on these series. If only a few simulations match or exceed your backtest results, your strategy is likely good.

Random Trades Method:
Randomly place trades on the actual price series. If only a few random trade sets perform as well as your strategy, it suggests your strategy is not just lucky. Different methods can provide different insights. It's crucial to consider multiple methods to get a well-rounded view of your strategy's statistical significance.

Heisenberg Uncertainty Principle in Backtesting:
Backtesting HFT strategies might not accurately reflect real-life profitability due to the Heisenberg uncertainty principle. The act of simulating market interactions can alter the outcomes, leading to discrepancies between backtest results and live trading performance.

9.3 multithreading in HFT

Multithreading is crucial for high-frequency trading (HFT), especially when dealing with multiple symbols simultaneously. In a multithreaded trading platform, multiple events (like the arrival of new ticks) can be processed concurrently. This ensures that trading decisions and actions for different symbols are not delayed due to the processing of other symbols
Modern programming languages like Java and Python natively support multithreading, making them ideal for developing robust HFT systems. MATLAB requires the Parallel Computing Toolbox for multithreading but is limited in the number of threads it can handle. However, MATLAB can still manage tick data efficiently using listeners, ensuring no ticks are lost even without multithreading. Writing stand-alone trading programs in languages that support multithreading natively provides the best performance and flexibility for high-frequency trading of multiple symbols.

9.4 Complex Event Processing (CEP)

Complex Event Processing (CEP) is a technology used to process and respond to multiple streams of events in real time. In the context of algorithmic trading, CEP allows a program to respond instantaneously to new events, such as the arrival of a new tick or a news item. This event-driven approach ensures that there is no delay between the occurrence of an event and the program's response.
CEP is essential for high-frequency trading and strategies that require instantaneous responses to market events. While simple trading rules can be implemented using callback functions provided by brokerage APIs, more complex rules benefit from CEP languages and technologies. Platforms like Progress Apama and certain open-source IDEs offer robust CEP capabilities, enabling the development of sophisticated, event-driven trading strategies.

10 The Basics of Mean Reversion

Mean reversion implies that after a significant deviation from an average or mean, values tend to move back towards the mean.
Price Series vs. Returns: Financial price series often follow geometric random walks, where prices do not revert to a historical mean. Returns: Instead of prices, it's the returns (percentage changes in prices) that tend to revert around a mean of zero due to their random distribution.
Stationarity and Trading Opportunities: Price series that exhibit mean reversion are termed stationary. These are rare among publicly traded assets. Statistical Tests: Tests like the Augmented Dickey-Fuller (ADF) test, Hurst exponent, and Variance Ratio test are used to determine stationarity.
Manufacturing Mean-Reverting Portfolios: While few individual assets exhibit

mean reversion, portfolios can be constructed to exhibit mean-reverting properties by combining assets that are not individually mean-reverting but are cointegrating.
Cointegration: Statistical tests like the CADF test and Johansen test identify combinations of assets whose net market value (prices) are mean-reverting.
Time Series Mean Reversion: Prices revert to a mean determined by their own historical prices.
Cross-Sectional Mean Reversion: Cumulative returns of instruments in a basket revert to the cumulative return of the entire basket. if an asset outperforms (has a positive relative return) compared to the basket, it tends to underperform (negative relative return) in the near future, and vice versa.
Linear Trading Strategies: These strategies exploit mean-reverting price series without needing complex parameters. They aim to buy assets when prices are below their historical mean and sell when prices revert upwards.
A mean-reverting price series suggests that the change in price in the next period is proportional to the difference between the current price and the mean price.

10.1 Hurst Exponent (H)

The Hurst exponent H is a measure used to determine the long-term memory of a time series.

Geometric Random Walk

$$H = 0.5$$

Indicates a random walk, meaning the price changes are independent of past values.

Mean Reverting

$$H < 0.5$$

Indicates that the series tends to revert to its mean.

Trending

$$H > 0.5$$

Indicates that the series tends to continue in its current direction.

10.2 Half-Life of Mean Reversion

The time it takes for the deviation from the mean to reduce by half. In practical trading, the half-life of mean reversion is a useful metric that helps traders understand how quickly a price series reverts to its mean. This measure can help in determining the suitability of a mean-reversion trading strategy.

Half-Life Calculation

The half-life ($T_{1/2}$) is given by:

$$T_{1/2} = -\frac{\log(2)}{\lambda}$$

Practical Application in Trading

- **Positive** λ: Indicates non-mean-reverting series; not suitable for mean-reversion strategies.
- **Small** λ **(close to zero)**: Implies long half-life; mean-reversion trades will be infrequent and potentially unprofitable.
- **Negative** λ: Useful for mean-reversion strategies. The half-life helps in determining the look-back period and expected holding period for trades.

Explanation of λ

λ (Lambda) is a coefficient that measures the speed of mean reversion. It is derived from the Augmented Dickey-Fuller (ADF) test.

10.3 Cointegration in Trading

1. Cointegration
If two or more non-stationary time series move together in such a way that a certain linear combination of them is stationary, then these time series are said to be cointegrated. This means that despite their individual trends, their relative relationship remains stable over time.
2. CADF Test
The Cointegrated Augmented Dickey-Fuller (CADF) test is used to determine if a linear combination of two price series is stationary. This involves the following steps:
Determine the Hedge Ratio: Run a linear regression between the two series to find the optimal hedge ratio.the hedge ratio, determines the proportion of capital allocation between the assets. This ratio minimizes the variance of the combined portfolio, ensuring effective mean reversion.
Form the Portfolio: Create a portfolio using this hedge ratio.
Test for Stationarity: Apply the ADF test to the residuals of the portfolio to check for stationarity.
Understanding the distinction between mean reversion and cointegration is crucial for developing robust trading strategies. While mean reversion indicates that a single asset's price tends to revert to its mean, cointegration pertains to a relationship between two or more non-stationary time series. This indicates that the price spread between the assets will revert to a mean.

this code identifies mean reverting stocks

```
import yfinance as yf
import pandas as pd
import statsmodels.api as sm
```

```python
from statsmodels.tsa.stattools import adfuller
from datetime import datetime, timedelta

def fetch_stock_data(symbol):
    try:
        data = yf.download(symbol, start='2022-01-01', end='
    2023-01-01')
        if data.empty:
            raise ValueError(f"No data found for symbol: {symbol}")
        return data
    except Exception as e:
        print(f"Error fetching data for {symbol}: {e}")
        return pd.DataFrame()

def adf_test(timeseries):
    result = adfuller(timeseries)
    return result[1]

def check_mean_reversion(stock_list):
    mean_reverting_stocks = []
    for stock in stock_list:
        data = fetch_stock_data(stock)
        if data.empty:
            continue
        closing_prices = data['Close']
        if closing_prices.empty:
            continue
        p_value = adf_test(closing_prices)
        if p_value < 0.05:
            mean_reverting_stocks.append(stock)
    return mean_reverting_stocks

if __name__ == "__main__":

    mean_reverting_stocks = check_mean_reversion(selected_stocks)
    print("Selected stocks and ETFs:", selected_stocks)
    print("Mean-reverting stocks:", mean_reverting_stocks)
```

11 Implementing Mean Reversion Strategies

11.1 Mean-Reverting Portfolio with Fixed Shares

Concept: Constructing a mean-reverting portfolio with a fixed number of shares means that once you decide on the quantity of each stock, it remains constant throughout the trade.

Hedge Ratios: To determine the hedge ratios (the proportions in which each stock is included in the portfolio), use the price series of the stocks. These ratios help balance the portfolio to achieve mean reversion.

11.2 Mean-Reverting Portfolio with Fixed Market Values

Concept: In this approach, each stock's contribution to the portfolio is based on its market value, which is kept constant during the trade.

Log Price Series: The hedge ratios are determined using the log price series of the stocks. Log prices help stabilize the ratios by accounting for exponential growth patterns in stock prices. Log Price spreads are useful when asset prices exhibit multiplicative changes rather than additive changes. Can stabilize variance and improve statistical properties for mean reversion analysis.

11.3 Ratio for Currency Pairs

Indicator: Instead of using spreads (differences in prices), the ratio of two currency prices is often a better indicator for trading currency pairs.

Reason: Ratios can more accurately reflect the relative value between currencies, which is crucial for spotting trading opportunities.

11.4 Hedge Ratio, Mean, and Standard Deviation Variability

Problem: Over time, the hedge ratio, mean, and standard deviation of a spread might change, which can affect the strategy's effectiveness.

Solutions: Moving Look-Back Period: Continuously update the statistical parameters (mean and standard deviation) using a fixed window of recent data.

Kalman Filter: A more sophisticated method to dynamically adjust the estimates based on new data, providing a way to track and predict changes more accurately.

11.5 Bollinger Bands with Scaling-In

Bollinger Bands: The linear strategy is parameterless but impractical because it doesn't limit the maximum capital deployed. There is no upper limit on the temporary deviation of the price from its average, leading to potential risk. Bollinger Bands address the limitations by defining entry and exit points based on standard deviations from the mean.

Entry and Exit Z-scores:

EntryZscore: The number of standard deviations the price must deviate from the mean to enter a position.

ExitZscore: The number of standard deviations from the mean to exit a position, which is less than the EntryZscore.

Trade Execution:

Long Entry: When the Z-score is less than the negative EntryZscore.

Long Exit: When the Z-score is greater than or equal to the negative ExitZscore.

Short Entry: When the Z-score is greater than the positive EntryZscore.

Short Exit: When the Z-score is less than or equal to the positive ExitZscore.

this code give the buy and sell signals for mean reverting stocks using bollinger bands

```
import yfinance as yf
import pandas as pd
import numpy as np

def fetch_stock_data(symbols, start_date='2022-01-01', end_date='
    2023-01-01'):
    all_data = {}
    for symbol in symbols:
        try:
            data = yf.download(f"{symbol}.NS", start=start_date,
    end=end_date)
            if data.empty:
                raise ValueError(f"No data found for symbol: {
    symbol}")
            all_data[symbol] = data
        except Exception as e:
            print(f"Error fetching data for {symbol}: {e}")

    return all_data

def calculate_moving_average_std(stock_data, window=20):
    stock_data['mean'] = stock_data['Close'].rolling(window=window)
    .mean()
    stock_data['std'] = stock_data['Close'].rolling(window=window).
    std()
    return stock_data

def simulate_mean_reversion(stock_data, entry_threshold=1.0,
    exit_threshold=0.5, initial_capital=100000):
    capital = initial_capital
    positions = 0
    trades = []
    total_profit = 0

    for date, row in stock_data.iterrows():
        if positions == 0:
            if row['Close'] < (row['mean'] - entry_threshold * row[
    'std']):
```

```
                buy_price = row['Close']
                shares = capital // buy_price
                capital -= shares * buy_price
                positions += shares
                trades.append((date, 'BUY', buy_price, shares))
            elif row['Close'] > (row['mean'] + entry_threshold *
    row['std']):
                sell_price = row['Close']
                shares = capital // sell_price
                capital += shares * sell_price
                positions -= shares
                trades.append((date, 'SELL', sell_price, shares))
        elif positions > 0:
            if row['Close'] >= (row['mean'] - exit_threshold * row[
    'std']):
                sell_price = row['Close']
                capital += positions * sell_price
                profit = positions * (sell_price - buy_price)
                total_profit += profit
                positions = 0
                trades.append((date, 'SELL', sell_price, positions)
    )
        elif positions < 0:
            if row['Close'] <= (row['mean'] + exit_threshold * row[
    'std']):
                buy_price = row['Close']
                capital -= abs(positions) * buy_price
                profit = abs(positions) * (sell_price - buy_price)
                total_profit += profit
                positions = 0
                trades.append((date, 'BUY', buy_price, positions))

    if positions != 0:
        final_price = stock_data['Close'].iloc[-1]
        if positions > 0:
            capital += positions * final_price
            profit = positions * (final_price - buy_price)
        elif positions < 0:
            capital -= abs(positions) * final_price
            profit = abs(positions) * (sell_price - final_price)
        total_profit += profit

    final_value = capital
    total_return = (final_value - initial_capital) /
    initial_capital

    return trades, final_value, total_return, total_profit


data = fetch_stock_data(mean_reverting_stocks)

entry_threshold = 2.0
exit_threshold = 1.0

results = {}
for stock in mean_reverting_stocks:
    stock_data = data[stock]
```

```
    stock_data = calculate_moving_average_std(stock_data)
    trades, final_value, total_return, total_profit =
    simulate_mean_reversion(stock_data, entry_threshold,
    exit_threshold)
    results[stock] = {
        'trades': trades,
        'final_value': final_value,
        'total_return': total_return,
        'total_profit': total_profit
    }

for stock, result in results.items():
    print(f"Stock: {stock}")
    print(f"Trades: {result['trades']}")
    print(f"Final Value: {result['final_value']}")
    print(f"Total Return: {result['total_return'] * 100:.2f}%")
    print(f"Total Profit: {result['total_profit']:.2f}")
    print("\n")
```

Scaling-In: Gradually increase your position as the price moves further from the mean. This approach can help manage risk and improve entry points.

11.6 Scaling-In for Live Trading

Live Trading Benefits: Although scaling-in might not always show optimal results in backtests, it can be useful in live trading.

Dynamic Markets: Volatilities and probabilities change in real-time, and scaling-in allows traders to adapt to these changes, potentially improving trade outcomes.

11.7 Kalman Filter for Dynamic Updates

Dynamic Price Updates: Use the Kalman filter to continuously estimate the dynamic hedge ratio involving iterative updates of expected values and covariances based on observable data (price series) and specified matrices (state transition and observation models). In Market-Making Model, we use the Kalman Filter to dynamically estimate the mean price and standard deviation of a single mean-reverting price series. The model helps in updating the estimate of the mean price and standard deviation based on new observations, thereby informing trading decisions. This real-time updating mechanism can enhance the accuracy of trading signals and adjustments.

11.8 Data Errors Impact on Backtests

Mean-Reverting Strategies: These strategies are particularly sensitive to data errors because they rely on precise price relationships.

Momentum Strategies: These strategies are less affected by data errors since they focus on the direction of price movements rather than exact price levels.

Sensitivity of Spread-Based Strategies

Spreads and Data Errors: Spread-based strategies (trading based on the difference between prices) can be highly sensitive to even small data inaccuracies. Such errors can distort the perceived relationship between instruments, leading to incorrect trading decisions.

12 Mean Reversion of Stocks and ETFs

The stock market is an ideal environment for applying mean-reverting trading strategies due to the large number of available instruments and their tendency to revert to mean values in the short term.

Challenges and Opportunities in Mean Reversion Strategies

Diversification:

Stocks can be paired across different sectors, enhancing diversification. Despite their long-term geometric random walk behavior, many stocks exhibit short-term mean-reverting properties under normal conditions.

Practical Difficulties:

Applying generic mean reversion techniques to stocks and ETFs can be challenging due to market dynamics and data quality issues. Strategies must be carefully designed to account for transaction costs and survivorship bias, which are often omitted in backtesting.

Mean Reversion in ETFs:

Simple mean-reverting strategies often perform better for ETF pairs and triplets compared to individual stocks. ETFs typically exhibit more stable mean-reverting behavior due to their diversified nature.

Mean Reversion Strategies for Stocks and ETFs

Short-Term Mean Reversion:

Most stocks exhibit mean-reverting properties in the short term. Strategies can exploit this behavior by identifying and trading deviations from the mean price.

Index Arbitrage:

Relies on the cointegration of stocks versus futures or ETFs. Traditional index arbitrage offers limited profit opportunities, but modified strategies can enhance returns.

Cross-Sectional Mean Reversion:

Involves the reversion of cumulative returns of instruments within a basket to the mean cumulative return of the basket. Easier to implement than time series mean reversion due to the prevalent cross-sectional mean-reverting patterns in baskets of stocks.

12.1 Challenges in Stock Pair Trading

Non-Stationarity of Stock Prices:

Individual stocks rarely meet the stationarity criteria, often following a geometric random walk. Once a stock deviates from its mean, it seldom reverts. Even logically paired stocks (e.g., Exxon vs. Chevron) often fail to maintain cointegration out-of-sample. The fortunes of individual companies can diverge due to management decisions, competition, and sector-specific dynamics.

Large Portfolio of Pairs:

Trading a large number of pairs does not necessarily mitigate risk. The small profits from successful pairs are often offset by substantial losses from pairs that fail. The expected return of individual pairs in out-of-sample periods is generally not positive, undermining the benefits of diversification.

Technical Difficulties

Short-Sale Constraints:

Shorting hard-to-borrow stocks is risky. A short squeeze can occur if positive news causes a stock price to spike, forcing lenders to recall their shares, resulting in forced liquidation at a loss.

Intraday Trading Issues:

With decreasing profit margins, traders aim to capture the best prices intraday, avoiding overnight positions that are susceptible to fundamental valuation changes. However, the national best bid and offer (NBBO) sizes for stocks have become very small, complicating intraday execution. Dark pools, iceberg orders, and high-frequency trading contribute to this challenge. Slippage and Execution:

Market orders based on NBBO prices can suffer significant slippage. Traders often need to use limit orders, actively managing cancellations and resubmissions, which introduces operational complexities.

12.2 Trading ETF Pairs (and Triplets)

Trading ETF pairs offers significant advantages over trading stock pairs due to their more stable and predictable nature. ETFs, being baskets of stocks, generally exhibit slower changes in their fundamental economics compared to individual stocks. This stability makes them more suitable for mean reversion strategies.

Trading mechanics for ETFs are similar to stock pairs but with generally larger NBBO sizes, reducing the impact of short-sale constraints.

12.3 Intraday Mean Reversion: Buy-on-Gap Model

Strategy Overview

This strategy aims to exploit intraday mean reversion by identifying stocks that experience a significant gap down at the open and are likely to recover during the trading day. The core idea is that panic selling at the open can lead to temporary price drops, which tend to revert over the course of the day.

Strategy Rules

Stock Selection at Market Open:

Identify stocks whose return from the previous day's low to the current day's open is lower than one standard deviation of their daily close-to-close returns over the last 90 days. These are the stocks that have "gapped down."

Filter Stocks by Moving Average:

Further narrow the list by selecting only those stocks whose opening prices are higher than their 20-day moving average of closing prices.

Buying Criteria:

From the filtered list, select the 10 stocks with the lowest returns from their previous day's lows. If there are fewer than 10 stocks, buy all available stocks that meet the criteria.

Liquidation:

Sell all positions at the market close.

Challenges and Practical Considerations

Using Open Prices for Signals:

The strategy requires the use of preopen prices to determine trading signals since actual open prices are not available before trading begins. The difference between preopen signals and actual open prices introduces "signal noise," but this is typically manageable.

Backtesting Accuracy:

The backtest should use primary exchange prices rather than consolidated prices to avoid inaccuracies discussed in earlier.
Short Sale Constraints:

A similar strategy can be applied to short stocks that gap up, but short-selling comes with additional risks and constraints, such as borrow availability and short squeezes.

Variations and Enhancements

Simultaneous Long and Short:

Trade both the long and short versions of the strategy simultaneously to potentially enhance returns.

Hedging:

Implement a hedged version by being long stocks and short stock index futures to mitigate market risk.

Sector Restrictions:

Limit the number of stocks from the same sector to diversify risk.

Extended Buying Period:

Extend the buying period beyond the market open to capture more opportunities.

Profit Caps:

Impose intraday profit caps to lock in gains and reduce risk.

12.4 Arbitrage between an ETF and Its Component Stocks

To implement the arbitrage strategy between an ETF and its component stocks, we follow these steps:

Data Collection: Gather historical price data for the ETF and its component stocks.

Cointegration Test: Use the Johansen test to find stocks that are cointegrated with the ETF.

Portfolio Construction: Use a genetic algorithm to optimize hedge ratios for the long-only portfolio to minimize the difference between the portfolio price series and the ETF price series.

Backtesting: Implement a linear mean reversion strategy based on the portfolio's deviation from the ETF.

12.5 Cross-Sectional Mean Reversion Strategy

In a cross-sectional mean reversion strategy, we focus on the relative returns of stocks within a certain universe (e.g., S&P 500) rather than their absolute returns. The goal is to exploit the short-term reversals in relative performance among stocks.

Key Concepts:

Relative Return: The return of a stock compared to the average return of all stocks in the universe.

Weight Allocation: The capital allocated to each stock is based on its relative return, with the aim of buying underperforming stocks and shorting overperforming stocks.

Daily Rebalancing: The portfolio is rebalanced daily based on the updated relative returns.

Steps:

Calculate Daily Returns: Compute the daily returns for each stock in the universe.

Compute Average Daily Return: Calculate the average daily return of all stocks.

Determine Weights: Calculate the weight for each stock based on its relative return.

Normalize Weights: Ensure the total gross capital invested is constant by normalizing the weights.

12.6 Intraday Linear Long-Short Model on Stocks

This enhanced strategy involves using the return from the previous close to today's open to determine the weights for entry at the open. Positions are then liquidated at the market close, turning it into an intraday strategy.

Key Steps:

Calculate Returns from Previous Close to Today's Open: Compute the returns based on the previous day's closing prices and today's opening prices.

Compute Market Return: Calculate the average return from the previous close to today's open for all stocks.

Determine Weights: Calculate the weight for each stock based on its relative return to the market return.

Normalize Weights: Ensure the total capital invested is constant by normalizing the weights.

Calculate Intraday Returns: Apply the strategy over the intraday period from open to close and evaluate performance.

13 Mean Reversion of Currencies and Futures

spot price : current market price

roll return : profit or loss from rolling over a futures contract to the next expiration

Backwardation: When the futures price is lower than the expected future spot price. This typically occurs when the current demand is higher than the future supply.

Contango: When the futures price is higher than the expected future spot price. This is often due to storage costs and other carrying costs being priced into the futures contracts.

13.1 Trading Currency Cross-Rates

Trading currency cross-rates involves forming a portfolio of foreign currencies that are expected to maintain a stable relationship over time. This stability can be due to similar economic fundamentals between the countries of the currencies involved. For example, the Australian dollar (AUD) and the Canadian dollar (CAD) might cointegrate due to the similarities in the economies of Australia and Canada, such as their reliance on natural resource exports.

Advantages of Trading Currency Pairs:

Higher Liquidity: Currencies typically have high liquidity, especially in terms of the best bid/ask sizes, reducing transaction costs.

Higher Leverage: Currency trading allows for higher leverage, although this also increases risk.

No Short-Sale Constraints: Unlike some stock markets, there are no restrictions on short-selling currencies.

24/5 Trading: Currency markets are open 24 hours a day, five days a week, providing more trading opportunities and the ability to set stop losses effectively.

Currency Trading Mechanics:

Base and Quote Currency: In a currency pair like AUD.ZAR, AUD is the base currency, and ZAR is the quote currency. If AUD.ZAR is quoted at 9.58, it means 1 AUD is worth 9.58 ZAR.

Synthetic Pairs: If a direct cross-rate (e.g., AUD.ZAR) is not available, traders can create a synthetic pair by trading related pairs through a common base currency like USD or EUR. For instance, trading AUD.USD and USD.ZAR to achieve the equivalent of trading AUD.ZAR.

13.2 Pair Trading USD.AUD versus USD.CAD Using the Johansen Eigenvector

Strategy Setup:

Use daily price series for USD.AUD and USD.CAD. Apply a fixed training set of 250 days for computing the hedge ratio.

Use a look-back period of 20 days for calculating the moving average and standard deviation.

Data Preparation:

Convert USD.CAD to CAD.USD for consistency. Create a matrix y with the price series of AUD.USD and CAD.USD.

Hedge Ratio Calculation:

Use the Johansen cointegration test on the log prices of AUD.USD and CAD.USD to determine the hedge ratio. The hedge ratio is the eigenvector from the Johansen test.

Position Sizing and Trading:

Calculate the market value of a unit portfolio over the look-back period.

Compute the moving average and standard deviation of unit portfolio.

Determine the z-score and the number of units to trade.

Positions in AUD.USD and CAD.USD are determined by the number of units and the hedge ratio.

Profit and Loss (P&L) Calculation:

Compute the daily P&L as the sum of the market value of each instrument times their returns.

Calculate the daily return of the portfolio as the P&L divided by the total gross market value of the portfolio at the end of the previous day.

13.3 Rollover Interests in Currency Trading

Rollover Interest Definition:

When trading currencies, holding a position overnight (past 5:00 p.m. ET) incurs interest costs or earns interest depending on the interest rates of the currencies involved.

If you hold a long position in a currency pair B.Q overnight, the interest differential is iB - iQ, where iB is the daily interest rate for currency B and iQ is for currency Q.

Interest Differential:

If iQ (interest rate of the quote currency) is greater than iB (interest rate of the base currency), then you have a debit interest (you pay interest).

Multiple Days:

If a position is held past 5 p.m. ET and the next trading day is a weekend or holiday, the rollover interest is multiplied by one plus the number of days the market remains closed. This accounts for the fact that you are effectively holding the position for multiple days.

13.4 Trading Futures Calendar Spread

Calendar Spread: A trading strategy involving futures contracts with different expiration dates (or maturities). These contracts have different prices and slightly different returns, creating opportunities for trading spreads between them.

Roll Return:

This is the return that arises because futures contracts with different maturities have different prices. Even if the underlying spot price remains unchanged, a futures position will have a non-zero return because futures prices converge towards the spot price as they approach expiration.

Backwardation: Near-expiration contracts have higher prices than far-expiration contracts, leading to positive roll returns.

Contango: Near-expiration contracts have lower prices than far-expiration contracts, leading to negative roll returns.

13.5 Estimating Spot and Roll Returns Using the Constant Returns Model

The process involves using linear regression to estimate the spot and roll returns of various futures contracts. This approach assumes that spot and roll returns are constant over time.

Spot Return Estimation:

Regress the log of the spot prices against time to estimate the average annualized spot return.

Roll Return Estimation:

Pick a fixed point in time and regress the futures prices of the nearest contracts against their time to maturity. This helps estimate the roll return, which will depend on the selected time and the available contracts.

13.6 Calendar Spreads

A calendar spread involves holding a long position in one futures contract and a short position in another futures contract with the same underlying asset but different expiration months. Given that both legs track the same underlying asset, calendar spreads might seem like ideal candidates for mean reversion. However, roll returns complicate this intuition. Calendar spreads is influenced solely by the roll return and not by the spot price.

Calendar spreads can be suitable candidates for mean-reversion trading strategies, primarily driven by the roll returns rather than the spot prices. Accurate estimation of the roll returns and understanding the market dynamics are crucial for the success of the strategy.

13.7 Futures Intermarket Spreads

Intermarket spreads involve trading pairs of futures contracts based on different underlying assets. Finding such spreads that exhibit mean-reverting properties can be challenging. However, there are some markets where intermarket spreads are closely related and provide potential opportunities for mean-reversion trading strategies.

13.8 Volatility Futures versus Equity Index Futures

Traders often observe that volatility is inversely correlated with the stock equity market index: when the market declines, volatility tends to rise sharply, and to a lesser extent, when the market rises, volatility tends to decrease. This relationship can be visualized by plotting the prices of E-mini S&P 500 futures (ES) against VIX futures (VX). Linear Regression: Established a linear relationship between ES and VX futures

Mean-Reverting Strategy: Constructed a mean-reverting trading strategy based on deviations from the regression model.

14 Interday Momentum Strategies

Momentum in trading refers to the phenomenon where asset prices tend to continue moving in the same direction for some time. There are four main causes of momentum:

Persistence of Roll Returns: For futures, the roll returns tend to persist, especially in their signs.

Slow Diffusion of Information: New information is gradually analyzed and accepted by the market.

Forced Transactions: Certain funds are forced to buy or sell assets, creating momentum.

Market Manipulation: High-frequency traders can manipulate market movements.

Types of Momentum

Researchers classify momentum into two types:

Time Series Momentum: Past returns of a single asset predict its future returns. If a price has been rising, it is likely to continue rising.

Cross-Sectional Momentum: Relative performance of assets predicts future performance. Assets that have performed well compared to others will likely continue to do so.

Measuring Time Series Momentum

To measure time series momentum, you can calculate the correlation coefficient between past and future returns of an asset.

Select Time Frames: Choose specific look-back and holding periods. The look-back period is the historical period used to calculate past returns, and the holding period is the future period for which you want to predict returns.

Calculate Returns: Compute the returns for the look-back period and the future holding period.

Correlation Coefficient: Calculate the correlation coefficient between these past and future returns. The correlation coefficient (ranging from -1 to 1) measures the strength and direction of the relationship between the two sets of returns.

P-Value: The p-value indicates the probability that the observed correlation occurred by chance. A low p-value (typically less than 0.05) suggests that the correlation is statistically significant.

Optimal Periods: Identify the pair of look-back and holding periods that produce the highest positive correlation. This optimal pair can be used for a momentum-based trading strategy.

Long-Term Trending Behavior

To detect long-term trending behavior and distinguish it from random price movements

The Hurst exponent (H) measures the tendency of a time series to either regress to the mean ($H < 0.5$), exhibit random walk behavior ($H \approx 0.5$), or show trending behavior ($H > 0.5$). A value of H greater than 0.5 indicates a tendency towards trending.

Variance Ratio Test: This test compares the variance of returns over different time intervals. If the price series is a random walk, the ratio of variances should be 1. A significant deviation from 1 suggests trending or mean-reverting behavior.

Serial Correlations in Futures Returns

The serial correlations in futures returns, particularly over longer time scales, can often be attributed to the persistence of roll returns. Futures contracts tend to stay in contango or backwardation for extended periods, creating a consistent roll return component. Meanwhile, spot returns can vary rapidly in both sign and magnitude. When the average roll returns dominate the total returns over a long period, we observe serial correlations in total returns, explaining the effectiveness of these momentum strategies.

Future versus ETF Arbitrage

In futures trading, total returns are composed of spot returns and roll returns. This can be exploited through arbitrage strategies between futures and ETFs, particularly when the roll returns are persistent.

Extracting Roll Returns

To extract roll returns, consider the following strategies based on the roll return's sign:

Contango (Negative Roll Return): Buy the underlying asset and short the futures.

Backwardation (Positive Roll Return): Short the underlying asset and buy the futures.

This works as long as the sign of the roll return doesn't change quickly, which is often the case.

Asynchronicity Issue

Futures (GC) and ETFs (GLD) often have different closing times. For example, GC's closing prices are recorded at 1:30 p.m. ET, while GLD's are recorded at 4:00 p.m. ET. Although this asynchronicity might affect some strategies, it does not impact strategies where trading signals are generated based on futures closing prices alone.

Volatility Futures versus Equity Index Futures:

The VX futures (CBOE Volatility Index futures) exhibit substantial roll returns, which can be as low as -50% annualized. These futures are also highly anti-correlated with the S&P 500 futures (ES), boasting a daily return correlation coefficient of -75%. Leveraging this anti-correlation and the disparity in roll return magnitudes, a momentum strategy can be developed to exploit these market characteristics.

This strategy, proposed by Simon and Campasano (2012), involves the following steps:

When in Contango:

Condition: If the price of the front contract of VX exceeds the VIX by 0.1 points times the number of trading days until settlement.

Action: Short 0.3906 front contracts of VX and short 1 front contract of ES.

Hold: Maintain the position for one day. When in Backwardation:

Condition: If the price of the front contract of VX is lower than the VIX by 0.1 points times the number of trading days until settlement.

Action: Buy 0.3906 front contracts of VX and buy 1 front contract of ES.

Hold: Maintain the position for one day.

Rationale

Contango and Backwardation: When the front contract price of VX is higher than the spot VIX price, the roll return is negative (contango), and vice versa for backwardation. The strategy involves taking positions based on the calculated roll return.

Hedge Ratio: The hedge ratio (0.3906 VX contracts for every ES contract) is determined by the regression fit between VX and ES prices.

The strategy focuses on managing risk and capturing potential gains based on the anticipated movements in VX futures.

Shorting or long both VX and ES futures simultaneously can be seen as a risk management approach. It aims to hedge against broader market movements while still benefiting from the anticipated changes in volatility (VX).

14.1 Cross-Sectional Strategies

The strategy aims to profit from roll returns by:

Buying futures contracts that are in backwardation (where futures prices are below expected spot prices).

Simultaneously shorting futures contracts that are in contango (where futures prices are above expected spot prices).

The expectation is that while spot price movements might cancel each other out due to positive correlations with economic indicators, roll returns remain favorable.

Factor-Based Approaches:

The strategy can be enhanced by incorporating factor models, such as fundamental factors (like earnings growth or book-to-price ratio) or statistical factors derived from techniques like Principal Component Analysis (PCA).

14.2 Mutual Fund Behavior and Stock Prices

Mutual funds, usually fully invested, adjust stock positions based on investor redemptions or inflows.

Large redemptions lead to selling stocks, which can depress prices.

Inflows result in increased stock holdings, boosting prices.

Impact on Stock Price Momentum:

Stocks held by funds facing redemptions may see negative returns due to selling pressure.

Stocks held by funds with inflows may experience upward momentum due to increased demand.

Construction of the PRESSURE Factor:

Researchers Coval and Stafford propose the PRESSURE factor to measure buying or selling pressure on stocks.

PRESSURE calculates the net percentage of funds experiencing significant redemptions or inflows, indicating the difference between funds selling and buying a stock.

Investment Strategies Based on PRESSURE Factor:

A market-neutral strategy involves shorting stocks with high selling pressure and buying those with high buying pressure.

15 Intraday Momentum Strategies

15.1 Opening Gap Strategy for Futures and Currencies

The momentum-based strategy involves buying when the instrument gaps up and shorting when it gaps down.

Entry Signal:

Longs: Buy when the open price (op) is greater than the previous day's high (hi), adjusted by a small multiple of the standard deviation.

Shorts: Short when the open price (op) is lower than the previous day's low (lo), adjusted by a small multiple of the standard deviation.

Positions:

Set to 1 for long positions and -1 for short positions based on the entry signals. Currency Strategy Adjustment

For currencies, the concept of daily "open" and "close" needs to be defined differently. In the example provided, the close is set at 5:00 p.m. and the open is set at 5:00 a.m.

Explanation for Gap Momentum

The overnight or weekend gap can trigger momentum due to the extended period without trading. The opening price often differs significantly from the closing price, leading to the triggering of stop orders at various levels. The cascading effect of these stop orders creates momentum, as additional stop orders further away from the open price are executed. Significant events occurring overnight can also contribute to this momentum.

15.2 News-Driven Momentum Strategy

The news-driven momentum strategy leverages the idea that news dissemination is often gradual, allowing traders to benefit from price movements in the initial moments following significant news events. One well-known example of this is the post-earnings announcement drift (PEAD) model, which has been studied extensively and remains profitable despite its long-standing recognition in financial markets.

Entry at Market Open: Enter a trade at the market open following an earnings announcement made after the previous day's close.

Trade Direction:

Buy if the stock's return is very positive.

Short if the stock's return is very negative.

Exit: Liquidate the position at the same day's close.

This approach does not require interpreting the earnings announcement as "good" or "bad," nor does it necessitate knowing whether the earnings exceeded or fell short of analysts' expectations. The strategy relies on market reactions to guide trading decisions.

15.3 Leveraged ETF Strategy

The concept revolves around maintaining a constant leverage ratio in a portfolio. This requirement to keep leverage constant leads to buying and selling actions based on the daily performance of the index, which in turn can induce momentum in the market.

Constant Leverage: Maintaining constant leverage in a portfolio requires buying or selling to adjust positions based on daily market performance.

Market Impact: Large rebalancing actions can create market momentum, especially near the market close.

Trading Strategy:

Entry Criteria:

Buy if the return from the previous day's close to 15 minutes before market close is greater than 2Sell if the return is smaller than -2Exit Criteria:

Exit the position at the market close.

15.4 High-Frequency Strategies

High-frequency momentum strategies involve trading based on short-term predictions of price movements derived from the order book's information. These strategies capitalize on the imbalance between bid and ask sizes to anticipate price changes.Research has shown a linear relationship between bid-ask size imbalance and short-term price changes, especially for lower-volume stocks.

Ratio Trade:

Join the bid when the bid size is significantly larger than the ask size.

When the price ticks up, sell at the best ask.

If the price doesn't move up, sell at the original bid to minimize losses.

Ticking (Quote Matching):

In markets with a bid-ask spread larger than two ticks, place a buy order at the best bid plus one tick.

If filled, place a sell order at the best ask minus one tick.

If the sell order isn't filled, sell at the original best bid, incurring a loss of one tick plus commissions.

Practical Considerations:

Commission Costs:

Ensure round-trip commissions are less than the spread minus two ticks for the ticking strategy to be profitable.

Market Conditions:

These strategies work best in highly liquid markets where bid-ask imbalances frequently lead to predictable price movements.

Risks in Ticking:

Front-Running: If the trader who placed the original best bid cancels their order, you may have to sell at a lower price.

Trap: The original bidder might have intended to sell at a better price than their own bid, leading to a loss when you have to sell.

Momentum Ignition and Flipping

Momentum Ignition:

Concept: Create the illusion of buying pressure by placing a large buy limit order at the best bid and a small sell limit order at the best ask.

Process: Traders buy at the ask price expecting an uptick. After the small sell order is filled, cancel the large buy order and potentially profit from subsequent selling pressure as traders exit their positions at a loss.

Flipping:

Risk: If someone fills your large buy order, you might have to sell at a loss.

Counter-Strategy: Sell to flippers if you suspect a large buy order is artificial, aiming to buy back at a lower price after they capitulate.

15.5 Impact of High-Frequency Strategies

Market Impact:

Reduced Quote Sizes: Traditional market makers avoid large quotes due to HFT strategies. This results in smaller National Best Bid and Offer (NBBO) sizes.

Fragmentation of Orders: Large institutional orders are broken into smaller "child orders" to avoid being exploited by high-frequency traders.

16 Clustering in Stock Analysis

Clustering Analysis:

Definition: The process of grouping stock symbols into clusters based on their similarity, using various metrics to measure this proximity or closeness.

Purpose: Understand the natural structure of data, summarize data, define useful subsets, or determine the belongingness of items to a set.

Stock Personality: Refers to the distinctive trading characteristics of a stock.

Proximity: Key factor for grouping stocks. Stocks with similar metrics are grouped together.

Algorithmic Responses: Stocks within the same cluster may respond similarly to trading algorithms, providing opportunities for strategy development.

Sector Membership: Clustering often correlates with sector membership, as stocks within the same sector exhibit similar behaviors.

16.1 Selecting a Cohort of Trading Stocks

A cohort in this context refers to a group of stock symbols that exhibit similar trading characteristics as defined by specific metrics.

Metrics for Selecting Trading Stocks

1. Price Tier Metric

Preference for Higher-Priced Stocks: Stocks priced between $75 to $100 are preferred due to their distinct ownership profile.

Capital Requirements: Ensure sufficient capital to trade 1000 share lots, but start with 500 share lots if capital is limited.

2. Volume Level Metric

Liquidity Proxy: Trade stocks with a three-session average volume of at least

1 million shares per day. Avoid stocks with less than 750,000 shares traded in any of the three sessions.

3. %RANGE Metric

Contextual Analysis: Compare %RANGE in the context of other stocks in the sector. Look for trends over a series of days. Intraday Indicator: %RANGE 200T should ideally be over 0.002 at least half the time.

Avoid Low %RANGE Stocks: Do not trade stocks with an end-of-day %RANGE of less than 0.002.

4. Traverse Metric

Movement Indicator: Measures the amount of 'movement' in the stock price end-of-day (EOD). This can define a 'roughness' characteristic.

Absolute Traverse: Measures price motion over nT (n = 100 and 200 ticks) and EOD.

17 Optimal Leverage

Determining the optimal leverage for a trading strategy or portfolio is crucial yet challenging. The goal is to maximize net worth or compounded growth rate while considering drawdowns and return volatilities. Various methods exist for computing optimal leverage, each with its own assumptions and limitations.

Methods for Computing Optimal Leverage

1. Kelly Formula (Gaussian Assumption)

The Kelly formula is a widely recognized method for determining optimal leverage under the assumption of Gaussian-distributed returns. It seeks to maximize the compounded growth rate, but requires the stringent assumption of normally distributed returns.

2. Max Drawdown Constraint

This method involves setting a maximum allowable drawdown, which serves as an additional constraint in the leverage optimization problem. The optimal leverage should ensure that the portfolio does not experience drawdowns exceeding this threshold.

3. Constant Leverage Application

Implementing constant leverage in practice can be counterintuitive. It requires adjusting the portfolio size in response to gains or losses to maintain a fixed leverage ratio. This can lead to behaviors like selling assets during a loss (to reduce exposure) and buying assets after a gain (to increase exposure).

18 References

1. Chan, E. P. *Algorithmic Trading: Winning Strategies and Their Rationale.*

2. Leshik, E., & Cralle, J. *An Introduction to Algorithmic Trading: Basic to Advanced Strategies.*

3. freeCodeCamp.org. *Data Analysis with Python Course - Numpy, Pandas, Data Visualization* [Video]. YouTube. https://www.youtube.com/watch?v=GPVsHOlRBBI

4. freeCodeCamp.org. *Algorithmic Trading Using Python - Full Course* [Video]. YouTube. https://www.youtube.com/watch?v=xfzGZB4HhEE

5. freeCodeCamp.org. *Python for Beginners – Full Course [Programming Tutorial]* [Video]. YouTube. https://youtu.be/eWRfhZUzrAc?si=9MzcTdX5n1HxSY22

6. Coursera. *Machine Learning Specialization.* https://www.coursera.org/specializations/machine-learning