Mean Reversion Strategies (Expanded Theoretical Explanation with Mathematical Expressions)

Mean reversion strategies are built on the statistical observation that prices and returns eventually revert to their long-term mean or equilibrium level. This behavior is typically modeled using **stochastic processes**, such as the **Ornstein-Uhlenbeck process**, which is a common model for mean-reverting behavior. In this process, prices tend to drift towards a central value, and deviations from this value are temporary.

Mean Reversion Formula

A basic mathematical expression for mean reversion can be expressed as:

$$X_{t+1} = X_t + \theta(\mu - X_t) + \epsilon_t$$

Where:

- X_t is the asset price at time t.
- μ is the long-term mean (average price).
- θ is the mean reversion speed, determining how fast the price reverts to the mean.
- ϵ_t is a random noise term, often assumed to follow a normal distribution, $\epsilon_t \sim N(0, \sigma^2)$.

The term $\theta(\mu - X_t)$ represents the pull of the price towards the mean. When X_t is far from μ, the price adjustment is stronger. As the price approaches the mean, the adjustment becomes smaller, reflecting the idea that large deviations are temporary and will be corrected over time.

In **mean reversion trading strategies**, this model suggests:

- **Buy** when the price falls significantly below the mean.
- **Sell** when the price rises significantly above the mean.

The goal is to exploit temporary price deviations with the expectation that prices will revert to their average value.

Z-score Strategy (Expanded Theoretical Explanation with Mathematical Expressions)

The **Z-score strategy** applies statistical measures to quantify how far the current price is from its historical mean in terms of standard deviations. The Z-score helps determine if the price is "overbought" or "oversold" relative to its historical distribution.

Z-score Calculation Formula

The Z-score Z_t is calculated as:

$$Z_t = \frac{X_t - \mu_t}{\sigma_t}$$

Where:

- Z_t is the Z-score at time t.
- X_t is the price at time t.
- μ_t is the moving average (mean) of prices over a specified lookback period (e.g., 20 days).
- σ_t is the standard deviation of prices over the same lookback period.

The Z-score effectively normalizes the deviation of the current price from the mean by expressing it in terms of

how many standard deviations X_t is away from μ_t. This helps assess whether a price is in an extreme condition:

- If $Z_t < -1.5$, the price is considered **oversold** (trading below its mean), implying a potential buying opportunity.
- If $Z_t > 1.5$, the price is considered **overbought** (trading above its mean), implying a potential selling opportunity.

Trading Rule Based on Z-score

$$\text{Position}_t = \begin{cases} 1 & \text{if } Z_t < -1.5 \text{ (Buy Signal)} \\ -1 & \text{if } Z_t > 1.5 \text{ (Sell Signal)} \\ 0 & \text{otherwise (Hold)} \end{cases}$$

Where:

- **1** represents a long position (buy).
- **-1** represents a short position (sell).
- **0** represents no position (hold).

This approach assumes that extreme price movements are temporary and will revert to the mean, making it possible to profit by betting on the price correction.

Backtesting and Cumulative Return Graphs

Z-score Plot: The first plot visualizes the Z-score over time and highlights the buy and sell thresholds. Whenever the Z-score crosses these thresholds, it triggers buy (below -1.5) or sell (above 1.5) signals.

Cumulative Return Plot: The second plot compares the cumulative returns of the Z-score-based strategy against a buy-and-hold benchmark strategy. It shows how the strategy performs relative to simply holding the asset throughout the period.

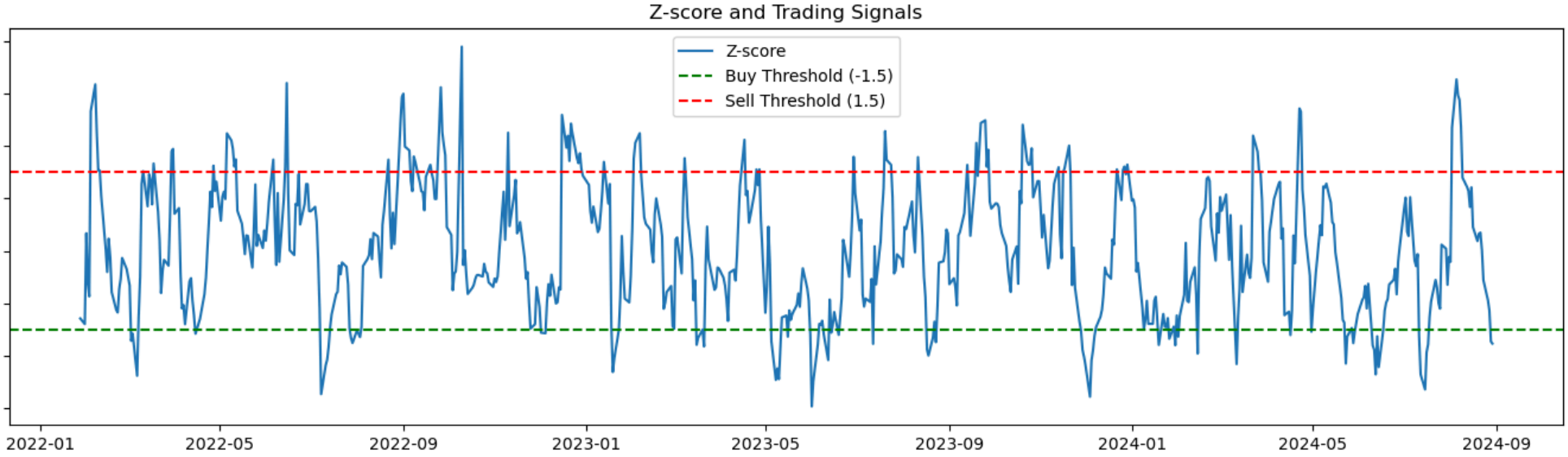
By analyzing the performance of the strategy through this backtest, one can evaluate the effectiveness of mean reversion and Z-score strategies in real market conditions.

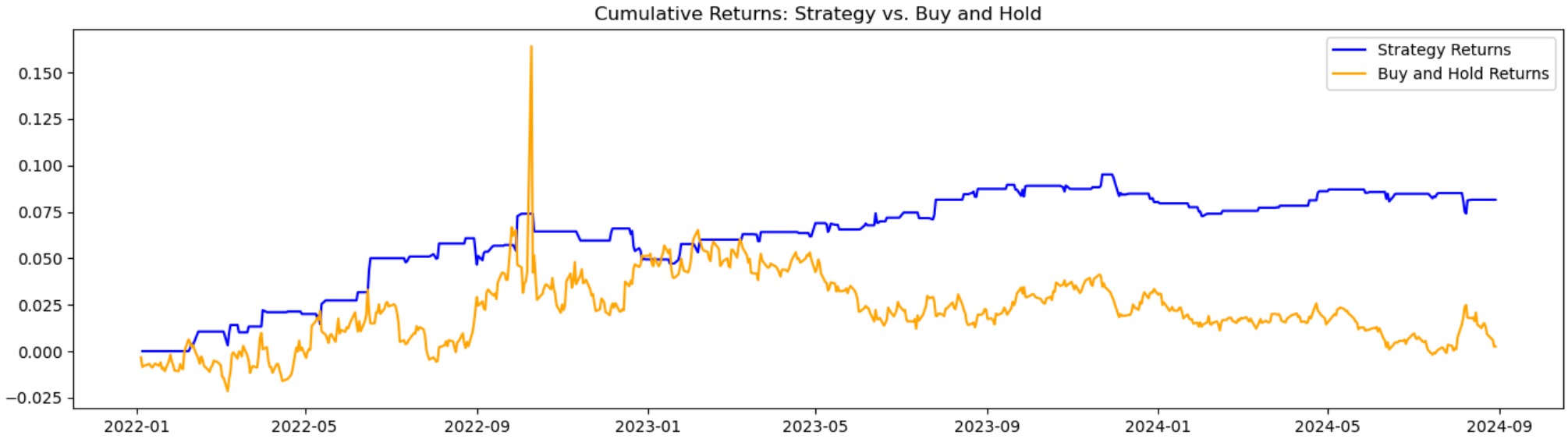
In [1]:

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

# Fetch data for EURGBP
ticker = 'EURGBP=X'
start_date = '2022-01-01'
end_date = '2024-08-30'
df = yf.download(ticker, start=start_date, end=end_date)

# Calculate mean and standard deviation (20-day rolling window)
window = 20
df['Mean'] = df['Close'].rolling(window=window).mean()
df['Std'] = df['Close'].rolling(window=window).std()

# Calculate Z-score
df['Z-score'] = (df['Close'] - df['Mean']) / df['Std']

# Define entry and exit thresholds
entry_threshold = -1.5  # Buy when Z-score < -1.5
exit_threshold = 1.5    # Sell when Z-score > 1.5

# Generate trading signals based on Z-score
df['Position'] = 0  # Default position is 0 (no position)
df.loc[df['Z-score'] < entry_threshold, 'Position'] = 1   # Long position (Buy)
df.loc[df['Z-score'] > exit_threshold, 'Position'] = -1   # Short position (Sell)
df['Position'] = df['Position'].shift()  # Shift position by one day

# Backtest - calculate daily returns
df['Strategy_Returns'] = df['Position'].shift(1) * df['Close'].pct_change()

# Calculate cumulative returns for strategy and buy-and-hold benchmark
df['Cumulative_Strategy_Returns'] = (1 + df['Strategy_Returns']).cumprod() - 1
df['Cumulative_Buy_and_Hold_Returns'] = (1 + df['Close'].pct_change()).cumprod() - 1

# Plot Z-score and trading signals
plt.figure(figsize=(14, 8))

plt.subplot(2, 1, 1)
plt.plot(df.index, df['Z-score'], label='Z-score')
plt.axhline(entry_threshold, color='green', linestyle='--', label='Buy Threshold (-1.5)')
```

```
plt.axhline(exit_threshold, color='red', linestyle='--', label='Sell Threshold (1.5)')
plt.title('Z-score and Trading Signals')
plt.legend()

# Plot cumulative returns for strategy and buy-and-hold
plt.subplot(2, 1, 2)
plt.plot(df.index, df['Cumulative_Strategy_Returns'], label='Strategy Returns', color='blue')
plt.plot(df.index, df['Cumulative_Buy_and_Hold_Returns'], label='Buy and Hold Returns', color='orange')
plt.title('Cumulative Returns: Strategy vs. Buy and Hold')
plt.legend()

plt.tight_layout()
plt.show()
```

```
[**********100%**********]  1 of 1 completed
```


Z-score and Trading Signals
Z-score
Buy Threshold (-1.5)
Sell Threshold (1.5)
4
3
2
1
0
−1
−2
−3
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01
2024-05
2024-09


Cumulative Returns: Strategy vs. Buy and Hold
Strategy Returns
Buy and Hold Returns
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01
2024-05
2024-09

In [ ]: