ES 5-Year Data Analysis

Project Introduction

This project provides an in-depth analysis of the ES index data over a five-year period. By leveraging historical price and volume data, the analysis seeks to uncover key trends, volatility patterns, and potential future price movements. The project utilizes various statistical methods, visualization techniques, and predictive modeling to achieve the following objectives:

- Understand the historical price behavior and trading volume trends of the ES index.
- Identify periods of high volatility and significant market movements.
- Develop predictive models to forecast future prices.
- Provide actionable insights and recommendations based on the findings.

The analysis will be useful for investors, financial analysts, and stakeholders interested in gaining insights into the ES market behavior.

In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.arima.model import ARIMA
import warnings

# Suppress specific warnings
warnings.filterwarnings('ignore')
```

Data Preprocessing

In this section, we preprocess the ES data to ensure it is ready for analysis. We convert the time column to a datetime format, set the frequency, and aggregate the data to a daily frequency. This prepares the dataset for exploratory analysis and modeling.

In [2]:
```python
# Load ES data
es_data = pd.read_csv('ES_5Years_8_11_2024.csv')
es_data['Time'] = pd.to_datetime(es_data['Time'], format='%m/%d/%Y %H:%M')
es_data.set_index('Time', inplace=True)

# Aggregate to daily frequency
es_daily = es_data.resample('D').agg({
    'Open': 'first',
    'High': 'max',
    'Low': 'min',
    'Close': 'last',
    'Volume': 'sum'
})

# Ensure the date index has a frequency set
es_daily.index = es_daily.index.to_period('D')

# Calculate Daily Returns
es_daily['Daily Return'] = es_daily['Close'].pct_change()

es_daily.head()
```

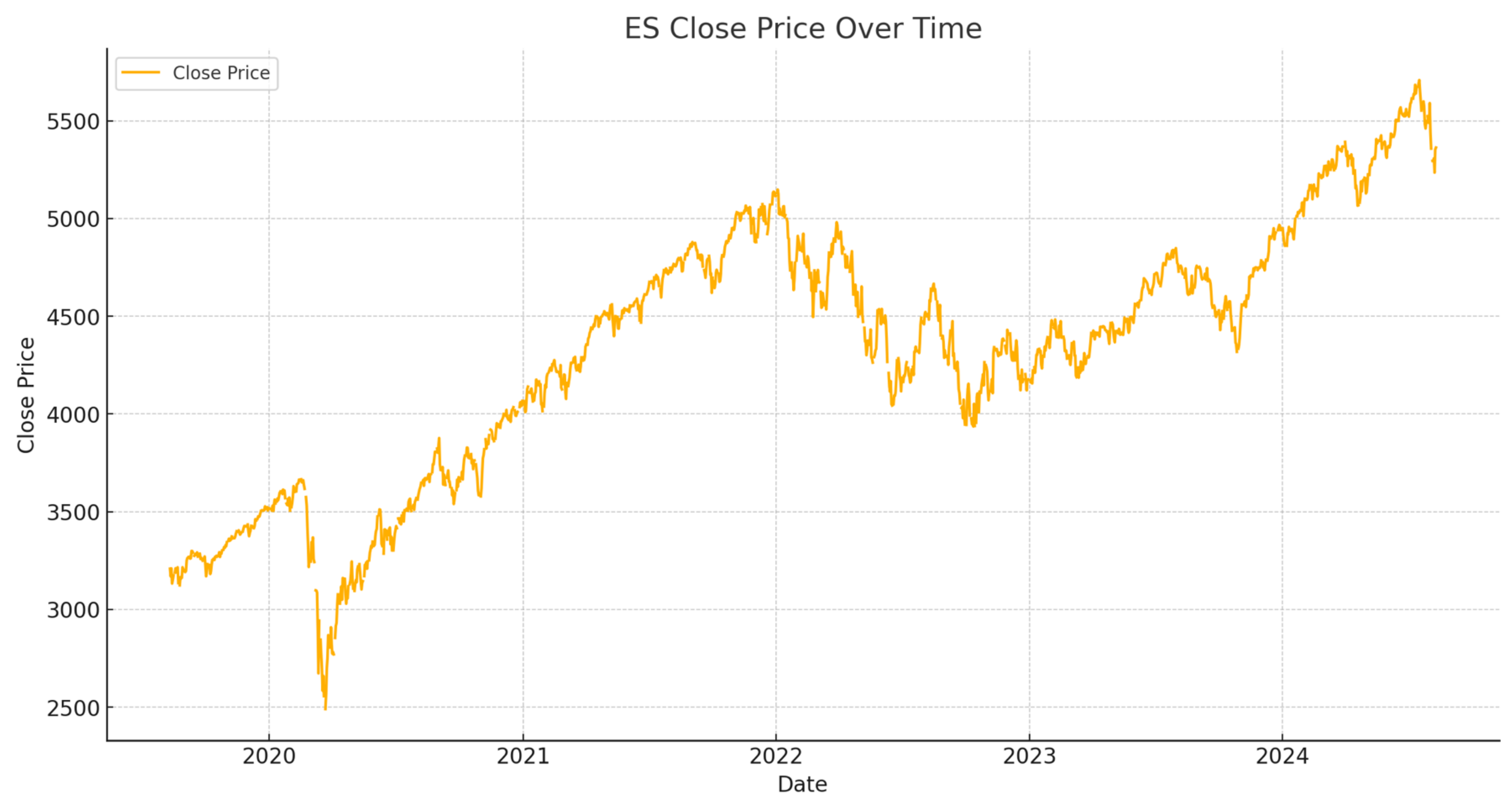
Out[2]:

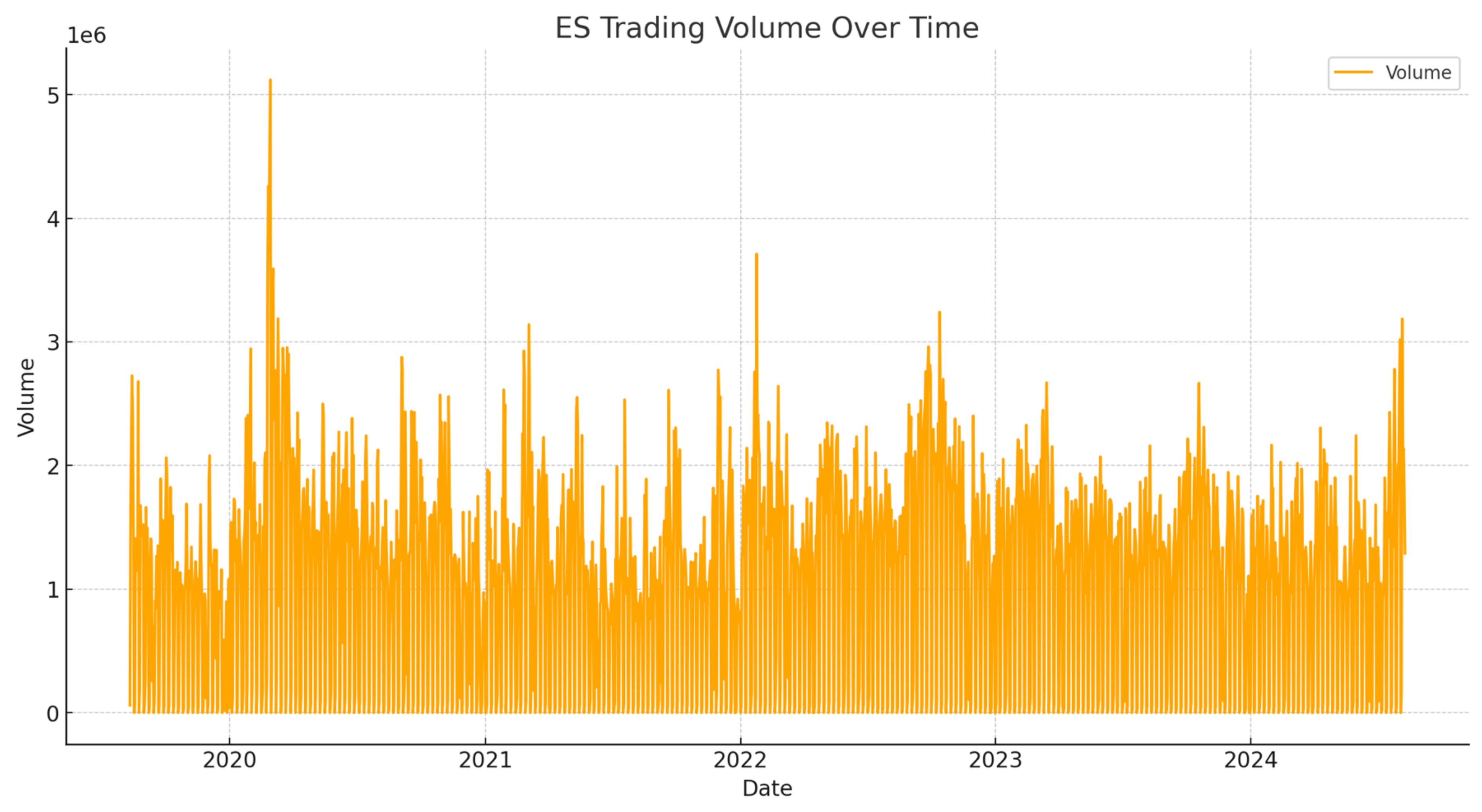
Time	Open	High	Low	Close	Volume	Daily Return
2019-08-11	3201.50	3210.75	3195.75	3208.25	60219	NaN
2019-08-12	3208.25	3213.75	3155.25	3168.50	1669830	-0.012390
2019-08-13	3168.25	3227.00	3149.50	3209.50	2137261	0.012940
2019-08-14	3209.25	3216.00	3116.50	3132.00	2727073	-0.024147
2019-08-15	3132.00	3154.50	3100.50	3150.50	2429154	0.005907

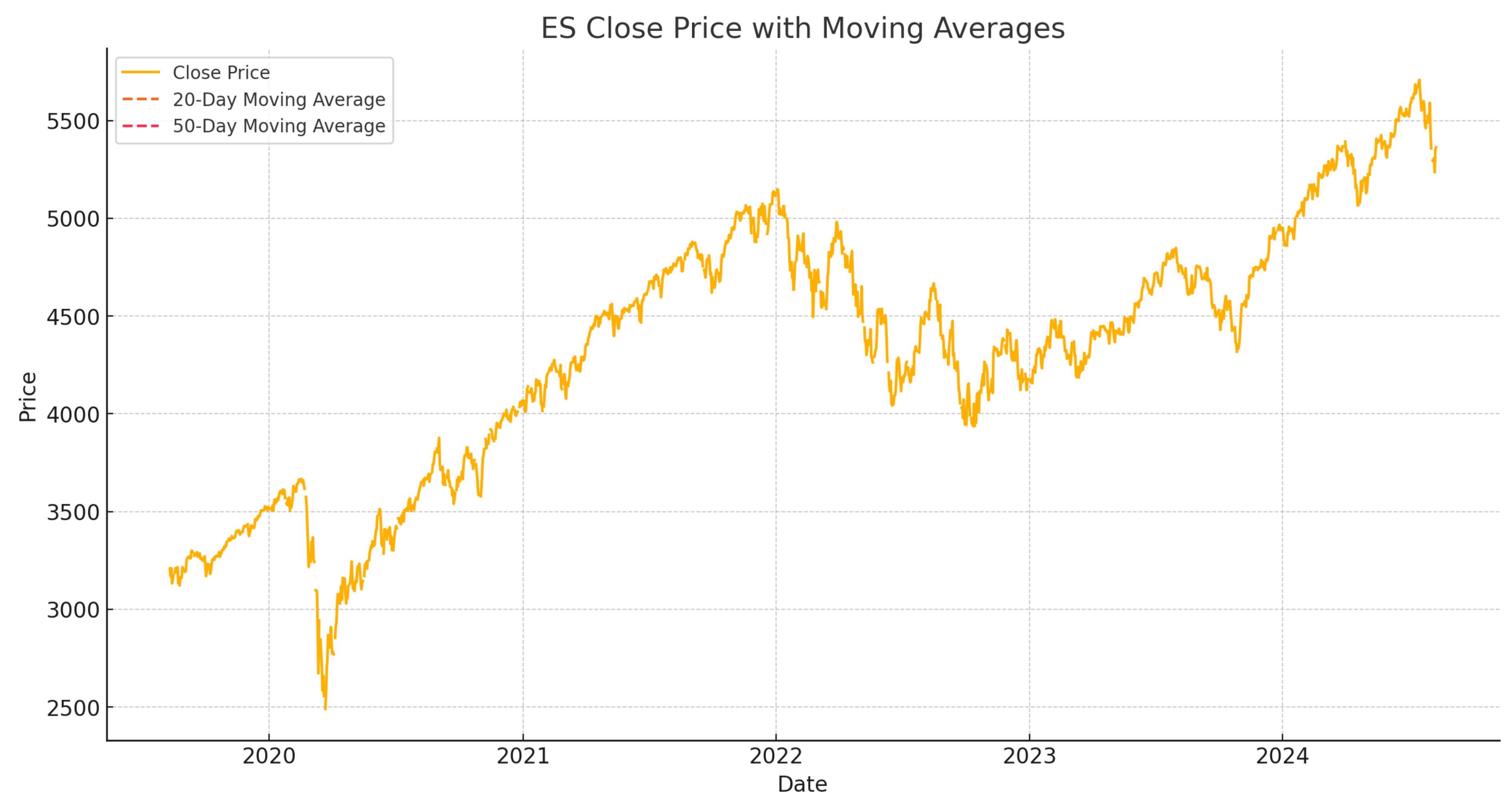
Exploratory Data Analysis (EDA)

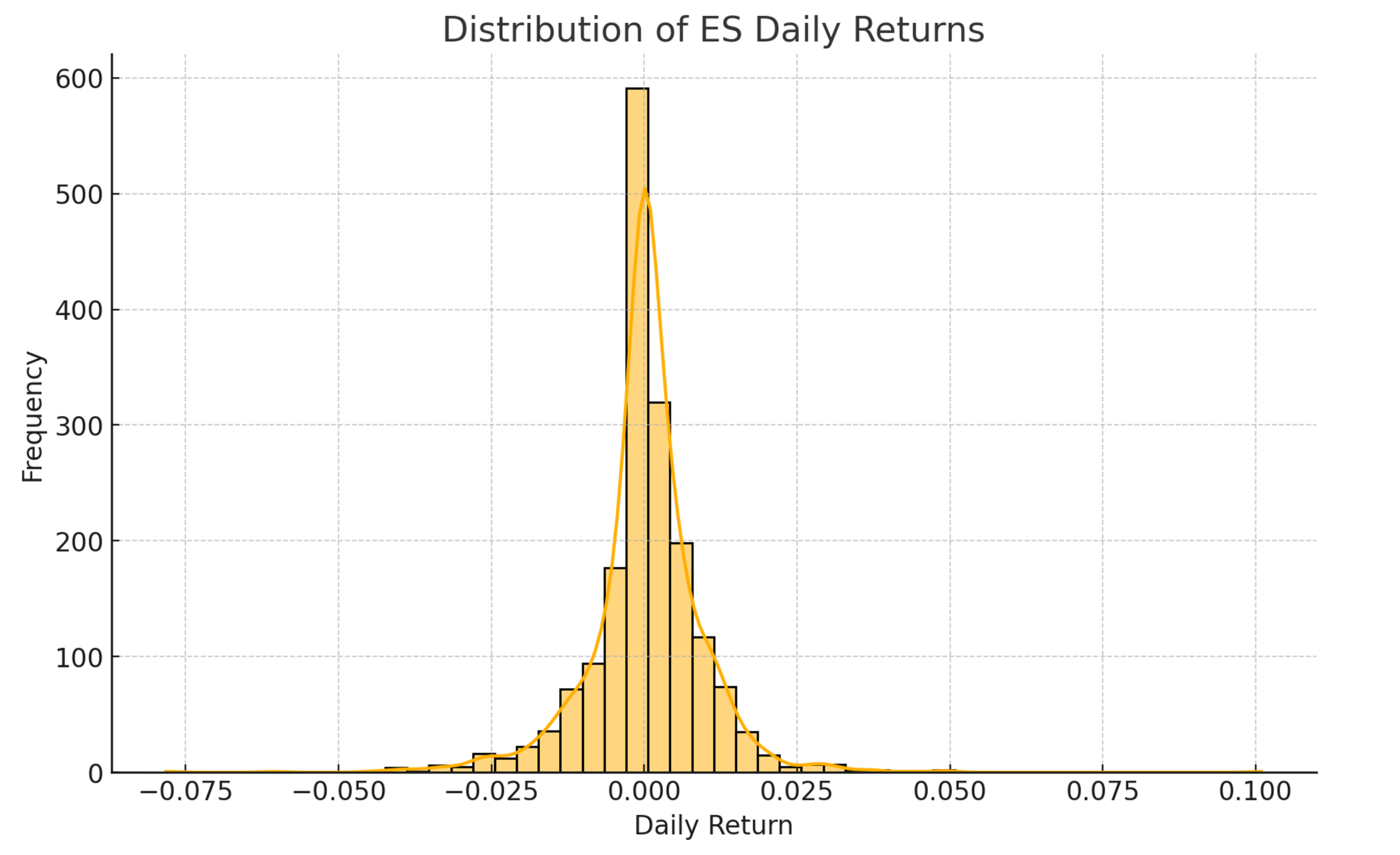
We perform EDA to gain insights into the ES price movements and trading volumes over time. This includes visualizing the closing prices, trading volumes, moving averages, and daily returns.

In [3]:
```python
# Convert Period index back to datetime for plotting
es_daily.index = es_daily.index.to_timestamp()

# Plotting ES Close Price over time
plt.figure(figsize=(14, 7))
plt.plot(es_daily['Close'], label='Close Price')
plt.title('ES Close Price Over Time')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.legend()
plt.show()

# Plotting ES Trading Volume over time
plt.figure(figsize=(14, 7))
plt.plot(es_daily['Volume'], label='Volume', color='orange')
plt.title('ES Trading Volume Over Time')
plt.xlabel('Date')
plt.ylabel('Volume')
plt.legend()
plt.show()

# Moving averages
es_daily['20_Day_MA'] = es_daily['Close'].rolling(window=20).mean()
es_daily['50_Day_MA'] = es_daily['Close'].rolling(window=50).mean()

plt.figure(figsize=(14, 7))
plt.plot(es_daily['Close'], label='Close Price')
plt.plot(es_daily['20_Day_MA'], label='20-Day Moving Average', linestyle='--')
plt.plot(es_daily['50_Day_MA'], label='50-Day Moving Average', linestyle='--')
plt.title('ES Close Price with Moving Averages')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

# Volatility Analysis: Daily Returns
plt.figure(figsize=(10, 6))
sns.histplot(es_daily['Daily Return'].dropna(), bins=50, kde=True)
plt.title('Distribution of ES Daily Returns')
plt.xlabel('Daily Return')
plt.ylabel('Frequency')
plt.show()
```





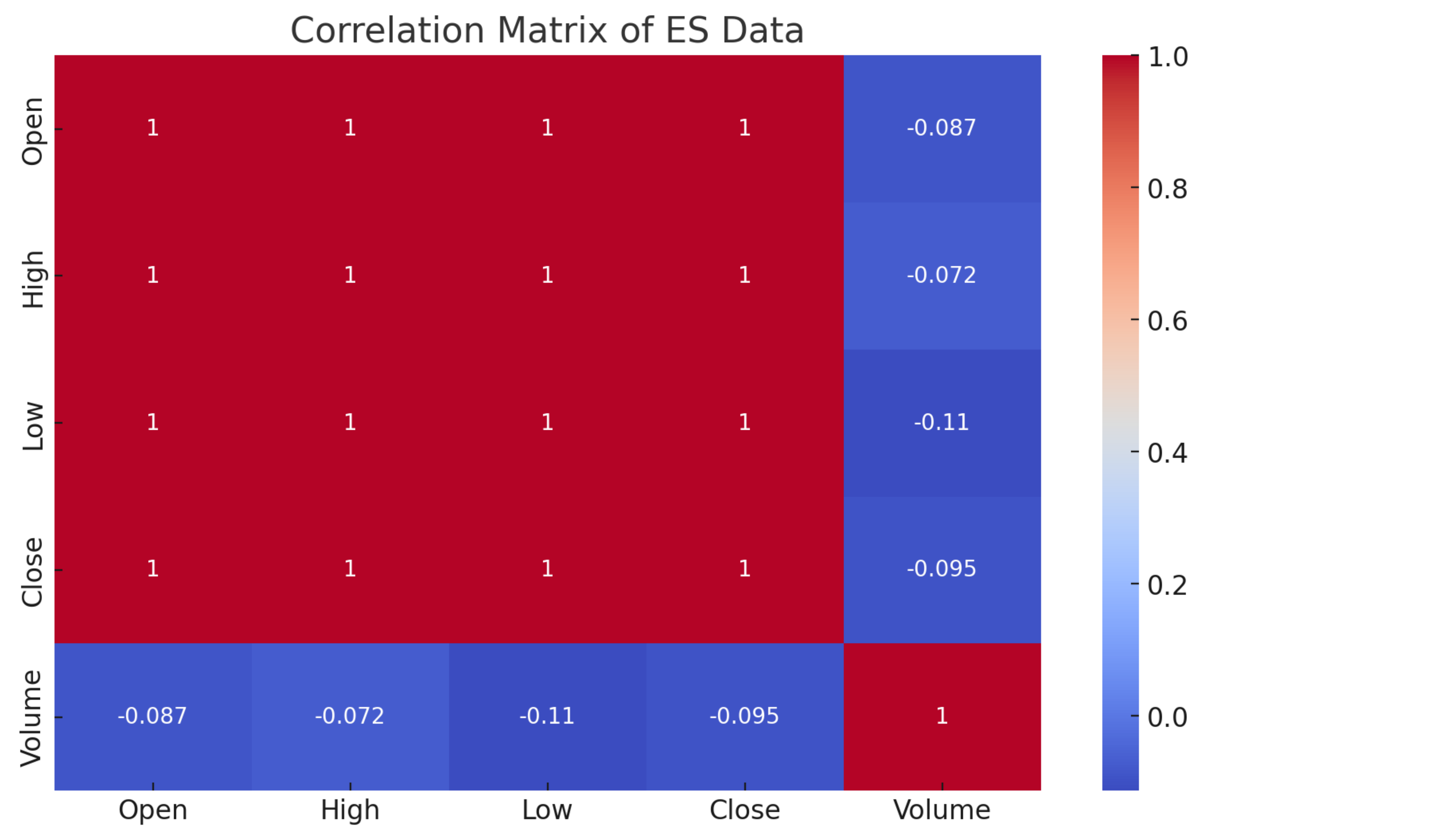




Statistical Analysis

We perform statistical analysis to understand the relationships between different features in the ES dataset. This includes correlation analysis and analyzing the distribution of daily returns to identify major market movements.

In [4]:
```python
# Correlation analysis
correlation_matrix = es_daily[['Open', 'High', 'Low', 'Close', 'Volume']].corr()
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of ES Data')
plt.show()

# Analyzing major market movements
significant_changes = es_daily[es_daily['Daily Return'].abs() > 0.02]
significant_changes[['Close', 'Daily Return']]
```



Out[4]:

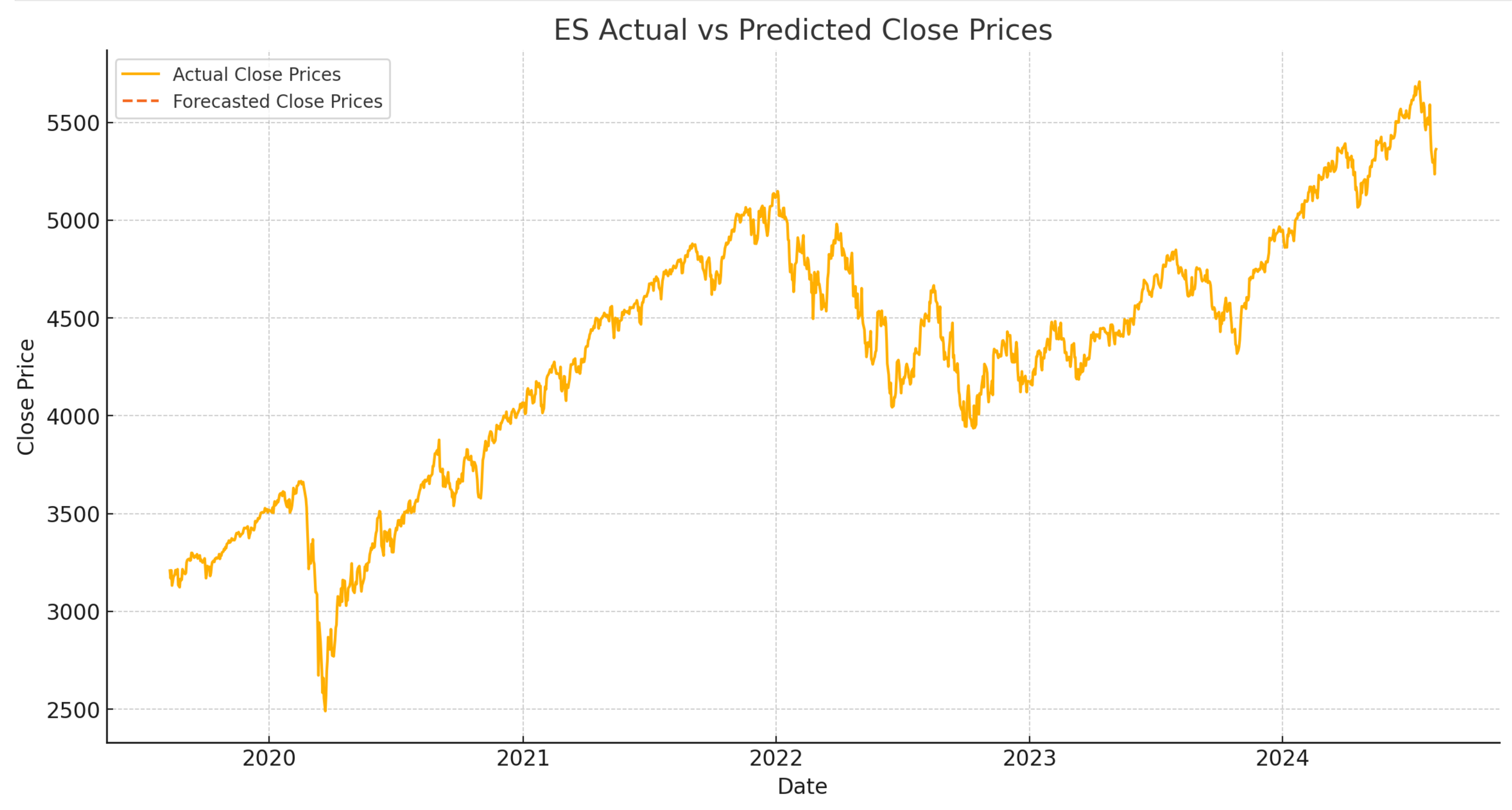
Time	Close	Daily Return
2019-08-14	3132.00	-0.024147
2019-08-23	3134.25	-0.025192
2020-02-25	3429.00	-0.028818
2020-02-26	3340.25	-0.025882
2020-02-27	3216.75	-0.036973
...	...	...
2022-12-15	4232.75	-0.024824
2023-03-09	4196.25	-0.023844
2023-11-14	4702.50	0.020231
2024-08-01	5440.75	-0.026874
2024-08-08	5354.25	0.022682

90 rows × 2 columns

Predictive Modeling

In this section, we apply predictive models to forecast future ES prices based on historical data. We use ARIMA (AutoRegressive Integrated Moving Average) as a simple time series model to predict future closing prices.

In [5]:
```python
# Prepare data for ARIMA model
es_close = es_daily['Close'].dropna()

# Fit ARIMA model
model = ARIMA(es_close, order=(5, 1, 0))
arima_result = model.fit()

# Predict future values
es_daily['Forecast'] = arima_result.predict(start=len(es_close), end=len(es_close) + 30, dynamic=True)

# Plot actual vs predicted
plt.figure(figsize=(14, 7))
plt.plot(es_close, label='Actual Close Prices')
plt.plot(es_daily['Forecast'], label='Forecasted Close Prices', linestyle='--')
plt.title('ES Actual vs Predicted Close Prices')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.legend()
plt.show()
```



Conclusion

Insights and Observations:

- The ES index shows significant periods of volatility, indicated by daily returns and moving averages.
- Correlation analysis highlights strong relationships between open, high, low, and close prices.
- The ARIMA model provides a basic forecast, suggesting potential trends in future prices. However, more complex models may improve accuracy.

Final Thoughts:

Continuous monitoring and modeling adjustments are crucial for capturing the market's behavior accurately. Including external factors such as macroeconomic indicators and sentiment analysis could further enhance forecasting models.