In [ ]:
```
pip install pandas numpy matplotlib seaborn plotly statsmodels sktime pmdari
```

```
Requirement already satisfied: pandas in /Library/Frameworks/Python.framewor
k/Versions/3.12/lib/python3.12/site-packages (2.2.2)
Requirement already satisfied: numpy in /Library/Frameworks/Python.framewor
k/Versions/3.12/lib/python3.12/site-packages (1.26.4)
Requirement already satisfied: matplotlib in /Library/Frameworks/Python.fram
ework/Versions/3.12/lib/python3.12/site-packages (3.9.0)
Requirement already satisfied: seaborn in /Library/Frameworks/Python.framewo
rk/Versions/3.12/lib/python3.12/site-packages (0.13.2)
Requirement already satisfied: plotly in /Library/Frameworks/Python.framewor
k/Versions/3.12/lib/python3.12/site-packages (5.24.1)
Requirement already satisfied: statsmodels in /Library/Frameworks/Python.fra
mework/Versions/3.12/lib/python3.12/site-packages (0.14.4)
Requirement already satisfied: sktime in /Library/Frameworks/Python.framewor
k/Versions/3.12/lib/python3.12/site-packages (0.34.0)
Requirement already satisfied: pmdarima in /Library/Frameworks/Python.framew
ork/Versions/3.12/lib/python3.12/site-packages (2.0.4)
Requirement already satisfied: tensorflow in /Library/Frameworks/Python.fram
ework/Versions/3.12/lib/python3.12/site-packages (2.16.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /Users/Jestin/Libra
ry/Python/3.12/lib/python/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /Library/Frameworks/Python.fr
amework/Versions/3.12/lib/python3.12/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /Library/Frameworks/Python.
framework/Versions/3.12/lib/python3.12/site-packages (from pandas) (2024.1)
Requirement already satisfied: contourpy>=1.0.1 in /Library/Frameworks/Pytho
n.framework/Versions/3.12/lib/python3.12/site-packages (from matplotlib) (1.
2.1)
Requirement already satisfied: cycler>=0.10 in /Library/Frameworks/Python.fr
amework/Versions/3.12/lib/python3.12/site-packages (from matplotlib) (0.12.
1)
Requirement already satisfied: fonttools>=4.22.0 in /Library/Frameworks/Pyth
on.framework/Versions/3.12/lib/python3.12/site-packages (from matplotlib)
(4.52.4)
Requirement already satisfied: kiwisolver>=1.3.1 in /Library/Frameworks/Pyth
on.framework/Versions/3.12/lib/python3.12/site-packages (from matplotlib)
(1.4.5)
Requirement already satisfied: packaging>=20.0 in /Users/Jestin/Library/Pyth
on/3.12/lib/python/site-packages (from matplotlib) (24.0)
Requirement already satisfied: pillow>=8 in /Library/Frameworks/Python.frame
work/Versions/3.12/lib/python3.12/site-packages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /Library/Frameworks/Pytho
n.framework/Versions/3.12/lib/python3.12/site-packages (from matplotlib) (3.
1.2)
Requirement already satisfied: tenacity>=6.2.0 in /Library/Frameworks/Pytho
n.framework/Versions/3.12/lib/python3.12/site-packages (from plotly) (9.0.0)
Requirement already satisfied: scipy!=1.9.2,>=1.8 in /Library/Frameworks/Pyt
hon.framework/Versions/3.12/lib/python3.12/site-packages (from statsmodels)
(1.13.1)
Requirement already satisfied: patsy>=0.5.6 in /Library/Frameworks/Python.fr
amework/Versions/3.12/lib/python3.12/site-packages (from statsmodels) (0.5.
6)
Requirement already satisfied: joblib<1.5,>=1.2.0 in /Library/Frameworks/Pyt
hon.framework/Versions/3.12/lib/python3.12/site-packages (from sktime) (1.4.
2)
Requirement already satisfied: scikit-base<0.12.0,>=0.6.1 in /Library/Framew
orks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from sktim
```

```
e) (0.11.0)
Requirement already satisfied: scikit-learn<1.6.0,>=0.24 in /Library/Framewo
rks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from sktim
e) (1.5.2)
Requirement already satisfied: Cython!=0.29.18,!=0.29.31,>=0.29 in /Library/
Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from
pmdarima) (3.0.11)
Requirement already satisfied: urllib3 in /Library/Frameworks/Python.framewo
rk/Versions/3.12/lib/python3.12/site-packages (from pmdarima) (2.2.1)
Requirement already satisfied: setuptools!=50.0.0,>=38.6.0 in /Library/Frame
works/Python.framework/Versions/3.12/lib/python3.12/site-packages (from pmda
rima) (70.0.0)
Requirement already satisfied: absl-py>=1.0.0 in /Library/Frameworks/Python.
framework/Versions/3.12/lib/python3.12/site-packages (from tensorflow) (2.1.
0)
Requirement already satisfied: astunparse>=1.6.0 in /Library/Frameworks/Pyth
on.framework/Versions/3.12/lib/python3.12/site-packages (from tensorflow)
(1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in /Library/Frameworks/P
ython.framework/Versions/3.12/lib/python3.12/site-packages (from tensorflow)
(24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /Libra
ry/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (f
rom tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in /Library/Frameworks/Py
thon.framework/Versions/3.12/lib/python3.12/site-packages (from tensorflow)
(0.2.0)
Requirement already satisfied: h5py>=3.10.0 in /Library/Frameworks/Python.fr
amework/Versions/3.12/lib/python3.12/site-packages (from tensorflow) (3.12.
1)
Requirement already satisfied: libclang>=13.0.0 in /Library/Frameworks/Pytho
n.framework/Versions/3.12/lib/python3.12/site-packages (from tensorflow) (1
8.1.1)
Requirement already satisfied: ml-dtypes~=0.3.1 in /Library/Frameworks/Pytho
n.framework/Versions/3.12/lib/python3.12/site-packages (from tensorflow) (0.
3.2)
Requirement already satisfied: opt-einsum>=2.3.2 in /Library/Frameworks/Pyth
on.framework/Versions/3.12/lib/python3.12/site-packages (from tensorflow)
(3.4.0)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!
=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in /Library/Frameworks/Python.framework/
Versions/3.12/lib/python3.12/site-packages (from tensorflow) (4.25.5)
Requirement already satisfied: requests<3,>=2.21.0 in /Library/Frameworks/Py
thon.framework/Versions/3.12/lib/python3.12/site-packages (from tensorflow)
(2.32.2)
Requirement already satisfied: six>=1.12.0 in /Users/Jestin/Library/Python/
3.12/lib/python/site-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in /Library/Frameworks/Pytho
n.framework/Versions/3.12/lib/python3.12/site-packages (from tensorflow) (2.
5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /Library/Framewor
ks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from tensorf
low) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in /Library/Frameworks/Python.f
ramework/Versions/3.12/lib/python3.12/site-packages (from tensorflow) (1.16.
0)
```

```
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from tensorflow) (1.67.1)
Requirement already satisfied: tensorboard<2.17,>=2.16 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from tensorflow) (2.16.2)
Requirement already satisfied: keras>=3.0.0 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from tensorflow) (3.6.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from astunparse>=1.6.0->tensorflow) (0.45.0)
Requirement already satisfied: rich in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from keras>=3.0.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from keras>=3.0.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from keras>=3.0.0->tensorflow) (0.13.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from requests<3,>=2.21.0->tensorflow) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from requests<3,>=2.21.0->tensorflow) (3.7)
Requirement already satisfied: certifi>=2017.4.17 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from requests<3,>=2.21.0->tensorflow) (2024.2.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from scikit-learn<1.6.0,>=0.24->sktime) (3.5.0)
Requirement already satisfied: markdown>=2.6.8 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from tensorboard<2.17,>=2.16->tensorflow) (3.7)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from tensorboard<2.17,>=2.16->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from tensorboard<2.17,>=2.16->tensorflow) (3.0.6)
Requirement already satisfied: MarkupSafe>=2.1.1 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from werkzeug>=1.0.1->tensorboard<2.17,>=2.16->tensorflow) (3.0.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from rich->keras>=3.0.0->tensorflow) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /Users/Jestin/Library/Python/3.12/lib/python/site-packages (from rich->keras>=3.0.0->tensorflow) (2.18.0)
Requirement already satisfied: mdurl~=0.1 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from markdown-it-py>=2.2.0->rich->keras>=3.0.0->tensorflow) (0.1.2)

[notice] A new release of pip is available: 24.0 -> 24.3.1
```

```
[notice] To update, run: pip3 install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
```

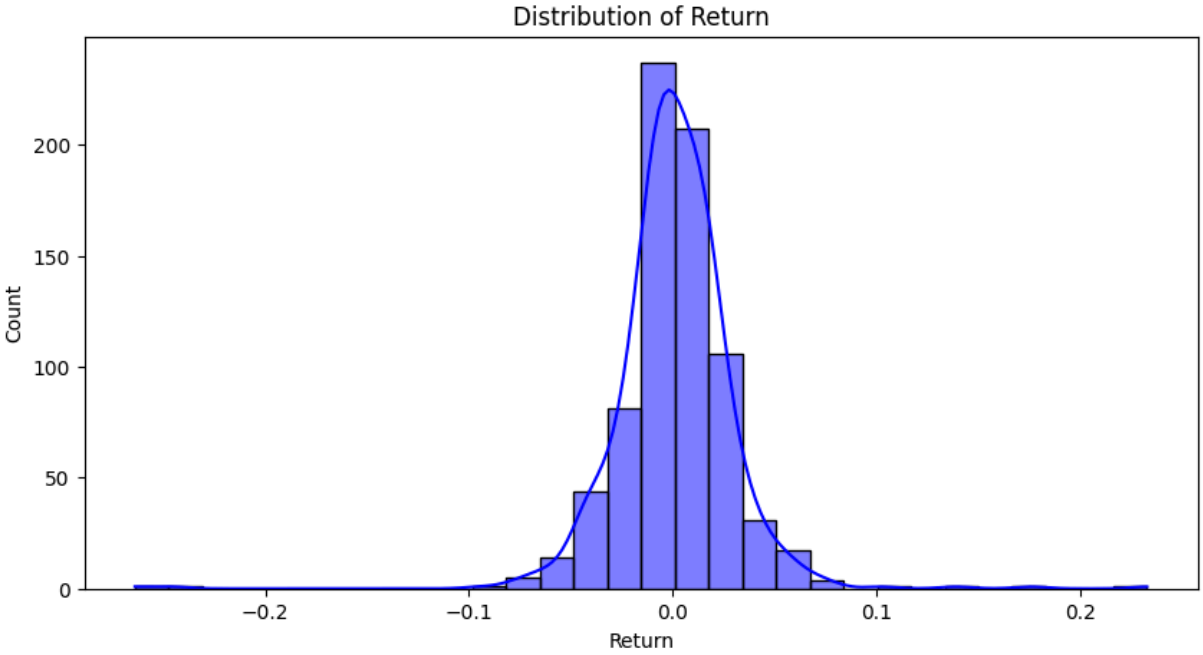
In [ ]:

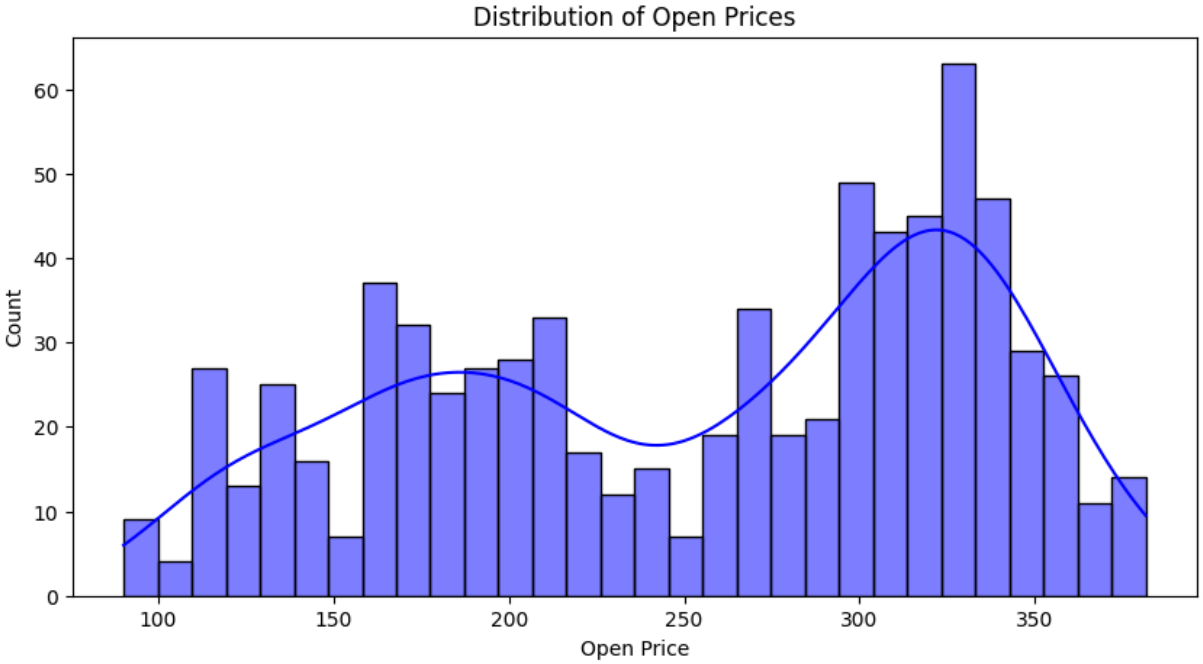
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import statsmodels.api as sm
from sktime.forecasting.model_selection import temporal_train_test_split
from sktime.forecasting.arima import AutoARIMA
from pmdarima import auto_arima
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_scor
from statsmodels.tsa.arima.model import ARIMA

# Load the data
data = pd.read_excel('/Users/Jestin/Desktop/Finance/Quant Finance/Meta/Meta

# Convert 'Date' column to datetime format
data['Date'] = pd.to_datetime(data['Date'])

# Basic Summary
print("Data Summary:")
print(data.info())
print("\nData Description:")
print(data.describe())

# Remove extra space from columns
data.columns = data.columns.str.strip()

# Distribution of Adjusted Close Prices
plt.figure(figsize=(10, 5))
sns.histplot(data['Return'], bins=30, kde=True, color='blue')
plt.title('Distribution of Return')
plt.xlabel('Return')
plt.show()

# Distribution of Open Prices
plt.figure(figsize=(10, 5))
sns.histplot(data['Open'], bins=30, kde=True, color='blue')
plt.title('Distribution of Open Prices')  # Corrected title
plt.xlabel('Open Price')
plt.show()

# Volume vs Adjusted Close Price
plt.figure(figsize=(12, 6))
sns.scatterplot(x=data['Volume'], y=data['Return'])
plt.title('Volume vs Adjusted Close Price')
plt.xlabel('Volume')
plt.ylabel('Return')
plt.show()

# Visualize stock price over time
```

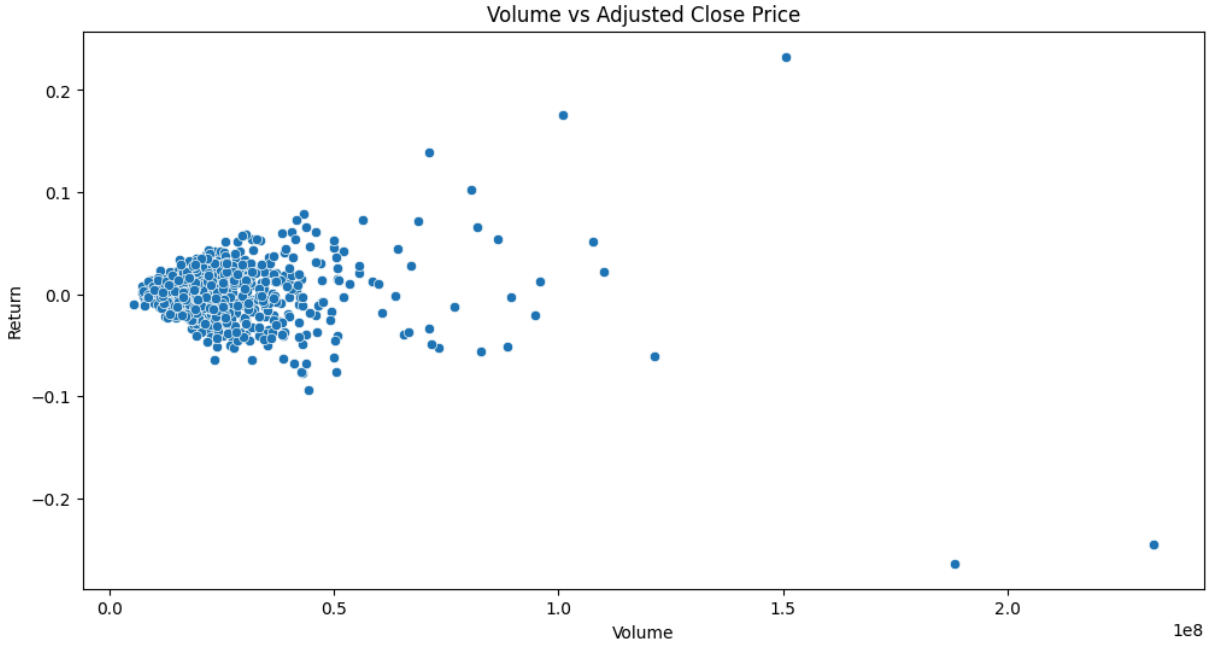
```
plt.figure(figsize=(14, 7))
plt.plot(data['Date'], data['Return'], label='Return', color='blue')
plt.title('Meta Returns Over Time')
plt.xlabel('Date')
plt.ylabel('Returns')
plt.legend()
plt.show()

# Checking for Seasonality and Trends
from statsmodels.tsa.seasonal import seasonal_decompose

# Decompose Return column (additive model)
decomposition_adj_close = seasonal_decompose(data['Return'], model='additive
decomposition_adj_close.plot()
plt.show()

# Decompose Open column (additive model)
decomposition_open = seasonal_decompose(data['Open'], model='additive', peri
decomposition_open.plot()
plt.show()

# Earnings dates (ensure they fall within the data range)
earnings_dates = [
    '2023-12-31', '2023-09-30', '2023-06-30', '2023-03-31',
    '2022-12-31', '2022-09-30', '2022-06-30', '2022-03-31',
    '2021-12-31', '2021-09-30', '2021-06-30', '2021-03-31'
]

earnings_dates = pd.to_datetime(earnings_dates)

# Insert a number for 'window_before_after'

# Adj Close Price

# Extracting the data for windows around earnings announcements
window_before_after = 10 # 10 days before and after earnings
price_windows = []

for earnings_date in earnings_dates:
    start_date = earnings_date - pd.Timedelta(days=window_before_after)
    end_date = earnings_date + pd.Timedelta(days=window_before_after)

    # Extract the stock price data for this window, using 'Date' as the colu
    window_data = data[(data['Date'] >= start_date) & (data['Date'] <= end_d
    price_windows.append(window_data)

# Combine all the windows into one DataFrame
price_windows_df = pd.concat(price_windows)

# Optionally, reset index for cleaner DataFrame
price_windows_df = price_windows_df.reset_index(drop=True)

# Display the combined price window DataFrame
print(price_windows_df)

# Plot stock price before and after earnings
```

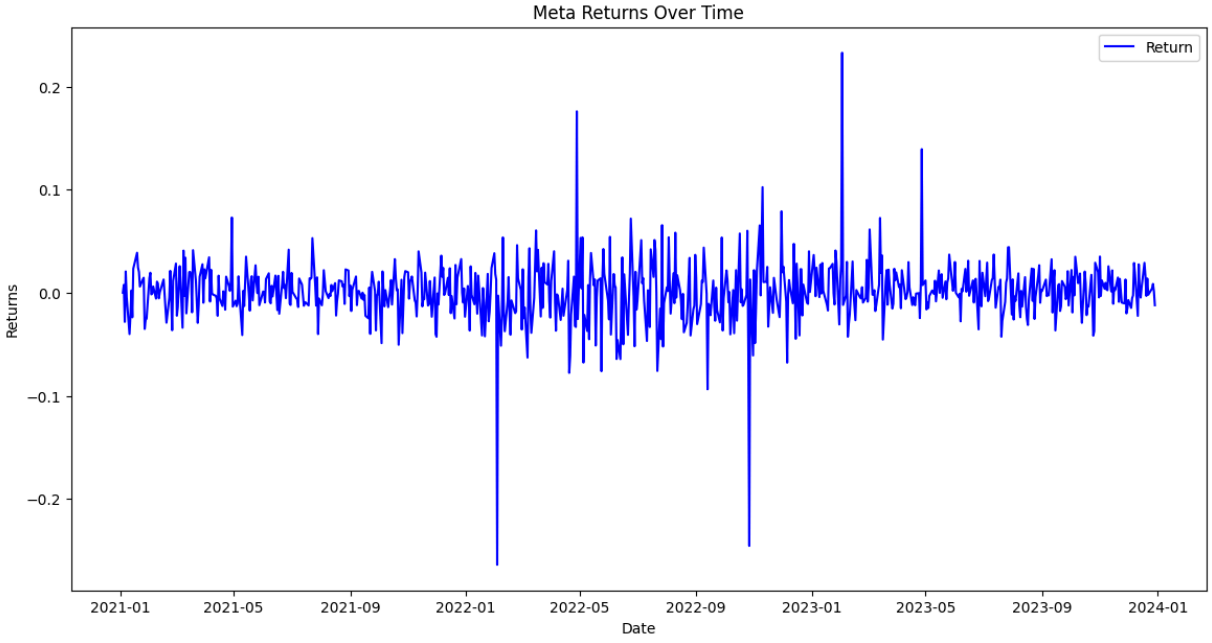
```
plt.figure(figsize=(10, 6))

for earnings_date in earnings_dates:
    window_data = price_windows_df[price_windows_df['Date'].between(earnings
                                                                    earning
    plt.plot(window_data['Date'], window_data['Return'], label=f'Earnings Da

plt.title('Returns Before and After Earnings for Meta')
plt.xlabel('Date')
plt.ylabel('Stock Price (Return)')
plt.legend()
plt.show()
```

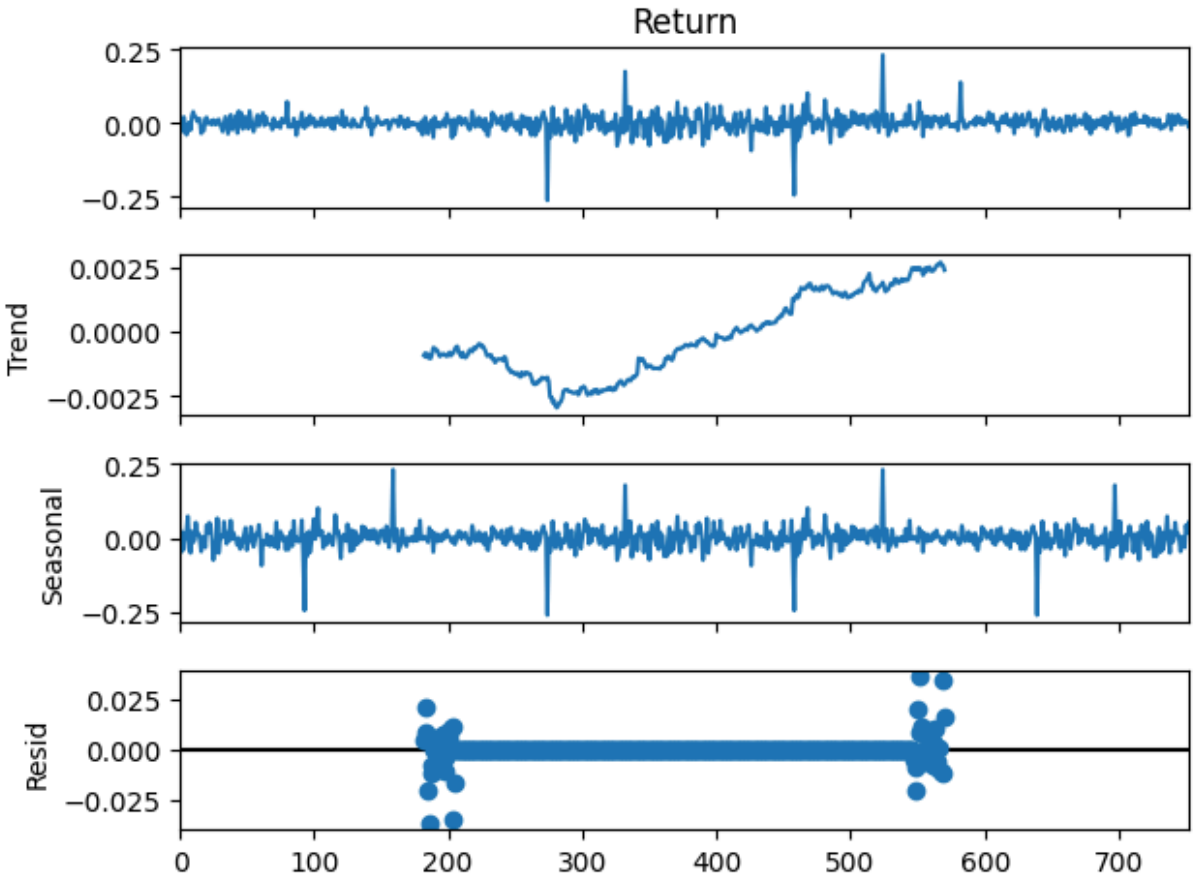
```
Data Summary:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 753 entries, 0 to 752
Data columns (total 8 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       753 non-null    datetime64[ns]
 1   Open       753 non-null    float64
 2   High       753 non-null    float64
 3   Low        753 non-null    float64
 4   Close      753 non-null    float64
 5   Adj Close  753 non-null    float64
 6   Volume     753 non-null    int64
 7   Return     753 non-null    float64
dtypes: datetime64[ns](1), float64(6), int64(1)
memory usage: 47.2 KB
None

Data Description:
                                Date        Open        High         Low  \
count                            753  753.000000  753.000000  753.000000
mean   2022-07-01 22:30:07.171314688  254.154197  257.952244  250.760823
min              2021-01-04 00:00:00   90.080000   90.460000   88.090000
25%              2021-10-01 00:00:00  186.130000  190.360000  182.260000
50%              2022-07-01 00:00:00  270.520000  274.250000  268.120000
75%              2023-03-31 00:00:00  323.690000  328.240000  319.600000
max              2023-12-29 00:00:00  381.680000  384.330000  378.810000
std                              NaN   78.398334   78.739298   77.835216

            Close   Adj Close        Volume      Return
count  753.000000  753.000000  7.530000e+02  753.000000
mean   254.338738  253.575246  2.620642e+07    0.000815
min     88.910000   88.640000  5.467500e+06   -0.263888
25%    184.900000  184.340000  1.658080e+07   -0.012052
50%    271.320000  270.510000  2.192460e+07    0.000408
75%    324.460000  323.490000  3.000160e+07    0.014862
max    382.180000  381.030000  2.323166e+08    0.232805
std     78.361148   78.125843  1.785264e+07    0.029652
```

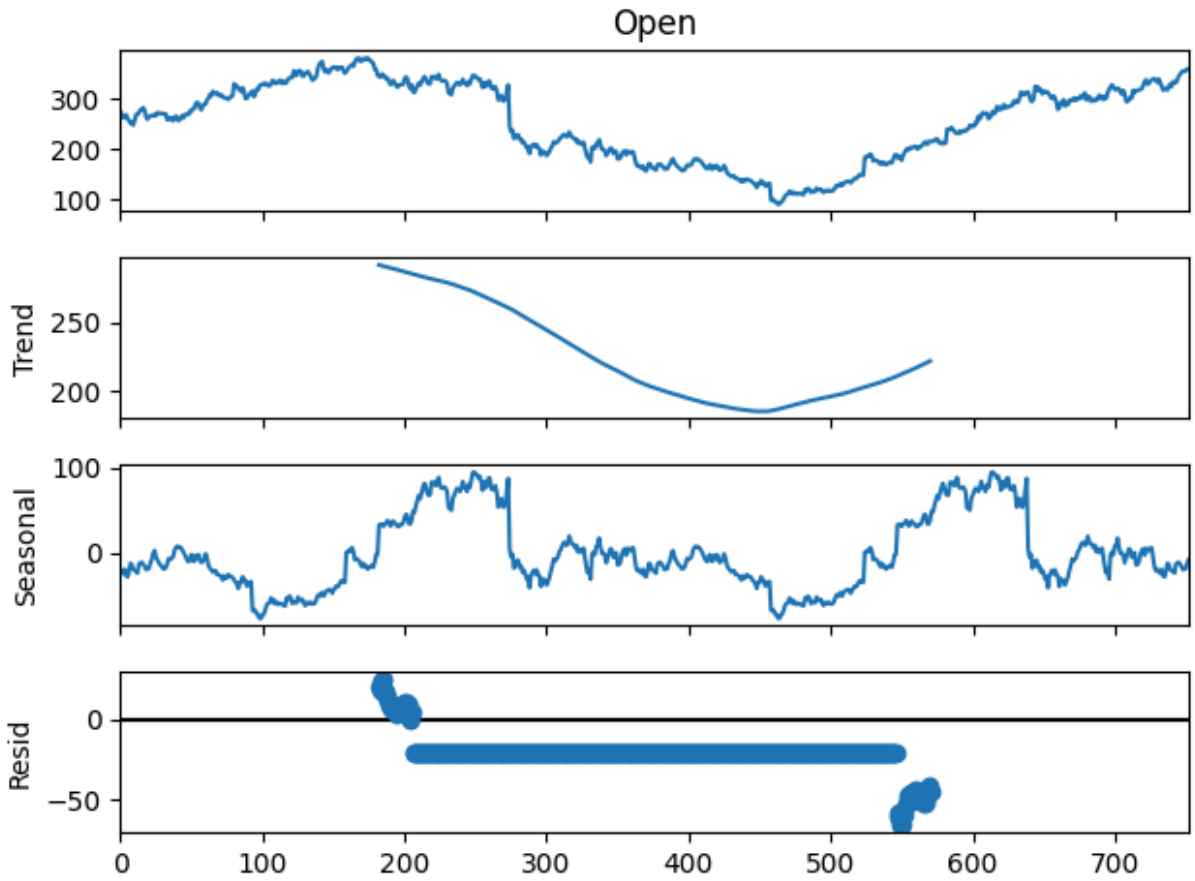

Distribution of Return
Count
Return
0
50
100
150
200
−0.2
−0.1
0.0
0.1
0.2


Distribution of Open Prices
Count
Open Price
0
10
20
30
40
50
60
100
150
200
250
300
350

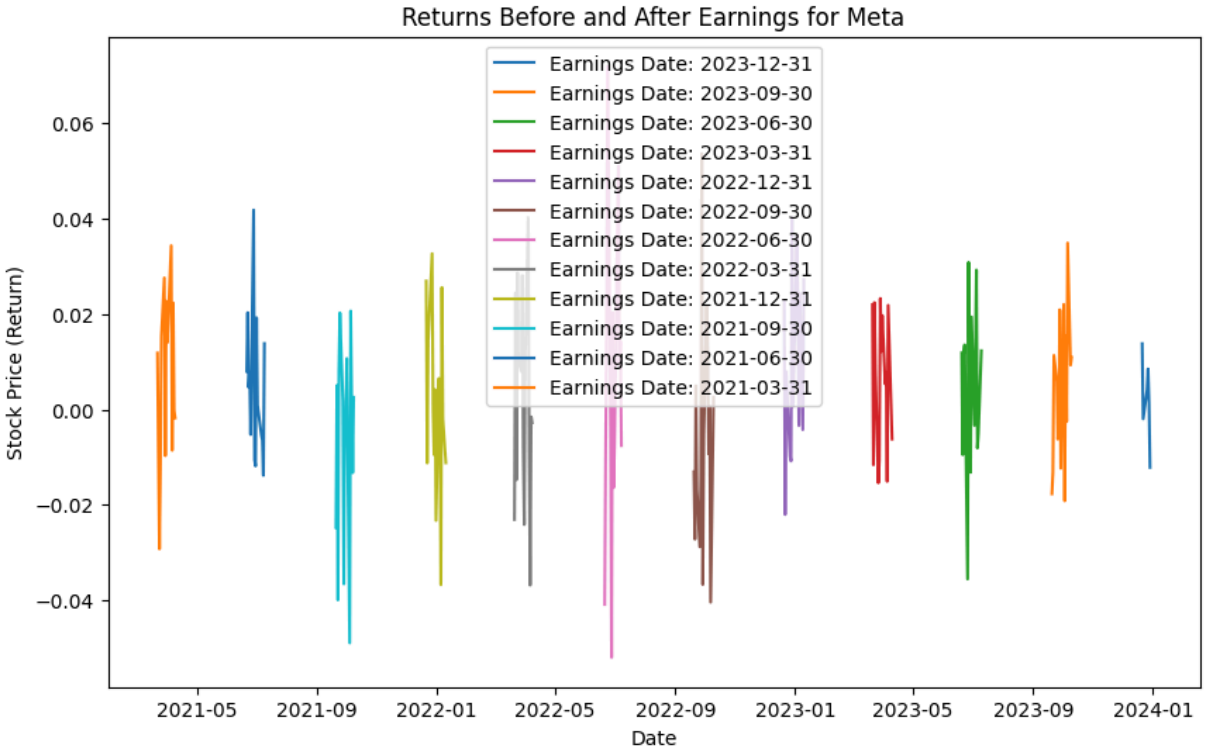

Volume vs Adjusted Close Price
Return
Volume
1e8
0.2
0.1
0.0
−0.1
−0.2
0.0
0.5
1.0
1.5
2.0


Meta Returns Over Time
Return
Returns
Date
0.2
0.1
0.0
−0.1
−0.2
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01


Return
0.25
0.00
−0.25
Trend
0.0025
0.0000
−0.0025
Seasonal
0.25
0.00
−0.25
Resid
0.025
0.000
−0.025
0
100
200
300
400
500
600
700

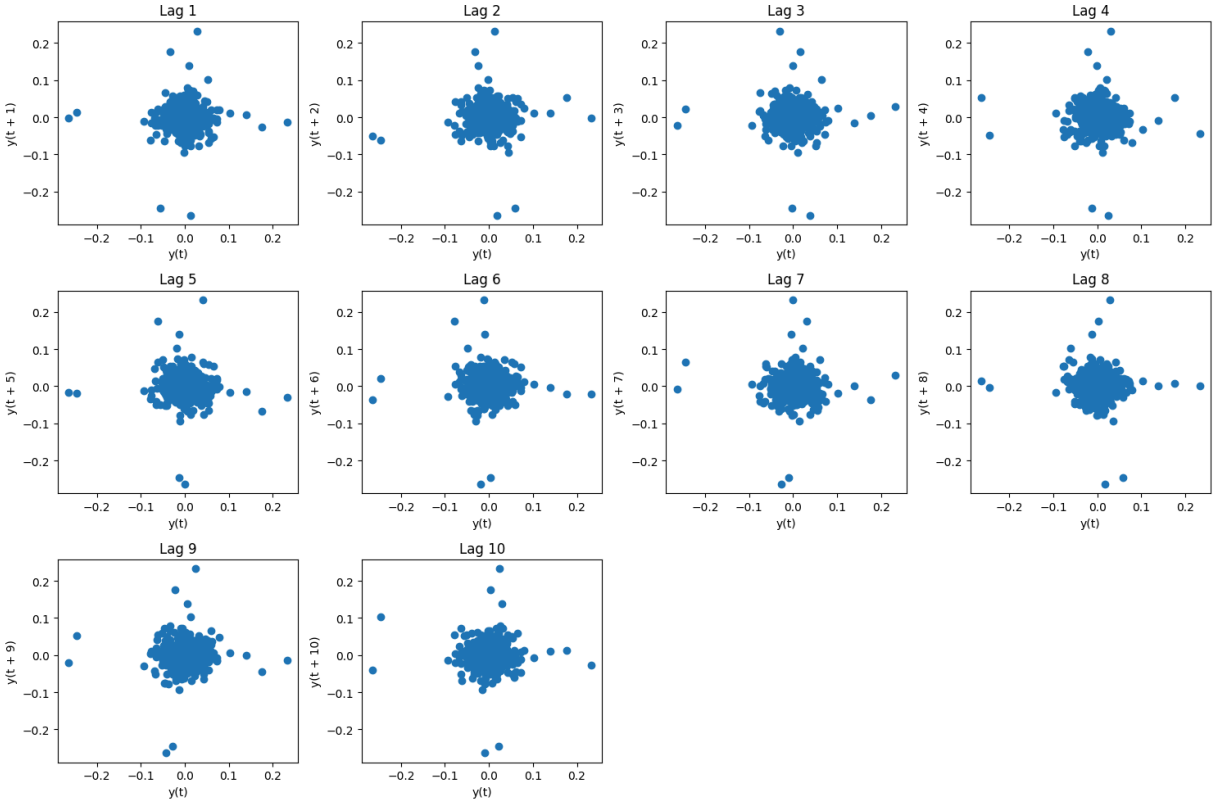

Open
300
200
100
Trend
250
200
Seasonal
100
0
Resid
0
−50
0
100
200
300
400
500
600
700

```
           Date    Open    High     Low   Close  Adj Close    Volume    Retur
n
0    2023-12-21  352.98  356.41  349.21  354.09     353.03  15289600  0.01378
4
1    2023-12-22  355.58  357.20  351.22  353.39     352.33  11764200 -0.00198
3
2    2023-12-26  354.99  356.98  353.45  354.83     353.76   9898600  0.00405
9
3    2023-12-27  356.07  359.00  355.31  357.83     356.76  13207900  0.00848
0
4    2023-12-28  359.70  361.90  357.81  358.32     357.24  11798800  0.00134
5
..          ...     ...     ...     ...     ...        ...       ...
...
157 2021-04-05  300.89  310.77  300.68  308.91     307.98  28237000  0.03432
3
158 2021-04-06  308.84  311.35  305.25  306.26     305.34  17335200 -0.00857
2
159 2021-04-07  306.34  314.25  305.50  313.09     312.15  22855200  0.02230
3
160 2021-04-08  314.85  315.88  310.05  313.02     312.08  20894100 -0.00022
4
161 2021-04-09  311.40  314.74  310.33  312.46     311.52  15988600 -0.00179
4

[162 rows x 8 columns]
```



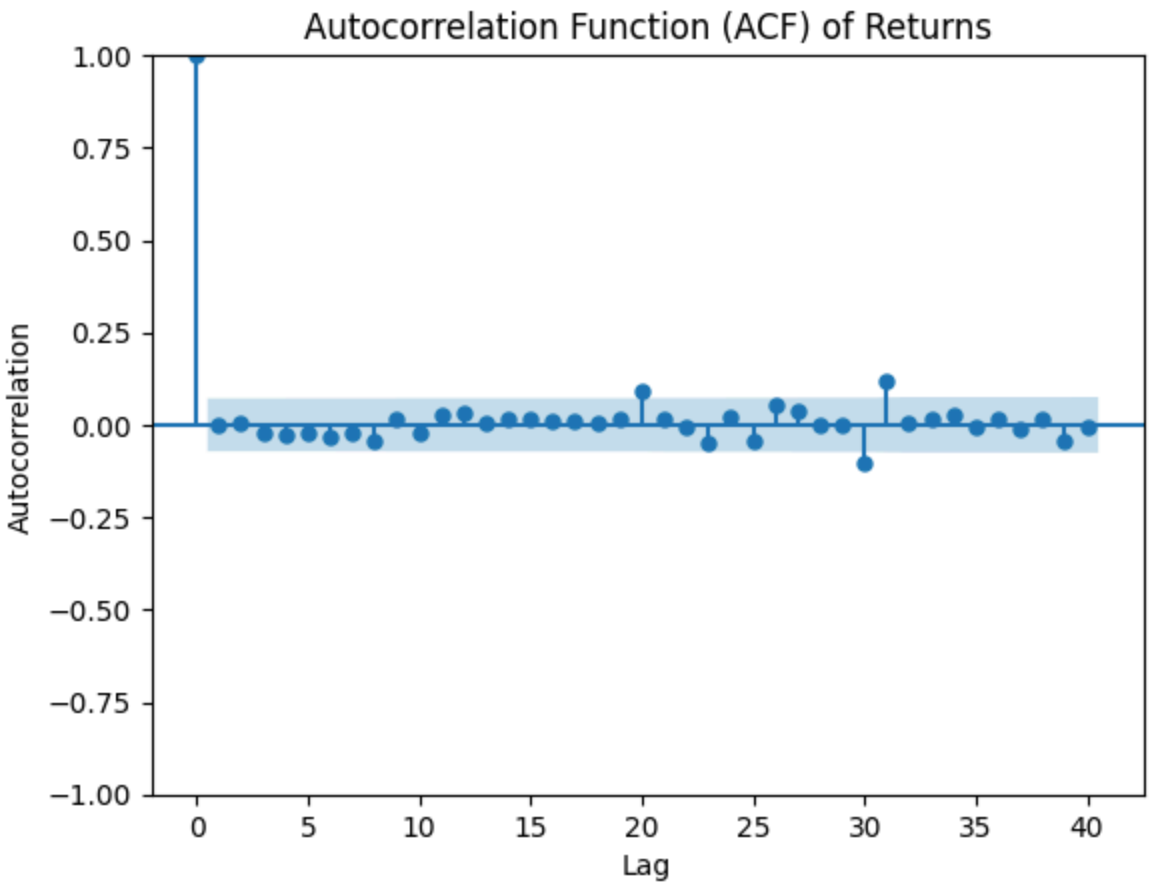
In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pandas.plotting import lag_plot

# Create lag plots
```

```
def plot_lag(data, lags=10):
    plt.figure(figsize=(15, 10))
    for i in range(1, lags + 1):
        plt.subplot(3, 4, i)  # Grid of subplots, adjust as needed
        lag_plot(data['Return'], lag=i)
        plt.title(f'Lag {i}')
    plt.tight_layout()
    plt.show()

# Plot lag plots for the Adjusted Close price column
plot_lag(data, lags=10)
```



In [ ]:
```
from statsmodels.graphics.tsaplots import plot_acf

# Plot Autocorrelation
def plot_autocorrelation(data, lags=40):
    plt.figure(figsize=(10, 6))
    plot_acf(data['Return'], lags=lags)
    plt.title(f'Autocorrelation Function (ACF) of Returns')
    plt.xlabel('Lag')
    plt.ylabel('Autocorrelation')
    plt.show()

# Plot the autocorrelation for the Adjusted Close column with 40 lags
plot_autocorrelation(data, lags=40)

from statsmodels.tsa.stattools import adfuller

result = adfuller(data['Return'])
print(f'ADF Statistic: {result[0]}')
print(f'p-value: {result[1]}')
```

```
<Figure size 1000x600 with 0 Axes>
```




```
ADF Statistic: -27.378764345007337
p-value: 0.0
```

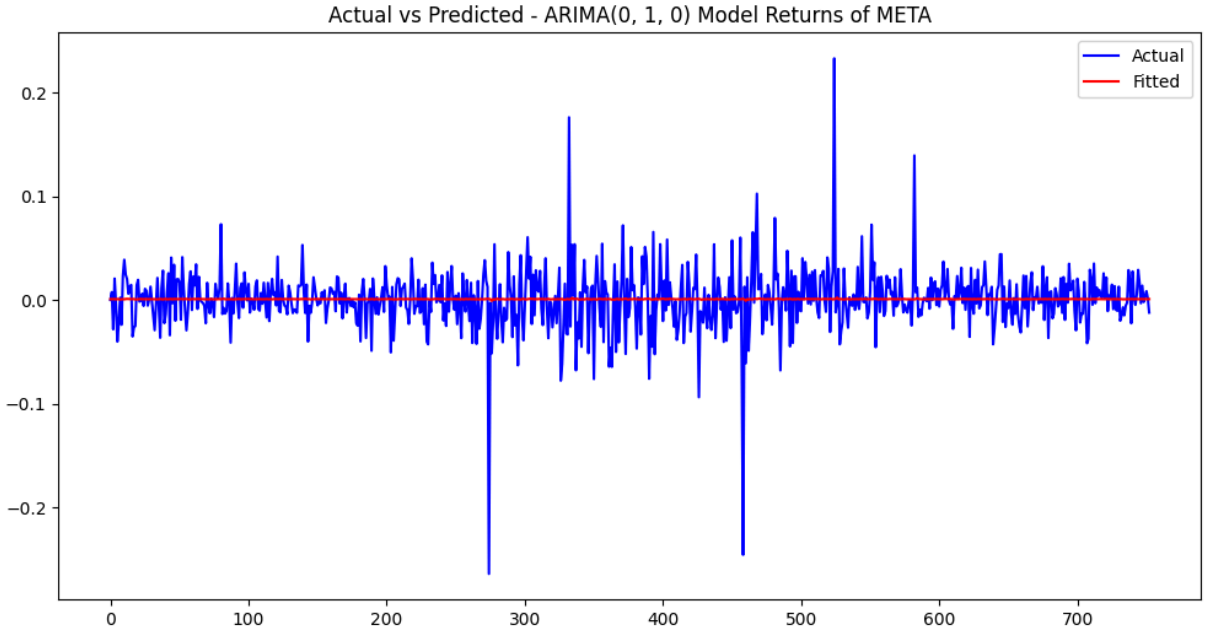
In [ ]:
```python
from statsmodels.tsa.stattools import adfuller

adf_test = adfuller(data['Return'].dropna())
print(f"ADF Statistic: {adf_test[0]}")
print(f"p-value: {adf_test[1]}")

if adf_test[1] < 0.05:
    print("Data is stationary.")
else:
    print("Data is not stationary. Differencing may be required.")
```

```
ADF Statistic: -27.378764345007337
p-value: 0.0
Data is stationary.
```

In [ ]:
```python
# Fit ARIMA model on the differenced data
from statsmodels.tsa.arima.model import ARIMA
import matplotlib.pyplot as plt

# Assuming 'data' is your dataframe and 'Adj Close Diff' is the differenced
arima_model = ARIMA(data['Return'].dropna(), order=(2, 0, 2))  # ARIMA(0, 1,
arima_result = arima_model.fit()

# Get the predicted values from the ARIMA model
predicted_values = arima_result.fittedvalues
```

```
# Plotting the actual vs predicted values
plt.figure(figsize=(12, 6))
plt.plot(data['Return'], label='Actual', color='blue')
plt.plot(predicted_values, label='Fitted', color='red')
plt.title('Actual vs Predicted - ARIMA(0, 1, 0) Model Returns of META')
plt.legend()
plt.show()

# Print model summary
print(arima_result.summary())
```

```
/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packa
ges/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary s
tarting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packa
ges/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible s
tarting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
```

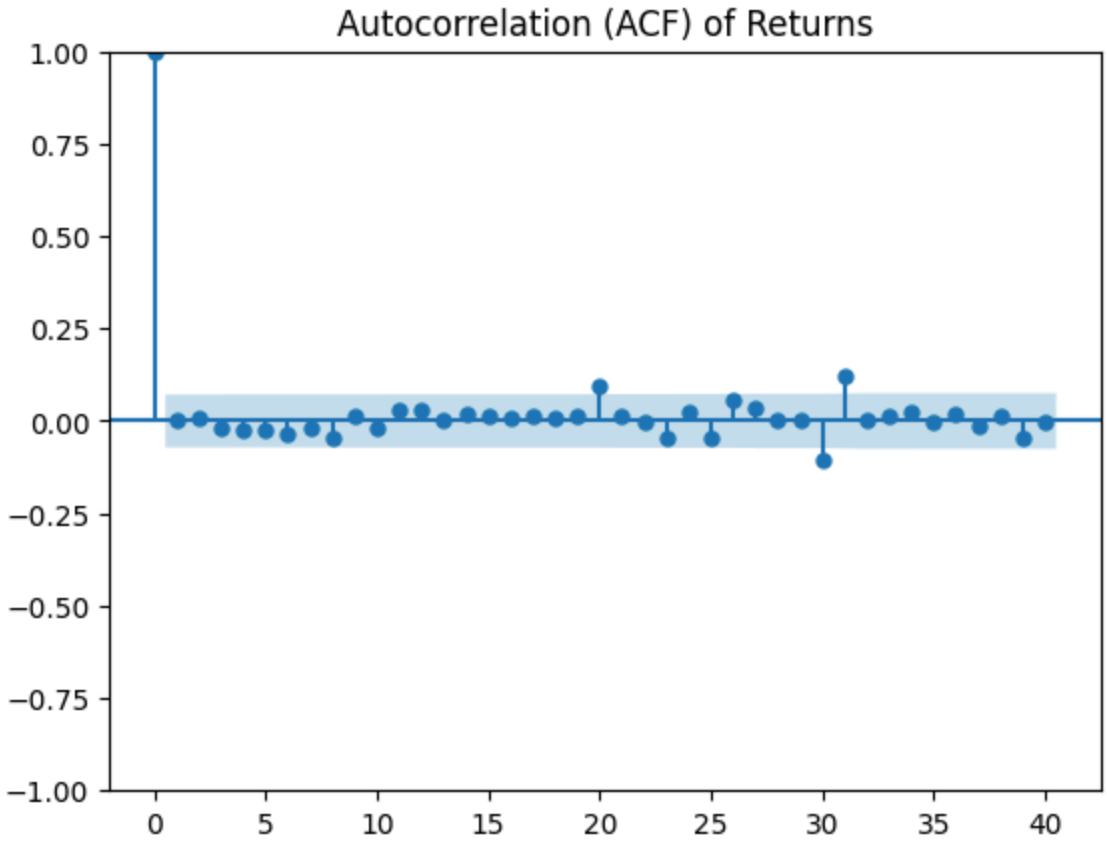


```
                               SARIMAX Results
==========================================================================
==
Dep. Variable:                  Return   No. Observations:                  7
53
Model:                 ARIMA(2, 0, 2)   Log Likelihood                1581.2
78
Date:                Sat, 16 Nov 2024   AIC                          -3150.5
57
Time:                        16:01:02   BIC                          -3122.8
12
Sample:                             0   HQIC                         -3139.8
68
                                - 753
Covariance Type:                  opg
==========================================================================
==
                 coef    std err          z      P>|z|      [0.025      0.97
5]
--------------------------------------------------------------------------
--
const          0.0008      0.001      0.736      0.462      -0.001       0.0
03
ar.L1       7.644e-05      6.055   1.26e-05      1.000     -11.867      11.8
67
ar.L2          0.0032      4.496      0.001      0.999      -8.809       8.8
15
ma.L1       7.386e-05      6.059   1.22e-05      1.000     -11.876      11.8
76
ma.L2          0.0032      4.498      0.001      0.999      -8.813       8.8
19
sigma2         0.0009   1.58e-05     55.454      0.000       0.001       0.0
01
==========================================================================
=======
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):             1
3749.78
Prob(Q):                              1.00   Prob(JB):
0.00
Heteroskedasticity (H):               1.86   Skew:
-0.70
Prob(H) (two-sided):                  0.00   Kurtosis:
23.89
==========================================================================
=======

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (compl
ex-step).
```

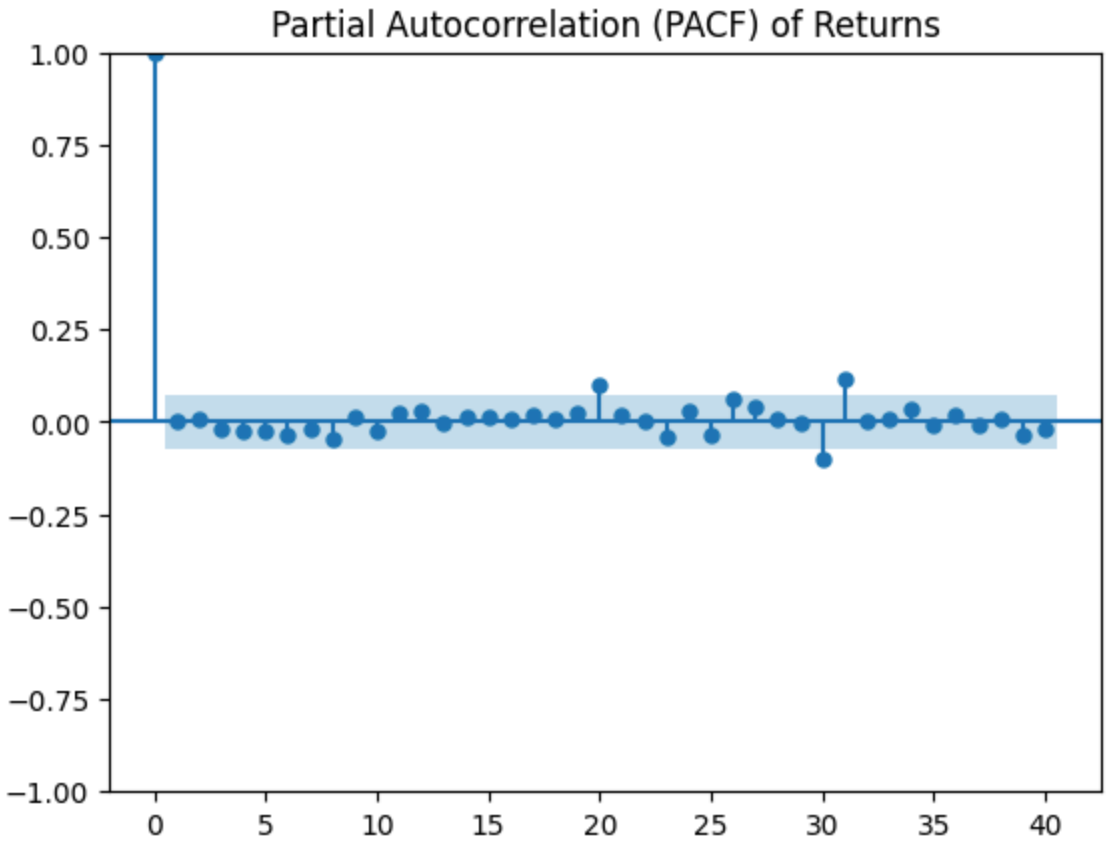
In [ ]:
```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import matplotlib.pyplot as plt

# Assuming 'data' is your dataframe and 'Return' is the column with returns
# Ensure that the data is differenced if needed and drop NaN values
returns = data['Return'].dropna()
```

```
# Plot the ACF (Autocorrelation Function)
plt.figure(figsize=(12, 6))
plot_acf(returns, lags=40, title='Autocorrelation (ACF) of Returns')
plt.show()

# Plot the PACF (Partial Autocorrelation Function)
plt.figure(figsize=(12, 6))
plot_pacf(returns, lags=40, title='Partial Autocorrelation (PACF) of Returns
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```




```
<Figure size 1200x600 with 0 Axes>
```


Partial Autocorrelation (PACF) of Returns
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25
30
35
40