

In [3]: !pip install yfinance pandas numpy matplotlib arch statsmodels

```
Collecting yfinance
  Downloading yfinance-0.2.41-py2.py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: pandas in c:\users\vnihar\anaconda3\lib\site-packages (2.1.4)
Requirement already satisfied: numpy in c:\users\vnihar\anaconda3\lib\site-packages (1.26.4)
Requirement already satisfied: matplotlib in c:\users\vnihar\anaconda3\lib\site-packages (3.8.0)
Collecting arch
  Downloading arch-7.0.8-cp311-win_amd64.whl.metadata (13 kB)
Requirement already satisfied: statsmodels in c:\users\vnihar\anaconda3\lib\site-packages (0.14.0)
Requirement already satisfied: requests>=2.31 in c:\users\vnihar\anaconda3\lib\site-packages (from yfinance) (2.31.0)
Collecting multitasking==0.0.7 (from yfinance)
  Downloading multitasking-0.0.7-py2.py3-none-any.whl.metadata (5.5 kB)
Requirement already satisfied: pillow<=9.2.0 in c:\users\vnihar\anaconda3\lib\site-packages (from multitasking==0.0.7) (10.2.0)
Requirement already satisfied: platformdirs<=2.0.0 in c:\users\vnihar\anaconda3\lib\site-packages (from multitasking==0.0.7) (3.10.0)
Requirement already satisfied: pytz<=2022.5 in c:\users\vnihar\anaconda3\lib\site-packages (from yfinance) (2023.3.post1)
Requirement already satisfied: frozendict<=3.4 in c:\users\vnihar\anaconda3\lib\site-packages (from yfinance) (2.4.2)
Collecting peewee<=3.17.0 (from yfinance)
  Downloading peewee-3.17.0.tar.gz (3.0 MB)
----- 0.0/3.0 MB ? eta -:-:--
----- 0.2/3.0 MB 5.3 MB/s eta 0:00:01
----- 0.5/3.0 MB 5.4 MB/s eta 0:00:01
----- 0.8/3.0 MB 5.3 MB/s eta 0:00:01
----- 1.5/3.0 MB 9.5 MB/s eta 0:00:01
----- 2.2/3.0 MB 9.0 MB/s eta 0:00:01
----- 2.7/3.0 MB 10.2 MB/s eta 0:00:01
----- 3.0/3.0 MB 9.9 MB/s eta 0:00:00
Installing build dependencies: started
Installing build dependencies: finished with status 'done'
Getting requirements to build wheel: started
Getting requirements to build wheel: finished with status 'done'
Preparing metadata (pyproject.toml): started
Preparing metadata (pyproject.toml): finished with status 'done'
Requirement already satisfied: beautifulsoup4<=4.11.1 in c:\users\vnihar\anaconda3\lib\site-packages (from yfinance) (4.12.2)
Collecting html5lib<=1.1 (from yfinance)
  Downloading html5lib-1.1-py2.py3-none-any.whl.metadata (16 kB)
Requirement already satisfied: python-dateutil<=2.8.2 in c:\users\vnihar\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: tzdata<=2022.1 in c:\users\vnihar\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: contourpy<=1.0.1 in c:\users\vnihar\anaconda3\lib\site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler<=0.10 in c:\users\vnihar\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools<=4.52.0 in c:\users\vnihar\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver<=1.0.1 in c:\users\vnihar\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging<=20.0 in c:\users\vnihar\anaconda3\lib\site-packages (from matplotlib) (23.1)
Requirement already satisfied: pillow<=9.2.0 in c:\users\vnihar\anaconda3\lib\site-packages (from matplotlib) (10.2.0)
Requirement already satisfied: pyparsing<=2.3.1 in c:\users\vnihar\anaconda3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: scipy<=1.8 in c:\users\vnihar\anaconda3\lib\site-packages (from arch) (1.11.4)
Requirement already satisfied: patsy<=0.5.2 in c:\users\vnihar\anaconda3\lib\site-packages (from statsmodels) (0.5.3)
Requirement already satisfied: soupsieve<1.2 in c:\users\vnihar\anaconda3\lib\site-packages (from beautifulsoup4<=4.11.1->yfinance) (2.5)
Requirement already satisfied: llvmlite<=0.40.0 in c:\users\vnihar\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (0.4)
Requirement already satisfied: webencodings in c:\users\vnihar\anaconda3\lib\site-packages (from html5lib<=1.1->yfinance) (0.5.1)
Requirement already satisfied: charset-normalizer<=4.>=2 in c:\users\vnihar\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (2.0.4)
Requirement already satisfied: idna<=3.5 in c:\users\vnihar\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (3.4)
Requirement already satisfied: urllib3<3.>=1.21.1 in c:\users\vnihar\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (2.0.7)
Requirement already satisfied: certifi<=2021.4.17 in c:\users\vnihar\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (2024.7.4)
Download yfinance-0.2.41-py2.py3-none-any.whl (73 kB)
----- 0.0/73.5 KB ? eta -:-:--
----- 12.5/73.5 KB 4.0 MB/s eta 0:00:00
Download arch-7.0.8-cp311-win_amd64.whl (924 kB)
----- 0.0/924.9 KB ? eta -:-:--
----- 458.6/924.9 KB 14.2 MB/s eta 0:00:01
----- 921.6/924.9 KB 11.7 MB/s eta 0:00:01
----- 924.9/924.9 KB 8.4 MB/s eta 0:00:00
Download html5lib-1.1-py2.py3-none-any.whl (112 kB)
----- 0.0/112.2 KB ? eta -:-:--
----- 112.2/112.2 KB 7 eta 0:00:00
Download multitasking-0.0.11-py3-none-any.whl (8.5 kB)
Building wheels for collected packages: peewee
Building wheel for peewee (pyproject.toml): started
Building wheel for peewee (pyproject.toml): finished with status 'done'
Created wheel for peewee: filename=peewee-3.17.0-py3-none-any.whl size=188939 sha256=3c1a80580c1a6089e4a5c2c0f10d0b0ef7041cef5c5d0f0f50b808a8b00f6c7
Stored in directory: c:\users\vnihar\AppData\Local\Temp\pip\Cache\wheels\1c\0897e9f9f659f0e248edc1722a142c1d74427aa09914a8efc131858
Successfully built peewee
Installing collected packages: peewee, multitasking, html5lib, yfinance, arch
Successfully installed arch-7.0.8 html5lib-1.1 multitasking-0.0.11 peewee-3.17.0 yfinance-0.2.41
```

```
In [5]: # Import necessary libraries
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from arch import arch_model
from statsmodels.tsa.stattools import adfuller
from statsmodels.stats.diagnostic import acorr_ljungbox
```

```
In [7]: # Define the ticker symbol and the date range
ticker = "HUDCO.NS"
start_date = "2019-04-01"
end_date = "2024-03-31"
```

```
In [9]: # Download the data
data = yf.download(ticker, start=start_date, end=end_date)

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

```
In [11]: # Display the first few rows of the data
print(data.head())
```

	Open	High	Low	Close	Adj Close	Volume
Date						
2019-04-01	45.290990	46.750000	45.090998	45.750000	33.416367	1444688
2019-04-02	45.750000	46.400000	45.250000	45.750000	33.416367	637084
2019-04-03	46.000000	46.549999	45.049999	45.200001	33.014645	957089
2019-04-04	45.049999	45.290999	44.599998	44.799999	31.725273	538443
2019-04-05	44.780001	45.099998	44.780001	45.099998	32.941601	490880

```
In [13]: # Check for missing values
missing_values = data.isnull().sum().sum()
print(f"Total number of missing values: {missing_values}")

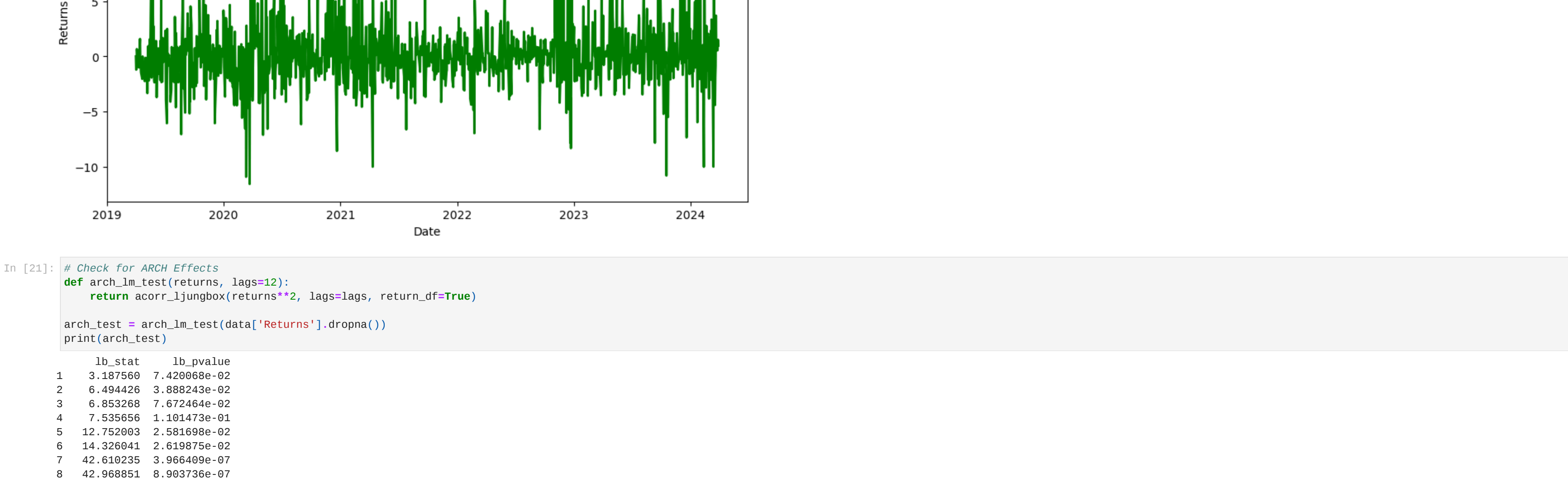
Total number of missing values: 0
```

```
In [15]: # Plot the adjusted closing price for the entire period
ticker = "HUDCO.NS"
plt.figure(figsize=(10, 6))
plt.plot(data[Adj Close], color='blue', lw=2)
plt.title("HUDCO Adjusted Closing Prices (2019-2024)")
plt.xlabel("Date")
plt.ylabel("Adjusted Price")
plt.show()
```



```
In [17]: # Calculate Returns (Using Adjusted Prices)
data['Returns'] = data[Adj Close].pct_change().dropna() * 100
```

```
In [19]: # Plot the returns
plt.figure(figsize=(10, 6))
plt.plot(data[Returns], color='green', lw=2)
plt.title("HUDCO Daily Returns (Adjusted Prices)")
plt.xlabel("Date")
plt.ylabel("Returns (%)")
plt.show()
```



```
In [21]: # Check for ARCH Effects
def arch_lm_test(returns, lags=12):
    return acorr_ljungbox(returns**2, lags=lags, return_df=True)

arch_test = arch_lm_test(data[Returns].dropna())
print(arch_test)
```

	lb_stat	lb_pvalue
1	3.187560	7.420088e-02
2	6.484480	3.885243e-02
3	6.853268	7.672484e-02
4	7.538558	1.101479e-01
5	12.752080	2.581698e-02
6	14.326041	2.619875e-02
7	42.610235	3.966409e-07
8	42.968851	8.903736e-07
9	53.989541	1.896490e-08
10	54.260380	4.322500e-08
11	54.443489	9.796174e-08
12	54.509445	2.215726e-07

```
In [23]: # Fit an ARCH Model and Plot Conditional Volatility
an_arch = arch_model(data[Returns].dropna(), vol='ARCH', p=1)
res_arch = an_arch.fit(disp='off')
print(res_arch.summary())
```

Constant Mean - ARCH Model Results					
Dep. Variable:	Returns	R-squared:	0.000		
Mean Model:	Constant	Mean	Adj. R-squared:	0.000	
Vol Model:	ARCH	Log-likelihood:	-3861.45		
Distribution:	Normal	AIC:	6128.91		
Method:	Maximum Likelihood	BIC:	6144.26		
Date:	Thu, Jul 25 2024	Of Residuals:	1233		
Time:	23:01:12	Of Model:	1		
Mean Model					
coef	std err	t	P> t	95.0% Conf. Int.	
mu	0.1561	8.324e-02	1.004	7.130e-02	[-1.302e-02, 0.313]
Volatility Model					
coef	std err	t	P> t	95.0% Conf. Int.	
omega	7.0545	0.860	8.280	2.875e-16	[-5.370, 0.791]
alpha[1]	0.2280	0.118	1.932	5.334e-02	[-3.285e-03, 0.459]

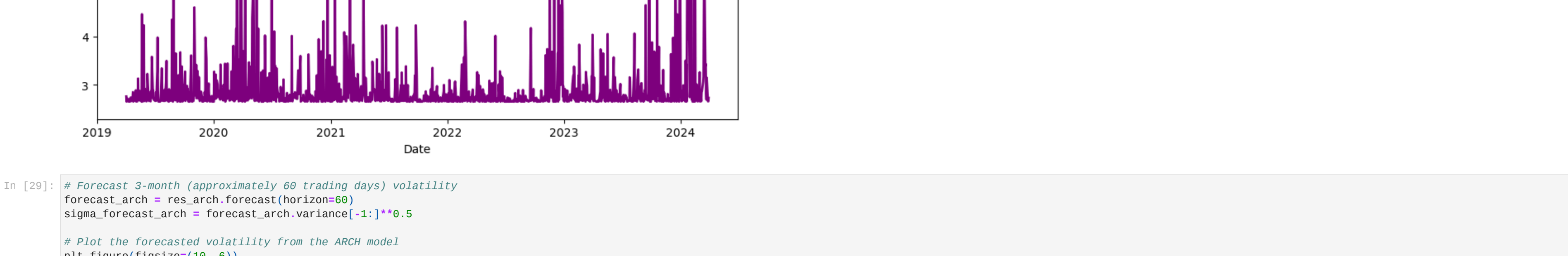
```
In [25]: # Plot the conditional volatility from the ARCH model
plt.figure(figsize=(10, 6))
plt.plot(res_arch.conditional_volatility, color='purple', lw=2)
plt.title("Conditional Volatility (ARCH Model)")
plt.xlabel("Date")
plt.ylabel("Volatility")
plt.show()
```



```
In [28]: # Forecast 3-month (approximately 60 trading days) volatility
forecast_arch = res_arch.forecast(horizon=60)
sigma_forecast_arch = forecast_arch.variance[1:]**0.5
```

```
# Plot the forecasted volatility from the ARCH model
plt.figure(figsize=(10, 6))
plt.plot(sigma_forecast_arch.T, color='purple', lw=2)
plt.title("Forecasted Volatility for 3 Months (ARCH)")
plt.xlabel("Days")
plt.ylabel("Volatility")
```

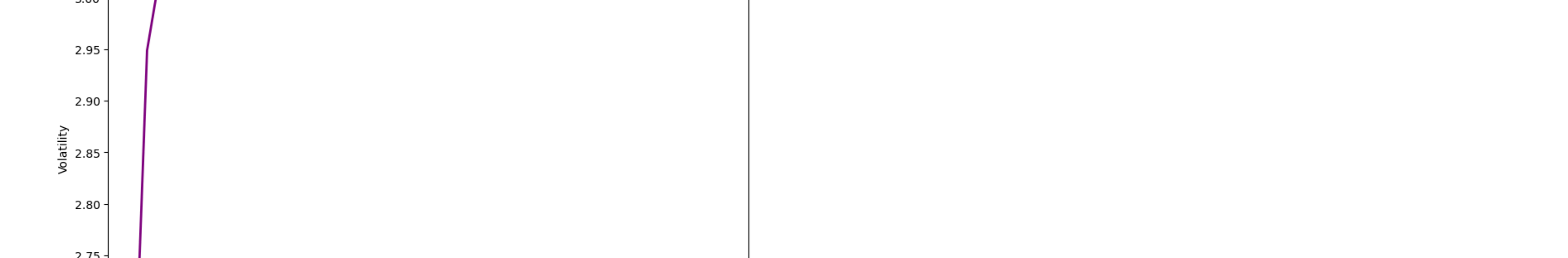
```
# Adjust x-axis to show fewer labels
plt.xticks(ticks=np.arange(0, 60, step=10), labels=np.arange(1, 61, step=10))
plt.show()
```



```
In [31]: # Fit a GARCH Model and Plot Conditional Volatility
an_garch = arch_model(data[Returns].dropna(), vol='GARCH', p=1, q=1)
res_garch = an_garch.fit(disp='off')
print(res_garch.summary())
```

Constant Mean - GARCH Model Results					
Dep. Variable:	Returns	R-squared:	0.000		
Mean Model:	Constant	Mean	Adj. R-squared:	0.000	
Vol Model:	GARCH	Log-likelihood:	-2999.58		
Distribution:	Normal	AIC:	6007.15		
Method:	Maximum Likelihood	BIC:	6027.62		
Date:	Thu, Jul 25 2024	Of Residuals:	1233		
Time:	23:01:13	Of Model:	1		
Mean Model					
coef	std err	t	P> t	95.0% Conf. Int.	
mu	0.1060	7.127e-02	1.520	0.128	[-3.136e-02, 0.248]
Volatility Model					
coef	std err	t	P> t	95.0% Conf. Int.	
omega	0.1750	0.165	1.061	0.289	[-0.148, 0.498]
alpha[1]	0.0764	5.224e-02	1.901	0.133	[-2.397e-02, 0.351]
beta[1]	0.9099	5.937e-02	15.327	5.043e-53	[-0.794, 1.026]

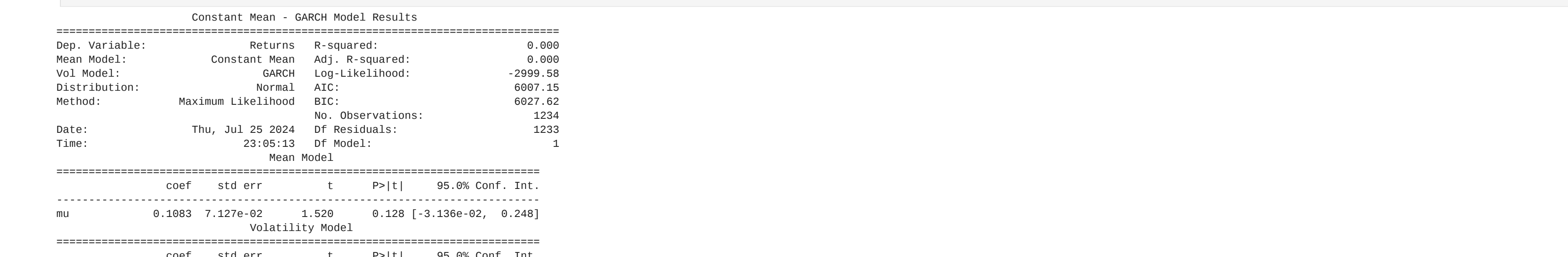
```
In [33]: # Plot the conditional volatility from the GARCH model
plt.figure(figsize=(10, 6))
plt.plot(res_garch.conditional_volatility, color='red', lw=2)
plt.title("Conditional Volatility (GARCH Model)")
plt.xlabel("Date")
plt.ylabel("Volatility")
plt.show()
```



```
In [35]: # Forecast 3-month (approximately 60 trading days) volatility
forecast_garch = res_garch.forecast(horizon=60)
sigma_forecast_garch = forecast_garch.variance[1:]**0.5
```

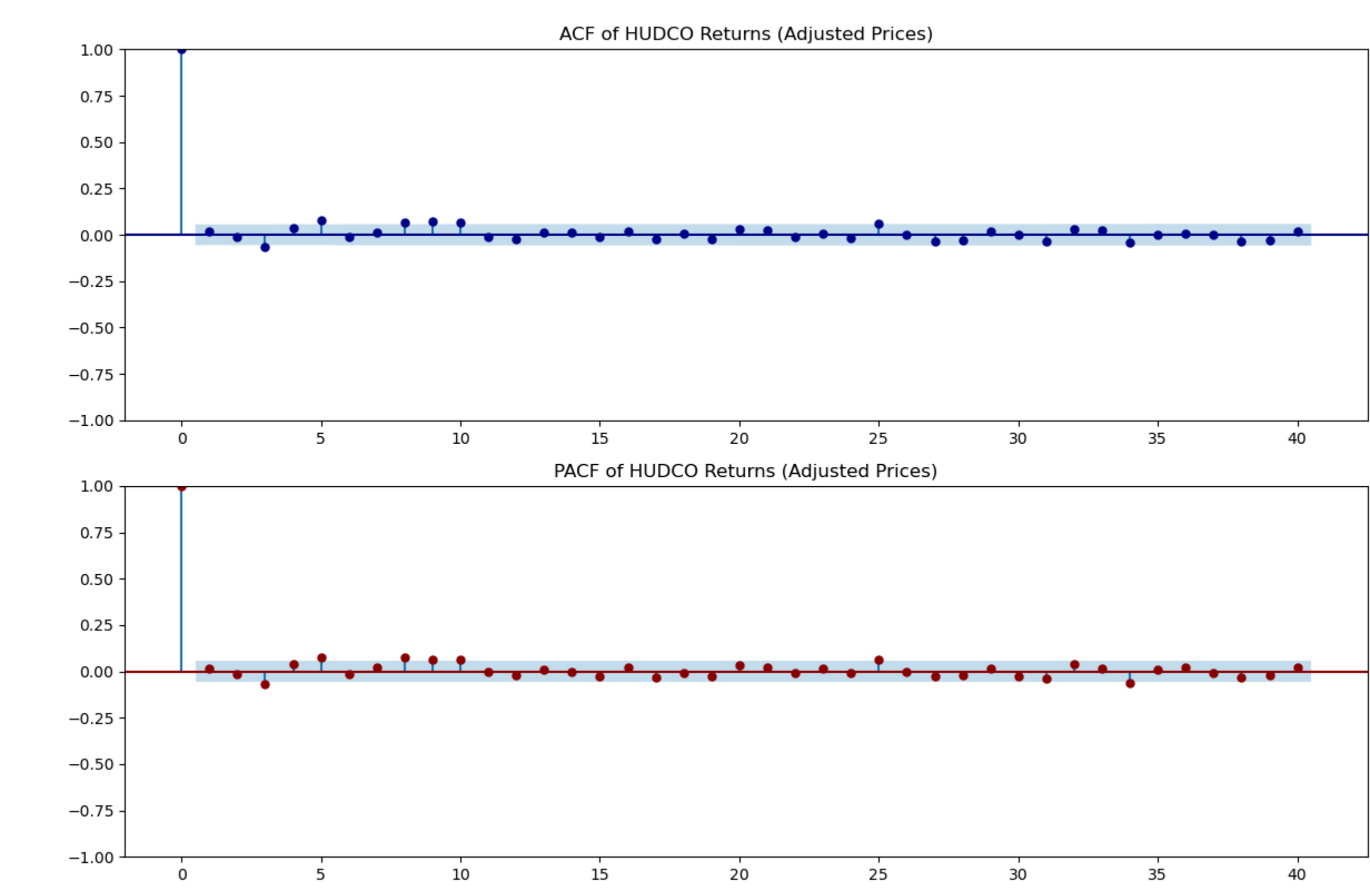
```
In [38]: # Plot the forecasted volatility from the GARCH model
plt.figure(figsize=(10, 6))
plt.plot(sigma_forecast_garch.T, color='red', lw=2)
plt.title("Forecasted Volatility for 3 Months (GARCH)")
plt.xlabel("Days")
plt.ylabel("Volatility")
```

```
# Adjust x-axis to show fewer labels
plt.xticks(ticks=np.arange(0, 60, step=10), labels=np.arange(1, 61, step=10))
plt.show()
```



```
In [43]: # Ensure to import the necessary functions
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

```
In [45]: # Additional Visualizations: ACF and PACF plots of the returns
plt.figure(figsize=(12, 8))
plt.subplot(211)
plt.acf(data[Returns].dropna(), lags=40, ax=plt.gca(), color='darkblue')
plt.title("ACF of HUDCO Returns (Adjusted Prices)")
plt.subplot(212)
plt.pacf(data[Returns].dropna(), lags=40, ax=plt.gca(), color='darkred')
plt.title("PACF of HUDCO Returns (Adjusted Prices)")
plt.tight_layout()
plt.show()
```

In []: