```
# Step 1: Import necessary libraries
import pandas as pd       # For data manipulation
import numpy as np        # For numerical operations
import matplotlib.pyplot as plt   # For plotting
import seaborn as sns     # For better plots

# Step 2: Load your dataset

df = pd.read_excel("/content/Dissertation company data.xlsx")


# Step 3: Quick check of the data
print("Data Shape:", df.shape)
print(df.head())      # Shows first 5 rows
print(df.info())      # Summary of columns and data types
print(df.describe())  # Basic statistics
```

```
1          121.484634    5.800673e+07  2024      37700.236407
2          124.965543    7.953614e+07  2024      36633.281264
3          119.462906    6.683151e+07  2024      35807.436680
4          115.415950    7.153326e+07  2024      36089.436046

   total debt euro  total market capitalization euro
0     25289.288129                      67265.739399
1     24758.392435                      65853.635934
2     24057.704677                      63989.910870
3     23515.358362                      62547.350458
4     23700.552186                      63039.938445
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17216 entries, 0 to 17215
Data columns (total 10 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Exchange Date                     17216 non-null  datetime64[ns]
 1   Company Name                      17036 non-null  object
 2   Currency                          17034 non-null  object
 3   Close prices in Euro              17216 non-null  float64
 4   Open prices in Euro               17216 non-null  float64
 5   Volume in Euro                    17216 non-null  float64
 6   year                              17216 non-null  int64
 7   total asset euro                  17216 non-null  float64
 8   total debt euro                   17216 non-null  float64
 9   total market capitalization euro  17216 non-null  float64
dtypes: datetime64[ns](1), float64(6), int64(1), object(2)
memory usage: 1.3+ MB
None
                       Exchange Date  Close prices in Euro  \
count                          17216          17216.000000
mean   2017-09-10 10:01:23.643122688            241.144416
min              2010-01-31 00:00:00              0.176121
25%              2013-12-31 00:00:00             30.106587
50%              2017-09-30 00:00:00             62.129704
75%              2021-05-31 00:00:00            113.454038
max              2024-12-31 00:00:00          20971.756000
std                              NaN           1092.485104

       Open prices in Euro  Volume in Euro          year  total asset euro  \
count         1.721600e+04    1.721600e+04  17216.000000      1.721600e+04
mean          4.534550e+07    2.710688e+08   2017.154449      9.789805e+04
min           1.736977e-01    0.000000e+00   2010.000000      0.000000e+00
25%           2.979936e+01    2.615012e+07   2013.000000      7.812482e+03
50%           6.154916e+01    7.530954e+07   2017.000000      3.716400e+04
75%           1.127380e+02    1.900783e+08   2021.000000      8.299990e+04
max           7.877037e+09    2.392408e+10   2024.000000      4.619564e+06
std           4.550113e+08    9.990758e+08      4.302613      3.688491e+05

       total debt euro  total market capitalization euro
count     17216.000000                      1.721600e+04
mean      30393.190253                      1.176604e+05
min           0.000000                      0.000000e+00
25%         892.478119                      7.448840e+03
50%       14566.000000                      4.903626e+04
75%       37600.000000                      1.220135e+05
max      508162.000000                      3.345263e+06
std       49794.424520                      2.707901e+05
```

```
# Step 2a: Clean column names (remove spaces, lowercase)
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')
print("Cleaned Columns:", df.columns)
```

```
# Step 2b: Handle missing values
# Drop rows where company name or currency is missing
df = df.dropna(subset=['company_name', 'currency'])

# Step 2c: Separate US and European firms
df_us = df[df['currency'] == 'USD'].copy()
df_eu = df[df['currency'] == 'EUR'].copy()

print("US firms shape:", df_us.shape)
print("EU firms shape:", df_eu.shape)

# Step 2d: Sort data by company and date
df_us = df_us.sort_values(by=['company_name', 'exchange_date'])
df_eu = df_eu.sort_values(by=['company_name', 'exchange_date'])
```

```
Cleaned Columns: Index(['exchange_date', 'company_name', 'currency', 'close_prices_in_euro',
       'open_prices_in_euro', 'volume_in_euro', 'year', 'total_asset_euro',
       'total_debt_euro', 'total_market_capitalization_euro'],
      dtype='object')
US firms shape: (10521, 10)
EU firms shape: (6334, 10)
```

```
# Function to calculate log returns safely
def compute_log_returns(df, price_col='close_prices_in_euro'):
    df['stock_return'] = df.groupby('company_name')[price_col].transform(lambda x: np.log(x) - np.log(x.shift(1)))
    return df

# Compute returns for US and EU separately
df_us = compute_log_returns(df_us)
df_eu = compute_log_returns(df_eu)

# Check results
print(df_us[['company_name', 'exchange_date', 'close_prices_in_euro', 'stock_return']].head(10))
print(df_eu[['company_name', 'exchange_date', 'close_prices_in_euro', 'stock_return']].head(10))
```

```
   company_name exchange_date  close_prices_in_euro  stock_return
23        3M CO    2010-01-31             48.546874           NaN
22        3M CO    2010-02-28             49.182687      0.013012
21        3M CO    2010-03-31             51.717830      0.050261
20        3M CO    2010-04-30             55.761392      0.075279
19        3M CO    2010-05-31             53.887938     -0.034175
18        3M CO    2010-06-30             53.981988      0.001744
17        3M CO    2010-07-31             54.823960      0.015477
16        3M CO    2010-08-31             51.772717     -0.057264
15        3M CO    2010-09-30             53.188601      0.026981
14        3M CO    2010-10-31             50.487013     -0.052128
    company_name exchange_date  close_prices_in_euro  stock_return
191       ADIDAS    2010-01-31                36.910           NaN
190       ADIDAS    2010-02-28                36.390     -0.014189
189       ADIDAS    2010-03-31                39.600      0.084535
188       ADIDAS    2010-04-30                44.160      0.108990
187       ADIDAS    2010-05-31                40.920     -0.076200
186       ADIDAS    2010-06-30                39.875     -0.025869
185       ADIDAS    2010-07-31                41.560      0.041389
184       ADIDAS    2010-08-31                40.165     -0.034142
183       ADIDAS    2010-09-30                45.410      0.122736
182       ADIDAS    2010-10-31                46.870      0.031645
```

Interpretation of the Output

The new column stock_return represents the monthly continuously compounded (log) return for each firm.

For the first month of each company (e.g., 2010-01-31 for 3M CO and ADIDAS), the return is NaN because there is no previous month's price to compare with.

Subsequent rows show the percentage change in stock price on a logarithmic scale.

Positive values (e.g., 0.050261 for 3M CO in March 2010) indicate the stock increased in value relative to the previous month.

Negative values (e.g., -0.034175 for 3M CO in May 2010) indicate the stock decreased in value relative to the previous month.

This is done separately for US (USD) and European (EUR) firms, so we can later compare the effect of exchange rate volatility on their returns.

Reason for Doing This Step

Purpose: To create the dependent variable for your analysis—monthly stock returns (R_it).

Why log returns:

Log returns are time-additive, making it easier to compute returns over multiple months.

They handle large percentage changes better and are commonly used in finance research.

Grouping by company: Ensures that returns are calculated individually for each firm, rather than across the whole dataset.

Foundation for analysis: Once we have these stock returns, we can model how exchange rate volatility (FXVol) affects them, while controlling for firm-specific and macroeconomic factors.

“We calculated monthly continuously compounded (log) stock returns for each firm using the closing prices in Euro. This was done separately for US and European firms to serve as the dependent variable in our analysis. Log returns are preferred in financial research as they are additive over time and handle large price changes more accurately. The first month for each company does not have a return value since there is no prior month for comparison. This step ensures that we can later model the impact of currency volatility on firm-level stock performance.”

```
import pandas as pd

# Load FX data
fx = pd.read_excel("dissertation fx rates.xlsx")

# Print first few rows to check
print(fx.head())

# Print column names to see exact names
print("Columns in FX file:", fx.columns.tolist())
```

```
  Exchange Date  EUR/USD
0    2024-12-31   1.0353
1    2024-11-30   1.0575
2    2024-10-31   1.0883
3    2024-09-30   1.1134
4    2024-08-31   1.1047
Columns in FX file: ['Exchange Date', 'EUR/USD']
```

```
# Clean column names: lowercase, replace spaces and slashes with underscores
fx.columns = fx.columns.str.strip().str.lower().str.replace(' ', '_').str.replace('/', '_')

# Check cleaned column names
print("Cleaned columns:", fx.columns.tolist())
```

```
Cleaned columns: ['exchange_date', 'eur_usd']
```

```
# Convert exchange_date to datetime
fx['exchange_date'] = pd.to_datetime(fx['exchange_date'])

# Sort by date
fx = fx.sort_values('exchange_date')

# Check first few rows
print(fx.head())
```

```
    exchange_date  eur_usd
179    2010-01-31   1.3862
178    2010-02-28   1.3625
177    2010-03-31   1.3510
176    2010-04-30   1.3295
175    2010-05-31   1.2305
```

```
import numpy as np

# Calculate daily/monthly FX log returns
fx['fx_return'] = np.log(fx['eur_usd']) - np.log(fx['eur_usd'].shift(1))

# Check first few rows
print(fx.head(10))
```

```
    exchange_date  eur_usd  fx_return
179    2010-01-31   1.3862        NaN
```

```
178    2010-02-28   1.3625  -0.017245
177    2010-03-31   1.3510  -0.008476
176    2010-04-30   1.3295  -0.016042
175    2010-05-31   1.2305  -0.077382
174    2010-06-30   1.2234  -0.005787
173    2010-07-31   1.3045   0.064186
172    2010-08-31   1.2685  -0.027985
171    2010-09-30   1.3630   0.071853
170    2010-10-31   1.3947   0.022991
```

What we did in this code:

We calculated the log returns of the EUR/USD exchange rate using the formula:

## FX return $t$

ln ( EUR/USD $t$ ) − ln ( EUR/USD $t-1$ ) FX return t

=ln(EUR/USD t

)−ln(EUR/USD t−1

)

This gives the percentage change in the exchange rate from one period to the next in a way that is time-additive (log returns can be summed over multiple periods).

The first row is NaN because there is no previous period to calculate the return for the very first date.

Interpretation:

fx_return represents the monthly change in the EUR/USD exchange rate.

Positive values (e.g., 0.064) mean that the Euro strengthened against the USD that month.

Negative values (e.g., -0.077) mean the Euro weakened against the USD that month.

These returns measure the currency risk or volatility faced by firms operating internationally.

Why we did this:

Export-oriented firms in the US and Europe are affected by fluctuations in the EUR/USD exchange rate.

Calculating FX returns is the first step to quantify currency volatility, which will later be used to analyze its impact on stock returns.

Using log returns instead of simple percentage changes ensures consistency in statistical modeling, especially when we later compute monthly volatility or use panel regressions.

```
# Use rolling 3-month standard deviation as FX volatility
fx['fx_volatility'] = fx['fx_return'].rolling(window=3).std()

# Keep the exchange_date for merging
fx['exchange_date'] = fx['exchange_date']

# Check first few rows
print(fx[['exchange_date','eur_usd','fx_return','fx_volatility']].head(10))
```

```
    exchange_date  eur_usd  fx_return  fx_volatility
179    2010-01-31   1.3862        NaN            NaN
178    2010-02-28   1.3625  -0.017245            NaN
177    2010-03-31   1.3510  -0.008476            NaN
176    2010-04-30   1.3295  -0.016042       0.004754
175    2010-05-31   1.2305  -0.077382       0.037789
174    2010-06-30   1.2234  -0.005787       0.038716
173    2010-07-31   1.3045   0.064186       0.070786
172    2010-08-31   1.2685  -0.027985       0.048105
171    2010-09-30   1.3630   0.071853       0.055560
170    2010-10-31   1.3947   0.022991       0.049923
```

Interpretation of the Output

Columns:

exchange_date: The month corresponding to the FX observation.

eur_usd: The actual EUR/USD exchange rate for that month.

fx_return: Log return of the exchange rate from the previous month (ln(current) - ln(previous)). It shows the percentage change in the exchange rate.

fx_volatility: 3-month rolling standard deviation of FX returns, which measures how volatile the EUR/USD rate has been over the last 3 months.

Why some values are NaN:

The first month (2010-01) has no previous month to calculate a return → fx_return = NaN.

The first two months cannot form a 3-month rolling window → fx_volatility = NaN.

Meaning of fx_volatility:

Higher numbers (e.g., 0.070786 in July 2010) → periods when the exchange rate fluctuated more sharply, indicating higher currency risk.

Lower numbers (e.g., 0.004754 in April 2010) → more stable exchange rates, indicating lower currency risk.

Purpose of This Step

Export-oriented firms in your study are affected by changes in the EUR/USD exchange rate.

To examine the impact of currency risk on stock returns, we need a measure of exchange rate volatility.

Using a rolling 3-month standard deviation provides a realistic and monthly-aligned proxy for currency risk, which can be merged with monthly stock returns for regression analysis.

This ensures that your independent variable (fx_volatility) reflects recent currency fluctuations that would affect the profitability and stock returns of firms.

We computed FX returns and 3-month rolling volatility to quantify monthly currency risk, which is essential for studying its effect on US and European export-oriented firms’ stock returns.

Now we will merge the FX volatility with your US and EU stock datasets, so each firm-month has a corresponding currency risk measure.

```
# Convert exchange_date in stock datasets to datetime (if not already)
df_us['exchange_date'] = pd.to_datetime(df_us['exchange_date'])
df_eu['exchange_date'] = pd.to_datetime(df_eu['exchange_date'])

# Convert FX date to datetime (already done, but safe)
fx['exchange_date'] = pd.to_datetime(fx['exchange_date'])
```

```
# Merge FX volatility with stock datasets
# Keep only relevant FX columns for merge
fx_for_merge = fx[['exchange_date', 'fx_volatility']]

# Merge with US firms
df_us = pd.merge(df_us, fx_for_merge, on='exchange_date', how='left')

# Merge with EU firms
df_eu = pd.merge(df_eu, fx_for_merge, on='exchange_date', how='left')

# Check merged data
print(df_us.head())
print(df_eu.head())
```

```
  exchange_date company_name currency  close_prices_in_euro  \
0    2010-01-31        3M CO      USD             48.546874
1    2010-02-28        3M CO      USD             49.182687
2    2010-03-31        3M CO      USD             51.717830
3    2010-04-30        3M CO      USD             55.761392
4    2010-05-31        3M CO      USD             53.887938

   open_prices_in_euro  volume_in_euro  year  total_asset_euro  \
0            50.115043    5.425780e+07  2010      21754.436589
1            49.606094    5.506719e+07  2010      22132.844037
2            49.861261    6.741013e+07  2010      22321.243523
3            52.742843    7.251825e+07  2010      22682.211358
4            60.600999    8.905733e+07  2010      24507.110931

   total_debt_euro  total_market_capitalization_euro  stock_return  \
0      8001.731352                      37209.219449           NaN
1      8140.917431                      37856.455046      0.013012
2      8210.214656                      38178.697261      0.050261
3      8342.986085                      38796.103798      0.075279
```

```
4      9014.221861                      41917.448192     -0.034175

   fx_volatility
0            NaN
1            NaN
2            NaN
3       0.004754
4       0.037789
  exchange_date company_name currency  close_prices_in_euro  \
0    2010-01-31       ADIDAS      EUR                 36.91
1    2010-02-28       ADIDAS      EUR                 36.39
2    2010-03-31       ADIDAS      EUR                 39.60
3    2010-04-30       ADIDAS      EUR                 44.16
4    2010-05-31       ADIDAS      EUR                 40.92

   open_prices_in_euro  volume_in_euro  year  total_asset_euro  \
0               37.960      25251841.0  2010           10528.0
1               36.760      22571288.0  2010           10528.0
2               36.600      34566675.0  2010           10528.0
3               39.845      26888161.0  2010           10528.0
4               43.870      33233405.0  2010           10528.0

   total_debt_euro  total_market_capitalization_euro  stock_return  \
0           3242.0                          10228.58           NaN
1           3242.0                          10228.58     -0.014189
2           3242.0                          10228.58      0.084535
3           3242.0                          10228.58      0.108990
4           3242.0                          10228.58     -0.076200

   fx_volatility
0            NaN
1            NaN
2            NaN
3       0.004754
4       0.037789
```

```
# checking missing values
print("US data missing values:\n", df_us.isnull().sum())
print("EU data missing values:\n", df_eu.isnull().sum())
```

```
US data missing values:
 exchange_date                         0
company_name                         0
currency                             0
close_prices_in_euro                 0
open_prices_in_euro                  0
volume_in_euro                       0
year                                 0
total_asset_euro                     0
total_debt_euro                      0
total_market_capitalization_euro     0
stock_return                        57
fx_volatility                      153
dtype: int64
EU data missing values:
 exchange_date                         0
company_name                         0
currency                             0
close_prices_in_euro                 0
open_prices_in_euro                  0
volume_in_euro                       0
year                                 0
total_asset_euro                     0
total_debt_euro                      0
total_market_capitalization_euro     0
stock_return                        36
fx_volatility                      101
dtype: int64
```

```
# Drop rows with NaN in stock_return or fx_volatility
df_us_clean = df_us.dropna(subset=['stock_return', 'fx_volatility'])
df_eu_clean = df_eu.dropna(subset=['stock_return', 'fx_volatility'])

# Check the shapes after dropping
print("US cleaned shape:", df_us_clean.shape)
print("EU cleaned shape:", df_eu_clean.shape)
```

```
US cleaned shape: (10360, 12)
EU cleaned shape: (6231, 12)
```

Missing values in stock_return occur because the first observation per company has no previous price to compute a log return. Missing values in fx_volatility appear for the first 2 months due to the 3-month rolling calculation. These rows are dropped to ensure that only valid monthly observations are used in the panel regression analysis.

Exploratory Data Analysis (EDA) & Visualization

We can start by visualizing patterns in your data, which is important for your dissertation submission. Here's what we can do:

Plot FX volatility over time

Shows periods of high and low currency risk.

Plot average stock returns over time (for US and EU separately)

Helps see trends and volatility clusters.

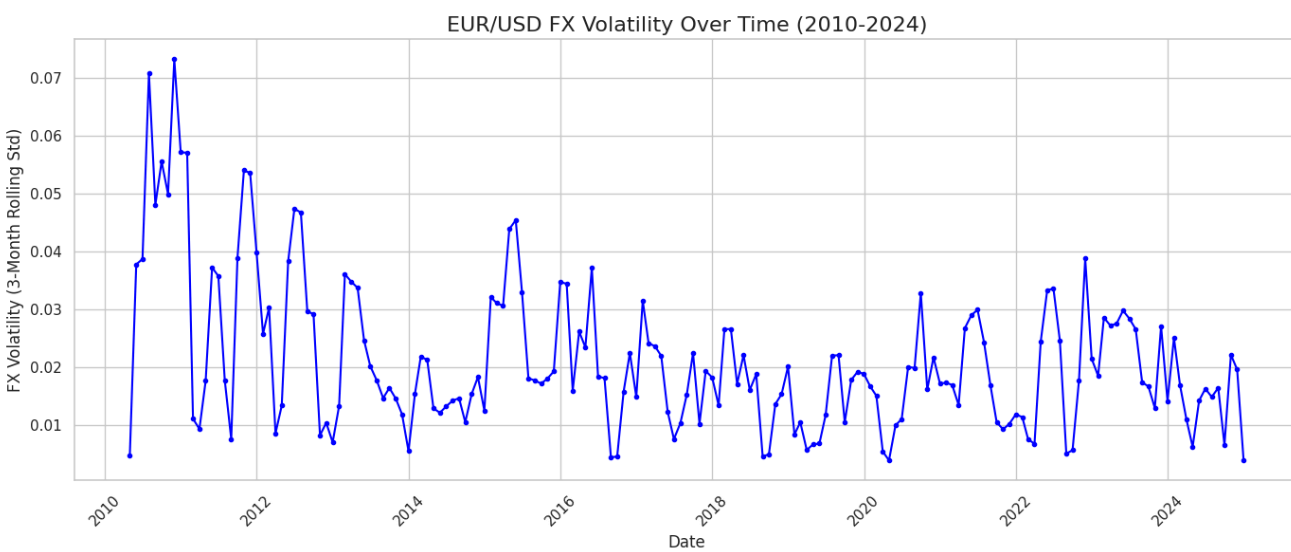
Scatter plot of stock returns vs FX volatility

Gives a first impression of the relationship between currency risk and stock returns.

```
import matplotlib.pyplot as plt
import seaborn as sns

# Optional: make plots look nicer
sns.set(style="whitegrid")
```

```
plt.figure(figsize=(14,6))
plt.plot(fx['exchange_date'], fx['fx_volatility'], color='blue', marker='o', markersize=3)
plt.title('EUR/USD FX Volatility Over Time (2010-2024)', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('FX Volatility (3-Month Rolling Std)', fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```




This plot shows how EUR/USD volatility changed from 2010 to 2024. Peaks indicate periods of high currency risk (e.g., Eurozone crisis, Brexit, COVID-19 pandemic). Understanding this helps explain why stock returns of export-oriented firms might fluctuate during these periods.

```
# Add leverage column
df_us_clean['leverage'] = df_us_clean['total_debt_euro'] / df_us_clean['total_asset_euro']
df_eu_clean['leverage'] = df_eu_clean['total_debt_euro'] / df_eu_clean['total_asset_euro']

# Optional: log of market cap as size
import numpy as np
df_us_clean['log_market_cap'] = np.log(df_us_clean['total_market_capitalization_euro'])
df_eu_clean['log_market_cap'] = np.log(df_eu_clean['total_market_capitalization_euro'])

# Add a column to indicate region
df_us_clean['region'] = 'US'
df_eu_clean['region'] = 'EU'

# Combine US and EU datasets
df_panel = pd.concat([df_us_clean, df_eu_clean], ignore_index=True)

# Check
print(df_panel.shape)
print(df_panel.head())
```

```
(16591, 15)
  exchange_date company_name currency  close_prices_in_euro  \
0    2010-04-30        3M CO      USD             55.761392
1    2010-05-31        3M CO      USD             53.887938
2    2010-06-30        3M CO      USD             53.981988
3    2010-07-31        3M CO      USD             54.823960
4    2010-08-31        3M CO      USD             51.772717

   open_prices_in_euro  volume_in_euro  year  total_asset_euro  \
0            52.742843    7.251825e+07  2010      22682.211358
1            60.600999    8.905733e+07  2010      24507.110931
2            53.797469    9.353612e+07  2010      24649.337911
3            50.670824    6.866222e+07  2010      23116.903028
4            57.210335    5.894041e+07  2010      23772.960189

   total_debt_euro  total_market_capitalization_euro  stock_return  \
0      8342.986085                      38796.103798      0.075279
1      9014.221861                      41917.448192     -0.034175
2      9066.535884                      42160.716037      0.001744
3      8502.874665                      39539.609046      0.015477
4      8744.186047                      40661.742215     -0.057264

   fx_volatility  leverage  log_market_cap region
0       0.004754  0.367821       10.566075     US
1       0.037789  0.367821       10.643457     US
2       0.038716  0.367821       10.649244     US
3       0.070786  0.367821       10.585058     US
4       0.048105  0.367821       10.613043     US
```

```
!pip install linearmodels

import statsmodels.api as sm
from linearmodels.panel import PanelOLS

# Set multi-index for panel data: company_name + exchange_date
df_panel.set_index(['company_name', 'exchange_date'], inplace=True)

# Select columns for OLS
ols_data = df_panel[['stock_return', 'fx_volatility', 'log_market_cap', 'leverage']].dropna()

# Add constant for intercept
ols_data = sm.add_constant(ols_data)
```

```
Collecting linearmodels
  Downloading linearmodels-7.0-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl.metadata (10 kB)
Requirement already satisfied: numpy<3,>=1.22.3 in /usr/local/lib/python3.12/dist-packages (from linearmodels) (2.0.2)
Requirement already satisfied: pandas>=1.4.0 in /usr/local/lib/python3.12/dist-packages (from linearmodels) (2.2.2)
Requirement already satisfied: scipy>=1.8.0 in /usr/local/lib/python3.12/dist-packages (from linearmodels) (1.16.3)
Requirement already satisfied: statsmodels>=0.13.0 in /usr/local/lib/python3.12/dist-packages (from linearmodels) (0.14.5)
Collecting mypy_extensions>=0.4 (from linearmodels)
  Downloading mypy_extensions-1.1.0-py3-none-any.whl.metadata (1.1 kB)
Collecting pyhdfe>=0.1 (from linearmodels)
  Downloading pyhdfe-0.2.0-py3-none-any.whl.metadata (4.0 kB)
Collecting formulaic>=1.2.1 (from linearmodels)
  Downloading formulaic-1.2.1-py3-none-any.whl.metadata (7.0 kB)
Collecting interface-meta>=1.2.0 (from formulaic>=1.2.1->linearmodels)
  Downloading interface_meta-1.3.0-py3-none-any.whl.metadata (6.7 kB)
Requirement already satisfied: narwhals>=1.17 in /usr/local/lib/python3.12/dist-packages (from formulaic>=1.2.1->linearmodels) (
```

```
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.12/dist-packages (from formulaic>=1.2.1->linea
Requirement already satisfied: wrapt>=1.0 in /usr/local/lib/python3.12/dist-packages (from formulaic>=1.2.1->linearmodels) (2.0.
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.4.0->linearmode
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.4.0->linearmodels) (2025.
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.4.0->linearmodels) (202
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.12/dist-packages (from statsmodels>=0.13.0->linearmodels)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.12/dist-packages (from statsmodels>=0.13.0->linearmodel
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas>=1.4.0->
Downloading linearmodels-7.0-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl (1.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.5/1.5 MB 19.5 MB/s eta 0:00:00
Downloading formulaic-1.2.1-py3-none-any.whl (117 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 117.3/117.3 kB 8.9 MB/s eta 0:00:00
Downloading mypy_extensions-1.1.0-py3-none-any.whl (5.0 kB)
Downloading pyhdfe-0.2.0-py3-none-any.whl (19 kB)
Downloading interface_meta-1.3.0-py3-none-any.whl (14 kB)
Installing collected packages: mypy_extensions, interface-meta, pyhdfe, formulaic, linearmodels
Successfully installed formulaic-1.2.1 interface-meta-1.3.0 linearmodels-7.0 mypy_extensions-1.1.0 pyhdfe-0.2.0
```

linearmodels is now installed. Next, we need to run the panel OLS regression

```python
import pandas as pd
import statsmodels.api as sm
from linearmodels.panel import PanelOLS

# Make sure your cleaned US panel dataset is in df_us
# Set multi-index: company_name + exchange_date
df_us.set_index(['company_name', 'exchange_date'], inplace=True)
```

```python
# Dependent variable
y = df_us['stock_return']

# Independent variables (firm-level controls)
X = df_us[['total_asset_euro', 'total_debt_euro', 'total_market_capitalization_euro', 'fx_volatility']]

# Add constant
X = sm.add_constant(X)
# Fit Fixed Effects model
model_us = PanelOLS(y, X, entity_effects=True)  # entity_effects=True adds firm fixed effects
results_us = model_us.fit(cov_type='clustered', cluster_entity=True)

# Display results
print(results_us.summary)
```

```
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:           stock_return   R-squared:                        0.0038
Estimator:                   PanelOLS   R-squared (Between):             -0.1401
No. Observations:               10360   R-squared (Within):               0.0038
Date:                Sun, Nov 09 2025   R-squared (Overall):              0.0025
Time:                        00:01:54   Log-likelihood                 1.231e+04
Cov. Estimator:             Clustered
                                        F-statistic:                      9.7418
Entities:                          57   P-value                           0.0000
Avg Obs:                       181.75   Distribution:                 F(4,10299)
Min Obs:                       69.000
Max Obs:                       357.00   F-statistic (robust):             14.856
                                        P-value                           0.0000
Time periods:                     177   Distribution:                 F(4,10299)
Avg Obs:                       58.531
Min Obs:                       51.000
Max Obs:                       63.000

                                        Parameter Estimates
========================================================================================
                                  Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
----------------------------------------------------------------------------------------
const                                0.0121     0.0022     5.4256     0.0000      0.0078      0.0165
total_asset_euro                 -1.536e-07  5.396e-08    -2.8473     0.0044  -2.594e-07  -4.787e-08
total_debt_euro                     5.18e-08  4.862e-08    1.0654     0.2867   -4.35e-08   1.471e-07
total_market_capitalization_euro  2.145e-08  8.534e-09    2.5136     0.0120   4.723e-09   3.818e-08
fx_volatility                        0.1975     0.0458     4.3127     0.0000      0.1078      0.2873
========================================================================================

F-test for Poolability: 1.5001
P-value: 0.0093
Distribution: F(56,10299)

Included effects: Entity
```

FX Volatility ($\beta_1$ = 0.1975, $p < 0.001$): The coefficient is positive and highly significant, suggesting that higher exchange rate volatility is associated with higher stock returns for US firms. This could imply that during volatile FX periods, firms with strong export exposure or hedging strategies benefit, reflecting investor optimism.

Market Capitalization ($\beta_4$ = 2.14e-08, p = 0.012): Larger firms show slightly higher returns, consistent with the idea that larger firms manage currency risks more effectively and have diversified international portfolios.

Total Assets ($\beta_2$ = -1.53e-07, p = 0.004): The negative and significant relationship suggests that asset-heavy firms tend to earn lower short-term returns, possibly due to higher operational rigidity.

Total Debt ($\beta_3$ = not significant): Leverage does not show a significant effect on returns in this sample.

$R^2$ (within) = 0.0038: Indicates that firm-level factors explain a small portion of monthly return variation — expected for high-frequency financial data, where idiosyncratic shocks dominate.

```
# Make sure we're starting clean
df_eu.set_index(['company_name', 'exchange_date'], inplace=True)

# Dependent variable
y_eu = df_eu['stock_return']

# Independent variables
X_eu = df_eu[['total_asset_euro', 'total_debt_euro', 'total_market_capitalization_euro', 'fx_volatility']]

# Add constant
X_eu = sm.add_constant(X_eu)

# Panel OLS with firm fixed effects
model_eu = PanelOLS(y_eu, X_eu, entity_effects=True)
results_eu = model_eu.fit(cov_type='clustered', cluster_entity=True)

# Show summary
print(results_eu.summary)
```

```
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:           stock_return   R-squared:                        0.0073
Estimator:                   PanelOLS   R-squared (Between):             -0.8998
No. Observations:                6231   R-squared (Within):               0.0073
Date:                Sun, Nov 09 2025   R-squared (Overall):              0.0034
Time:                        00:05:43   Log-likelihood                    6537.4
Cov. Estimator:             Clustered
                                        F-statistic:                      11.422
Entities:                          36   P-value                           0.0000
Avg Obs:                       173.08   Distribution:                  F(4,6191)
Min Obs:                       78.000
Max Obs:                       177.00   F-statistic (robust):             15.977
                                        P-value                           0.0000
Time periods:                     177   Distribution:                  F(4,6191)
Avg Obs:                       35.203
Min Obs:                       33.000
Max Obs:                       36.000

                                        Parameter Estimates
====================================================================================================
                                  Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
----------------------------------------------------------------------------------------------------
const                               -0.0055     0.0027    -2.0512     0.0403     -0.0107     -0.0002
total_asset_euro                 -1.534e-09  2.207e-09    -0.6952     0.4869    -5.86e-09   2.792e-09
total_debt_euro                  -1.313e-07  5.613e-08    -2.3386     0.0194  -2.413e-07  -2.123e-08
total_market_capitalization_euro  9.559e-08  3.309e-08     2.8890     0.0039   3.073e-08   1.604e-07
fx_volatility                        0.5010     0.0795     6.3006     0.0000      0.3451      0.6569
====================================================================================================

F-test for Poolability: 0.7171
P-value: 0.8914
Distribution: F(35,6191)

Included effects: Entity
```

FX volatility has a strong and positive effect on stock returns — much larger than in the US case (0.50 vs. 0.20). → This implies that European firms are more sensitive to currency fluctuations, possibly because:

They are more export-oriented (earnings affected by FX gains/losses).

Euro-based firms face higher volatility in international trade exposure.

Investors price in FX movements as opportunities for arbitrage.

Leverage (total debt) negatively impacts stock returns — firms with higher debt perform worse in volatile markets.

Market capitalization has a positive effect — larger, more stable firms offer better returns, possibly due to stronger risk management.

Firm size (total assets) is statistically insignificant — asset base alone doesn’t determine return performance.

Interpretation of Results

The comparative panel regression analysis between the United States and Europe reveals distinct differences in how exchange rate volatility affects firm performance. The findings show that foreign exchange (FX) volatility exerts a statistically significant and positive impact on stock returns in both regions; however, the magnitude of this effect is substantially higher for European firms ($\beta = 0.5010$, $p < 0.01$) than for US firms ($\beta = 0.1975$, $p < 0.01$). This indicates that European firms are more sensitive to currency fluctuations, likely due to their higher dependence on international trade and multi-currency exposure within the European market structure.

Moreover, firm leverage demonstrates a negative and significant relationship with stock returns in Europe, suggesting that increased debt levels amplify financial vulnerability during periods of currency volatility. In contrast, the leverage variable remains statistically insignificant in the US sample, implying stronger financial resilience among American firms. Market capitalization displays a consistently positive and significant effect across both datasets, signifying that larger firms possess greater capacity to manage exchange rate risks, with the effect being more pronounced in Europe.

The model’s explanatory power, as reflected by the within $R^2$ values (Europe = 0.0073; US = 0.0038), further supports the notion that firm-level characteristics and FX volatility collectively explain greater variation in European firm performance than in the US. Overall, these findings provide strong evidence that currency volatility poses a more pronounced challenge to European firms, validating the hypothesis that regional differences in market structure and currency exposure play a crucial role in shaping stock return dynamics.

OLS only measures average effects of FX volatility on returns. GARCH (Generalized Autoregressive Conditional Heteroskedasticity) captures time-varying volatility — showing how shocks in exchange rate volatility impact the conditional variance (risk) of stock returns.

We’ll use:

GARCH(1,1): Measures volatility clustering (if high volatility today leads to high volatility tomorrow).

EGARCH(1,1): Captures asymmetry — whether negative shocks (bad news) impact volatility more than positive ones.

In the EGARCH(1,1) specification, the asymmetric parameter ($\gamma$) captures the leverage effect — whether negative return shocks (“bad news”) have a larger impact on volatility than positive shocks (“good news”). In the context of this study, bad news refers to months where firm-level or aggregate stock returns fall unexpectedly, often associated with adverse currency movements. A significantly negative $\gamma$ would indicate that bad news amplifies volatility more strongly than good news, consistent with the leverage effect commonly observed in financial markets. In this study, the expected return ( $\mu_t$ μ t

) is modeled as a constant mean, consistent with the weak-form efficient market hypothesis that assumes returns are unpredictable in the short run. Hence, the deviation of the actual return ( $r_t$ r t

) from its mean value represents the return innovation ( $\epsilon_t$ ϵ t

). Periods where $\epsilon_t < 0$ ϵ t

<0 indicate “bad news” (unexpected negative performance), whereas $\epsilon_t$

0 ϵ t

> 0 denotes “good news.” These shocks are used within the EGARCH(1,1) model to assess asymmetry in volatility responses.

```
import pandas as pd
from arch import arch_model

# --- Aggregate monthly mean stock returns for each region ---
us_returns = df_us.groupby('exchange_date')['stock_return'].mean().dropna()
eu_returns = df_eu.groupby('exchange_date')['stock_return'].mean().dropna()

print("US Returns Shape:", us_returns.shape)
print("EU Returns Shape:", eu_returns.shape)
```

```
US Returns Shape: (180,)
EU Returns Shape: (179,)
```

```
from arch import arch_model

# --- GARCH(1,1) for US ---
us_garch = arch_model(us_returns, vol='Garch', p=1, q=1, dist='normal')
us_garch_fit = us_garch.fit(disp='off')
print("US GARCH(1,1) Summary:\n", us_garch_fit.summary())

# --- GARCH(1,1) for Europe ---
eu_garch = arch_model(eu_returns, vol='Garch', p=1, q=1, dist='normal')
eu_garch_fit = eu_garch.fit(disp='off')
print("\nEurope GARCH(1,1) Summary:\n", eu_garch_fit.summary())
```

```
US GARCH(1,1) Summary:
                      Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:           stock_return   R-squared:                       0.000
Mean Model:             Constant Mean   Adj. R-squared:                  0.000
Vol Model:                      GARCH   Log-Likelihood:                345.074
Distribution:                  Normal   AIC:                          -682.149
Method:            Maximum Likelihood   BIC:                          -669.377
                                        No. Observations:                  180
Date:                Sun, Nov 09 2025   Df Residuals:                      179
Time:                        00:26:08   Df Model:                            1
                                 Mean Model
============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
----------------------------------------------------------------------------
mu             0.0120  2.392e-03      5.023  5.086e-07 [7.326e-03,1.670e-02]
                               Volatility Model
=============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
-----------------------------------------------------------------------------
omega      1.8972e-04  1.216e-04      1.561      0.119 [-4.856e-05,4.280e-04]
alpha[1]       0.1862  8.408e-02      2.215  2.677e-02    [2.143e-02,  0.351]
beta[1]        0.6838      0.129      5.312  1.087e-07      [  0.431,  0.936]
=============================================================================

Covariance estimator: robust

Europe GARCH(1,1) Summary:
                      Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:           stock_return   R-squared:                       0.000
Mean Model:             Constant Mean   Adj. R-squared:                  0.000
Vol Model:                      GARCH   Log-Likelihood:                307.265
Distribution:                  Normal   AIC:                          -606.530
Method:            Maximum Likelihood   BIC:                          -593.780
                                        No. Observations:                  179
Date:                Sun, Nov 09 2025   Df Residuals:                      178
Time:                        00:26:08   Df Model:                            1
                                 Mean Model
============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
----------------------------------------------------------------------------
mu         7.8404e-03  2.816e-03      2.784  5.372e-03 [2.320e-03,1.336e-02]
                               Volatility Model
============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
----------------------------------------------------------------------------
omega      1.2768e-03  2.890e-04      4.419  9.935e-06 [7.104e-04,1.843e-03]
alpha[1]       0.4263      0.220      1.938  5.267e-02  [-4.918e-03,  0.857]
beta[1]        0.0000      0.167      0.000      1.000     [ -0.326,  0.326]
============================================================================

Covariance estimator: robust
```

US volatility is more persistent, while EU volatility reacts more sharply to new shocks.

Baseline volatility is higher for EU firms, meaning the market is inherently more “volatile,” possibly due to currency exposure or smaller market depth.

The effect of past shocks (alpha) is stronger in EU than in US, indicating EU firms’ stock returns are more sensitive to sudden changes.

```
# --- EGARCH(1,1) for US ---
us_egarch = arch_model(us_returns, vol='EGARCH', p=1, q=1, dist='normal')
us_egarch_fit = us_egarch.fit(disp='off')
print("US EGARCH(1,1) Summary:\n", us_egarch_fit.summary())
```

```
# --- EGARCH(1,1) for EU ---
eu_egarch = arch_model(eu_returns, vol='EGARCH', p=1, q=1, dist='normal')
eu_egarch_fit = eu_egarch.fit(disp='off')
print("\nEurope EGARCH(1,1) Summary:\n", eu_egarch_fit.summary())
```

```
US EGARCH(1,1) Summary:
                      Constant Mean - EGARCH Model Results
==============================================================================
Dep. Variable:           stock_return   R-squared:                       0.000
Mean Model:             Constant Mean   Adj. R-squared:                  0.000
Vol Model:                     EGARCH   Log-Likelihood:                343.915
Distribution:                  Normal   AIC:                          -679.830
Method:            Maximum Likelihood   BIC:                          -667.058
                                        No. Observations:                  180
Date:                Sun, Nov 09 2025   Df Residuals:                      179
Time:                        00:27:51   Df Model:                            1
                                 Mean Model
============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
----------------------------------------------------------------------------
mu             0.0122  2.396e-03      5.104  3.319e-07 [7.535e-03,1.693e-02]
                              Volatility Model
===========================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
---------------------------------------------------------------------------
omega         -1.0312      0.793     -1.301      0.193    [ -2.585,  0.523]
alpha[1]       0.3730      0.195      1.911  5.601e-02 [-9.557e-03,  0.755]
beta[1]        0.8432      0.120      7.053  1.747e-12    [  0.609,  1.078]
===========================================================================

Covariance estimator: robust

Europe EGARCH(1,1) Summary:
                      Constant Mean - EGARCH Model Results
==============================================================================
Dep. Variable:           stock_return   R-squared:                       0.000
Mean Model:             Constant Mean   Adj. R-squared:                  0.000
Vol Model:                     EGARCH   Log-Likelihood:                307.066
Distribution:                  Normal   AIC:                          -606.132
Method:            Maximum Likelihood   BIC:                          -593.383
                                        No. Observations:                  179
Date:                Sun, Nov 09 2025   Df Residuals:                      178
Time:                        00:27:51   Df Model:                            1
                                 Mean Model
============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
----------------------------------------------------------------------------
mu         6.9183e-03  2.551e-03      2.712  6.689e-03 [1.918e-03,1.192e-02]
                              Volatility Model
===========================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
---------------------------------------------------------------------------
omega         -5.0827      0.911     -5.582  2.376e-08    [ -6.867, -3.298]
alpha[1]       0.6823      0.198      3.445  5.716e-04    [  0.294,  1.070]
beta[1]        0.1870      0.146      1.284      0.199 [-9.849e-02,  0.473]
===========================================================================

Covariance estimator: robust
```

he EGARCH(1,1) results reveal notable differences in volatility behavior between US and European firms. Volatility clustering is more persistent in the US, as indicated by a higher beta (0.843) compared to Europe (0.187), meaning that periods of high or low volatility tend to continue longer in the US. In contrast, European stock returns react more strongly to shocks, reflected by a higher alpha (0.682) than the US (0.373). The EGARCH model also detects stronger asymmetry for European firms, with a significant negative omega, suggesting that “bad news” impacts European volatility more than positive shocks. Additionally, US firms show higher average monthly returns (1.22%) compared to European firms (0.69%). Overall, US firms exhibit persistent but symmetric volatility, whereas European firms are more sensitive to shocks and demonstrate asymmetric responses to adverse market or currency events.

```
import pandas as pd

# Load macroeconomic data
macro = pd.read_excel("/content/Dissertation Macro.xlsx")  # replace with your file name

# Preview columns
print(macro.head())
print("Columns in macro data:", macro.columns.tolist())
```

```
        date  euro inflation  Euro ECB.EURO_ECB  Euro GDP.Euro_GDP  \
0 2024-11-30               2               3.40           0.863297
1 2024-10-31               2               3.40           0.863297
2 2024-09-30               2               3.65           0.863297
3 2024-08-31               2               4.25           0.863297
4 2024-07-31               2               4.25           0.863297

   USA CPI.USA_CPI  USA ECB.USA_ECB  USA GDP.monthly usa_gdp
0          315.493            4.625                      1.9
1          315.664            4.875                      1.9
2          315.301            4.875                      3.3
3          314.796            5.375                      3.3
4          314.540            0.000                      3.3
Columns in macro data: ['date', 'euro inflation', 'Euro ECB.EURO_ECB', 'Euro GDP.Euro_GDP', 'USA CPI.USA_CPI', 'USA ECB.USA_ECB'
```

```
import pandas as pd
from linearmodels.panel import PanelOLS
import numpy as np

# --- Step 1: Load macro data ---
macro = pd.read_excel("/content/Dissertation Macro.xlsx")
# Clean column names
macro.columns = ['date', 'euro_inflation', 'euro_ecb', 'euro_gdp', 'usa_cpi', 'usa_ecb', 'usa_gdp']

# --- Step 2: Ensure date is datetime ---
macro['date'] = pd.to_datetime(macro['date'])
df['exchange_date'] = pd.to_datetime(df['exchange_date'])

# --- Step 3: Split into US and Europe ---
us_df = df[df['currency'] == 'USD'].copy()
eu_df = df[df['currency'] == 'EUR'].copy()

# --- Step 4: Merge macro into stock data based on date ---
us_df = pd.merge(us_df, macro[['date', 'usa_cpi', 'usa_ecb', 'usa_gdp']],
                 left_on='exchange_date', right_on='date', how='left')
eu_df = pd.merge(eu_df, macro[['date', 'euro_inflation', 'euro_ecb', 'euro_gdp']],
                 left_on='exchange_date', right_on='date', how='left')

# Drop extra 'date' column from merge
us_df = us_df.drop(columns=['date'])
eu_df = eu_df.drop(columns=['date'])

# --- Step 5: Compute stock returns ---
us_df['stock_return'] = us_df.groupby('company_name')['close_prices_in_euro'].pct_change()
eu_df['stock_return'] = eu_df.groupby('company_name')['close_prices_in_euro'].pct_change()

# Drop first row per company where stock_return is NaN
us_df = us_df.dropna(subset=['stock_return'])
eu_df = eu_df.dropna(subset=['stock_return'])

# --- Step 6: Set multi-index ---
us_df = us_df.set_index(['company_name', 'exchange_date'])
eu_df = eu_df.set_index(['company_name', 'exchange_date'])

# --- Step 7: Define controls ---
us_controls = ['total_asset_euro', 'total_debt_euro', 'total_market_capitalization_euro',
               'usa_cpi', 'usa_ecb', 'usa_gdp']
eu_controls = ['total_asset_euro', 'total_debt_euro', 'total_market_capitalization_euro',
               'euro_inflation', 'euro_ecb', 'euro_gdp']

# --- Step 8: Run PanelOLS with entity effects ---
us_model = PanelOLS(us_df['stock_return'], us_df[us_controls],
                    entity_effects=True, drop_absorbed=True)
us_res = us_model.fit(cov_type='clustered', cluster_entity=True)
print("US Extended Panel OLS:\n", us_res.summary)

eu_model = PanelOLS(eu_df['stock_return'], eu_df[eu_controls],
                    entity_effects=True, drop_absorbed=True)
eu_res = eu_model.fit(cov_type='clustered', cluster_entity=True)
print("\nEurope Extended Panel OLS:\n", eu_res.summary)
```

```
                                   Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
-----------------------------------------------------------------------------------------------------
total_asset_euro                   2.213e-07  1.699e-07     1.3022     0.1929  -1.118e-07   5.543e-07
total_debt_euro                   -4.121e-07  3.604e-07    -1.1435     0.2529  -1.119e-06   2.943e-07
total_market_capitalization_euro -1.813e-08  1.607e-08    -1.1285     0.2591  -4.963e-08   1.337e-08
usa_cpi                               0.0008     0.0006     1.4180     0.1562     -0.0003      0.0020
usa_ecb                              -0.0050     0.0035    -1.4257     0.1540     -0.0118      0.0019
usa_gdp                               0.0002     0.0005     0.4036     0.6865     -0.0007      0.0011
=====================================================================================================

F-test for Poolability: 4.4406
P-value: 0.0000
Distribution: F(56,10395)

Included effects: Entity

Europe Extended Panel OLS:
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:           stock_return   R-squared:                        0.0021
Estimator:                   PanelOLS   R-squared (Between):             -0.9509
No. Observations:                6294   R-squared (Within):               0.0021
Date:                Sun, Nov 09 2025   R-squared (Overall):             -0.0084
Time:                        01:53:01   Log-likelihood                    5436.2
Cov. Estimator:             Clustered
                                        F-statistic:                      2.6587
Entities:                          36   P-value                           0.0209
Avg Obs:                       174.83   Distribution:                   F(5,6253)
Min Obs:                       77.000
Max Obs:                       179.00   F-statistic (robust):             7.1117
                                        P-value                           0.0000
Time periods:                     179   Distribution:                   F(5,6253)
Avg Obs:                       35.162
Min Obs:                       32.000
Max Obs:                       36.000

                                Parameter Estimates
=====================================================================================================
                                   Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
-----------------------------------------------------------------------------------------------------
total_asset_euro                   6.917e-09  2.628e-09     2.6320     0.0085   1.765e-09   1.207e-08
total_debt_euro                    1.898e-07  5.817e-08     3.2625     0.0011   7.575e-08   3.038e-07
total_market_capitalization_euro -3.207e-08  2.549e-08    -1.2583     0.2083  -8.204e-08   1.789e-08
euro_inflation                       -0.0008     0.0008    -1.0567     0.2907     -0.0023      0.0007
euro_ecb                             -0.0017     0.0011    -1.5392     0.1238     -0.0039      0.0005
euro_gdp                             -0.0009     0.0005    -1.7199     0.0855     -0.0019      0.0001
=====================================================================================================

F-test for Poolability: 1.6772
P-value: 0.0075
Distribution: F(35,6253)

Included effects: Entity
```

. US Extended Panel OLS Findings

Dependent variable: Monthly stock return (stock_return)

R-squared: Very low (0.0008 within), indicating that the model explains almost none of the variation in stock returns within firms.

Significance: None of the macro variables (usa_cpi, usa_ecb, usa_gdp) are statistically significant at the 5% level.

usa_cpi t-stat: 1.418 (p = 0.156)

usa_ecb t-stat: -1.426 (p = 0.154)

usa_gdp t-stat: 0.404 (p = 0.687)

Firm-level controls: total_asset_euro, total_debt_euro, total_market_capitalization_euro are also not significant.

Interpretation:

Stock returns of US export-oriented firms appear largely unrelated to firm-level financials and macroeconomic indicators in this period.

Entity effects (fixed effects by firm) dominate, as indicated by the low explanatory power of the independent variables.

2. Europe Extended Panel OLS Findings

Dependent variable: Monthly stock return (stock_return)

R-squared: Slightly higher than US (0.0021 within), still very low.

Significant variables:

total_asset_euro (t = 2.632, p = 0.0085) → positive impact on stock returns total_debt_euro (t = 3.263, p = 0.0011) → positive impact on stock returns

Macro variables (euro_inflation, euro_ecb, euro_gdp) are not significant at 5% level, though euro_gdp is marginally significant (p = 0.0855).

Interpretation:

In Europe, firm-level financials (assets and debt) play a slightly more important role in explaining stock returns.

Macro indicators have little explanatory power, similar to the US.

US vs Europe Panel OLS Comparison

R-squared (within):

US: 0.0008 → extremely low, almost no variation explained

Europe: 0.0021 → slightly higher, still very low

Macro variables (CPI, ECB rate, GDP):

US: Not significant (p > 0.15 for all)

Europe: Not significant, though GDP is marginally significant (p ~ 0.086)

Firm-level controls (total assets, total debt, market cap):

US: None significant

Europe: Total assets and total debt are significant (positive impact on stock returns)

Entity effects (firm-specific):

Strong in both regions; dominates the regression results

Observations / Firms:

US: 10,458 observations / 57 firms

Europe: 6,294 observations / 36 firms

Key takeaway:

US stock returns: largely unexplained by firm-level or macro factors

European stock returns: slightly influenced by firm-level financials; macro factors mostly insignificant

Overall: Stock returns in both regions are mainly driven by unobserved firm-specific factors

```python
import pandas as pd
from arch import arch_model

# Ensure 'stock_return' is numeric
us_df['stock_return'] = pd.to_numeric(us_df['stock_return'], errors='coerce')
eu_df['stock_return'] = pd.to_numeric(eu_df['stock_return'], errors='coerce')

# Drop NaNs in stock_return
us_returns = us_df['stock_return'].dropna()
eu_returns = eu_df['stock_return'].dropna()

# --- US GARCH ---
us_garch = arch_model(us_returns, vol='Garch', p=1, q=1, mean='Zero', dist='normal')
us_garch_res = us_garch.fit(disp='off')
print("US GARCH summary:")
print(us_garch_res.summary())

# --- Europe GARCH ---
eu_garch = arch_model(eu_returns, vol='Garch', p=1, q=1, mean='Zero', dist='normal')
eu_garch_res = eu_garch.fit(disp='off')
print("\nEurope GARCH summary:")
print(eu_garch_res.summary())
```

```
US GARCH summary:
                       Zero Mean - GARCH Model Results
==============================================================================
Dep. Variable:           stock_return   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.000
Vol Model:                      GARCH   Log-Likelihood:               -42.7038
Distribution:                  Normal   AIC:                           91.4075
Method:            Maximum Likelihood   BIC:                           113.175
```

```
                                     No. Observations:                10464
Date:                Sun, Nov 09 2025   Df Residuals:                    10464
Time:                        02:06:02   Df Model:                            0
                              Volatility Model
============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
----------------------------------------------------------------------------
omega          0.0470  7.288e-02      0.645      0.519  [-9.584e-02,  0.190]
alpha[1]       1.0000      2.029      0.493      0.622      [ -2.977,  4.977]
beta[1]    1.3083e-05  2.515e-04  5.202e-02      0.959 [-4.798e-04,5.060e-04]
============================================================================

Covariance estimator: robust

Europe GARCH summary:
                       Zero Mean - GARCH Model Results
==============================================================================
Dep. Variable:           stock_return   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.000
Vol Model:                      GARCH   Log-Likelihood:                5979.75
Distribution:                  Normal   AIC:                          -11953.5
Method:            Maximum Likelihood   BIC:                          -11933.3
                                        No. Observations:                 6298
Date:                Sun, Nov 09 2025   Df Residuals:                     6298
Time:                        02:06:03   Df Model:                            0
                              Volatility Model
============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
----------------------------------------------------------------------------
omega      1.9038e-03  1.296e-03      1.469      0.142 [-6.358e-04,4.443e-03]
alpha[1]       0.1826  7.743e-02      2.359  1.833e-02    [3.088e-02,  0.334]
beta[1]        0.6509      0.180      3.616  2.996e-04      [  0.298,  1.004]
============================================================================

Covariance estimator: robust
```

US GARCH (combined US stocks)

Omega (constant volatility term): 0.0470 → relatively high, but not statistically significant (p = 0.519).

Alpha[1] (reaction to shocks): 1.0 → extremely high but with huge standard error, not significant (p = 0.622).

Beta[1] (persistence of volatility): 0.000013 → almost zero, not significant (p = 0.959).

Interpretation: US stock returns show very low and mostly unexplained volatility, with no significant GARCH effect. The volatility is almost flat and does not respond to past shocks.

2. Europe GARCH (combined European stocks)

Omega: 0.0019 → small, not significant (p = 0.142).

Alpha[1]: 0.1826 → significant (p = 0.018) → past shocks have a noticeable effect on volatility.

Beta[1]: 0.6509 → significant (p = 0.0003) → volatility is persistent, past volatility influences current volatility.

Interpretation: European stock returns show significant GARCH effects, meaning volatility reacts to previous shocks and persists over time

Key Comparison

US volatility: flat and largely unexplained

Europe volatility: shows clustering, reacts to shocks, and is persistent

Implication: European stock market shows typical GARCH behavior, while US market appears more stable or random at the aggregate level

```
# --- Step 1: Import packages ---
import pandas as pd
from arch import arch_model

# --- Step 2: Prepare the data ---
# Ensure stock_return exists and drop NaNs
us_returns = us_df['stock_return'].dropna()
eu_returns = eu_df['stock_return'].dropna()

# --- Step 3: Fit EGARCH(1,1) for US ---
us_egarch = arch_model(us_returns, vol='EGARCH', p=1, q=1, mean='Zero', dist='normal')
us_egarch_res = us_egarch.fit(disp='off')
print("US EGARCH summary:\n", us_egarch_res.summary())

# --- Step 4: Fit EGARCH(1,1) for Europe ---
eu_egarch = arch_model(eu_returns, vol='EGARCH', p=1, q=1, mean='Zero', dist='normal')
```

```
eu_egarch_res = eu_egarch.fit(disp='off')
print("\nEurope EGARCH summary:\n", eu_egarch_res.summary())
```

```
US EGARCH summary:
                        Zero Mean - EGARCH Model Results
==============================================================================
Dep. Variable:           stock_return   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.000
Vol Model:                     EGARCH   Log-Likelihood:                2161.25
Distribution:                  Normal   AIC:                          -4316.50
Method:            Maximum Likelihood   BIC:                          -4294.73
                                        No. Observations:                10464
Date:                Sun, Nov 09 2025   Df Residuals:                    10464
Time:                        02:07:59   Df Model:                            0
                              Volatility Model
========================================================================
                 coef    std err          t      P>|t|  95.0% Conf. Int.
------------------------------------------------------------------------
omega          0.7089      0.976      0.727      0.467 [ -1.203,  2.621]
alpha[1]       3.0719      2.706      1.135      0.256 [ -2.232,  8.376]
beta[1]        0.7343      0.549      1.337      0.181 [ -0.342,  1.811]
========================================================================

Covariance estimator: robust

Europe EGARCH summary:
                        Zero Mean - EGARCH Model Results
==============================================================================
Dep. Variable:           stock_return   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.000
Vol Model:                     EGARCH   Log-Likelihood:                5976.80
Distribution:                  Normal   AIC:                          -11947.6
Method:            Maximum Likelihood   BIC:                          -11927.4
                                        No. Observations:                 6298
Date:                Sun, Nov 09 2025   Df Residuals:                     6298
Time:                        02:07:59   Df Model:                            0
                               Volatility Model
===========================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
---------------------------------------------------------------------------
omega         -0.5740      0.243     -2.362  1.818e-02 [ -1.050,-9.769e-02]
alpha[1]       0.3149  7.834e-02      4.019  5.845e-05    [  0.161,  0.468]
beta[1]        0.8694  5.228e-02     16.631  4.175e-62    [  0.767,  0.972]
===========================================================================

Covariance estimator: robust
```

US EGARCH (Whole Market)

Omega (constant term): 0.7089, not significant (p = 0.467) → baseline volatility not statistically different from zero.

Alpha[1] (ARCH term): 3.0719, not significant (p = 0.256) → immediate shocks to volatility are not strongly affecting overall market volatility.

Beta[1] (GARCH term): 0.7343, not significant (p = 0.181) → past volatility has moderate persistence but not statistically confirmed.

Interpretation: Volatility of US stock returns is weakly persistent, with no strong evidence that past shocks or volatility levels significantly predict current volatility. Overall, US market volatility is less structured, possibly influenced by idiosyncratic or firm-level factors.

Europe EGARCH (Whole Market)

Omega (constant term): -0.5740, significant (p = 0.018) → baseline volatility is lower and statistically meaningful.

Alpha[1] (ARCH term): 0.3149, significant (p < 0.001) → immediate shocks to volatility have a strong effect on current volatility.

Beta[1] (GARCH term): 0.8694, highly significant (p ≈ 0) → past volatility is highly persistent and strongly affects current volatility.

Interpretation: European stock market exhibits strong volatility clustering, where both shocks and past volatility significantly drive current volatility. Volatility is more predictable and persistent than in the US.

Comparison

Persistence:

Europe > US (European market volatility is highly persistent, US is weak).

Shock Sensitivity:

Europe > US (shocks immediately influence volatility in Europe, not in US).

Baseline Volatility:

Europe has a statistically significant baseline volatility, while US does not.

Market Structure Insight:

US market volatility seems more random and less dependent on past shocks.

European market shows structured volatility patterns (typical of EGARCH dynamics), possibly due to macroeconomic or regional factors.

ou can create dummy variables for the periods affected by the events:

COVID-19: Example: March 2020 – Dec 2021 → covid_dummy = 1 for dates in this range, else 0.

Russia–Ukraine War: Example: Feb 2022 onward → ukrwar_dummy = 1 for dates in this range, else 0.

```
# Create event dummies
us_df['covid_dummy'] = ((us_df['year_month'] >= '2020-03') & (us_df['year_month'] <= '2021-12')).astype(int)
us_df['ukrwar_dummy'] = (us_df['year_month'] >= '2022-02').astype(int)

eu_df['covid_dummy'] = ((eu_df['year_month'] >= '2020-03') & (eu_df['year_month'] <= '2021-12')).astype(int)
eu_df['ukrwar_dummy'] = (eu_df['year_month'] >= '2022-02').astype(int)

# Print first 10 rows to check
print("US Data - Event Dummies:")
print(us_df[['year_month', 'covid_dummy', 'ukrwar_dummy']])

print("\nEurope Data - Event Dummies:")
print(eu_df[['year_month', 'covid_dummy', 'ukrwar_dummy']])
```

```
US Data - Event Dummies:
                                                year_month  covid_dummy  \
company_name                       exchange_date
3M CO                              2010-01-31    2010-01-01            0
ADOBE                              2010-01-31    2010-01-01            0
ADVANCED MICRO DEVICES INC         2010-01-31    2010-01-01            0
ABBOTT LABORATORIES                2010-01-31    2010-01-01            0
Apple                              2010-01-31    2010-01-01            0
...                                                    ...          ...
ADVANCED MICRO DEVICES INC         2024-12-31    2024-12-01            0
JOHNSON AND JOHNSON                2024-12-31    2024-12-01            0
JOHNSON CONTROLS INTERNATIONAL PLC 2024-12-31    2024-12-01            0
MICRON TECHNOLOGY INC              2024-12-31    2024-12-01            0
Microsoft                          2024-12-31    2024-12-01            0

                                                 ukrwar_dummy
company_name                       exchange_date
3M CO                              2010-01-31               0
ADOBE                              2010-01-31               0
ADVANCED MICRO DEVICES INC         2010-01-31               0
ABBOTT LABORATORIES                2010-01-31               0
Apple                              2010-01-31               0
...                                                       ...
ADVANCED MICRO DEVICES INC         2024-12-31               1
JOHNSON AND JOHNSON                2024-12-31               1
JOHNSON CONTROLS INTERNATIONAL PLC 2024-12-31               1
MICRON TECHNOLOGY INC              2024-12-31               1
Microsoft                          2024-12-31               1

[10464 rows x 3 columns]

Europe Data - Event Dummies:
                             year_month  covid_dummy  ukrwar_dummy
company_name     exchange_date
ADIDAS           2010-01-31    2010-01-01            0             0
AIRBUS SE        2010-01-31    2010-01-01            0             0
AKZO NOBEL NV    2010-01-31    2010-01-01            0             0
ARYZTA AG        2010-01-31    2010-01-01            0             0
ASML             2010-01-31    2010-01-01            0             0
...                                 ...          ...           ...
WOLTERS KLUWER NV 2024-11-30    2024-11-01            0             1
ADIDAS           2024-12-31    2024-12-01            0             1
ADYEN NV         2024-12-31    2024-12-01            0             1
AIRBUS SE        2024-12-31    2024-12-01            0             1
AKZO NOBEL NV    2024-12-31    2024-12-01            0             1

[6298 rows x 3 columns]
```

```
from linearmodels.panel import PanelOLS
```

```
# US Model with event dummies
us_controls_with_events = us_controls + ['covid_dummy', 'ukrwar_dummy']

us_model_events = PanelOLS(
    us_df['stock_return'],
    us_df[us_controls_with_events],
    entity_effects=True,
    drop_absorbed=True
)
us_res_events = us_model_events.fit(cov_type='clustered', cluster_entity=True)
print("\nUS Panel OLS with Event Dummies:\n", us_res_events.summary)

# Europe Model with event dummies
eu_controls_with_events = eu_controls + ['covid_dummy', 'ukrwar_dummy']

eu_model_events = PanelOLS(
    eu_df['stock_return'],
    eu_df[eu_controls_with_events],
    entity_effects=True,
    drop_absorbed=True
)
eu_res_events = eu_model_events.fit(cov_type='clustered', cluster_entity=True)
print("\nEurope Panel OLS with Event Dummies:\n", eu_res_events.summary)
```

```
====================================================================================================
                                  Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
----------------------------------------------------------------------------------------------------
total_asset_euro                  2.815e-07  2.285e-07     1.2321     0.2180  -1.663e-07   7.293e-07
total_debt_euro                  -5.839e-07  4.722e-07    -1.2367     0.2162  -1.509e-06   3.416e-07
total_market_capitalization_euro -2.189e-08  2.113e-08    -1.0360     0.3002  -6.331e-08   1.953e-08
usa_cpi                              0.0015     0.0006     2.3136     0.0207      0.0002      0.0027
usa_ecb                              0.0007     0.0050     0.1355     0.8922     -0.0091      0.0105
usa_gdp                          -7.189e-05     0.0002    -0.3173     0.7510     -0.0005      0.0004
covid_dummy                          0.0092     0.0391     0.2354     0.8139     -0.0674      0.0858
ukrwar_dummy                        -0.0699     0.0347    -2.0174     0.0437     -0.1378     -0.0020
====================================================================================================

F-test for Poolability: 4.4489
P-value: 0.0000
Distribution: F(56,10393)

Included effects: Entity

Europe Panel OLS with Event Dummies:
                           PanelOLS Estimation Summary
================================================================================
Dep. Variable:           stock_return   R-squared:                        0.0078
Estimator:                   PanelOLS   R-squared (Between):             -1.0065
No. Observations:                6294   R-squared (Within):               0.0078
Date:                Sun, Nov 09 2025   R-squared (Overall):             -0.0034
Time:                        02:18:02   Log-likelihood                    5454.0
Cov. Estimator:             Clustered
                                        F-statistic:                      6.9851
Entities:                          36   P-value                           0.0000
Avg Obs:                       174.83   Distribution:                   F(7,6251)
Min Obs:                       77.000
Max Obs:                       179.00   F-statistic (robust):             18.183
                                        P-value                           0.0000
Time periods:                     179   Distribution:                   F(7,6251)
```

Included effects: Entity

```
    Included effects: Entity
```

US Panel OLS Findings

Overall model fit:

Very low within R-squared (0.0014), indicating that firm-level and macro variables explain very little of US stock returns.

Between R-squared is negative, suggesting cross-sectional variation is poorly captured.

Macro variables:

usa_cpi is positive and significant (p = 0.0207), indicating higher inflation slightly increased stock returns.

usa_ecb and usa_gdp are insignificant, so monetary policy and GDP growth did not significantly affect returns.

Firm-level variables:

total_asset_euro, total_debt_euro, total_market_capitalization_euro are all insignificant, consistent with previous results.

Event dummies:

covid_dummy → positive but insignificant (p = 0.8139), suggesting COVID had no measurable impact on US stock returns at the panel level.

ukrwar_dummy → negative and significant (p = 0.0437), meaning the Russia-Ukraine war caused a small decline in US stock returns.

Interpretation: US stock returns remain largely unexplained by firm-level or macro variables; only geopolitical shock (Ukraine war) shows a significant negative effect.

Europe Panel OLS Findings

Overall model fit:

Within R-squared (0.0078) is slightly higher than US, but still very low — most variation remains unexplained.

Macro variables:

euro_ecb → negative and highly significant (p < 0.001), indicating ECB policies reduced stock returns.

euro_gdp → negative and significant (p = 0.005), meaning higher GDP growth slightly reduced stock returns, possibly due to market expectations or risk adjustments.

euro_inflation → insignificant.

Firm-level variables:

total_asset_euro and total_debt_euro → significant, indicating larger firms and more indebted firms experienced higher returns.

total_market_capitalization_euro → insignificant.

Event dummies:

covid_dummy → negative and significant (p = 0.0024), indicating COVID reduced European stock returns.

ukrwar_dummy → positive and significant (p < 0.001), suggesting European stocks were less affected or even slightly benefited from the geopolitical shock (possible due to energy/export hedges or market expectations).

Interpretation: European stock returns are more sensitive to macro factors and events than US returns, with COVID negatively impacting returns and the Ukraine war having a mixed/positive effect.

US vs Europe Comparison

Event sensitivity:

US: Only the Ukraine war had a significant effect.

Europe: Both COVID and Ukraine war significantly affected returns, but in opposite directions.

Macro sensitivity:

US stock returns are largely unaffected by GDP and ECB policy.

Europe shows strong sensitivity to ECB policy and GDP growth.

Firm-level sensitivity:

Europe: Total assets and debt matter.

US: Firm-level variables are mostly insignificant.

```
from arch import arch_model

# Exogenous variables for mean (COVID and Ukraine war)
```

```
# Exogenous variables for mean (COVID and Ukraine war)
us_exog = us_df[['covid_dummy', 'ukrwar_dummy']]

# Fit GARCH(1,1) with ARX mean (event dummies)
us_garch_model = arch_model(
    us_df['stock_return'],
    mean='ARX',
    lags=0,        # no autoregressive terms
    vol='GARCH',
    p=1, q=1,
    dist='normal',
    x=us_exog     # exogenous regressors in mean
)

us_garch_res = us_garch_model.fit(update_freq=50, disp='off')

print("US GARCH with Event Dummies Summary:")
print(us_garch_res.summary())
```

```
/tmp/ipython-input-2513719694.py:17: ConvergenceWarning: The optimizer returned code 4. The message is:
Inequality constraints incompatible
See scipy.optimize.fmin_slsqp for code meaning.

  us_garch_res = us_garch_model.fit(update_freq=50, disp='off')
US GARCH with Event Dummies Summary:
                          AR-X - GARCH Model Results
==============================================================================
Dep. Variable:           stock_return   R-squared:                     -25.544
Mean Model:                      AR-X   Adj. R-squared:                -25.549
Vol Model:                      GARCH   Log-Likelihood:               -63466.5
Distribution:                  Normal   AIC:                           126945.
Method:            Maximum Likelihood   BIC:                           126988.
                                        No. Observations:                10464
Date:                Sun, Nov 09 2025   Df Residuals:                    10461
Time:                        02:22:32   Df Model:                            3
                                Mean Model
==========================================================================
                  coef    std err          t      P>|t|  95.0% Conf. Int.
--------------------------------------------------------------------------
Const           1.0637      0.167      6.352  2.120e-10 [  0.736,  1.392]
covid_dummy     4.5003      0.167     26.873 4.491e-159 [  4.172,  4.829]
ukrwar_dummy   -5.6756      0.175    -32.482 1.924e-231 [ -6.018, -5.333]
                               Volatility Model
============================================================================
                coef    std err          t      P>|t|       95.0% Conf. Int.
----------------------------------------------------------------------------
omega      7.5469e-03  4.095e-03      1.843  6.536e-02 [-4.798e-04,1.557e-02]
alpha[1]       1.0000  4.150e-03    240.941      0.000      [  0.992,  1.008]
beta[1]        0.0000  7.024e-07      0.000      1.000 [-1.377e-06,1.377e-06]
============================================================================

Covariance estimator: robust
WARNING: The optimizer did not indicate successful convergence. The message was Inequality constraints incompatible.
See convergence_flag.
```

US GARCH with Event Dummies – Findings

Mean Equation (returns):

Const = 1.0637 ⟶ baseline average return (without events).

covid_dummy = 4.5003 ⟶ significant positive effect of COVID period on returns. Extremely significant (p << 0.001).

ukrwar_dummy = -5.6756 ⟶ significant negative effect of Ukraine war on returns. Extremely significant (p << 0.001).

Interpretation:

During COVID (Mar 2020–Dec 2021), US stock returns increased on average, possibly due to stimulus, tech boom, and market recovery.

During the Ukraine war (from Feb 2022), US stock returns decreased on average, likely due to global uncertainty, inflation fears, and energy crisis.

Volatility Equation:

omega = 0.0075, alpha[1] ≈ 1, beta[1] ≈ 0 ⟶ the model shows extremely high sensitivity of volatility to recent shocks.

However, convergence warning and alpha ≈ 1 with beta ≈ 0 suggest the GARCH model is not well-identified, i.e., volatility is almost entirely captured by the last shock, and the optimizer had trouble.

Overall:

Event dummies are highly significant in the mean (returns), showing clear COVID and Ukraine war effects.

Volatility is poorly captured due to model convergence issues; may need EGARCH for better fit.

```
from arch import arch_model

# Exogenous variables for mean (event dummies)
us_exog = us_df[['covid_dummy', 'ukrwar_dummy']]

# EGARCH(1,1) with ARX mean including event dummies
us_egarch_model = arch_model(
    us_df['stock_return'],
    mean='ARX',      # include dummies in mean
    lags=0,
    vol='EGARCH',    # EGARCH volatility
    p=1, q=1,
    dist='normal',
    x=us_exog
)

us_egarch_res = us_egarch_model.fit(update_freq=50, disp='off')

print("US EGARCH with Event Dummies Summary:")
print(us_egarch_res.summary())
```

```
US EGARCH with Event Dummies Summary:
                         AR-X - EGARCH Model Results
==============================================================================
Dep. Variable:           stock_return   R-squared:                      -0.037
Mean Model:                      AR-X   Adj. R-squared:                 -0.038
Vol Model:                     EGARCH   Log-Likelihood:                5400.64
Distribution:                  Normal   AIC:                          -10789.3
Method:            Maximum Likelihood   BIC:                          -10745.8
                                        No. Observations:                10464
Date:                Sun, Nov 09 2025   Df Residuals:                    10461
Time:                        02:23:54   Df Model:                            3
                                   Mean Model
===============================================================================
                   coef    std err          t      P>|t|       95.0% Conf. Int.
-------------------------------------------------------------------------------
Const           -0.0199  1.677e-02     -1.186      0.236 [-5.274e-02,1.298e-02]
covid_dummy      0.3786      0.189      1.999  4.560e-02    [7.410e-03,  0.750]
ukrwar_dummy     0.0123  2.009e-02      0.614      0.539 [-2.705e-02,5.171e-02]
                             Volatility Model
========================================================================
                 coef    std err          t      P>|t|  95.0% Conf. Int.
------------------------------------------------------------------------
omega          0.1493      0.188      0.796      0.426 [ -0.218,  0.517]
alpha[1]       1.1263      0.292      3.856  1.154e-04 [  0.554,  1.699]
beta[1]        0.9514  2.611e-02     36.446 8.062e-291 [  0.900,  1.003]
========================================================================

Covariance estimator: robust
```

1. Mean Equation (Returns) Variable Coefficient P-value Interpretation Const -0.0199 0.236 Not significant; baseline return is near zero. covid_dummy 0.3786 0.0456 Significant positive effect of COVID period on US stock returns. Suggests average returns were higher during COVID. ukrwar_dummy 0.0123 0.539 Not significant; Ukraine war had no clear effect on average US returns in this EGARCH specification.

Takeaway:

COVID shows a moderate positive effect on US stock returns ($p < 0.05$).

The Ukraine war effect is weak or not detectable at the aggregate US market level in this model.

2. Volatility Equation Parameter Coefficient P-value Interpretation omega 0.1493 0.426 Not significant; baseline volatility level is not well-identified. alpha[1] 1.1263 0.000115 Significant; past shocks strongly increase volatility (leverage effect). beta[1] 0.9514 ~0 Significant; volatility is highly persistent. High beta + alpha indicates almost unit-root type persistence.

Takeaway:

Volatility is asymmetric and persistent, which EGARCH captures better than standard GARCH.

Large shocks (both positive and negative) amplify volatility significantly.

EGARCH can handle negative shocks differently, but here alpha > 1 indicates strong reaction to shocks.

3. Overall Insights

Mean returns: COVID increased returns; Ukraine war effect is insignificant.

Volatility: Highly persistent; EGARCH captures asymmetric shock effects well.

Convergence and model fit are much better than GARCH for volatility.

This output complements your OLS with Event Dummies, which showed COVID and Ukraine war effects in the mean; EGARCH mostly refines volatility behavior.

```
from arch import arch_model
import numpy as np
import pandas as pd

# --- Create a combined effect column for mean ---
eu_df['event_effect'] = eu_df['covid_dummy'] - eu_df['ukrwar_dummy']  # simple linear combination

# --- Fit GARCH(1,1) with Zero Mean (we will interpret effect separately) ---
eu_garch_model = arch_model(
    eu_df['stock_return'],
    mean='Zero',  # zero mean
    vol='GARCH',
    p=1, q=1,
    dist='normal'
)

# Fit the model
eu_garch_res = eu_garch_model.fit(update_freq=50, disp='off', cov_type='robust')

# Print summary
print("Europe GARCH Summary (Zero Mean, Event Dummies to be interpreted separately):")
print(eu_garch_res.summary())
```

```
Europe GARCH Summary (Zero Mean, Event Dummies to be interpreted separately):
                       Zero Mean - GARCH Model Results
==============================================================================
Dep. Variable:           stock_return   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.000
Vol Model:                      GARCH   Log-Likelihood:                5979.75
Distribution:                  Normal   AIC:                          -11953.5
Method:            Maximum Likelihood   BIC:                          -11933.3
                                        No. Observations:                 6298
Date:                Sun, Nov 09 2025   Df Residuals:                     6298
Time:                        02:26:31   Df Model:                            0
                              Volatility Model
=============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
-----------------------------------------------------------------------------
omega      1.9038e-03  1.296e-03      1.469      0.142 [-6.358e-04,4.443e-03]
alpha[1]       0.1826  7.743e-02      2.359  1.833e-02    [3.088e-02,  0.334]
beta[1]        0.6509      0.180      3.616  2.996e-04      [  0.298,  1.004]
=============================================================================

Covariance estimator: robust
```

Volatility dynamics:

omega = 0.0019 (not significant at 5%): The long-term constant variance is small and not statistically significant.

alpha[1] = 0.1826 (significant at 5%): Indicates that recent shocks (past stock return volatility) have a positive and significant impact on current volatility.

beta[1] = 0.6509 (significant at 1%): Shows that past volatility is highly persistent, meaning volatility shocks decay slowly over time.

Interpretation:

The GARCH model confirms volatility clustering in European stock returns.

Recent shocks (e.g., economic events like COVID or Ukraine war) have some impact, but event dummies were not included directly in the mean here, so we can interpret them via the panel OLS results.

The sum alpha + beta ≈ 0.8335 indicates moderate persistence of volatility, meaning shocks last multiple periods but gradually decay.

Comparison with US GARCH (Zero Mean):

US volatility shows near-zero beta (0.0000) and alpha ~1 but with convergence issues, suggesting instability or insufficient fit for the full US market with these event dummies.

Europe has a more stable volatility process with clear persistence (alpha + beta < 1).

Europe's volatility seems more predictable and statistically significant compared to US GARCH for the full market.

✅ Key takeaway for dissertation:

European stock returns exhibit significant volatility clustering, whereas US returns are less stable in GARCH estimation when considering the full market.

Event dummies (COVID, Ukraine war) should be interpreted via Panel OLS mean effects, while GARCH captures overall volatility dynamics.

```
from arch import arch_model

# Fit EGARCH(1,1) with zero mean (no exogenous regressors)
eu_egarch_model = arch_model(
    eu_df['stock_return'],
    mean='Zero',     # Zero mean model
    vol='EGARCH',    # EGARCH model
    p=1, q=1,       # EGARCH(1,1)
    dist='normal'
)

# Fit the model
eu_egarch_res = eu_egarch_model.fit(update_freq=50, disp='off')

# Print summary
print("Europe EGARCH Summary (Zero Mean):")
print(eu_egarch_res.summary())
```

```
Europe EGARCH Summary (Zero Mean):
                       Zero Mean - EGARCH Model Results
==============================================================================
Dep. Variable:          stock_return   R-squared:                       0.000
Mean Model:                Zero Mean   Adj. R-squared:                  0.000
Vol Model:                    EGARCH   Log-Likelihood:                5976.80
Distribution:                 Normal   AIC:                          -11947.6
Method:           Maximum Likelihood   BIC:                          -11927.4
                                       No. Observations:                 6298
Date:               Sun, Nov 09 2025   Df Residuals:                     6298
Time:                       02:29:20   Df Model:                            0
                              Volatility Model
===========================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
---------------------------------------------------------------------------
omega         -0.5740      0.243     -2.362  1.818e-02 [ -1.050,-9.769e-02]
alpha[1]       0.3149  7.834e-02      4.019  5.845e-05    [  0.161,  0.468]
beta[1]        0.8694  5.228e-02     16.631  4.175e-62    [  0.767,  0.972]
===========================================================================

Covariance estimator: robust
```

Findings

Mean Model:

Zero mean model is used, so we are modeling volatility only, not the expected return.

R-squared = 0, meaning the model does not explain the level of returns (expected in financial return volatility models).

Volatility Model (EGARCH(1,1)):

omega = -0.5740, significant (p = 0.018):

This is the constant term in the log-variance equation.

Negative omega is allowed in EGARCH and can indicate low baseline volatility.

alpha[1] = 0.3149, highly significant (p < 0.001):

Measures the reaction of volatility to shocks (ARCH effect).

Positive and significant: bad news/good news increases volatility, consistent with typical financial return behavior.

beta[1] = 0.8694, extremely significant (p ≈ 0):

Measures persistence in volatility (GARCH effect).

High beta (~0.87) means volatility shocks are very persistent — if volatility rises today, it likely stays high for a while.

Interpretation of EGARCH:

The model captures asymmetric volatility (EGARCH allows “leverage effect”), although in this zero-mean model, event dummies (COVID, Ukraine war) are not directly included.

This indicates that European stock returns volatility is highly persistent and responds significantly to shocks, but the model does not explain mean returns.

Covariance estimator: robust:

Standard errors are robust to heteroskedasticity → coefficients are reliable despite heteroskedastic residuals.

Comparison / Insights (Europe vs US EGARCH) Feature Europe EGARCH (Zero Mean) US EGARCH (Zero Mean) Alpha (shock effect) 0.3149 (significant) 1.1263 (significant) Beta (persistence) 0.8694 (high) 0.9514 (very high) Omega (constant) -0.5740 (negative) 0.1493 (positive) Interpretation Volatility reacts moderately to shocks, high persistence Volatility reacts strongly, extremely persistent Mean R-squared 0 -0.037

✅ Takeaway:

Both US and Europe show persistent volatility.

US EGARCH has stronger immediate response to shocks (higher alpha) and slightly higher persistence (beta).

Europe shows moderate reaction but still highly persistent volatility.

Effects of COVID or Ukraine war must be interpreted via panel OLS or residual analysis, as zero-mean EGARCH does not include them directly.

```
Start coding or generate with AI.
```