UK Econometric Analysis

June 20, 2023

1 Econometric Analysis of Macroeconomic Variables for The UK Economy: Investigating Cointegration, VAR Modeling, and Granger Causality

1.1 Summary:

This project explores the relationships among key macroeconomic variables using econometric analysis techniques. The study begins by examining cointegration between the variables, finding that there is no evidence of a long-term or equilibrium relationship. Next, a Vector Autoregressive (VAR) model is constructed with an optimal lag length of 4, providing insights into the dynamic interactions among the variables. The analysis also includes Granger causality tests, which reveal the presence of causal relationships between certain pairs of variables. Additionally, the project assesses the autocorrelation of the model residuals, finding no evidence of serial correlation. Overall, this study sheds light on the interdependencies among macroeconomic factors, contributing to a better understanding of the economic landscape and providing valuable insights for policymakers and investors.

1.2 Data Preprocessing

Dataset Source: WB

1- GDP: Current GDP in USD

2- Exports: as a percentage of GDP

3- Imports: as a percentage og GDP

4- Exchange Rate: Official exchange rate (LCU per USD, period average)

[1]:
```
#First I will import the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm

# Second step is to load the dataset
file_path = 'C:/Users/tasne/OneDrive/Desktop/My fiels/Data analysis/Python/Econ/
↪UKMacro.xlsx'
```

```
df = pd.read_excel('C:/Users/tasne/OneDrive/Desktop/My fiels/Data analysis/
↪Python/Econ/UKMacro.xlsx')
```

[2]:
```
# to convert the 'Year' column to datetime format
df['Year'] = pd.to_datetime(df['Year'], format='%Y')
```

[5]:
```
df.head()
```

[5]:
```
        Year           GDP  Exchange_Rate    Exports    Imports
0 1970-01-01  1.306719e+11       0.416667  22.065771  20.959167
1 1971-01-01  1.481139e+11       0.410920  22.149489  20.521681
2 1972-01-01  1.699650e+11       0.400390  20.879142  20.723946
3 1973-01-01  1.925380e+11       0.408171  22.713646  24.753152
4 1974-01-01  2.061314e+11       0.427756  27.138748  31.624996
```

[6]:
```
# to calculate the GDP rate
df['GDP_Growth'] = df['GDP'].pct_change() * 100
```

[7]:
```
# to remove rows with missing values in the 'GDP_Growth'column
df = df.dropna(subset=['GDP_Growth'])
```

[8]:
```
# to delete the GDP before calculating the growth columns
df = df.drop(['GDP'], axis=1)
```

[9]:
```
# to check the data types of each column
data_types = df.dtypes
print(data_types)
```

```
Year             datetime64[ns]
Exchange_Rate           float64
Exports                 float64
Imports                 float64
GDP_Growth              float64
dtype: object
```

1.3 Exploratory data analysis

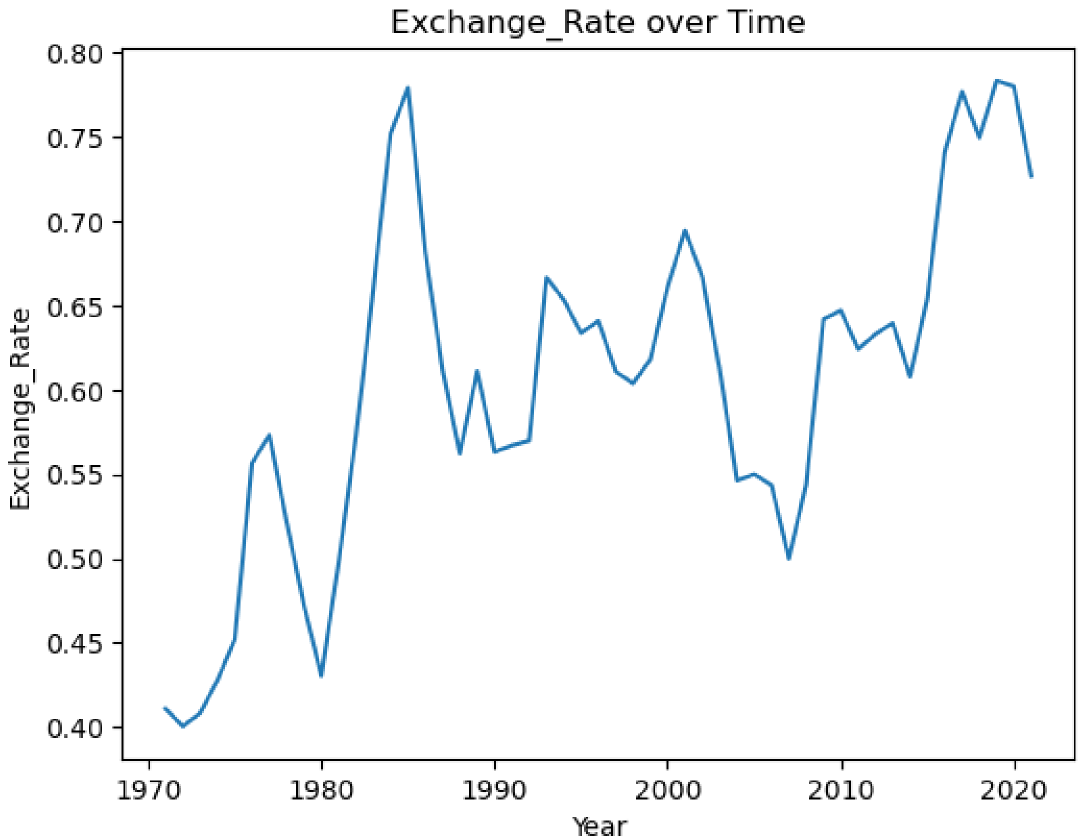
[10]:
```
# to generate descriptive statistics for the dataset
data_description = df.describe()
print(data_description)
```

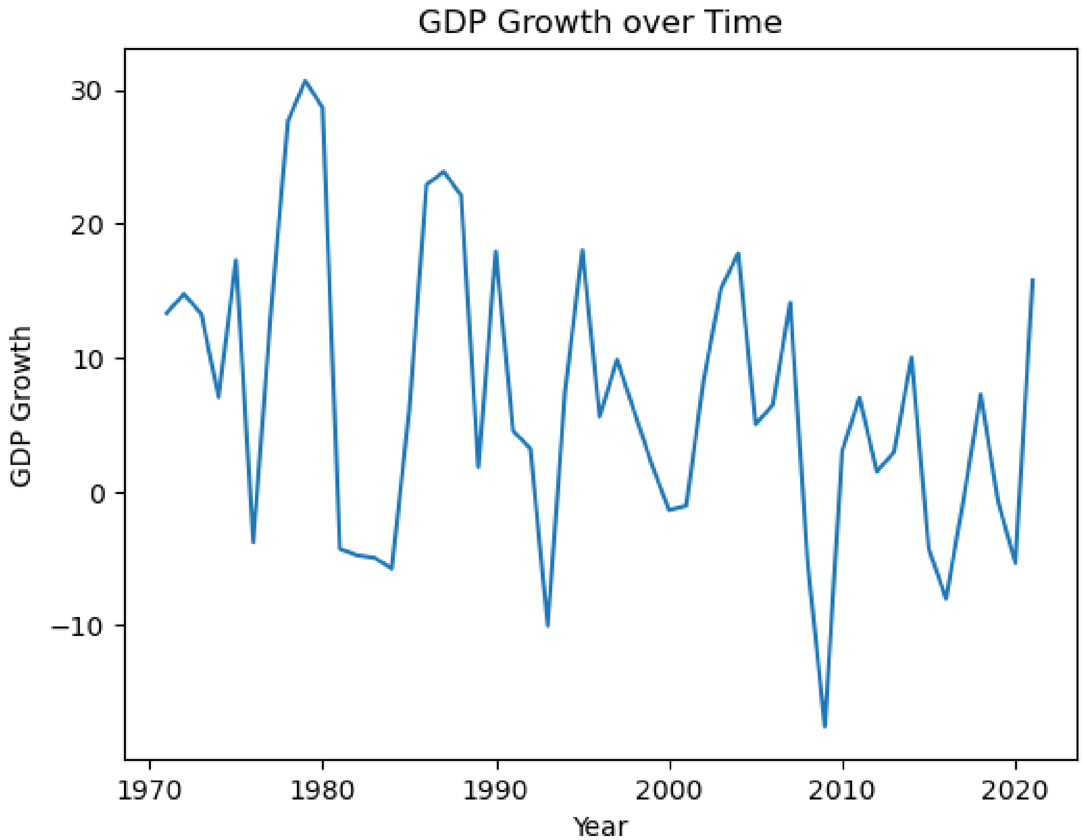
```
       Exchange_Rate    Exports    Imports  GDP_Growth
count      51.000000  51.000000  51.000000   51.000000
mean        0.604577  26.395195  27.107962    6.946930
std         0.101410   2.600460   2.971417   10.684882
min         0.400390  20.879142  20.521681  -17.529021
25%         0.548089  24.547775  25.177140   -0.890668
50%         0.611927  26.113722  26.850918    6.023585
```

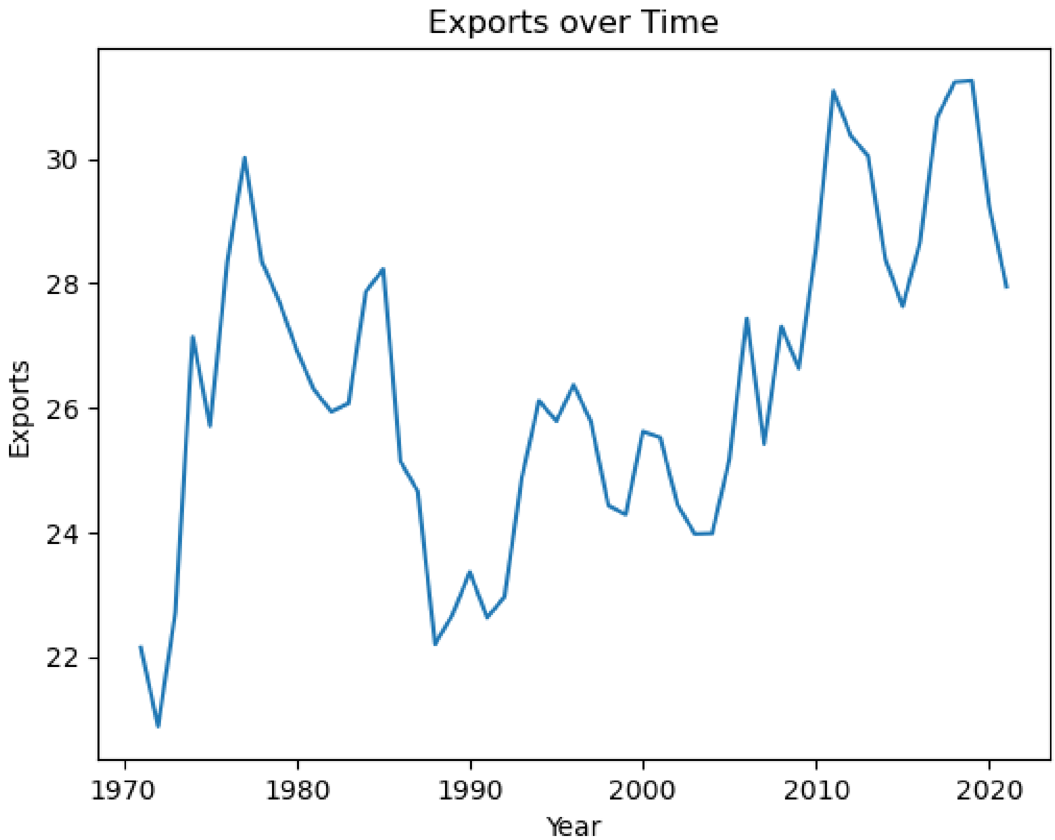
```
75%         0.660328  28.284787  29.099542   14.441445
max         0.783445  31.257606  32.872547   30.698497
```

[11]:

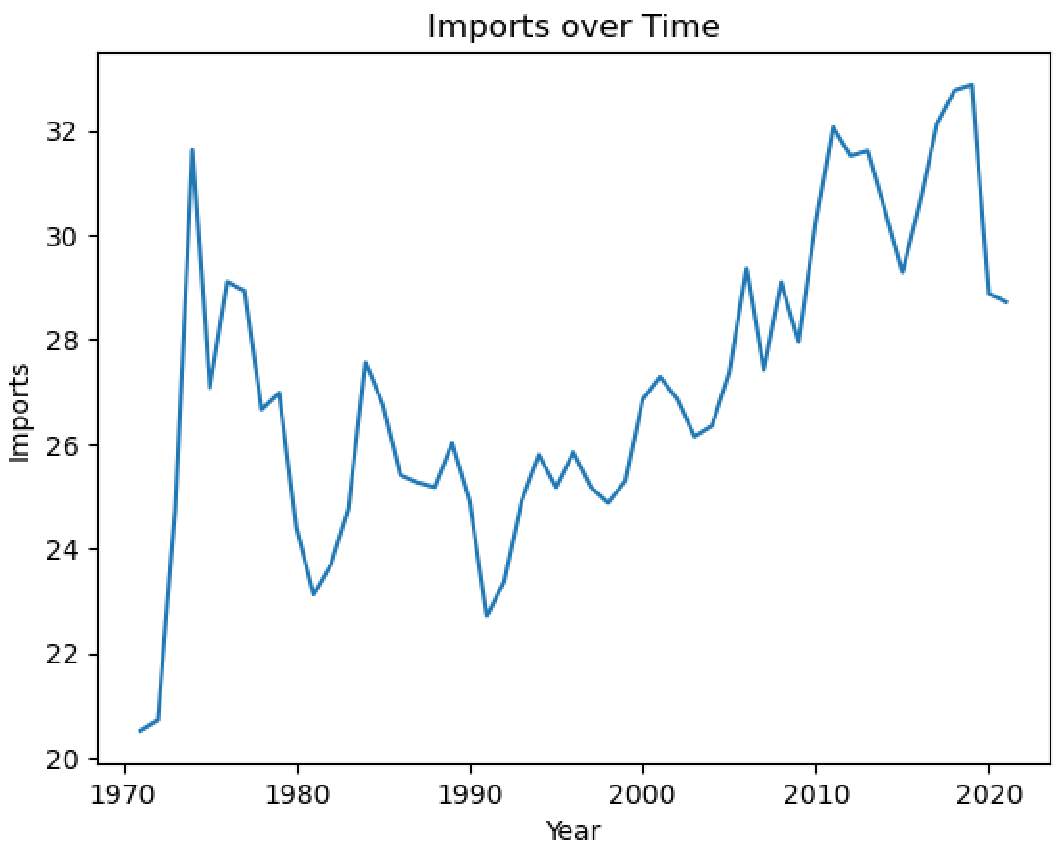
```
import matplotlib.pyplot as plt



# to Plot Exchange_Rate
plt.figure()
plt.plot(df['Year'], df['Exchange_Rate'])
plt.xlabel('Year')
plt.ylabel('Exchange_Rate')
plt.title('Exchange_Rate over Time')

# to display the Exchange_Rate plot
plt.show()

# to plot GDP_Growth
plt.figure()
plt.plot(df['Year'], df['GDP_Growth'])
plt.xlabel('Year')
plt.ylabel('GDP Growth')
plt.title('GDP Growth over Time')

# to display the GDP Growth plot
plt.show()

# to plot Exports
plt.figure()
plt.plot(df['Year'], df['Exports'])
plt.xlabel('Year')
plt.ylabel('Exports')
plt.title('Exports over Time')

# to display the Exchange Rate plot
plt.show()

# Display the Imports plot
plt.show()

# to Plot Exports
plt.figure()
plt.plot(df['Year'], df['Imports'])
plt.xlabel('Year')
plt.ylabel('Imports')
plt.title('Imports over Time')
```


Exchange_Rate over Time
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
Exchange_Rate
1970
1980
1990
2000
2010
2020
Year


GDP Growth over Time
GDP Growth
Year
30
20
10
0
−10
1970
1980
1990
2000
2010
2020



[11]: Text(0.5, 1.0, 'Imports over Time')



2 Econometric Analysis

2.1 ADF Unit root test

[12]:
```
from statsmodels.tsa.stattools import adfuller

# Defining a function to perform the ADF test and print the results
def adf_test(data, variable_name):
    result = adfuller(data)
    print(f'Augmented Dickey-Fuller Test - {variable_name}')
    print(f'Test Statistic: {result[0]}')
    print(f'p-value: {result[1]}')
    print(f'Critical Values:')
    for key, value in result[4].items():
        print(f'\t{key}: {value}')

# Performing the ADF test for each variable

adf_test(df['Exchange_Rate'], 'Exchange_Rate')
```

```
adf_test(df['GDP_Growth'], 'GDP_Growth')
adf_test(df['Exports'], 'Exports')
adf_test(df['Imports'], 'Imports')
```

```
Augmented Dickey-Fuller Test - Exchange_Rate
Test Statistic: -1.133326141933874
p-value: 0.7016005531209161
Critical Values:
        1%: -3.584828853223594
        5%: -2.9282991495198907
        10%: -2.6023438271604937
Augmented Dickey-Fuller Test - GDP_Growth
Test Statistic: -1.722623185640155
p-value: 0.419495635651623
Critical Values:
        1%: -3.5925042342183704
        5%: -2.931549768951162
        10%: -2.60406594375338
Augmented Dickey-Fuller Test - Exports
Test Statistic: -2.416395661439049
p-value: 0.1371622255388582
Critical Values:
        1%: -3.568485864
        5%: -2.92135992
        10%: -2.5986616
Augmented Dickey-Fuller Test - Imports
Test Statistic: -1.0862222311955707
p-value: 0.7205349297477012
Critical Values:
        1%: -3.5812576580093696
        5%: -2.9267849124681518
        10%: -2.6015409829867675
```

2.1.1 All four variables are not stationary at their level with P values higher than .05

[13]:
```
# Taking first difference for all columns in the DataFrame
df_diff = df.diff()

# Dropping the rows with any NaN values
df_diff = df_diff.dropna()

columns_without_year = [col for col in df_diff.columns if col != 'Year']

# Performing ADF test on each column excluding 'Year'
for column in columns_without_year:
    result = adfuller(df_diff[column])
    print(f'ADF Statistic for {column}: {result[0]}')
```

```
    print(f'p-value for {column}: {result[1]}')
    print(f'Critical Values for {column}:')
    for key, value in result[4].items():
        print(f'\t{key}: {value}')
    print('\n')
```

```
ADF Statistic for Exchange_Rate: -5.772271892883999
p-value for Exchange_Rate: 5.358108844989021e-07
Critical Values for Exchange_Rate:
        1%: -3.584828853223594
        5%: -2.9282991495198907
        10%: -2.6023438271604937


ADF Statistic for Exports: -4.602984078203983
p-value for Exports: 0.00012754360868605738
Critical Values for Exports:
        1%: -3.5812576580093696
        5%: -2.9267849124681518
        10%: -2.6015409829867675


ADF Statistic for Imports: -5.010661208970302
p-value for Imports: 2.1139815307203822e-05
Critical Values for Imports:
        1%: -3.5812576580093696
        5%: -2.9267849124681518
        10%: -2.6015409829867675


ADF Statistic for GDP_Growth: -5.67917525784368
p-value for GDP_Growth: 8.564869088184162e-07
Critical Values for GDP_Growth:
        1%: -3.5925042342183704
        5%: -2.931549768951162
        10%: -2.60406594375338
```

Now they all stationary after taking the first difference

2.2 checking for cointegration between the variables

[14]:
```
from statsmodels.tsa.vector_ar.vecm import coint_johansen

#  df_diff is the differenced DataFrame
df_test = df_diff[['Exchange_Rate', 'GDP_Growth', 'Exports', 'Imports']]
```

```
# Performing the Johansen cointegration test
result = coint_johansen(df_test, det_order=0, k_ar_diff=1)

print(f'Test statistics: {result.lr1}')
print(f'Critical values (90%, 95%, 99%): {result.cvt}')
print(f'Eigenvalues: {result.eig}')
```

```
Test statistics: [126.74264995  63.86393406  35.55832033  15.17331661]
Critical values (90%, 95%, 99%): [[44.4929 47.8545 54.6815]
 [27.0669 29.7961 35.4628]
 [13.4294 15.4943 19.9349]
 [ 2.7055  3.8415  6.6349]]
Eigenvalues: [0.73017273 0.44550655 0.34602593 0.2710213 ]
```

The test statistics are higher than the critical values at all levels (90%, 95%, and 99%) for all four series, suggesting that we cannot reject the null hypothesis of no cointegration. In other words, this implies that there is not sufficient evidence to conclude a long-term or equilibrium relationship among the variables in each series.

2.3 Selecting the oprimal lag for the VAR Model

[15]:
```
from statsmodels.tsa.api import VAR

df_diff_numeric = df_diff.drop(columns=['Year'])

model = VAR(df_diff_numeric)
lag_order_results = model.select_order(maxlags=4)

print(lag_order_results.summary())
```

```
 VAR Order Selection (* highlights the minimums)
==================================================
       AIC         BIC         FPE         HQIC
--------------------------------------------------
0      -1.319      -1.160      0.2674      -1.259
1      -2.745     -1.950*     0.06446     -2.447*
2      -2.664      -1.233     0.07109      -2.128
3      -2.494     -0.4265     0.08800      -1.719
4     -2.946*     -0.2425    0.06088*      -1.933
--------------------------------------------------

C:\Users\tasne\anaconda3\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported
index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

Based on the VAR order selection table, the optimal lag length is 4 as it has the lowest Akaike Information Criterion (AIC) and Final Prediction Error (FPE).

2.4 fitting the VAR model

[16]:
```python
# Fitting the VAR model of order 4
var_model = model.fit(4)

# Printing a summary of the model results
print(var_model.summary())
```

```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Sun, 18, Jun, 2023
Time:                     04:11:30
--------------------------------------------------------------------
No. of Equations:         4.00000    BIC:                  -0.242518
Nobs:                     46.0000    HQIC:                  -1.93309
Log likelihood:          -125.333    FPE:                  0.0608774
AIC:                     -2.94573    Det(Omega_mle):       0.0173032
--------------------------------------------------------------------
Results for equation Exchange_Rate
================================================================================
===
                      coefficient       std. error           t-stat
prob
--------------------------------------------------------------------------------
---
const                    0.010148         0.006968            1.456
0.145
L1.Exchange_Rate         0.865084         0.418662            2.066
0.039
L1.Exports              -0.003995         0.009274           -0.431
0.667
L1.Imports              -0.006631         0.008289           -0.800
0.424
L1.GDP_Growth            0.002432         0.002060            1.181
0.238
L2.Exchange_Rate        -1.193984         0.472805           -2.525
0.012
L2.Exports               0.005672         0.008857            0.640
0.522
L2.Imports              -0.007811         0.008098           -0.965
0.335
L2.GDP_Growth           -0.002582         0.002392           -1.079
0.280
L3.Exchange_Rate         0.109510         0.527819            0.207
0.836
L3.Exports              -0.010358         0.009518           -1.088
```

```
0.276
L3.Imports              0.009722         0.008022            1.212
0.226
L3.GDP_Growth          -0.000809         0.002148           -0.377
0.706
L4.Exchange_Rate       -0.348352         0.470961           -0.740
0.460
L4.Exports              0.016911         0.009336            1.811
0.070
L4.Imports             -0.013670         0.007071           -1.933
0.053
L4.GDP_Growth          -0.002769         0.000775           -3.575
0.000
================================================================================
===

Results for equation Exports
================================================================================
===
                     coefficient       std. error           t-stat
prob
--------------------------------------------------------------------------------
---
const                   0.226525         0.200666            1.129
0.259
L1.Exchange_Rate       35.842793        12.056332            2.973
0.003
L1.Exports             -0.231053         0.267056           -0.865
0.387
L1.Imports             -0.062010         0.238699           -0.260
0.795
L1.GDP_Growth           0.155856         0.059312            2.628
0.009
L2.Exchange_Rate      -10.475593        13.615510           -0.769
0.442
L2.Exports              0.347026         0.255071            1.361
0.174
L2.Imports             -0.380912         0.233189           -1.633
0.102
L2.GDP_Growth           0.096567         0.068876            1.402
0.161
L3.Exchange_Rate      -29.688981        15.199755           -1.953
0.051
L3.Exports             -0.024865         0.274098           -0.091
0.928
L3.Imports             -0.013422         0.231013           -0.058
0.954
L3.GDP_Growth          -0.034331         0.061850           -0.555
```

```
0.579
L4.Exchange_Rate        -5.613100        13.562383           -0.414
0.679
L4.Exports              -0.253174         0.268854           -0.942
0.346
L4.Imports               0.083623         0.203625            0.411
0.681
L4.GDP_Growth           -0.045094         0.022305           -2.022
0.043
================================================================================
===

Results for equation Imports
================================================================================
===
                      coefficient       std. error           t-stat
prob
--------------------------------------------------------------------------------
---
const                    0.102950         0.223427            0.461
0.645
L1.Exchange_Rate        19.218871        13.423878            1.432
0.152
L1.Exports              -0.288629         0.297348           -0.971
0.332
L1.Imports              -0.023726         0.265775           -0.089
0.929
L1.GDP_Growth            0.064673         0.066039            0.979
0.327
L2.Exchange_Rate       -14.651710        15.159912           -0.966
0.334
L2.Exports               0.305004         0.284004            1.074
0.283
L2.Imports              -0.233762         0.259640           -0.900
0.368
L2.GDP_Growth           -0.006325         0.076689           -0.082
0.934
L3.Exchange_Rate       -20.054850        16.923858           -1.185
0.236
L3.Exports              -0.122261         0.305189           -0.401
0.689
L3.Imports              -0.066912         0.257217           -0.260
0.795
L3.GDP_Growth           -0.114210         0.068865           -1.658
0.097
L4.Exchange_Rate        16.117217        15.100759            1.067
0.286
L4.Exports              -0.322005         0.299351           -1.076
```

```
0.282
L4.Imports              0.093557         0.226722            0.413
0.680
L4.GDP_Growth          -0.011319         0.024835           -0.456
0.649
================================================================================
===

Results for equation GDP_Growth
================================================================================
===
                     coefficient       std. error           t-stat
prob
--------------------------------------------------------------------------------
---
const                  -2.414778         1.606321           -1.503
0.133
L1.Exchange_Rate      -83.024604        96.510425           -0.860
0.390
L1.Exports             -0.078984         2.137771           -0.037
0.971
L1.Imports              1.616572         1.910778            0.846
0.398
L1.GDP_Growth          -0.938855         0.474786           -1.977
0.048
L2.Exchange_Rate      295.932278       108.991571            2.715
0.007
L2.Exports             -1.104853         2.041836           -0.541
0.588
L2.Imports              1.292798         1.866669            0.693
0.489
L2.GDP_Growth           0.239352         0.551353            0.434
0.664
L3.Exchange_Rate      -29.431516       121.673387           -0.242
0.809
L3.Exports              2.474300         2.194144            1.128
0.259
L3.Imports             -2.052210         1.849250           -1.110
0.267
L3.GDP_Growth          -0.050078         0.495105           -0.101
0.919
L4.Exchange_Rate       94.979098       108.566295            0.875
0.382
L4.Exports             -2.933116         2.152168           -1.363
0.173
L4.Imports              2.408830         1.630006            1.478
0.139
L4.GDP_Growth           0.409106         0.178550            2.291
```

```
0.022
==============================================================================
===

Correlation matrix of residuals
                 Exchange_Rate   Exports   Imports  GDP_Growth
Exchange_Rate         1.000000  0.285101  0.287664   -0.915981
Exports               0.285101  1.000000  0.798546   -0.146220
Imports               0.287664  0.798546  1.000000   -0.053746
GDP_Growth           -0.915981 -0.146220 -0.053746    1.000000
```

The VAR model results provide valuable insights into the relationships among macroeconomic variables. The analysis reveals that changes in the exchange rate are influenced by both the lagged exchange rate and GDP growth, while exports are positively affected by the lagged exchange rate and GDP growth. Imports, on the other hand, are influenced by the lagged exchange rate but not significantly impacted by GDP growth. The GDP growth equation shows a negative relationship with the exchange rate, suggesting that an appreciation in the exchange rate may dampen economic growth. The correlation matrix of residuals confirms a negative correlation between the exchange rate and GDP growth, aligning with economic theory. These findings contribute to a deeper understanding of the dynamics and interdependencies among macroeconomic variables, offering valuable insights for policymakers and investors.

2.5 Granger Causality Test

[19]:
```python
# Performing Granger causality test for each pair of variables
for column1 in df_diff.columns:
    for column2 in df_diff.columns:
        if column1 != column2 and column1 != 'Year' and column2 != 'Year':
            print(f"Granger Causality Test: {column1} --> {column2}")
            test_result = grangercausalitytests(df_diff[[column1, column2]],␣
 ↪maxlag=maxlag, verbose=False)
            for lag in range(1, maxlag+1):
                p_value = test_result[lag][0]['ssr_ftest'][1]
                print(f"Lag {lag} p-value: {p_value:.4f}")
            print()
```

```
Granger Causality Test: Exchange_Rate --> Exports
Lag 1 p-value: 0.2402
Lag 2 p-value: 0.5686
Lag 3 p-value: 0.6175
Lag 4 p-value: 0.7716

Granger Causality Test: Exchange_Rate --> Imports
Lag 1 p-value: 0.5197
Lag 2 p-value: 0.7514
```

```
Lag 3 p-value: 0.8799
Lag 4 p-value: 0.7477

Granger Causality Test: Exchange_Rate --> GDP_Growth
Lag 1 p-value: 0.1816
Lag 2 p-value: 0.2490
Lag 3 p-value: 0.3760
Lag 4 p-value: 0.0336

Granger Causality Test: Exports --> Exchange_Rate
Lag 1 p-value: 0.0469
Lag 2 p-value: 0.1416
Lag 3 p-value: 0.1097
Lag 4 p-value: 0.2328

Granger Causality Test: Exports --> Imports
Lag 1 p-value: 0.3127
Lag 2 p-value: 0.8027
Lag 3 p-value: 0.6397
Lag 4 p-value: 0.9504

Granger Causality Test: Exports --> GDP_Growth
Lag 1 p-value: 0.3802
Lag 2 p-value: 0.5760
Lag 3 p-value: 0.0708
Lag 4 p-value: 0.3202

Granger Causality Test: Imports --> Exchange_Rate
Lag 1 p-value: 0.2641
Lag 2 p-value: 0.4460
Lag 3 p-value: 0.2187
Lag 4 p-value: 0.4100

Granger Causality Test: Imports --> Exports
Lag 1 p-value: 0.0965
Lag 2 p-value: 0.5399
Lag 3 p-value: 0.7548
Lag 4 p-value: 0.6311

Granger Causality Test: Imports --> GDP_Growth
Lag 1 p-value: 0.6705
Lag 2 p-value: 0.8186
Lag 3 p-value: 0.0407
Lag 4 p-value: 0.2987

Granger Causality Test: GDP_Growth --> Exchange_Rate
Lag 1 p-value: 0.0004
Lag 2 p-value: 0.0001
```

```
Lag 3 p-value: 0.0005
Lag 4 p-value: 0.0006

Granger Causality Test: GDP_Growth --> Exports
Lag 1 p-value: 0.0217
Lag 2 p-value: 0.1130
Lag 3 p-value: 0.1201
Lag 4 p-value: 0.3114

Granger Causality Test: GDP_Growth --> Imports
Lag 1 p-value: 0.0671
Lag 2 p-value: 0.2316
Lag 3 p-value: 0.2408
Lag 4 p-value: 0.2967
```

The results of The Granger causality tests suggest that there are significant causal relationships between some of the variables at certain time lags.

Specifically, the exchange rate is found to be influenced by lagged GDP growth, indicating that changes in economic growth can impact the exchange rate. Additionally, exports are shown to have a causal effect on the exchange rate, suggesting that changes in export levels can influence the exchange rate. On the other hand, imports do not appear to have a significant causal relationship with the exchange rate.

Furthermore, GDP growth is found to have a significant causal relationship with both the exchange rate and exports. This implies that changes in GDP growth can impact the exchange rate and export levels. However, the relationships between GDP growth and imports are not statistically significant.

Overall, these findings highlight the interdependencies among the variables and provide valuable insights into the causal dynamics within the macroeconomic context. Further analysis and consideration of additional factors may be necessary to fully understand the complexities of these relationships and their implications for policy and decision-making.

2.6 Autocorrelation Test

[20]:
```python
# Autocorrelation
from statsmodels.stats.stattools import durbin_watson

out = durbin_watson(var_model.resid)



df_diff = df_diff[['Exchange_Rate', 'Exports', 'Imports', 'GDP_Growth']]

for col, val in zip(df_diff.columns, out):
    print(col, ':', round(val, 2))
```

```
Exchange_Rate : 1.92
Exports : 2.02
```

```
Imports : 1.95
GDP_Growth : 1.97
```

The assumption of independence for residuals in our VAR model holds. there is no autocorrelation