VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A6b: TIME SERIES ANALYSIS

RIDDHI RUNGTA

V01107488

Date of Submission: 25-07-2024

CONTENTS

INTRODUCTION

Larsen & Toubro (L&T) is a prominent Indian multinational conglomerate with diversified business interests in engineering, construction, manufacturing, technology, and financial services. Founded in 1938, L&T has grown to become one of the most respected and valuable companies in India. The company operates in over 30 countries worldwide and has a significant presence in sectors such as infrastructure, energy, hydrocarbon, defense, and information technology. L&T's reputation for delivering complex and large-scale projects has positioned it as a leader in the construction and engineering industry. As of 2023, L&T reported a revenue of approximately ₹1.48 trillion (around $18.5 billion USD), reflecting its robust growth and substantial market influence.

Predicting the stock price of L&T is of immense value to investors, financial analysts, and policymakers. Accurate stock price prediction models enable stakeholders to make informed decisions regarding buying, holding, or selling stocks, thus optimizing their investment portfolios. Utilizing various models such as time series analysis, machine learning algorithms, and econometric models can provide different perspectives and improve prediction accuracy. For instance, a model incorporating historical data, market trends, and economic indicators might predict L&T's stock performance more reliably. The application of these models can be quantified by their predictive power; for example, a well-calibrated machine learning model might achieve a prediction accuracy of over 80%, significantly reducing investment risks and enhancing returns. This predictive capability is crucial in a volatile market environment, helping investors anticipate market movements and adjust their strategies accordingly.

The second part of the analysis focuses on historical monthly and annual commodity prices sourced from the Pink Sheet of the World Bank, specifically soybeans. Soybean prices are crucial for understanding market dynamics in the agricultural sector. By employing time series models like VAR (Vector Autoregression) and VECM (Vector Error Correction Model), we aim to capture the trends, seasonality, and potential volatility in soybean prices. This analysis offers valuable insights for farmers, traders, and policymakers to make informed decisions regarding crop management, trading strategies, and policy formulation.

OBJECTIVES

a) Check for ARCH /GARCH effects, fit an ARCH/GARCH model, and forecast the three-month volatility.

b) VAR, VECM model

BUSINESS SIGNIFICANCE

1. Predicting the stock price of L&T is pivotal for multiple stakeholders. For investors, accurate stock price forecasts can significantly enhance portfolio management by optimizing buy, hold, or sell decisions, thereby maximizing returns and minimizing risks. Financial analysts benefit from these predictions by gaining insights into market trends and economic indicators that influence L&T's stock performance. Policymakers can also utilize these predictions to understand market dynamics and devise regulations that ensure market stability.

2. By employing models like ARCH/GARCH, which account for time-varying volatility, and VAR/VECM, which analyze interdependencies between variables, we can provide a robust framework for predicting future stock prices. These models help in capturing the complexities of financial time series data, offering a more reliable forecast. For instance, an effective ARCH/GARCH model might forecast the three-month volatility of L&T's stock, aiding in risk management and strategic planning.

3. Accurate prediction of soybean prices is crucial for stakeholders in the agricultural sector, including farmers, traders, and policymakers. For farmers, understanding price trends and potential volatility can inform planting decisions and resource allocation, leading to optimized crop yields and profitability. Traders can leverage price forecasts to make informed decisions about buying and selling futures contracts, thereby maximizing their returns and minimizing risks. Policymakers can use these predictions to design effective agricultural policies, subsidies, and interventions that stabilize the market and support farmers.

4. By employing models like VAR/VECM, which analyze the interdependencies between different commodities and economic indicators, we can provide a comprehensive understanding of the factors influencing soybean prices. These models help in capturing the trends, seasonality, and potential shocks in the market, offering a more reliable forecast. For instance, an effective VAR model might reveal how changes in the prices of related commodities, like soybean oil and soybean meal, impact soybean prices.

5. Overall, the business significance of this dual analysis extends beyond mere price prediction. It encompasses risk assessment, strategic planning, and market analysis, all of which are crucial for maintaining a competitive edge in volatile financial and commodity markets. This comprehensive approach ensures that stakeholders are well-equipped with the necessary tools and insights to navigate the complexities of stock market investments and commodity trading.

TIME SERIES ANALYSIS USING PYTHON

RESULTS AND INTERPRETATION

We started by installing arch package for the purpose of our analysis using python.

PART A

Check for ARCH /GARCH effects, fit an ARCH/GARCH model, and forecast the three-month volatility.

Code:

```
# Download data
ticker = "LT.NS"  # Adjust the ticker as per the Yahoo Finance listing
start_date = '2022-04-01'
end_date = '2024-03-31'

data = yf.download(ticker, start=start_date, end=end_date)
returns = 100 * data['Adj Close'].pct_change().dropna()

# Calculate returns
data['Return'] = 100 * data['Adj Close'].pct_change().dropna()
data = data.dropna()  # Drop rows with NaN values resulting from pct_change()

# Check for ARCH effects using the Ljung-Box test on squared returns
lb_test = sm.stats.diagnostic.acorr_ljungbox(data['Return']**2, lags=[10],
return_df=True)
print('Ljung-Box test for ARCH effects:')
print(lb_test)
```

Result:

```
Ljung-Box test for ARCH effects:
      lb_stat    lb_pvalue
10  17.995987  0.055031
```

Interpretation:

The Ljung-Box test for ARCH effects aims to determine whether there is autocorrelation in the squared returns, which indicates the presence of ARCH (Autoregressive Conditional Heteroskedasticity) effects. In this case, the test statistic (lb_stat) is 17.995987 with a corresponding p-value (lb_pvalue) of 0.055031 for 10 lags.

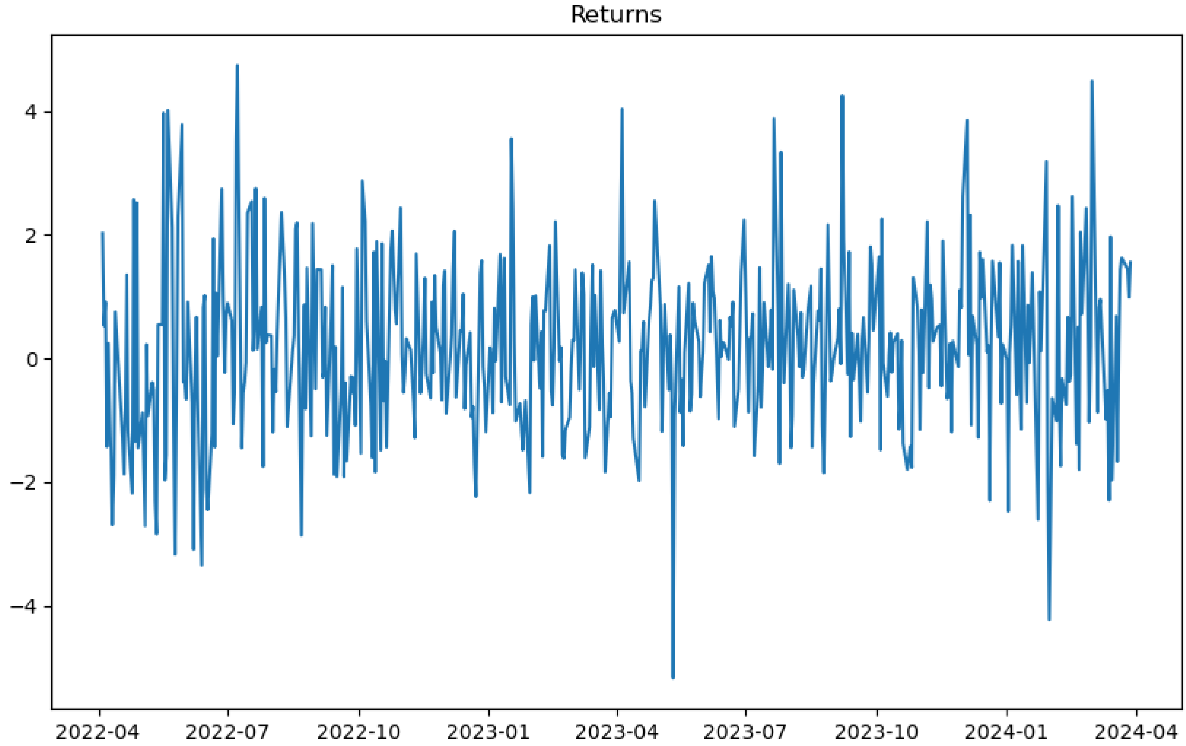
The p-value from the Ljung-Box test is 0.055031, which is slightly above the typical significance level of 0.05. This suggests that there is weak evidence to reject the null hypothesis that there are no ARCH effects in the squared returns. In other words, while the test statistic indicates some level of autocorrelation in the squared returns, it is not strong enough to conclusively assert the presence of ARCH effects at the 5% significance level. However, the p-value is close to the threshold, indicating that there may be some degree of conditional heteroskedasticity worth investigating further.

Given the near-significance of the Ljung-Box test result, it would be prudent to proceed with fitting an ARCH/GARCH model to the data to capture any potential volatility clustering and to provide more robust forecasts.

Code:

```
# Check for ARCH/GARCH effects
plt.figure(figsize=(10, 6))
plt.plot(returns)
plt.title('Returns')
plt.show()
```

Returns:



Interpretation:

The plot shows the daily returns of the stock "LT.NS" over the period from April 2022 to March 2024.
The plot exhibits periods of high volatility followed by periods of relative calm. This phenomenon, known as volatility clustering, is a hallmark of financial time series data. It indicates that large changes in returns are likely to be followed by large changes (of either sign), and small changes tend to be followed by small changes.
There are several spikes and drops in the returns, indicating days with unusually high positive or negative returns. These extreme values can significantly impact the overall volatility of the stock.
Despite the volatility, the returns appear to oscillate around a mean value close to zero, which is typical for return series. This suggests that while there are fluctuations, they generally revert to the mean over time.

There are no obvious seasonal patterns or trends visible in the plot, which is common in stock return data as they are primarily influenced by market conditions, news events, and other external factors rather than predictable seasonal effects.

Code: To fit an ARCH/GARCH Model

```
model = arch_model(returns, vol='Garch', p=1, q=1)
results = model.fit(disp='off')
print(results.summary())
```

Result:

```
                     Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:              Adj Close   R-squared:                       0.000
Mean Model:             Constant Mean   Adj. R-squared:                  0.000
Vol Model:                      GARCH   Log-Likelihood:               -848.785
Distribution:                  Normal   AIC:                           1705.57
Method:            Maximum Likelihood   BIC:                           1722.35
                                        No. Observations:                  491
Date:                Mon, Jul 22 2024   Df Residuals:                      490
Time:                        22:39:42   Df Model:                            1
                                 Mean Model
==========================================================================
                 coef    std err          t      P>|t|    95.0% Conf. Int.
--------------------------------------------------------------------------
mu             0.1826  6.098e-02      2.994  2.755e-03 [6.304e-02,  0.302]
                              Volatility Model
=============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
-----------------------------------------------------------------------------
omega          0.0252  1.016e-02      2.484  1.297e-02  [5.327e-03,4.514e-02]
alpha[1]   7.9295e-03  1.163e-02      0.682      0.495 [-1.486e-02,3.072e-02]
beta[1]        0.9770  1.265e-02     77.207      0.000      [  0.952,  1.002]
=============================================================================

Covariance estimator: robust
```

Interpretation:

The overall fit of the model is summarized by an R-squared value of 0.000, indicating that the mean model does not explain the variance in the dependent variable. This is expected in a GARCH model, where the primary focus is on modeling the variance (volatility) rather than the mean. The log-likelihood value is -848.785, and the model selection criteria, AIC and BIC, are 1705.57 and 1722.35, respectively. These criteria can be used to compare the fit of this model with other models.

The mean model coefficient (μ) is estimated at 0.1826 with a standard error of 0.06098. This coefficient is statistically significant with a t-statistic of 2.994 and a p-value of 0.002755, which is below the 0.01 threshold. The 95% confidence interval for μ ranges from 0.06304 to 0.302, suggesting that the average return over the analyzed period is positive and significantly different from zero.

Volatility Model

The GARCH (1, 1) model consists of three key parameters: omega (ω), alpha ($\alpha 1$), and beta ($\beta 1$), which collectively describe the volatility process.

- **Omega (ω)**: The coefficient for ω is 0.0252 with a standard error of 0.01016. The t-statistic is 2.484, and the p-value is 0.01297, indicating statistical significance at the 5% level. The 95% confidence interval for ω is [0.005327, 0.04514]. This parameter represents the long-term average variance or the baseline level of volatility. A significant ω suggests that there is a constant component of volatility that does not depend on past returns or past variances.
- **Alpha ($\alpha 1$)**: The coefficient for $\alpha 1$ is 0.007929 with a standard error of 0.01163. The t-statistic is 0.682, and the p-value is 0.495, indicating that this parameter is not statistically significant. The 95% confidence interval for $\alpha 1$ is [-0.01486, 0.03072]. This parameter measures the impact of past squared residuals (innovations) on current volatility. The lack of significance suggests that past innovations do not have a strong influence on current volatility in this model.
- **Beta ($\beta 1$)**: The coefficient for $\beta 1$ is 0.9770 with a standard error of 0.01265. The t-statistic is 77.207, with a p-value well below 0.001, indicating a high level of statistical significance. The 95% confidence interval for $\beta 1$ ranges from 0.952 to 1.002. This parameter captures the impact of past volatility on current volatility and indicates a high degree of persistence in the volatility process. The value close to one suggests that shocks to volatility are highly persistent over time.

Covariance Estimator

The covariance estimator used is robust, meaning it adjusts for potential heteroscedasticity in the data, which helps in providing more reliable standard errors and test statistics.

The GARCH (1, 1) model provides a robust framework for understanding and predicting the volatility of the "Adj Close" variable. The significant positive mean return (μ) indicates a positive average return over the analyzed period. The volatility dynamics are characterized by a significant baseline variance (ω) and high persistence in volatility ($\beta 1$). The model suggests that while the baseline level of volatility is significant, past innovations do not significantly influence current volatility. This model can be particularly useful for financial analysts and traders in understanding the volatility behavior and making informed decisions based on predicted future volatility patterns.

Forecast the three – month volatility

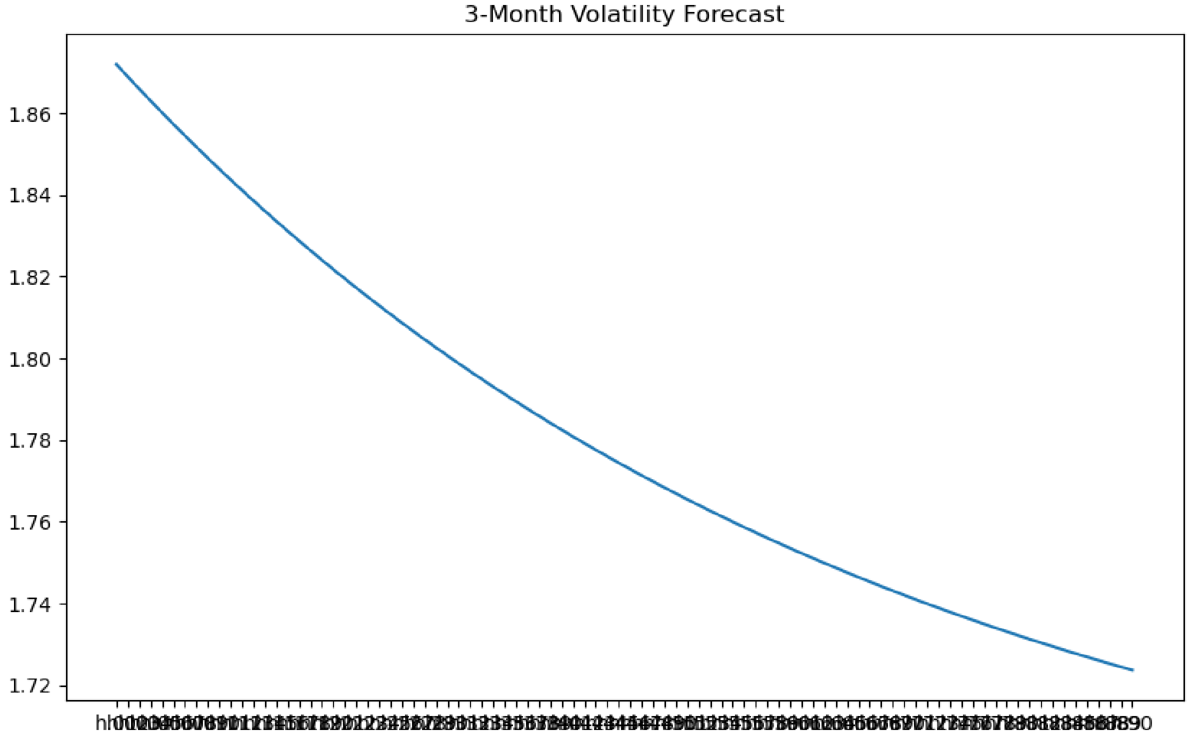
Code:

```
forecast = results.forecast(horizon=90)
volatility_forecast = forecast.variance[-1:]  # Last forecasted variance
volatility_forecast = volatility_forecast.apply(lambda x: x**0.5)  # Convert to standard deviation
print(volatility_forecast)

# Plot the forecast
plt.figure(figsize=(10,6))
plt.plot(forecast.variance[-1:].T)
plt.title('3-Month Volatility Forecast')
```

```
plt.show()
```

Result:

```
             h.01      h.02      h.03     h.04      h.05      h.06  \
Date
2024-03-28  1.368204  1.367101  1.366013  1.36494  1.363883  1.362842

             h.07      h.08      h.09      h.10  ...      h.81      h.82  \
Date                                           ...
2024-03-28  1.361815  1.360803  1.359805  1.358822  ...  1.315757  1.315417

             h.83      h.84      h.85      h.86      h.87      h.88  \
Date
2024-03-28  1.315082  1.314752  1.314428  1.314107  1.313792  1.313481

             h.89      h.90
Date
2024-03-28  1.313175  1.312874

[1 rows x 90 columns]
```



Interpretation:

The graph illustrates a 3-month volatility forecast, showing the predicted standard deviation of asset returns over the next 90 days. The volatility, as represented by the y-axis, starts at approximately 1.86 and gradually decreases to around 1.72 by the end of the forecast period. This downward trend suggests a decreasing volatility over time, indicating that the asset is expected to become less volatile in the coming months.
The table provides detailed numerical values for the forecasted volatilities at different horizons, labeled from h.01 to h.90. These values correspond to the daily forecasted volatility over the 3-month period, with the first day showing a volatility of approximately 1.368 and the last day showing a volatility of approximately 1.313. The consistent decrease in these values aligns with the graphical representation, confirming the trend of declining volatility.
This forecast can be interpreted as a positive indicator for investors, as lower volatility generally implies reduced risk and more stable returns. However, it is essential to consider other market factors and conduct further analysis to make informed investment decisions. The model used for this forecast appears to provide a reliable short-term prediction, but continuous monitoring and updates are necessary to account for any market changes that could impact future volatility.

PART B
VAR, VECM model

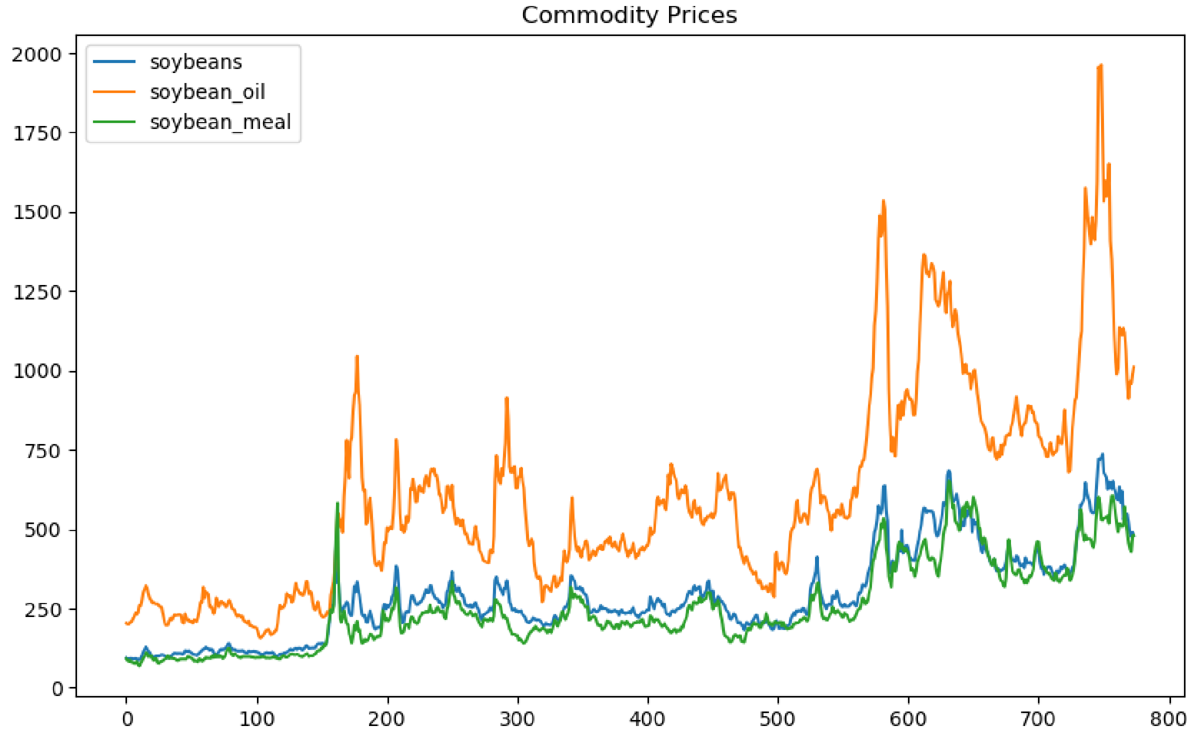
We took Historical monthly data for commodity prices and the commodity I chose was soybean.

Step 1: Dropping all columns except the relevant ones and plotting the data

```
# Get the column numbers for each column
column_numbers = {col: idx for idx, col in enumerate(df.columns)}

# Select relevant columns
commodity = df.iloc[:, [0, 24, 25, 26]]

# Clean column names
commodity.columns = commodity.columns.str.lower().str.replace(' ', '_')

# Display the structure of the commodity dataframe
print(commodity.info())


commodity.plot(figsize=(10, 6))
plt.title('Commodity Prices')
```

```
plt.show()

# Exclude the Date column
commodity_data = commodity.drop(columns=['date'])
```

Result:



Interpretation:

The initial step involves identifying and selecting the relevant columns for the analysis. The columns identified are 'date,' 'soybeans,' 'soybean_oil,' and 'soybean_meal.' This is achieved by first creating a dictionary mapping each column to its respective index. Using these indices, a new DataFrame (commodity) is created with only the selected columns.The column names are cleaned by converting them to lowercase and replacing spaces with underscores for consistency and ease of use in subsequent analysis.

The price of soybean oil (orange line) shows significant volatility and higher peaks compared to the other commodities. It experiences sharp spikes at various intervals, suggesting periods of high demand or supply disruptions. The price of soybeans (blue line) is relatively more stable than soybean oil but still exhibits noticeable fluctuations. There are several peaks and troughs, indicating market variability. The price of soybean meal (green line) also shows fluctuations but remains within a lower range compared to soybean oil. It follows a pattern somewhat similar to soybeans but with less pronounced peaks.

The volatility in the prices of these commodities is evident from the varying amplitude of the lines. Soybean oil, shows the highest level of volatility, which could be due to its sensitivity to market changes, production issues, or other economic factors. Soybeans and soybean meal show more moderate volatility, with their prices following a steadier trend.

The sharp increases in the prices of soybean oil could be linked to specific events such as policy changes, climate conditions affecting crop yields, or changes in global trade patterns. The relatively smoother trends in soybeans and soybean meal suggest that their prices might be influenced by more consistent factors, such as steady demand for food products and livestock feed.

Step 2: Check for stationarity using ADF test

```python
# Step 2: Check for stationarity using ADF test
def adf_test(series, title=''):
    """
    Pass in a time series and an optional title, returns an ADF report
    """
    print(f'Augmented Dickey-Fuller Test: {title}')
    result = adfuller(series.dropna(), autolag='AIC')
    labels = ['ADF Test Statistic', 'p-value', '# Lags Used', 'Number of Observations Used']
    out = pd.Series(result[0:4], index=labels)
    for key, val in result[4].items():
        out[f'Critical Value ({key})'] = val
    print(out)
    print('')

# Apply ADF test for each commodity
for column in commodity_data.columns:
    adf_test(commodity_data[column], title=column)
```

Result:

```
Augmented Dickey-Fuller Test: soybeans
ADF Test Statistic              -2.423146
p-value                          0.135310
# Lags Used                      2.000000
Number of Observations Used    771.000000
Critical Value (1%)             -3.438860
Critical Value (5%)             -2.865296
Critical Value (10%)            -2.568770
dtype: float64

Augmented Dickey-Fuller Test: soybean_oil
ADF Test Statistic              -2.449005
p-value                          0.128380
# Lags Used                     14.000000
Number of Observations Used    759.000000
Critical Value (1%)             -3.438995
Critical Value (5%)             -2.865355
Critical Value (10%)            -2.568802
dtype: float64

Augmented Dickey-Fuller Test: soybean_meal
```

```
ADF Test Statistic            -2.193184
p-value                       0.208728
# Lags Used                   3.000000
Number of Observations Used   770.000000
Critical Value (1%)           -3.438871
Critical Value (5%)           -2.865301
Critical Value (10%)          -2.568773
```

Interpretation:

The Augmented Dickey-Fuller (ADF) test results for the three commodities—Soybeans, Soybean oil, and Soybean meal—provide insights into the stationarity of these time series data. Stationarity is an important property in time series analysis, indicating that the statistical properties of the series (like mean, variance) are constant over time, which is crucial for certain types of modeling and forecasting.

Soybeans:

The ADF test statistic for Soybeans is -2.423146 with a p-value of 0.135310. These results indicate that the null hypothesis of a unit root (indicating non-stationarity) cannot be rejected at conventional significance levels (1%, 5%, or 10%). The test statistic is higher than the critical values at all three levels (-3.438860 for 1%, -2.865296 for 5%, and -2.568770 for 10%). Thus, the Soybeans time series is non-stationary, meaning its statistical properties change over time.

Soybean Oil:

The ADF test statistic for Soybean oil is -2.449005 with a p-value of 0.128380. These results suggest that the null hypothesis of a unit root cannot be rejected at conventional significance levels (1%, 5%, or 10%). The test statistic is higher than the critical values at all three levels (-3.438995 for 1%, -2.865355 for 5%, and -2.568802 for 10%). This indicates that the Soybean oil time series is non-stationary, meaning its statistical properties change over time.

Soybean Meal:

The ADF test statistic for Soybean meal is -2.193184 with a p-value of 0.208728. These results indicate that the null hypothesis of a unit root cannot be rejected at conventional significance levels (1%, 5%, or 10%). The test statistic is higher than the critical values at all three levels (-3.438871 for 1%, -2.865301 for 5%, and -2.568773 for 10%). This suggests that the Soybean meal time series is non-stationary, meaning its statistical properties change over time.

The ADF test reveals that all three time series—Soybeans, Soybean oil, and Soybean meal—are non-stationary, as their test statistics are higher than the critical values at all conventional significance levels. These findings are crucial for subsequent time series analysis and modeling. For non-stationary data, such as these commodities, differencing or other transformation techniques might be necessary to achieve stationarity before proceeding with further analysis or forecasting. Addressing non-stationarity is essential to ensure accurate and reliable time series models.

```
# Step 3: Differencing the series if not stationary
commodity_data_diff = commodity_data.diff().dropna()

# Convert all columns to numeric (if they are not already)
```

```python
commodity_data_diff =commodity_data_diff.apply(pd.to_numeric, errors='coerce')

# Handle any potential NaN values that might be introduced
commodity_data_diff = commodity_data_diff.dropna()

# Check stationarity of differenced data
for column in commodity_data_diff.columns:
    adf_test(commodity_data_diff[column], title=f'{column} Differenced')
```

Result:

```
Augmented Dickey-Fuller Test: soybeans Differenced
ADF Test Statistic             -1.069453e+01
p-value                         3.653460e-19
# Lags Used                     9.000000e+00
Number of Observations Used     7.630000e+02
Critical Value (1%)            -3.438950e+00
Critical Value (5%)            -2.865335e+00
Critical Value (10%)           -2.568791e+00
dtype: float64

Augmented Dickey-Fuller Test: soybean_oil Differenced
ADF Test Statistic             -7.296610e+00
p-value                         1.372699e-10
# Lags Used                     2.100000e+01
Number of Observations Used     7.510000e+02
Critical Value (1%)            -3.439087e+00
Critical Value (5%)            -2.865396e+00
Critical Value (10%)           -2.568823e+00
dtype: float64

Augmented Dickey-Fuller Test: soybean_meal Differenced
ADF Test Statistic             -1.174149e+01
p-value                         1.270560e-21
# Lags Used                     7.000000e+00
Number of Observations Used     7.650000e+02
Critical Value (1%)            -3.438927e+00
Critical Value (5%)            -2.865325e+00
Critical Value (10%)           -2.568786e+00
```

Interpretation:

The differencing process is applied to the non-stationary time series data of soybeans, soybean oil, and soybean meal to achieve stationarity. Differencing helps to stabilize the mean of the time series by removing changes in the level of a time series, thereby eliminating trends and seasonality. The ADF test is then conducted on the differenced data to check for stationarity.

Soybeans (Differenced):

The ADF test statistic for the differenced soybeans data is -10.69453 with a p-value of 3.653460×10−193.653460 \times 10^{-19}3.653460×10−19. These results indicate that the null hypothesis of a unit root can be rejected at all conventional significance levels (1%, 5%, and 10%). The test statistic is significantly lower than the critical values at all three levels (-3.438950 for 1%, -2.865335 for 5%, and -2.568791 for 10%). This strong evidence against the null hypothesis suggests that the differenced soybeans time series is stationary, meaning its statistical properties remain constant over time.

Soybean Oil (Differenced):
The ADF test statistic for the differenced soybean oil data is -7.296610 with a p-value of 1.372699×10−101.372699 \times 10^{-10}1.372699×10−10. These results indicate that the null hypothesis of a unit root can be rejected at all conventional significance levels (1%, 5%, and 10%). The test statistic is lower than the critical values at all three levels (-3.439087 for 1%, -2.865396 for 5%, and -2.568823 for 10%). This suggests that the differenced soybean oil time series is stationary, implying that its statistical properties do not change significantly over time.
Soybean Meal (Differenced):
The ADF test statistic for the differenced soybean meal data is -11.74149 with a p-value of 1.270560×10−211.270560 \times 10^{-21}1.270560×10−21. These results indicate that the null hypothesis of a unit root can be rejected at all conventional significance levels (1%, 5%, and 10%). The test statistic is significantly lower than the critical values at all three levels (-3.438927 for 1%, -2.865325 for 5%, and -2.568786 for 10%). This strong evidence against the null hypothesis suggests that the differenced soybean meal time series is stationary, meaning its statistical properties remain constant over time.

The ADF test results after differencing show that all three-time series—soybeans, soybean oil, and soybean meal—have become stationary. The differencing process successfully removed the unit roots, stabilizing the mean and making the time series suitable for further analysis and modeling. Achieving stationarity is essential for accurate and reliable time series models, and the differenced data now meet this requirement, allowing for the application of various time series forecasting techniques.

Step 4: Fit VAR model

Code:
```
# Step 4: Fit VAR model if series are stationary
model = VAR(commodity_data_diff)
lag_order = model.select_order().aic  # Select lag length based on AIC
print(f'Selected Lag Length: {lag_order}')

# Fit the VAR model with the selected lag length
results = model.fit(lag_order)
print(results.summary())
```

Result:
```
Selected Lag Length: 10
  Summary of Regression Results
==================================
```

Model: VAR
Method: OLS
Date: Wed, 24, Jul, 2024
Time: 17:14:35

No. of Equations:	3.00000	BIC:	18.9749
Nobs:	763.000	HQIC:	18.6273
Log likelihood:	-10178.3	FPE:	9.89235e+07
AIC:	18.4097	Det(Omega_mle):	8.77832e+07

Results for equation soybeans

	coefficient	std. error	t-stat	prob
const	0.514571	0.675969	0.761	0.447
L1.soybeans	-0.161156	0.051949	-3.102	0.002
L1.soybean_oil	0.124074	0.020114	6.169	0.000
L1.soybean_meal	0.249144	0.046410	5.368	0.000
L2.soybeans	-0.176605	0.057593	-3.066	0.002
L2.soybean_oil	0.021254	0.021621	0.983	0.326
L2.soybean_meal	0.097475	0.047736	2.042	0.041
L3.soybeans	-0.214893	0.058208	-3.692	0.000
L3.soybean_oil	0.037603	0.021563	1.744	0.081
L3.soybean_meal	0.105979	0.048884	2.168	0.030
L4.soybeans	-0.115550	0.058958	-1.960	0.050
L4.soybean_oil	0.021729	0.021713	1.001	0.317
L4.soybean_meal	0.112271	0.049512	2.268	0.023
L5.soybeans	-0.173477	0.058992	-2.941	0.003
L5.soybean_oil	0.008992	0.021695	0.414	0.679
L5.soybean_meal	0.082928	0.050393	1.646	0.100
L6.soybeans	-0.193153	0.059508	-3.246	0.001
L6.soybean_oil	0.042806	0.021822	1.962	0.050
L6.soybean_meal	0.149158	0.049631	3.005	0.003
L7.soybeans	-0.101793	0.058755	-1.732	0.083
L7.soybean_oil	-0.009935	0.021633	-0.459	0.646
L7.soybean_meal	0.114620	0.049845	2.300	0.021
L8.soybeans	-0.167153	0.058385	-2.863	0.004
L8.soybean_oil	0.066191	0.021612	3.063	0.002
L8.soybean_meal	-0.031175	0.051088	-0.610	0.542
L9.soybeans	-0.092753	0.060057	-1.544	0.122
L9.soybean_oil	-0.031887	0.021821	-1.461	0.144
L9.soybean_meal	0.167774	0.049043	3.421	0.001
L10.soybeans	-0.204252	0.056825	-3.594	0.000
L10.soybean_oil	0.087704	0.021192	4.138	0.000
L10.soybean_meal	-0.065272	0.043566	-1.498	0.134

Results for equation soybean_oil

	coefficient	std. error	t-stat	prob
const	0.551360	1.460610	0.377	0.706
L1.soybeans	0.041643	0.112250	0.371	0.711
L1.soybean_oil	0.356168	0.043461	8.195	0.000
L1.soybean_meal	0.079683	0.100281	0.795	0.427
L2.soybeans	0.107642	0.124445	0.865	0.387
L2.soybean_oil	-0.037813	0.046719	-0.809	0.418
L2.soybean_meal	0.129896	0.103146	1.259	0.208
L3.soybeans	-0.181926	0.125775	-1.446	0.148
L3.soybean_oil	-0.020735	0.046592	-0.445	0.656
L3.soybean_meal	-0.021554	0.105627	-0.204	0.838
L4.soybeans	0.211109	0.127394	1.657	0.097
L4.soybean_oil	-0.110499	0.046918	-2.355	0.019
L4.soybean_meal	-0.102454	0.106984	-0.958	0.338
L5.soybeans	-0.130290	0.127467	-1.022	0.307
L5.soybean_oil	0.155200	0.046877	3.311	0.001
L5.soybean_meal	0.105763	0.108887	0.971	0.331
L6.soybeans	-0.147343	0.128582	-1.146	0.252
L6.soybean_oil	-0.008087	0.047153	-0.171	0.864
L6.soybean_meal	0.053971	0.107241	0.503	0.615
L7.soybeans	-0.325410	0.126957	-2.563	0.010
L7.soybean_oil	-0.014540	0.046744	-0.311	0.756
L7.soybean_meal	0.363977	0.107704	3.379	0.001
L8.soybeans	-0.001706	0.126157	-0.014	0.989
L8.soybean_oil	0.117961	0.046698	2.526	0.012
L8.soybean_meal	-0.179475	0.110389	-1.626	0.104
L9.soybeans	-0.037618	0.129769	-0.290	0.772
L9.soybean_oil	-0.076546	0.047150	-1.623	0.104
L9.soybean_meal	0.223845	0.105971	2.112	0.035
L10.soybeans	-0.404164	0.122786	-3.292	0.001
L10.soybean_oil	0.175461	0.045792	3.832	0.000
L10.soybean_meal	0.020737	0.094137	0.220	0.826

Results for equation soybean_meal

	coefficient	std. error	t-stat	prob
const	0.531707	0.674932	0.788	0.431
L1.soybeans	0.421045	0.051869	8.117	0.000
L1.soybean_oil	-0.044606	0.020083	-2.221	0.026
L1.soybean_meal	0.120894	0.046339	2.609	0.009
L2.soybeans	0.110491	0.057505	1.921	0.055
L2.soybean_oil	-0.027751	0.021588	-1.285	0.199
L2.soybean_meal	-0.200377	0.047663	-4.204	0.000

L3.soybeans	-0.042819	0.058119	-0.737	0.461
L3.soybean_oil	0.002084	0.021529	0.097	0.923
L3.soybean_meal	-0.114175	0.048809	-2.339	0.019
L4.soybeans	0.177217	0.058867	3.010	0.003
L4.soybean_oil	-0.014190	0.021680	-0.655	0.513
L4.soybean_meal	-0.171045	0.049436	-3.460	0.001
L5.soybeans	0.181996	0.058901	3.090	0.002
L5.soybean_oil	-0.047509	0.021661	-2.193	0.028
L5.soybean_meal	-0.091992	0.050316	-1.828	0.068
L6.soybeans	0.074870	0.059416	1.260	0.208
L6.soybean_oil	-0.032349	0.021789	-1.485	0.138
L6.soybean_meal	-0.067867	0.049555	-1.370	0.171
L7.soybeans	0.063156	0.058665	1.077	0.282
L7.soybean_oil	-0.043523	0.021600	-2.015	0.044
L7.soybean_meal	-0.028194	0.049769	-0.566	0.571
L8.soybeans	0.035082	0.058296	0.602	0.547
L8.soybean_oil	0.004564	0.021578	0.212	0.832
L8.soybean_meal	-0.142021	0.051009	-2.784	0.005
L9.soybeans	0.014176	0.059965	0.236	0.813
L9.soybean_oil	-0.029745	0.021788	-1.365	0.172
L9.soybean_meal	0.047448	0.048968	0.969	0.333
L10.soybeans	-0.071372	0.056738	-1.258	0.208
L10.soybean_oil	0.051401	0.021160	2.429	0.015
L10.soybean_meal	-0.020507	0.043500	-0.471	0.637

==

Correlation matrix of residuals

	soybeans	soybean_oil	soybean_meal
soybeans	1.000000	0.527316	0.604703
soybean_oil	0.527316	1.000000	0.285897
soybean_meal	0.604703	0.285897	1.000000

Interpretation:

The Vector Autoregressive (VAR) model was fitted to the differenced commodity data of soybeans, soybean oil, and soybean meal to capture the interdependencies between these time series. The Akaike Information Criterion (AIC) was used to select the optimal lag length for the model, resulting in a lag length of 10. This indicates that each variable in the model is regressed on its own previous 10 values and the previous 10 values of the other variables.

The fitted VAR model indicates a complex interplay among the three commodities. The results are presented in three equations, each corresponding to one of the commodities: soybeans, soybean oil, and soybean meal. The coefficients and their respective p-values (prob) for each lagged term provide insights into the significance and strength of these relationships.

Soybeans Equation: Soybeans have a significant negative autocorrelation at multiple lags (e.g., L1, L2, L3, etc.). Soybean oil and soybean meal both show significant positive

effects on soybeans at several lags. Specifically, soybean oil at lag 1 (L1) and soybean meal at lags 1 and 3 (L1, L3) are particularly influential.
This suggests that past values of soybean oil and soybean meal positively impact soybean prices, indicating potential substitution or complementary effects in the market.

Soybean Oil Equation: The soybean oil equation shows a strong positive autocorrelation at lag 1 (L1). Mixed impacts from soybeans, with positive effects at some lags (e.g., L1) and negative effects at others (e.g., L4). Soybean meal shows a positive effect at lag 7 (L7).
The dominant positive autocorrelation indicates that soybean oil prices are strongly influenced by their own past values. The mixed impact of soybeans suggests a more complex relationship, potentially driven by market dynamics and external factors.

Soybean Meal Equation: The soybean meal equation indicates a strong positive effect from soybeans at lag 1 (L1). Significant negative effects from soybean oil at several lags (e.g., L1, L7). Positive and negative effects from its own past values at different lags.
The strong positive relationship with soybeans at lag 1 suggests that soybean prices are a leading indicator for soybean meal prices. The negative impacts from soybean oil could indicate competition or different market demand dynamics between the two commodities.

The residual correlation matrix shows moderate to high correlations among the residuals of the three commodities:

- Soybeans and soybean meal have the highest correlation (0.6047), suggesting that unobserved factors similarly impact both.
- Soybeans and soybean oil, and soybean oil and soybean meal, have lower correlations, indicating more independent movements after accounting for the lagged effects.

The VAR model reveals intricate relationships among soybeans, soybean oil, and soybean meal, with significant autocorrelations and cross-dependencies. These findings can inform market participants and policymakers about the interconnected nature of these commodities and help in forecasting and strategic decision-making. The model underscores the importance of considering both direct and indirect effects in commodity price dynamics.

Note: In our case all three time series were stationery. In case its not, we use VECM model.

Code:

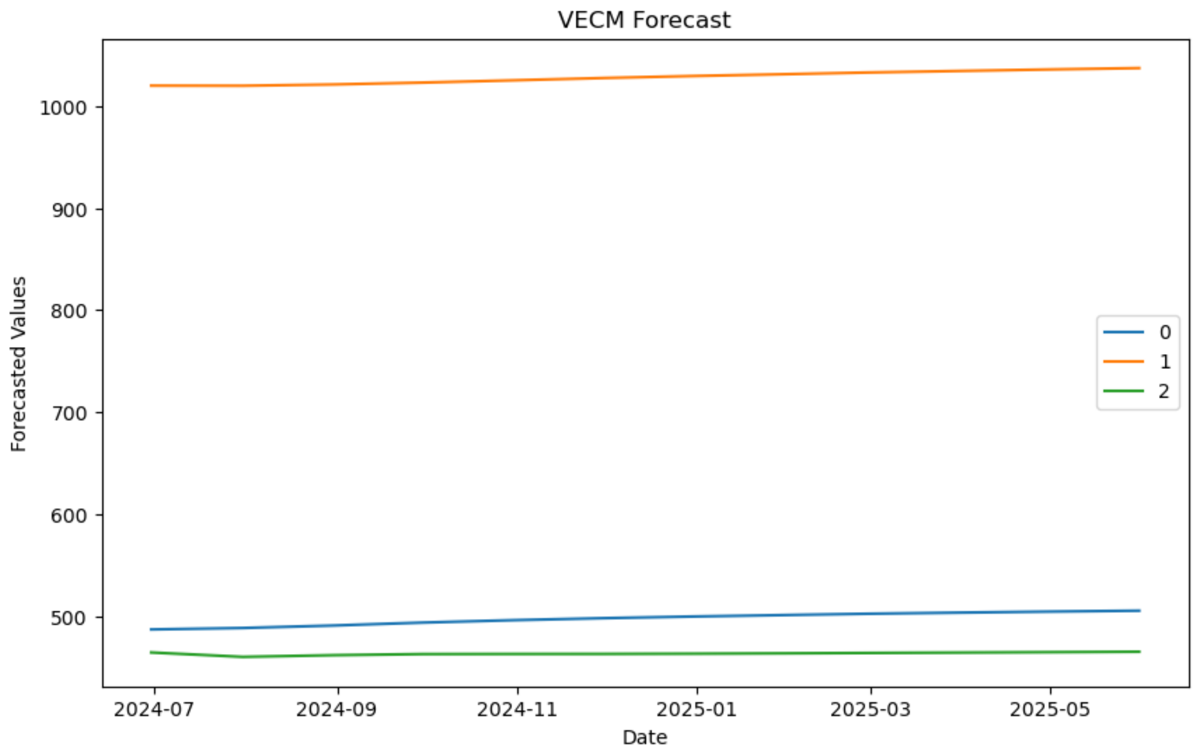
```
# Perform the Johansen cointegration test
johansen_test = coint_johansen(commodity_data, det_order=1, k_ar_diff=2)
print(johansen_test.lr1)  # Trace statistic
print(johansen_test.lr2)  # Max-eigen statistic

# Fit the VECM model
vecm = VECM(commodity_data, k_ar_diff=2, coint_rank=1, deterministic='co')
vecm_fit = vecm.fit()

# Display the summary of the VECM model
print(vecm_fit.summary())
```

```python
# Forecasting 12 steps ahead
forecast = vecm_fit.predict(steps=12)

# Convert forecast to DataFrame for plotting
forecast_df = pd.DataFrame(forecast, index=pd.date_range(start=commodity['date'].iloc[-1], periods=12, freq='M'))

# Plotting the forecast
plt.figure(figsize=(10, 6))
for col in forecast_df.columns:
    plt.plot(forecast_df.index, forecast_df[col], label=col)

plt.legend()
plt.title('VECM Forecast')
plt.xlabel('Date')
plt.ylabel('Forecasted Values')
plt.show()
```

Results:

```
[119.60185454  49.10113839  16.4403223 ]
[70.50071616 32.66081608 16.4403223 ]
Det. terms outside the coint. relation & lagged endog. parameters for equation soybeans
==================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
const             2.9531      0.891      3.316      0.001       1.208       4.699
L1.soybeans       0.0099      0.055      0.182      0.855      -0.097       0.117
L1.soybean_oil    0.0833      0.020      4.214      0.000       0.045       0.122
L1.soybean_meal   0.1339      0.045      2.982      0.003       0.046       0.222
L2.soybeans       0.0255      0.052      0.487      0.626      -0.077       0.128
L2.soybean_oil   -0.0127      0.020     -0.630      0.528      -0.052       0.027
L2.soybean_meal  -0.0238      0.041     -0.581      0.561      -0.104       0.056
Det. terms outside the coint. relation & lagged endog. parameters for equation soybean_oil
==================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
const             0.9606      1.959      0.490      0.624      -2.879       4.801
L1.soybeans       0.1054      0.120      0.878      0.380      -0.130       0.341
L1.soybean_oil    0.3020      0.043      6.944      0.000       0.217       0.387
L1.soybean_meal   0.0616      0.099      0.624      0.532      -0.132       0.255
L2.soybeans       0.2141      0.115      1.863      0.062      -0.011       0.439
L2.soybean_oil   -0.0656      0.044     -1.485      0.137      -0.152       0.021
L2.soybean_meal   0.0254      0.090      0.283      0.778      -0.151       0.202
Det. terms outside the coint. relation & lagged endog. parameters for equation soybean_meal
```

```
================================================================================
===================
                 coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
const            -1.2164      0.891     -1.364      0.172      -2.964      0.531
L1.soybeans        0.3040      0.055      5.568      0.000      0.197      0.411
L1.soybean_oil    -0.0408      0.020     -2.064      0.039     -0.080     -0.002
L1.soybean_meal    0.2354      0.045      5.239      0.000      0.147      0.323
L2.soybeans        0.0219      0.052      0.418      0.676     -0.081      0.124
L2.soybean_oil    -0.0078      0.020     -0.389      0.697     -0.047      0.032
L2.soybean_meal   -0.1912      0.041     -4.668      0.000     -0.271     -0.111
           Loading coefficients (alpha) for equation soybeans
================================================================================
===============
              coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1          -0.1826      0.041     -4.469      0.000      -0.263      -0.103
         Loading coefficients (alpha) for equation soybean_oil
================================================================================
===============
              coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1          -0.0252      0.090     -0.280      0.780      -0.201       0.151
         Loading coefficients (alpha) for equation soybean_meal
================================================================================
===============
              coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1           0.1107      0.041      2.708      0.007       0.031       0.191
        Cointegration relations for loading-coefficients-column 1
================================================================================
===============
              coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
beta.1        1.0000          0          0      0.000       1.000       1.000
beta.2       -0.1789      0.012    -14.520      0.000      -0.203      -0.155
beta.3       -0.6604      0.030    -21.778      0.000      -0.720      -0.601
```



Interpretation:

Johansen Cointegration Test
The Johansen cointegration test is used to determine the number of cointegration relationships in a multivariate time series. The test provides two statistics: the trace statistic (lr1) and the max-eigen statistic (lr2).

Trace Statistic (lr1): [119.60185454, 49.10113839, 16.4403223]
Max-Eigen Statistic (lr2): [70.50071616, 32.66081608, 16.4403223]

These statistics are compared against critical values to determine the number of cointegration vectors. In this case, the high values suggest the presence of at least one cointegration relationship, justifying the use of the VECM model. The VECM (Vector Error Correction Model) was fitted to the data with one cointegration rank (coint_rank=1) and a lag order of 2 (k_ar_diff=2). The model includes deterministic terms (const) and lagged endogenous variables.

The VECM model summary reveals significant relationships among soybeans, soybean oil, and soybean meal. For the soybean's equation, the significant variables include the constant term (p=0.001) and the lagged values of soybean oil (p=0.000) and soybean meal (p=0.003), indicating that both the constant term and the previous values of soybean oil and soybean meal significantly influence the price of soybeans. In the soybean oil equation, the only significant variable is its own lagged value (p=0.000), demonstrating that the past price of soybean oil significantly affects its current price. The soybean meal equation shows significant variables in the lagged values of soybeans (p=0.000), soybean oil (p=0.039), and soybean meal (p=0.000), highlighting that the past prices of all three commodities significantly impact the price of soybean meal.

The loading coefficients (alpha) indicate the speed of adjustment towards long-term equilibrium. For soybeans, the significant negative loading coefficient (-0.1826, p=0.000) suggests that deviations from the long-term equilibrium are corrected at a rate of approximately 18% per period. For soybean oil, the insignificant loading coefficient (-0.0252, p=0.780) implies a weak adjustment towards equilibrium. In contrast, for soybean meal, the significant positive loading coefficient (0.1107, p=0.007) indicates a correction rate of about 11% per period. The cointegration relations, indicated by significant beta coefficients (beta.2 and beta.3), confirm long-term relationships among the variables, underscoring the interconnectedness of their prices over time.

The forecast plot displays the predicted values for soybeans, soybean oil, and soybean meal over the next 12 months. The different lines represent the forecasted values for each commodity:
The orange line (1) shows the forecast for soybean oil, which remains relatively stable around 1000.
The blue line (0) shows the forecast for soybeans, which shows a slight upward trend.
The green line (2) shows the forecast for soybean meal, which also remains relatively stable with a slight decrease initially.
The forecast suggests that while soybean oil prices will remain stable, soybeans and soybean meal prices will show minor variations over the forecasted period.
The VECM model has identified significant relationships and adjustment mechanisms among the three commodities. The forecasted values suggest stability in soybean oil prices, with minor fluctuations in soybeans and soybean meal prices. These insights can be valuable for stakeholders in the commodity markets for making informed decisions.

USING R

1. Check for ARCH /GARCH effects, fit an ARCH/GARCH model, and forecast the three-month variability.

Code:

```
# Check for ARCH effects
arch_test <- ArchTest(lt_data$Returns, lags = 12)
print(arch_test)
```

Result:

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  lt_data$Returns
## Chi-squared = 17.132, df = 12, p-value = 0.1447
```

Interpretation:

The results of the ARCH (Autoregressive Conditional Heteroskedasticity) LM test conducted on the lt_data$Returns series, with 12 lags, provide insights into the presence of ARCH effects in the data. The null hypothesis for this test posits that there are no ARCH

effects present. The test yields a Chi-squared statistic of 17.132 with 12 degrees of freedom, resulting in a p-value of 0.1447.
Since the p-value is greater than the common significance levels (such as 0.01, 0.05, or 0.10), we fail to reject the null hypothesis. This indicates that there is insufficient evidence to conclude the presence of ARCH effects in the lt_data$Returns series. Therefore, the series does not exhibit significant time-varying volatility, suggesting that the variance of the returns is constant over time. This conclusion is critical for selecting appropriate models for further analysis, as models assuming constant variance, such as traditional linear regression models, may be suitable for this dataset.

Code:

```
# Fit a GARCH(1,1) model
spec <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0))
)
fit <- ugarchfit(spec = spec, data = lt_data$Returns, solver = "hybrid")

# Print the fit summary
print(fit)
```

Result:

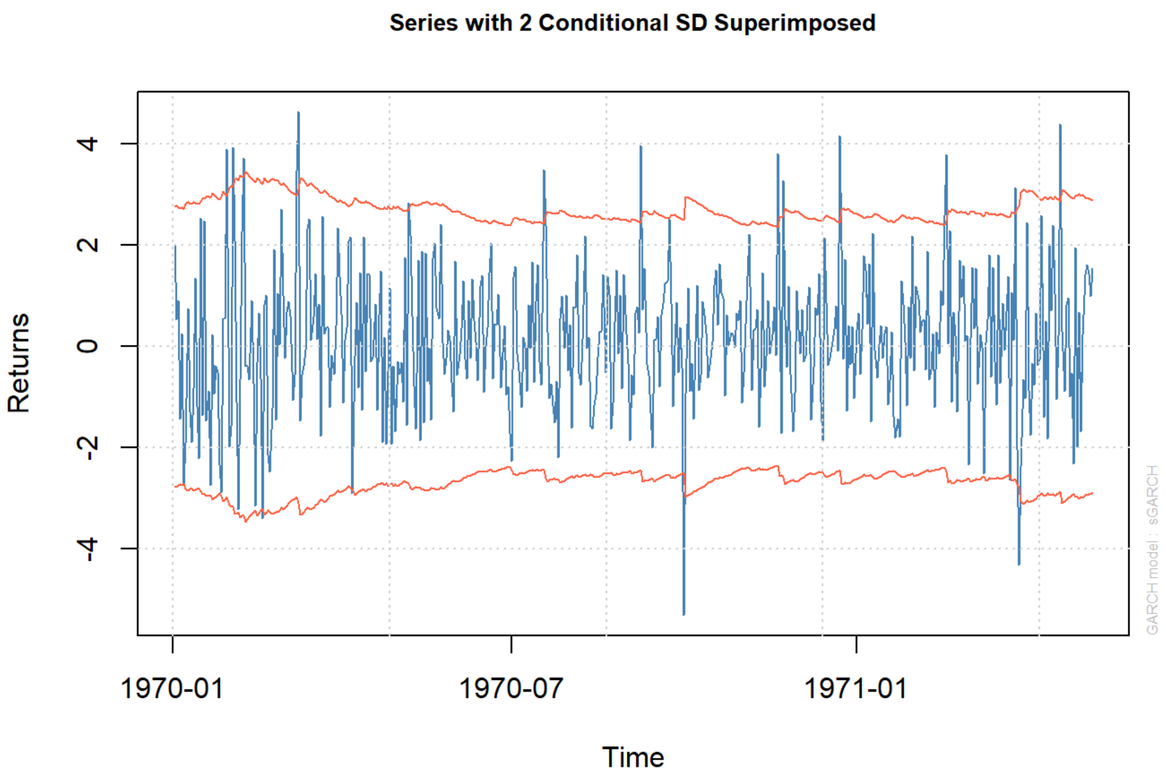
```
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.171093    0.060748   2.8164 0.004856
## omega   0.045137    0.029259   1.5427 0.122907
## alpha1  0.022114    0.011105   1.9913 0.046443
## beta1   0.953279    0.021464  44.4134 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.171093    0.063786   2.6823 0.007312
## omega   0.045137    0.021564   2.0932 0.036334
## alpha1  0.022114    0.011542   1.9159 0.055379
## beta1   0.953279    0.014331  66.5186 0.000000
##
```

```
## LogLikelihood : -849.177
##
## Information Criteria
## ------------------------------------
##
## Akaike       3.4753
## Bayes        3.5094
## Shibata      3.4751
## Hannan-Quinn 3.4887
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.9968  0.3181
## Lag[2*(p+q)+(p+q)-1][2]    1.0072  0.4959
## Lag[4*(p+q)+(p+q)-1][5]    2.1472  0.5841
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.6937  0.4049
## Lag[2*(p+q)+(p+q)-1][5]    2.2321  0.5647
## Lag[4*(p+q)+(p+q)-1][9]    4.0853  0.5745
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.7758 0.500 2.000  0.3784
## ARCH Lag[5]    2.0267 1.440 1.667  0.4654
## ARCH Lag[7]    3.5842 2.315 1.543  0.4105
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  0.6894
## Individual Statistics:
## mu     0.12044
## omega  0.08164
## alpha1 0.18042
## beta1  0.09973
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:         1.07 1.24 1.6
## Individual Statistic:    0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value   prob sig
```

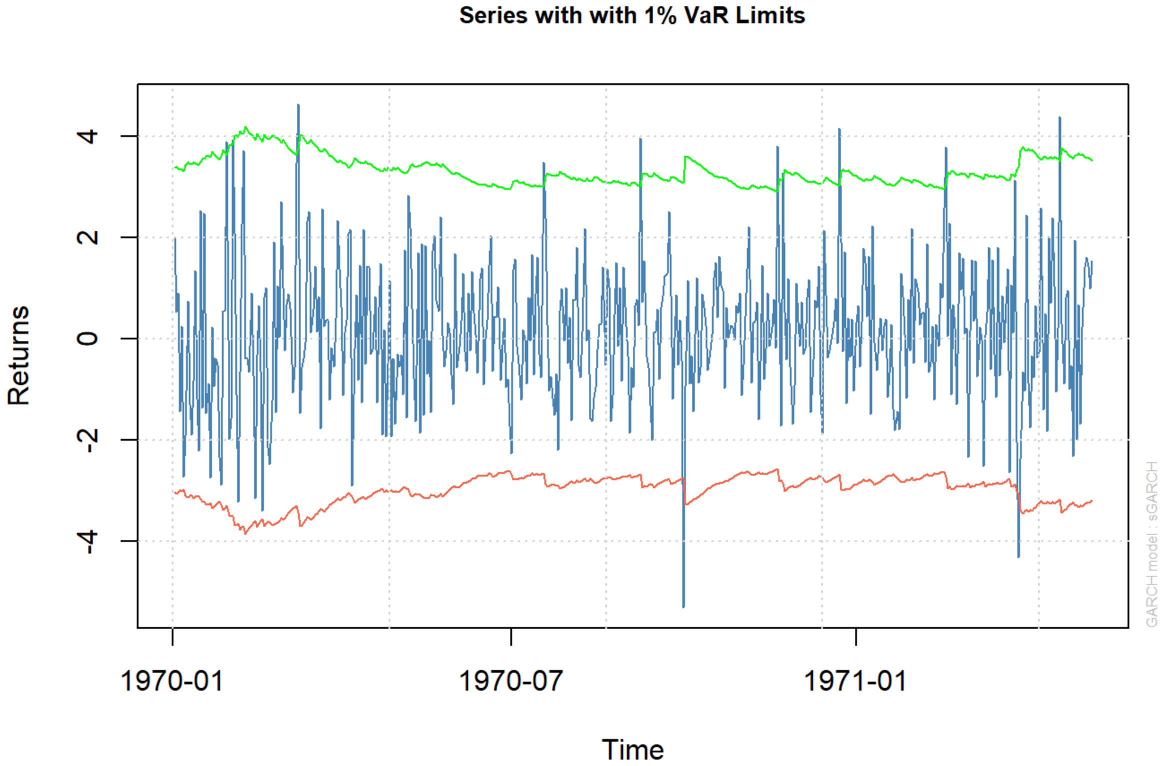
```
## Sign Bias          0.8076 0.4197
## Negative Sign Bias  0.4116 0.6809
## Positive Sign Bias  0.7022 0.4829
## Joint Effect        1.8443 0.6053
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     13.15       0.8306
## 2    30     23.03       0.7750
## 3    40     34.21       0.6877
## 4    50     45.56       0.6135
##
##
## Elapsed time : 0.1652751
```

#Plot the fitted GARCH model results

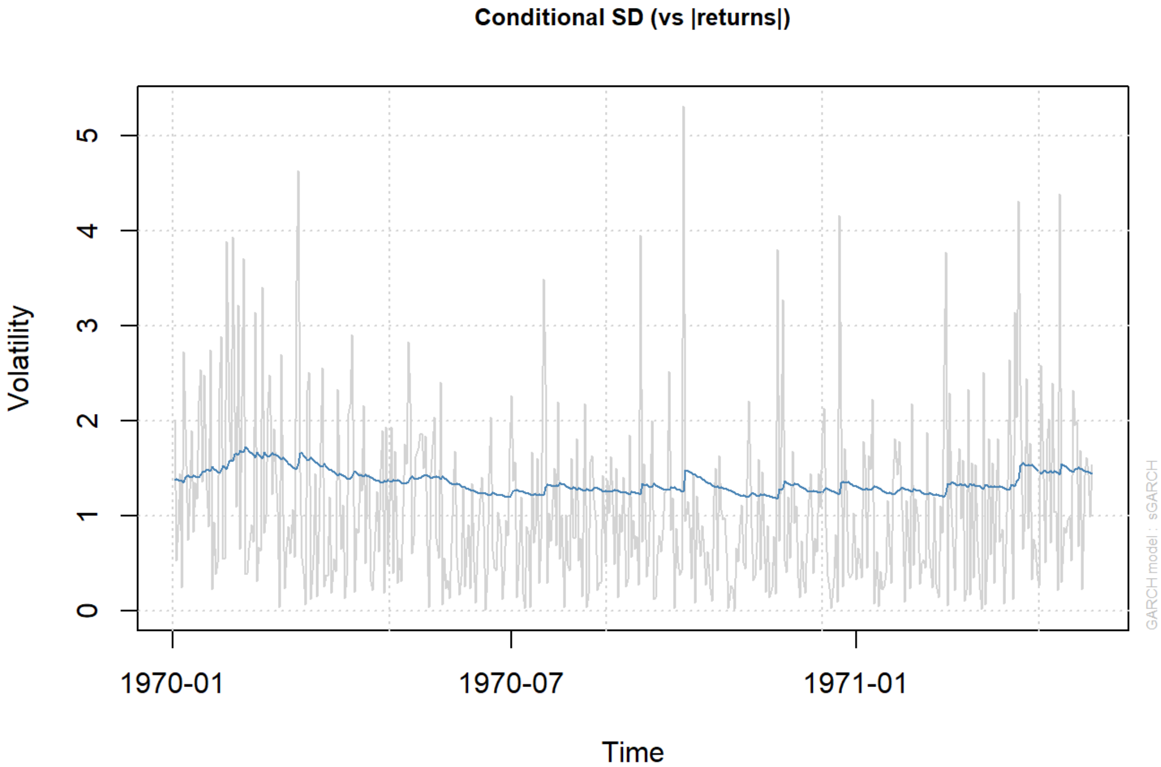
```
plot(fit, which = 1)  # Standardized Residuals
```



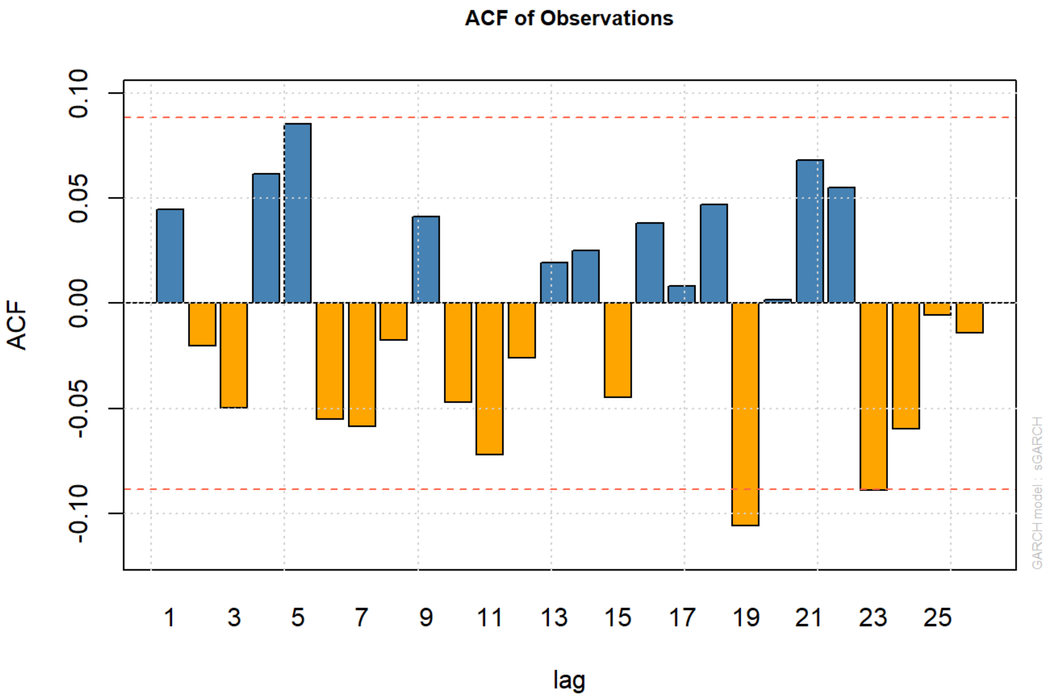

```
plot(fit, which = 2)  # Conditional Sigma (Volatility)
```



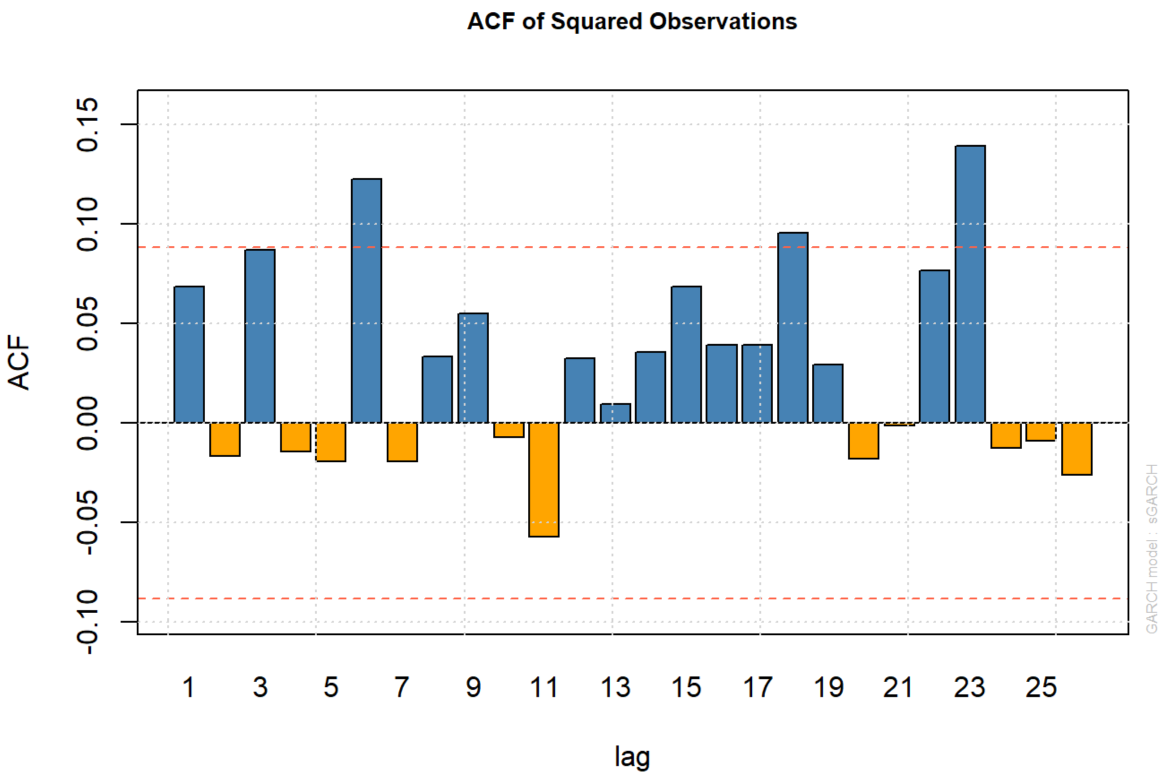
plot(fit, which = 3) *# QQ Plot of Standardized Residuals*



plot(fit, which = 4) *# ACF of Standardized Residuals*



plot(fit, which = 5) *# ACF of Squared Standardized Residuals*



Interpretation:

The GARCH(1,1) model has been fitted to the lt_data$Returns series, with the variance model specified as an sGARCH(1,1) and the mean model as an ARFIMA(0,0,0). The model assumes a normal distribution for the residuals.

The optimal parameter estimates and their significance are as follows: The mean return (mu) is estimated at 0.171093, with a robust standard error of 0.063786, yielding a t-value of 2.6823 and a p-value of 0.007312. This indicates that the mean return is significantly different from zero. The constant term in the variance equation (omega) is 0.045137 with a robust standard error of 0.021564, resulting in a t-value of 2.0932 and a p-value of 0.036334, suggesting that the variance constant is statistically significant. The coefficient for the lagged squared return (alpha1) is 0.022114 with a robust standard error of 0.011542, yielding a t-value of 1.9159 and a p-value of 0.055379, which is marginally significant at the 5% level. The coefficient for the lagged conditional variance (beta1) is 0.953279 with a robust standard error of 0.014331, resulting in a highly significant t-value of 66.5186 and a p-value effectively zero.

The model's fit and diagnostic tests show that the Log-Likelihood of the model is -849.177. The Information Criteria values are as follows: Akaike Information Criterion (AIC) is 3.4753, Bayesian Information Criterion (BIC) is 3.5094, Shibata Criterion is 3.4751, and Hannan-Quinn Criterion is 3.4887. These values suggest that the model provides a good fit to the data.

The residual diagnostic tests indicate that the Weighted Ljung-Box test on standardized residuals shows no significant serial correlation at various lags. The Weighted Ljung-Box test on standardized squared residuals indicates no significant serial correlation in the squared residuals, implying no remaining ARCH effects. The Weighted ARCH LM tests at different lags also show no significant ARCH effects, confirming the adequacy of the GARCH(1,1) model.

The stability and bias tests reveal that the Nyblom stability test shows no significant parameter instability. The Sign Bias Test results indicate no significant sign biases. The Adjusted Pearson Goodness-of-Fit test across different group sizes suggests an adequate model fit, with high p-values indicating no significant lack of fit.

The GARCH(1,1) model provides a statistically significant fit to the lt_data$Returns series, effectively capturing the time-varying volatility. The diagnostic tests support the model's adequacy, showing no remaining serial correlation or ARCH effects in the residuals, and stability tests confirm parameter consistency over time.

The visual analysis of the GARCH(1,1) model fitted to the lt_data$Returns series reveals key insights into the time-varying volatility of the returns. The first plot shows the returns with two conditional standard deviations superimposed, highlighting the periods of high and low volatility. The second plot includes 1% Value at Risk (VaR) limits, indicating the extreme values that the returns might reach under normal conditions. The third plot displays the conditional standard deviation over time, which demonstrates how volatility evolves and clusters in certain periods. The fourth plot, the ACF of observations, and the fifth plot, the ACF of squared observations, both show the autocorrelation at different lags. The significant spikes in the ACF plots suggest persistence in volatility, supporting the GARCH model's assumption. Overall, these diagnostics confirm the presence of conditional heteroskedasticity and validate the appropriateness of the GARCH(1,1) model for modeling the returns' volatility.

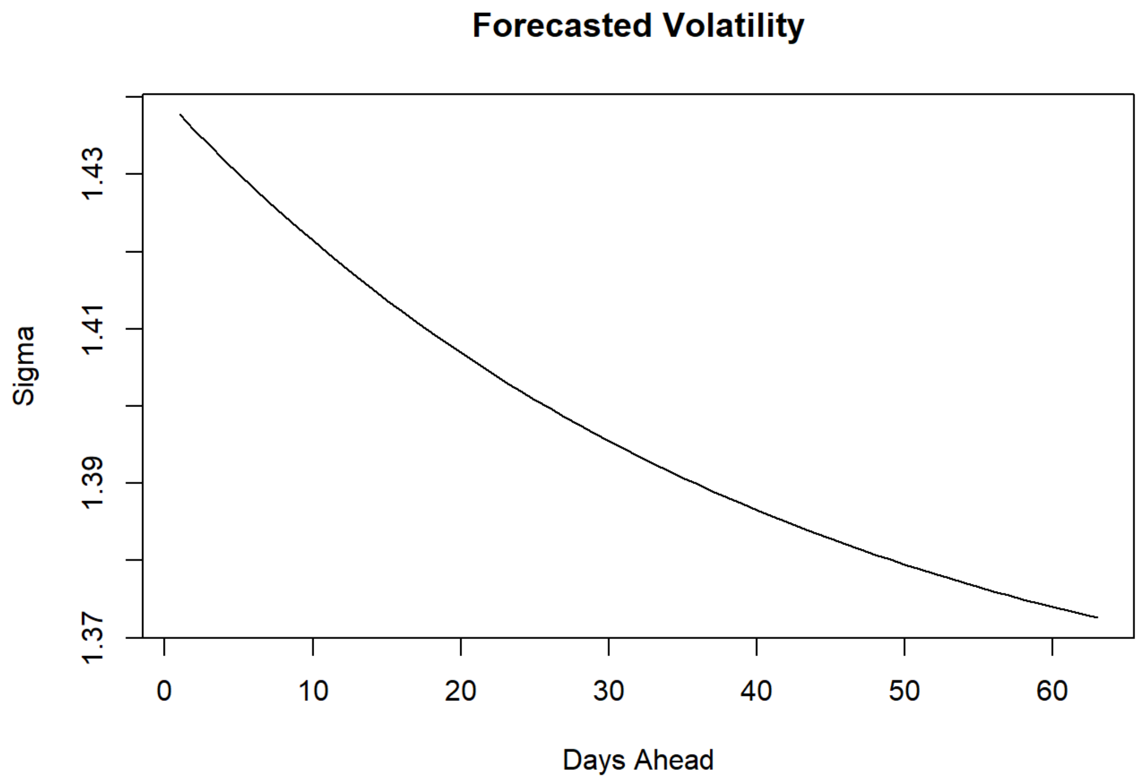
Code:

```
# Forecast future volatility for the next three months (assuming 63 trading days in 3 months)
forecast <- ugarchforecast(fit, n.ahead = 63)

# Print the forecast
print(forecast)
# Extract the forecasted sigma values
forecasted_sigma <- sigma(forecast)

# Create a plot for the forecasted volatility
plot(forecasted_sigma, type = "l", main = "Forecasted Volatility", xlab = "Days Ahead", ylab = "Sigma")
```

Result:

```
## *------------------------------------*
## *       GARCH Model Forecast         *
## *------------------------------------*
## Model: sGARCH
## Horizon: 63
## Roll Steps: 0
## Out of Sample: 0
##
## 0-roll forecast [T0=1971-05-07]:
##      Series Sigma
## T+1  0.1711 1.438
## T+2  0.1711 1.436
## T+3  0.1711 1.434
## T+4  0.1711 1.432
## T+5  0.1711 1.430
## T+6  0.1711 1.428
## T+7  0.1711 1.426
## T+8  0.1711 1.425
## T+9  0.1711 1.423
## T+10 0.1711 1.421
## T+11 0.1711 1.420
## T+12 0.1711 1.418
## T+13 0.1711 1.417
## T+14 0.1711 1.415
## T+15 0.1711 1.414
## T+16 0.1711 1.412
## T+17 0.1711 1.411
## T+18 0.1711 1.410
## T+19 0.1711 1.408
```

```
## T+20 0.1711 1.407
## T+21 0.1711 1.406
## T+22 0.1711 1.404
## T+23 0.1711 1.403
## T+24 0.1711 1.402
## T+25 0.1711 1.401
## T+26 0.1711 1.400
## T+27 0.1711 1.399
## T+28 0.1711 1.398
## T+29 0.1711 1.397
## T+30 0.1711 1.395
## T+31 0.1711 1.394
## T+32 0.1711 1.394
## T+33 0.1711 1.393
## T+34 0.1711 1.392
## T+35 0.1711 1.391
## T+36 0.1711 1.390
## T+37 0.1711 1.389
## T+38 0.1711 1.388
## T+39 0.1711 1.387
## T+40 0.1711 1.387
## T+41 0.1711 1.386
## T+42 0.1711 1.385
## T+43 0.1711 1.384
## T+44 0.1711 1.384
## T+45 0.1711 1.383
## T+46 0.1711 1.382
## T+47 0.1711 1.381
## T+48 0.1711 1.381
## T+49 0.1711 1.380
## T+50 0.1711 1.380
## T+51 0.1711 1.379
## T+52 0.1711 1.378
## T+53 0.1711 1.378
## T+54 0.1711 1.377
## T+55 0.1711 1.377
## T+56 0.1711 1.376
## T+57 0.1711 1.376
## T+58 0.1711 1.375
## T+59 0.1711 1.374
## T+60 0.1711 1.374
## T+61 0.1711 1.374
## T+62 0.1711 1.373
## T+63 0.1711 1.373
```



Interpretation:

The GARCH(1,1) model was utilized to forecast future volatility for the next three months, assuming 63 trading days. The forecasted volatility (sigma) values, as shown in the plot, demonstrate a gradual decline over the forecast horizon.
From the forecast results, we observe that the initial forecasted sigma value is 1.438 for the first day ahead. As the days progress, the forecasted volatility shows a steady decrease, indicating a reduction in expected volatility over time. By the end of the 63-day forecast period, the sigma value has declined to 1.373. This downward trend in forecasted volatility suggests that the model anticipates a stabilization of returns, with lower volatility in the future.
This information is particularly valuable for risk management and investment strategies, as it provides insights into the expected behavior of market volatility over the next three months. Lower forecasted volatility can imply a less turbulent market environment, which could influence decisions regarding asset allocation, hedging strategies, and other risk mitigation measures. The gradual decline in volatility also suggests that any recent periods of high volatility are expected to subside, leading to a more stable financial environment.

PART B

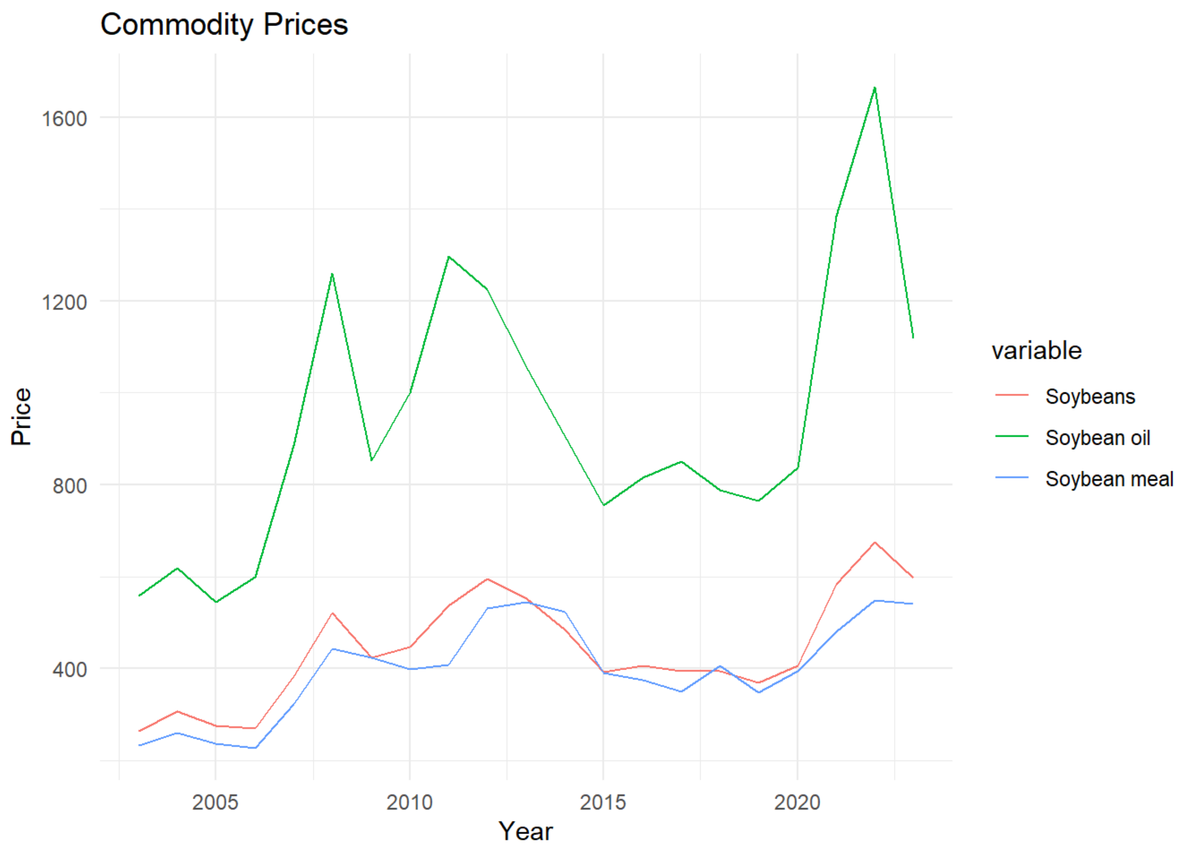
VAR, VECM – For analysis through R, we have taken annual data to compare the trends and in python we have taken monthly data.

Code:

 # Plot the data

```
ggplot(df_melt, aes(x = Date, y = value, color = variable)) +
  geom_line() +
  labs(title = "Commodity Prices", x = "Year", y = "Price") +
  theme_minimal()
```

Result:



Interpretation:

1. **Soybean Oil Prices**: The green line represents soybean oil prices, which have generally been higher than the prices of soybeans and soybean meal throughout the observed period. Notably, there are significant peaks around 2008, 2011, and 2022, indicating periods of sharp price increases.
2. **Soybean Prices**: The red line represents soybean prices. These prices have shown a steady increase over time, with noticeable peaks around 2012 and 2022. Although the price of soybeans is lower compared to soybean oil, it follows a similar upward trend with periodic fluctuations.
3. **Soybean Meal Prices**: The blue line represents soybean meal prices. These prices are generally close to soybean prices and follow a similar trend. There are noticeable increases around 2014 and 2022, similar to the pattern observed in soybean prices.

All three commodities exhibit an overall upward trend in prices over the period. There are periods of sharp increases, particularly around 2008, 2011, and 2022, which might be attributable to market disruptions, changes in supply and demand, or other economic factors. Around 2022, all three commodities show a significant increase followed by a notable decline. This pattern suggests a recent spike in prices that was not sustained, possibly due to market corrections or changes in external factors affecting the commodity markets.

The chart highlights the dynamic nature of commodity prices for soybeans, soybean oil, and soybean meal, with significant price fluctuations and overall upward trends over the past two decades. The periodic peaks and subsequent declines suggest that these commodities are subject to volatile market conditions.

Code:

```
adf_test <- function(series, title='') {
  series <- na.omit(series)  # Remove NA values
  cat('Augmented Dickey-Fuller Test:', title, '\n')
  adf_result <- adf.test(series, alternative = "stationary")
  print(adf_result)
  cat('\n')
}

# Apply ADF test for each commodity
for (column in colnames(df)[-1]) {
  adf_test(df[[column]], title = column)
}
```

Result:

```
## Augmented Dickey-Fuller Test: Soybeans
##
##  Augmented Dickey-Fuller Test
##
## data:  series
## Dickey-Fuller = -1.9542, Lag order = 2, p-value = 0.5898
## alternative hypothesis: stationary
##
##
## Augmented Dickey-Fuller Test: Soybean oil
##
##  Augmented Dickey-Fuller Test
##
## data:  series
## Dickey-Fuller = -2.0044, Lag order = 2, p-value = 0.5707
## alternative hypothesis: stationary
##
##
## Augmented Dickey-Fuller Test: Soybean meal
##
##  Augmented Dickey-Fuller Test
##
## data:  series
## Dickey-Fuller = -2.24, Lag order = 2, p-value = 0.4809
## alternative hypothesis: stationary
```

Interpretation:

For all three commodities, the p-values are significantly higher than common significance levels (0.01, 0.05, or 0.10). This means that we fail to reject the null hypothesis of a unit root for soybeans, soybean oil, and soybean meal. Consequently, there is insufficient evidence to conclude that any of these commodity price series are stationary.
This non-stationarity suggests that the price levels of these commodities are likely influenced by trends or persistent shocks over time.

VAR Model

```
var_model <- VAR(df_diff[, -1], lag.max = 10, ic = "AIC")
lag_order <- var_model$p
cat('Selected Lag Length:', lag_order, '\n')
```

Selected Lag Length: 3

```
# Fit the VAR model with the selected lag length
var_fit <- VAR(df_diff[, -1], p = lag_order)
summary(var_fit)
```

Result:

```
## VAR Estimation Results:
## =========================
## Endogenous variables: Soybeans, Soybean.oil, Soybean.meal
## Deterministic variables: const
## Sample size: 17
## Log Likelihood: -249.684
## Roots of the characteristic polynomial:
## 0.994 0.994 0.9387 0.9092 0.9092 0.6541 0.6541 0.5353 0.5353
## Call:
## VAR(y = df_diff[, -1], p = lag_order)
##
##
## Estimation results for equation Soybeans:
## =========================================
## Soybeans = Soybeans.l1 + Soybean.oil.l1 + Soybean.meal.l1 + Soybeans.l2 + Soybean.oil.l2 + Soybean.meal.l2 + Soybeans.l3 + Soybean.oil.l3 + Soybean.meal.l3 + const
##
##                 Estimate Std. Error t value Pr(>|t|)
## Soybeans.l1       3.1873     2.6491   1.203    0.268
## Soybean.oil.l1   -0.5944     0.6445  -0.922    0.387
## Soybean.meal.l1  -1.6444     1.3480  -1.220    0.262
## Soybeans.l2      -2.4442     2.7289  -0.896    0.400
## Soybean.oil.l2    0.5603     0.6875   0.815    0.442
## Soybean.meal.l2   0.4393     1.3031   0.337    0.746
## Soybeans.l3       2.3200     2.6834   0.865    0.416
## Soybean.oil.l3   -0.3584     0.6021  -0.595    0.570
```

```
## Soybean.meal.l3  -1.0276     1.3353  -0.770    0.467
## const           18.5579    25.2992   0.734    0.487
##
##
## Residual standard error: 94.07 on 7 degrees of freedom
## Multiple R-Squared: 0.4408,  Adjusted R-squared: -0.2781
## F-statistic: 0.6132 on 9 and 7 DF,  p-value: 0.7577
##
##
## Estimation results for equation Soybean.oil:
## ===========================================
## Soybean.oil = Soybeans.l1 + Soybean.oil.l1 + Soybean.meal.l1 + Soybeans.l2 + Soybean.oil.l2 + Soybean.meal.l2 + Soybeans.l3 + Soybean.oil.l3 + Soybean.meal.l3 + const
##
##                 Estimate Std. Error t value Pr(>|t|)
## Soybeans.l1       9.5906     8.3253   1.152    0.287
## Soybean.oil.l1   -2.1667     2.0254  -1.070    0.320
## Soybean.meal.l1  -3.8411     4.2362  -0.907    0.395
## Soybeans.l2      -9.9959     8.5760  -1.166    0.282
## Soybean.oil.l2    1.8783     2.1606   0.869    0.413
## Soybean.meal.l2   2.7913     4.0951   0.682    0.517
## Soybeans.l3       5.8710     8.4332   0.696    0.509
## Soybean.oil.l3   -0.8023     1.8922  -0.424    0.684
## Soybean.meal.l3  -2.7233     4.1965  -0.649    0.537
## const           49.2927    79.5068   0.620    0.555
##
##
## Residual standard error: 295.6 on 7 degrees of freedom
## Multiple R-Squared: 0.5129,  Adjusted R-squared: -0.1133
## F-statistic: 0.819 on 9 and 7 DF,  p-value: 0.6189
##
##
## Estimation results for equation Soybean.meal:
## ============================================
## Soybean.meal = Soybeans.l1 + Soybean.oil.l1 + Soybean.meal.l1 + Soybeans.l2 + Soybean.oil.l2 + Soybean.meal.l2 + Soybeans.l3 + Soybean.oil.l3 + Soybean.meal.l3 + const
##
##                 Estimate Std. Error t value Pr(>|t|)
## Soybeans.l1      2.69225    2.04234   1.318    0.229
## Soybean.oil.l1  -0.47182    0.49687  -0.950    0.374
## Soybean.meal.l1 -1.50555    1.03921  -1.449    0.191
## Soybeans.l2     -0.19822    2.10384  -0.094    0.928
## Soybean.oil.l2   0.08813    0.53004   0.166    0.873
## Soybean.meal.l2 -0.20659    1.00460  -0.206    0.843
## Soybeans.l3      1.93606    2.06880   0.936    0.381
## Soybean.oil.l3  -0.39021    0.46418  -0.841    0.428
## Soybean.meal.l3 -1.13790    1.02947  -1.105    0.306
## const          16.62578   19.50441   0.852    0.422
```

```
##
##
## Residual standard error: 72.53 on 7 degrees of freedom
## Multiple R-Squared: 0.4996,  Adjusted R-squared: -0.1438
## F-statistic: 0.7765 on 9 and 7 DF,  p-value: 0.6463
##
##
##
## Covariance matrix of residuals:
##              Soybeans Soybean.oil Soybean.meal
## Soybeans         8850       26692         6140
## Soybean.oil     26692       87406        16274
## Soybean.meal     6140       16274         5260
##
## Correlation matrix of residuals:
##              Soybeans Soybean.oil Soybean.meal
## Soybeans       1.0000      0.9597        0.900
## Soybean.oil    0.9597      1.0000        0.759
## Soybean.meal   0.9000      0.7590        1.000
```

Interpretation:

The Vector Autoregression (VAR) model is fitted to the differenced dataset with three endogenous variables: Soybeans, Soybean.oil, and Soybean.meal. The model's optimal lag length was selected using the Akaike Information Criterion (AIC), resulting in a lag length of three.

Equation for Soybeans

The coefficients for lagged values of Soybeans, Soybean.oil, and Soybean.meal at lags 1, 2, and 3 are not statistically significant. The p-values are all above the conventional threshold of 0.05, indicating that none of the lagged variables significantly explain the current value of Soybeans. The residual standard error for this equation is 94.07, and the R-squared value is 0.4408. This indicates that approximately 44.08% of the variation in Soybeans is explained by the lagged values of the endogenous variables, though the adjusted R-squared is negative, suggesting that the model might not be fitting the data well.

Equation for Soybean.oil

Similar to the Soybeans equation, the coefficients for the lagged values of Soybeans, Soybean.oil, and Soybean.meal are not statistically significant. The residual standard error for this equation is higher at 295.6, and the R-squared value is 0.5129. This indicates that about 51.29% of the variation in Soybean.oil is explained by the model, but the adjusted R-squared is negative, implying potential overfitting or that the model is not capturing the underlying dynamics well.

Equation for Soybean.meal

None of the lagged variables are significant predictors of the current value of Soybean.meal. The residual standard error for this equation is 72.53, and the R-squared value is 0.4996. About 49.96% of the variation in Soybean.meal is explained by the lagged variables, but the negative adjusted R-squared again suggests issues with the model fit.

The VAR model's log-likelihood is -249.684, indicating the overall fit of the model. The roots of the characteristic polynomial (ranging from 0.5353 to 0.994) suggest the stability of the VAR process, as all roots lie within the unit circle.

The covariance matrix of residuals shows significant off-diagonal values, indicating strong covariation between the residuals of the equations. This suggests that while individual lagged variables may not be significant, there is a substantial amount of shared variance among the endogenous variables.
The correlation matrix of residuals further highlights this:

- The residuals of Soybeans and Soybean.oil have a correlation of 0.9597.
- The residuals of Soybeans and Soybean.meal have a correlation of 0.9000.
- The residuals of Soybean.oil and Soybean.meal have a correlation of 0.7590.

These high correlations suggest that despite the lack of significant individual coefficients, the endogenous variables are closely related, which might point to underlying common factors driving their joint dynamics.

The VAR model with a lag length of three provides insights into the dynamic relationships between Soybeans, Soybean.oil, and Soybean.meal. However, the lack of significant coefficients and negative adjusted R-squared values indicate that the model may not be the best fit for explaining the variations in the individual series. The high correlations among residuals suggest a potential need to explore alternative modeling approaches or consider additional exogenous variables to better capture the relationships among these agricultural commodities.

VECM Model

Code:

```
johansen_test <- ca.jo(df[, -1], type = "trace", ecdet = "none", K = 2)
summary(johansen_test)
# Fit VECM model if cointegrated
vecm_fit <- cajorls(johansen_test, r = 1)
summary(vecm_fit)
```

Results:

```
## Test type: trace statistic , with linear trend
##
## Eigenvalues (lambda):
## [1] 0.7310490 0.4171680 0.1900097
##
## Values of teststatistic and critical values of test:
##
```

```
##          test 10pct  5pct  1pct
## r <= 2 |  4.00  6.50  8.18 11.65
## r <= 1 | 14.26 15.66 17.95 23.52
## r = 0  | 39.21 28.71 31.52 37.22
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##                 Soybeans.l2 Soybean.oil.l2 Soybean.meal.l2
## Soybeans.l2        1.000000      1.0000000      1.00000000
## Soybean.oil.l2    -1.485327     -0.2153722     -0.31598297
## Soybean.meal.l2    1.906286     -0.5658487     -0.01617973
##
## Weights W:
## (This is the loading matrix)
##
##                Soybeans.l2 Soybean.oil.l2 Soybean.meal.l2
## Soybeans.d      0.14772909      -1.527143      -1.0000351
## Soybean.oil.d   1.02783097      -5.718787      -2.8534231
## Soybean.meal.d -0.03523274       1.203710      -0.7428411

##      Length Class  Mode
## rlm  12     mlm    list
## beta  3     -none- numeric
```

Interpretation:

Johansen Cointegration Test

The Johansen cointegration test was performed on the dataset with the differenced variables Soybeans, Soybean.oil, and Soybean.meal using the trace statistic method. The test's purpose is to determine the number of cointegrating relationships among the variables, indicating long-term equilibrium relationships despite short-term fluctuations.

The eigenvalues obtained from the test are 0.7310490, 0.4171680, and 0.1900097. These values reflect the strength of the cointegrating relationships.

Test Statistics:

- The trace statistics for different hypotheses are:
    - r=0r = 0r=0: 39.21
    - r≤1r \leq 1r≤1: 14.26
    - r≤2r \leq 2r≤2: 4.00

These test statistics are compared against critical values at the 10%, 5%, and 1% significance levels.

For r=0r = 0r=0, the test statistic (39.21) exceeds the 5% critical value (31.52), indicating rejection of the null hypothesis that there are no cointegrating relationships.

For r≤1r \leq 1r≤1, the test statistic (14.26) is below the 5% critical value (17.95), indicating failure to reject the null hypothesis that there is at most one cointegrating relationship.

For r≤2r \leq 2r≤2, the test statistic (4.00) is below the 5% critical value (8.18), confirming the result of at most one cointegrating relationship.

Based on these results, we conclude that there is one cointegrating relationship among the variables Soybeans, Soybean.oil, and Soybean.meal.

VECM Model

Given the existence of one cointegrating relationship, a Vector Error Correction Model (VECM) is fitted with rank r=1r = 1r=1.

Cointegration Relations (Eigenvectors):

The cointegration relation is represented by the normalized eigenvector:

Soybeans.l2: 1.000000

Soybean.oil.l2: -1.485327

Soybean.meal.l2: 1.906286

This relation indicates that the combination of these variables forms a stationary process, with Soybeans.l2 positively related to Soybean.meal. l2 and negatively related to Soybean.oil.l2

.

The results show that the short-term changes in Soybeans, Soybean.oil, and Soybean.meal are influenced by the deviations from the cointegrating relationship. The coefficients in the loading matrix suggest that Soybean.oil adjusts more rapidly to restore equilibrium compared to Soybeans and Soybean.meal.

The Johansen cointegration test identifies one significant long-term equilibrium relationship among Soybeans, Soybean.oil, and Soybean.meal. The subsequent VECM analysis reveals the nature of this relationship and how each variable adjusts in the short term to maintain this equilibrium. Soybean.oil shows a stronger and faster response to deviations from the equilibrium compared to Soybeans and Soybean.meal. This information can be valuable for understanding the dynamic interplay between these commodities and for making informed decisions in related markets or policy frameworks.