Designed & Developed By

SALA . CHANTI SAI KUMAR

PROGRAM BOOK FOR PROJECT

Name of the Student	:	SALA . CHANTI SAI KUMAR
Name of the College	:	DVR&DR.HS MIC COLLAGE OF TECHNOLOGY
Registered Email	:	s.saikumar1243@gmail.com
Period of Project	:	From : 8-1-25 To : 9-1-25

INTELLIPAAT
2024-2025

An Project Report on

Global Commodity Prices

Submitted in accordance with the requirement for the degree of

Bachelor of Technology

Under the Faculty Guide of

Harsh

Department of

Computer Science and Engineering

Submitted by

SALA . CHANTI SAI KUMAR

Registered Email : s.saikumar1243@gmail.com

Department of

Computer Science and Engineering

DVR&DR.HS MIC COLLAGE OF TECHNOLOGY

YEAR: 2024-2025

CONTENTS

1. Problem Statement

The problem is to analyze commodity prices for various commodities using the commodity prices dataset. The goal is to leverage Python, data science techniques, statistical analysis and data modeling. Perform all necessary steps to get the key insights from the data.

2. Project Objective

1. The maximum price of Robusta coffe
2. 75th percentile of sugar prices in the Europen Union(EU)
3. The skewness of the price distribution for Arabica Coffee
4. How the distribution of sugar prices in the US significantly different from a normal distribution prices change
5. How many times does the price of Dubai oil exceed the price of Brent oil by a certain threshold of $10
6. What will be the overall price trend for each commodity
7. Which commodity experienced the highest price fluctuations during the observed period
8. How has brent oil prices vary on a quarterly basis since the last five years
9. How much the difference between global sugar prices and the prices of EU sugar and US sugar
10. The difference Between the distribution of sugar prices between Europe(EU) and the United States(US)

3. Data Description

Attributes	Description
date	The date of the recorded commodity price
oil_brent	The price of Brent oil ($/bbl)
Oil_Dubai	The price of Dubai oil ($/bbl)
Coffee_Arabica	The price of Arabica coffee ($/kg)
Coffee_Robustas	The price of Robusta coffee ($/kg)
Tea_Columbo	The price of Columbo tea ($/kg)
Tea_Kolkata	The price of Kolkata tea ($/kg)
Tea_Mombasa	The price of Mombasa tea ($/kg)
Sugar_EU	The price of EU sugar ($/kg)
Sugar_US	The price of US sugar ($/kg)
Sugar_World	The price of global sugar ($/kg)

Columns and Descriptions:

1. sl : Serial number (integer).
2. Date : Date in YYYY-MM-DD format (string).
3. oil_brent : Price of Brent oil (float).
4. oil_dubai : Price of Dubai oil (float).
5. coffee_arabica : Price of Arabica coffee (float).
6. coffee_robustas : Price of Robusta coffee (float).
7. tea_columbo : Price of tea from Colombo market (float).
8. tea_kolkata : Price of tea from Kolkata market (float).
9. tea_mombasa : Price of tea from Mombasa market (float).
10. sugar_eu : Price of sugar in the EU market (float).
11. sugar_us : Price of sugar in the US market (float).
12. sugar_world : World sugar price (float).

Data Types

```
data.info()
```

[6] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   sl                756 non-null    int64
 1   date              756 non-null    object
 2   oil_brent         756 non-null    float64
 3   oil_dubai         756 non-null    float64
 4   coffee_arabica    756 non-null    float64
 5   coffee_robustas   756 non-null    float64
 6   tea_columbo       756 non-null    float64
 7   tea_kolkata       756 non-null    float64
 8   tea_mombasa       756 non-null    float64
 9   sugar_eu          756 non-null    float64
 10  sugar_us          756 non-null    float64
 11  sugar_world       756 non-null    float64
dtypes: float64(10), int64(1), object(1)
memory usage: 71.0+ KB
```

No-Null Values in the DataSet:

```
data.notnull()
```

[18] ✓ 0.0s

	sl	date	oil_brent	oil_dubai	coffee_arabica	coffee_robustas	tea_columbo	tea_kolkata	tea_mombasa	sugar_eu	sugar_us	sugar_world	Quarter
0	True	True	True	True	True	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True	True	True	True	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...
751	True	True	True	True	True	True	True	True	True	True	True	True	True
752	True	True	True	True	True	True	True	True	True	True	True	True	True
753	True	True	True	True	True	True	True	True	True	True	True	True	True
754	True	True	True	True	True	True	True	True	True	True	True	True	True
755	True	True	True	True	True	True	True	True	True	True	True	True	True

756 rows × 13 columns

4. Data Pre-processing Steps and Inspiration

1. Loading the Data:

- The dataset is loaded using pd.read_csv().

```python
data = pd.read_csv('DataSet - Commodity_Prices.csv')
```

[5] ✓ 0.0s Python

2. Data Overview:

- Data structure and details are checked with data.info().

```python
data.info()
```

[6] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   sl               756 non-null    int64
 1   date             756 non-null    object
 2   oil_brent        756 non-null    float64
 3   oil_dubai        756 non-null    float64
 4   coffee_arabica   756 non-null    float64
 5   coffee_robustas  756 non-null    float64
 6   tea_columbo      756 non-null    float64
 7   tea_kolkata      756 non-null    float64
 8   tea_mombasa      756 non-null    float64
 9   sugar_eu         756 non-null    float64
 10  sugar_us         756 non-null    float64
 11  sugar_world      756 non-null    float64
dtypes: float64(10), int64(1), object(1)
memory usage: 71.0+ KB
```

3. Missing Data Handling:

- data.notnull() is used to examine missing data.
- Missing values in commodity columns are filled using forward-fill (fillna(method='ffill')).

```python
data.notnull()
```

[18] ✓ 0.0s

	sl	date	oil_brent	oil_dubai	coffee_arabica	coffee_robustas	tea_columbo	tea_kolkata	tea_mombasa	sugar_eu	sugar_us	sugar_world	Quarter
0	True	True	True	True	True	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True	True	True	True	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...
751	True	True	True	True	True	True	True	True	True	True	True	True	True
752	True	True	True	True	True	True	True	True	True	True	True	True	True
753	True	True	True	True	True	True	True	True	True	True	True	True	True
754	True	True	True	True	True	True	True	True	True	True	True	True	True
755	True	True	True	True	True	True	True	True	True	True	True	True	True

756 rows × 13 columns

4. Feature Exploration:

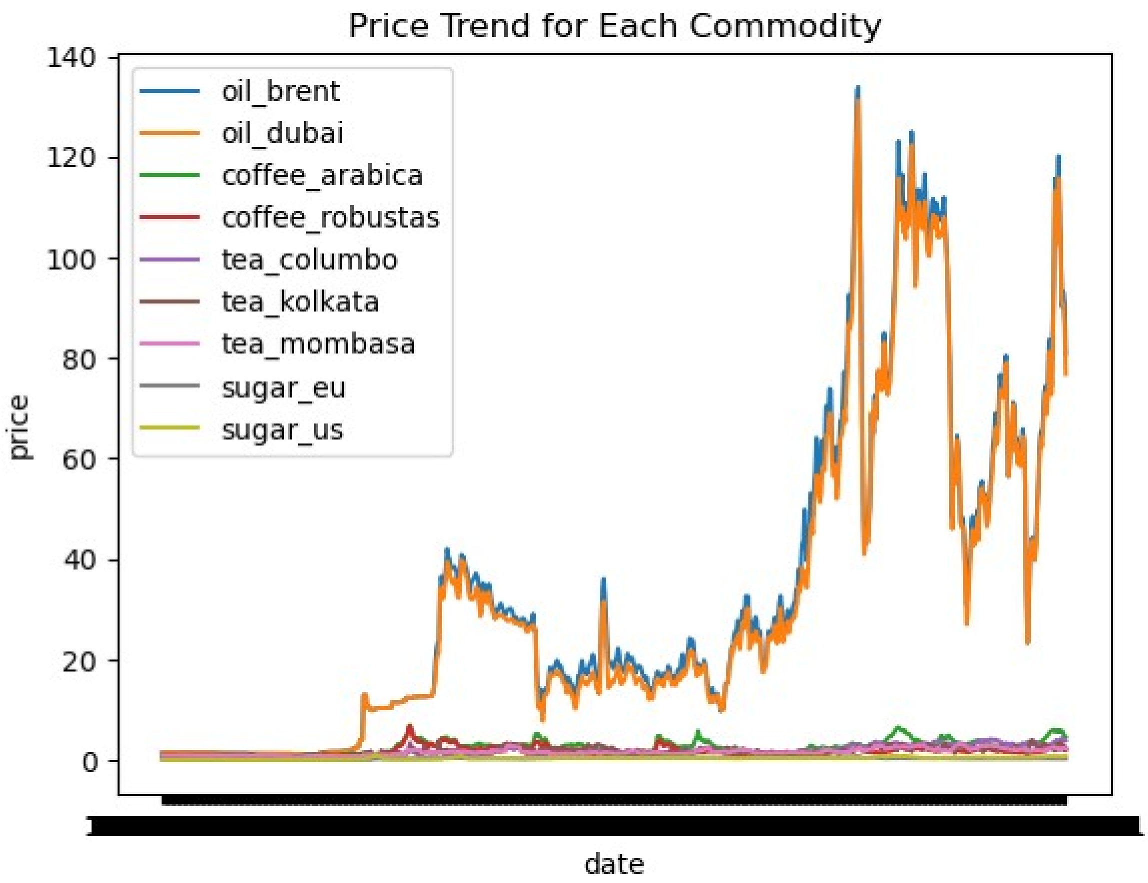
- Various features like maximum, percentiles, and skewness are calculated for specific columns.
- Statistical tests (e.g., normality test using normaltest) are performed on features to understand distributions.

5. Conditional Analysis:

- Certain conditions are evaluated, such as checking when one feature exceeds another by a specific threshold.

6. Data Visualization:

- Trends in commodity prices are visualized with line plots for different commodities.



5. Model Evaluation and Techniques

1. Data Analysis and Statistics:

- Descriptive Statistics:
    - Maximum value computation (max()).
    - Percentile computation using np.percentile().
- Skewness Measurement:
    - Skewness of distributions is calculated using scipy.stats.skew().
- Normality Test:
    - The normality of a distribution is tested using scipy.stats.normaltest().

2. Data Cleaning:

- Missing Value Handling:
    - Forward-fill imputation using fillna(method='ffill').

3. Conditional Analysis:

- Logical conditions and thresholds are used to analyze specific patterns (e.g., checking when one commodity price exceeds another).

4. Visualization:

- Line plots are created using matplotlib.pyplot to visualize trends in commodity prices.

5. Exploratory Data Analysis (EDA):

- Overview of the dataset using data.info() and null-value checks.

6. Conclusion

The analyzing commodity prices has the potential to generate valuable insights that can guide decision-making in industries reliant on commodities. By adding advanced data analysis, statistical modeling, and machine learning techniques, you can uncover patterns, predict future trends, and identify anomalies in pricing data. Exploring these possibilities will not only deepen your understanding of market dynamics but also provide practical solutions for managing risks and optimizing strategies in commodity trading and related fields.

7. Global Commodity Prices Project Code

commodity_analysis.ipynb

home > groot > Desktop > AI & DATA SCIENCE > Assignments > A1 > three > commodity_analysis.ipynb >

+ Code + Markdown | Run All Restart Clear All Outputs ... Python 3.12.3

```python
import pandas as pd
import numpy as np
from scipy.stats import skew, normaltest
import matplotlib.pyplot as plt
import seaborn as sns
```

[34] ✓ 0.0s Python

+ Code + Markdown

Add Markdown Cell

```python
data = pd.read_csv('DataSet - Commodity_Prices.csv')
```

[35] ✓ 0.0s Python

```python
data.info()
```

[36] ✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   sl               756 non-null    int64
 1   date             756 non-null    object
 2   oil_brent        756 non-null    float64
 3   oil_dubai        756 non-null    float64
 4   coffee_arabica   756 non-null    float64
 5   coffee_robustas  756 non-null    float64
 6   tea_columbo      756 non-null    float64
 7   tea_kolkata      756 non-null    float64
 8   tea_mombasa      756 non-null    float64
 9   sugar_eu         756 non-null    float64
 10  sugar_us         756 non-null    float64
 11  sugar_world      756 non-null    float64
dtypes: float64(10), int64(1), object(1)
memory usage: 71.0+ KB
```

```
data.notnull()
```

[18] ✓ 0.0s

	sl	date	oil_brent	oil_dubai	coffee_arabica	coffee_robustas	tea_columbo	tea_kolkata	tea_mombasa	sugar_eu	sugar_us	sugar_world	Quarter
0	True	True	True	True	True	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True	True	True	True	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...
751	True	True	True	True	True	True	True	True	True	True	True	True	True
752	True	True	True	True	True	True	True	True	True	True	True	True	True
753	True	True	True	True	True	True	True	True	True	True	True	True	True
754	True	True	True	True	True	True	True	True	True	True	True	True	True
755	True	True	True	True	True	True	True	True	True	True	True	True	True

756 rows × 13 columns

commodity_analysis.ipynb ●

home > groot > Desktop > AI & DATA SCIENCE > Assignments > A1 > three > commodity_analysis.ipynb >

+ Code + Markdown | Run All Restart Clear All Outputs ··· Python 3.12.3

1: What is the maximum price of Robusta coffee?

+ Code + Markdown

```python
max_robust_coffee = data['coffee_robustas'].max()
print("maximum price of Robusta coffee",max_robust_coffee)
```

[38] ✓ 0.0s Python

```
maximum price of Robusta coffee 6.883547
```

2: What is the 75th percentile of sugar prices in the European Union (EU)?

```python
percentile75_sug_eu = np.percentile(data['sugar_eu'], 75)
print("75th percentile of EU sugar prices: ",percentile75_sug_eu)
```

[39] ✓ 0.0s Python

```
75th percentile of EU sugar prices:  0.56951937998
```

3: What is the skewness of the price distribution for Arabica coffee?

```python
skewness_arabica_coffee = skew(data['coffee_arabica'])
print("Skewness of Arabica coffee prices: ",skewness_arabica_coffee
```

[40] ✓ 0.0s Python

```
Skewness of Arabica coffee prices:  0.5892256240650386
```

4: Is the distribution of sugar prices in the US significantly different from a normal distribution?

commodity_analysis.ipynb

home > groot > Desktop > AI & DATA SCIENCE > Assignments > A1 > three > commodity_analysis.ipynb >

+ Code + Markdown | Run All Restart Clear All Outputs ··· Python 3.12.3

4: Is the distribution of sugar prices in the US significantly different from a normal distribution?

```
us_sugar_prices = data['sugar_us']
stat, p_value = normaltest(us_sugar_prices)
if p_value < 0.05:
    print("The us sugar prices is different from a normal distribut
else:
    print("The US sugar prices is not different from a normal distr
```

[41] ✓ 0.0s Python

```
The US sugar prices is not different from a normal distribution.
```

5: How many times does the price of Dubai oil exceed the price of Brent oil by a certain threshold $10?

```
threshold_exceed_count = ((data['oil_dubai'] - data['oil_brent']) >
if threshold_exceed_count>10:
    print(f"Dubai oil price exceeds Brent oil price by $10 or more
else:
    print("The Dubai oil price is not exceeds Brent oil price")
```

[42] ✓ 0.0s Python

```
The Dubai oil price is not exceeds Brent oil price
```

6: What is the overall price trend for each commodity?

```
commodity_columns = data.columns[2:11]
```

commodity_analysis.ipynb ●

three > commodity_analysis.ipynb > M↓ 7: Which commodity experienced the highest price fluctuations during the observed period?

+ Code + Markdown | ▷ Run All ↺ Restart ☰ Clear All Outputs | Jupyter Variables ··· Python 3.12.3

6: What is the overall price trend for each commodity?

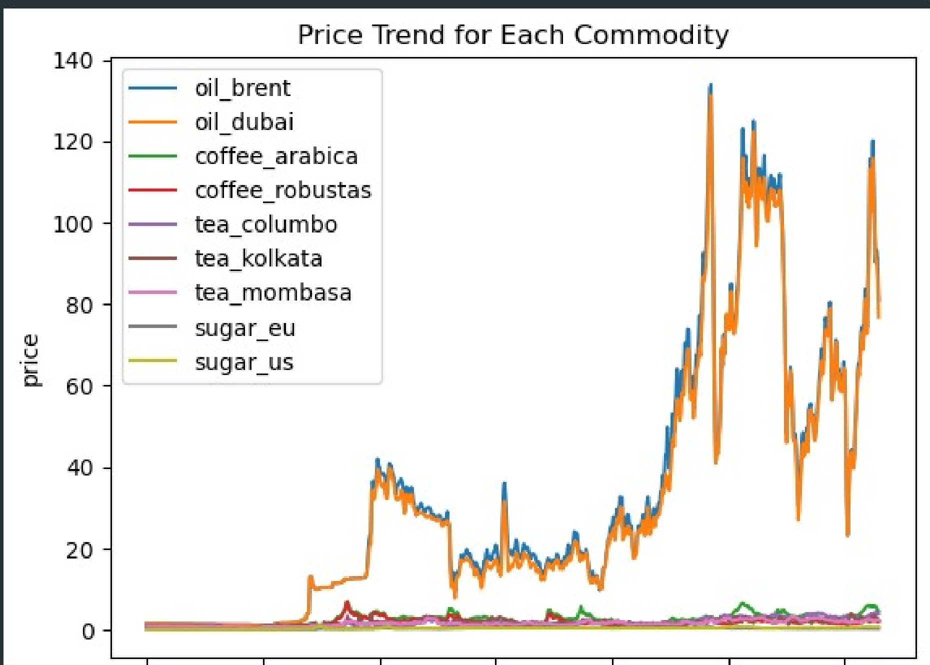
```python
commodity_columns = data.columns[2:11]
for column in commodity_columns:
    data[column].fillna(method='ffill', inplace=True)
    plt.plot(data['date'], data[column], label=column)
plt.xlabel('date')
plt.ylabel('price')
plt.title('Price Trend for Each Commodity')
plt.legend()
plt.show()
```

[48] ✓ 0.1s Python

```
/tmp/ipykernel_85106/2554639718.py:3: FutureWarning: A value is trying to be set on a c
The behavior will change in pandas 3.0. This inplace method will never work because the

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({co


  data[column].fillna(method='ffill', inplace=True)
/tmp/ipykernel_85106/2554639718.py:3: FutureWarning: Series.fillna with 'method' is dep
  data[column].fillna(method='ffill', inplace=True)
```



0 ⚠ 6 0 Spaces: 4 Cell 17 of 24

commodity_analysis.ipynb

home > groot > Desktop > AI & DATA SCIENCE > Assignments > A1 > three > commodity_analysis.ipynb > data['date'] = pd.to_datetime(data['date'])

\+ Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ... Python 3.12.3

7: Which commodity experienced the highest price fluctuations during the observed period?

```
numeric_data = data.select_dtypes(inclu (parameter) columns: Hashable | Sequence[Hashable] | Index[Any]
fluctuations = numeric_data.std()
max_fluctuation_commodity = fluctuations.idxmax()
print(f"During the observed period the {max_fluctuation_commodity} has highest price fluctuations")
```

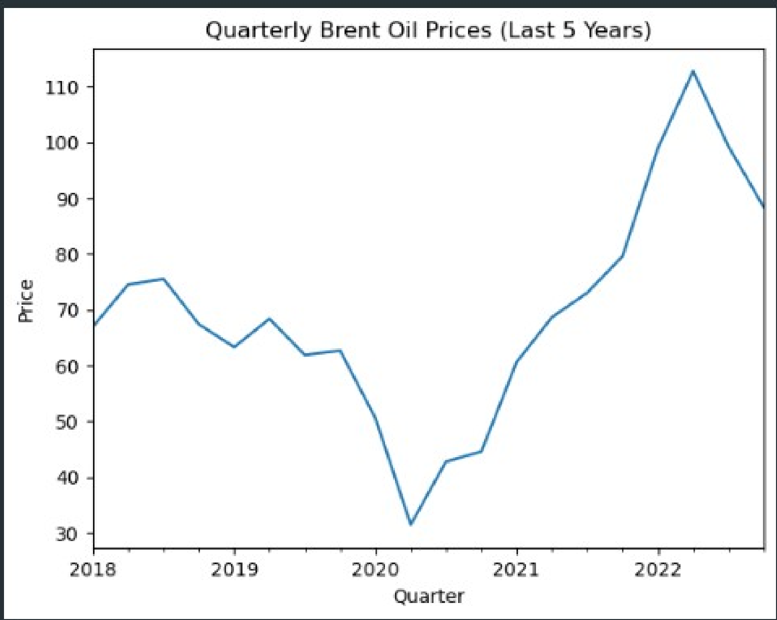
[44] ✓ 0.0s Python

```
During the observed period the oil_brent has highest price fluctuations
```

8: How has Brent oil prices varied on a quarterly basis since the last five years?

```
data['date'] = pd.to_datetime(data['date'])
data['Quarter'] = data['date'].dt.to_period('Q')
last_five_years = data[data['date'] > (data['date'].max() - pd.DateOffset(years=5))]
brent_quarterly = last_five_years.groupby('Quarter')['oil_brent'].mean()
brent_quarterly.plot(kind='line', title='Quarterly Brent Oil Prices (Last 5 Years)')
plt.xlabel('Quarter')
plt.ylabel('Price')
plt.show()
```

[45] ✓ 0.0s Python



Cell 20 of 24

commodity_analysis.ipynb ●

home > groot > Desktop > AI & DATA SCIENCE > Assignments > A1 > three > commodity_analysis.ipynb > data['date'] = pd.to_datetime(data['date'])

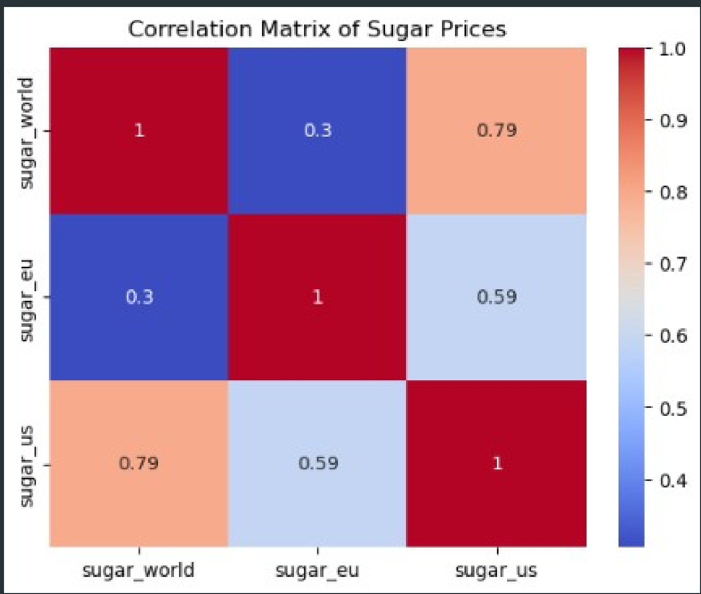
+ Code + Markdown | ▷ Run All ⟲ Restart ☰ Clear All Outputs | Jupyter Variables ☰ Outline ··· Python 3.12.3

9: Is there a correlation between global sugar prices and the prices of EU sugar and US sugar?

```python
correlation = data[['sugar_world', 'sugar_eu', 'sugar_us']].corr()
print("Correlation between sugar prices:\n", correlation)
sns.heatmap(correlation, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of Sugar Prices')
plt.show()
```

[46] ✓ 0.0s Python

```
Correlation between sugar prices:
              sugar_world  sugar_eu  sugar_us
sugar_world     1.000000  0.304783  0.794707
sugar_eu        0.304783  1.000000  0.590955
sugar_us        0.794707  0.590955  1.000000
```



10: Is there a significant difference in the distribution of sugar prices between Europe (EU) and the United States (US)?

```python
from scipy.stats import ttest_ind
eu_sugar_prices = data['sugar_eu']
us_sugar_prices = data['sugar_us']
t_stat, p_value = ttest_ind(eu_sugar_prices, us_sugar_prices, equal_var=False)
if p_value < 0.05:
    print("There is a difference in the sugar price distributions between EU and US.")
else:
    print("There is no difference in the sugar price distributions between EU and US.")
```

[47] ✓ 0.0s Python

```
There is a difference in the sugar price distributions between EU and US.
```

Cell 20 of 24