In [6]:
```
print("Shape of dataset:", df.shape)
df.info()
df.describe()
```

```
Shape of dataset: (1107, 19)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1107 entries, 0 to 1106
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype         
---  ------                                --------------  -----         
 0   Date                                  1107 non-null   datetime64[ns]
 1   Coal_RB_4800_FOB_London_Close_USD     1068 non-null   float64       
 2   Coal_RB_5500_FOB_London_Close_USD     1068 non-null   float64       
 3   Coal_RB_5700_FOB_London_Close_USD     1068 non-null   float64       
 4   Coal_RB_6000_FOB_CurrentWeek_Avg_USD  1084 non-null   float64       
 5   Coal_India_5500_CFR_London_Close_USD  1068 non-null   float64       
 6   Price_WTI                             1099 non-null   float64       
 7   Price_Brent_Oil                       1096 non-null   float64       
 8   Price_Dubai_Brent_Oil                 1067 non-null   float64       
 9   Price_ExxonMobil                      1067 non-null   float64       
 10  Price_Shenhua                         1042 non-null   float64       
 11  Price_All_Share                       1058 non-null   float64       
 12  Price_Mining                          1058 non-null   float64       
 13  Price_LNG_Japan_Korea_Marker_PLATTS   1066 non-null   float64       
 14  Price_ZAR_USD                         1107 non-null   float64       
 15  Price_Natural_Gas                     1099 non-null   float64       
 16  Price_ICE                             1067 non-null   float64       
 17  Price_Dutch_TTF                       1076 non-null   float64       
 18  Price_Indian_en_exg_rate              1047 non-null   float64       
dtypes: datetime64[ns](1), float64(18)
memory usage: 164.4 KB
```

Out[6]:

	Date	Coal_RB_4800_FOB_London_Close_USD	Coal_RB_5500_FOB_London_Close_USD	Coal_RB_5700_FOB_London_Close_US
count	1107	1068.000000	1068.000000	1068.00000
mean	2022-05-16 14:22:26.341463296	87.953184	113.353155	134.97180
min	2020-04-02 00:00:00	26.750000	34.560000	44.69000
25%	2021-04-24 12:00:00	50.370000	67.880000	83.40000
50%	2022-05-17 00:00:00	76.190000	95.800000	108.91000
75%	2023-06-07 12:00:00	104.180000	129.260000	155.60000
max	2024-06-28 00:00:00	301.090000	360.240000	412.35000
std	NaN	47.303473	63.105405	77.47239

In [4]:
```
import pandas as pd
file_path = r"C:\Users\Prerna Pandey\Downloads\Data set (2).xlsx"
df = pd.read_excel(file_path)
df.head()
```

Out[4]:

	Date	Coal_RB_4800_FOB_London_Close_USD	Coal_RB_5500_FOB_London_Close_USD	Coal_RB_5700_FOB_London_Close_USD	Coal_RB_6000
0	2020-04-02	41.00	53.22	64.7	
1	2020-04-03	40.34	52.36	63.1	
2	2020-04-06	40.34	52.36	63.1	
3	2020-04-07	40.34	52.36	63.1	
4	2020-04-08	40.34	52.36	63.1	

In [4]:

```
import pandas as pd
file_path = r"C:\Users\Prerna Pandey\Downloads\Data set (2).xlsx"
df = pd.read_excel(file_path)
df.head()
```

Out[4]:

	Date	Coal_RB_4800_FOB_London_Close_USD	Coal_RB_5500_FOB_London_Close_USD	Coal_RB_5700_FOB_London_Close_USD	Coal_RB_6000
0	2020-04-02	41.00	53.22	64.7	
1	2020-04-03	40.34	52.36	63.1	
2	2020-04-06	40.34	52.36	63.1	
3	2020-04-07	40.34	52.36	63.1	
4	2020-04-08	40.34	52.36	63.1	

In [4]:

```
import pandas as pd

df = pd.read_excel(r"C:\Users\Prerna Pandey\Downloads\Data set (2).xlsx")
df.fillna(df.mean(numeric_only=True), inplace=True)
df.head()
```

Out[4]:

	Date	Coal_RB_4800_FOB_London_Close_USD	Coal_RB_5500_FOB_London_Close_USD	Coal_RB_5700_FOB_London_Close_USD	Coal_RB_6000_
0	2020-04-02	41.00	53.22	64.7	
1	2020-04-03	40.34	52.36	63.1	
2	2020-04-06	40.34	52.36	63.1	
3	2020-04-07	40.34	52.36	63.1	
4	2020-04-08	40.34	52.36	63.1	

In [5]:
```
import numpy as np
num_df = df.select_dtypes(include=[np.number])
num_df.head()
```

Out[5]:

	Coal_RB_4800_FOB_London_Close_USD	Coal_RB_5500_FOB_London_Close_USD	Coal_RB_5700_FOB_London_Close_USD	Coal_RB_6000_FOB_Cu
0	41.00	53.22	64.7	
1	40.34	52.36	63.1	
2	40.34	52.36	63.1	
3	40.34	52.36	63.1	
4	40.34	52.36	63.1	

In [10]:
```
# Mean
mean_values = num_df.mean()
print("Mean:\n", mean_values)

# Median
```

```
median_values = num_df.median()
print("\nMedian:\n", median_values)

# Mode
mode_values = num_df.mode().iloc[0]
print("\nMode:\n", mode_values)
```

```
Mean:
 Coal_RB_4800_FOB_London_Close_USD          87.953184
Coal_RB_5500_FOB_London_Close_USD          113.353155
Coal_RB_5700_FOB_London_Close_USD          134.971807
Coal_RB_6000_FOB_CurrentWeek_Avg_USD       144.351494
Coal_India_5500_CFR_London_Close_USD       123.759710
Price_WTI                                   72.345469
Price_Brent_Oil                             76.283020
Price_Dubai_Brent_Oil                       74.752493
Price_ExxonMobil                            80.871012
Price_Shenhua                               21.274702
Price_All_Share                          68736.904811
Price_Mining                             52972.178374
Price_LNG_Japan_Korea_Marker_PLATTS         17.622529
Price_ZAR_USD                                0.059943
Price_Natural_Gas                            3.697999
Price_ICE                                  112.756111
Price_Dutch_TTF                             57.384856
Price_Indian_en_exg_rate                   141.616027
dtype: float64

Median:
 Coal_RB_4800_FOB_London_Close_USD          76.19000
Coal_RB_5500_FOB_London_Close_USD          95.80000
Coal_RB_5700_FOB_London_Close_USD          108.91000
Coal_RB_6000_FOB_CurrentWeek_Avg_USD       115.27000
Coal_India_5500_CFR_London_Close_USD       111.20000
Price_WTI                                   75.67000
Price_Brent_Oil                             79.96000
Price_Dubai_Brent_Oil                       78.79000
Price_ExxonMobil                            86.41000
Price_Shenhua                               22.32500
Price_All_Share                          70089.08500
Price_Mining                             53270.24500
Price_LNG_Japan_Korea_Marker_PLATTS         12.90250
Price_ZAR_USD                                0.05872
Price_Natural_Gas                            2.86300
Price_ICE                                  111.96000
Price_Dutch_TTF                             35.83850
Price_Indian_en_exg_rate                   140.90000
dtype: float64
```

```
Mode:
 Coal_RB_4800_FOB_London_Close_USD          33.86000
Coal_RB_5500_FOB_London_Close_USD           47.56000
Coal_RB_5700_FOB_London_Close_USD           50.12000
Coal_RB_6000_FOB_CurrentWeek_Avg_USD       115.27000
Coal_India_5500_CFR_London_Close_USD        41.81000
Price_WTI                                   78.26000
Price_Brent_Oil                             72.22000
Price_Dubai_Brent_Oil                       43.78000
Price_ExxonMobil                            64.31000
Price_Shenhua                               24.75000
Price_All_Share                          44598.70000
Price_Mining                             35258.30000
Price_LNG_Japan_Korea_Marker_PLATTS          2.00000
Price_ZAR_USD                                0.05273
Price_Natural_Gas                            2.49200
Price_ICE                                   90.40000
Price_Dutch_TTF                             13.90000
Price_Indian_en_exg_rate                   145.10000
Name: 0, dtype: float64
```

In [11]:

```python
# Variance
variance = num_df.var()
print("Variance:\n", variance)

# Standard Deviation
std_dev = num_df.std()
print("\nStandard Deviation:\n", std_dev)

# Range (max - min)
range_ = num_df.max() - num_df.min()
print("\nRange:\n", range_)
```

```
Variance:
 Coal_RB_4800_FOB_London_Close_USD       2.237619e+03
Coal_RB_5500_FOB_London_Close_USD       3.982292e+03
Coal_RB_5700_FOB_London_Close_USD       6.001971e+03
Coal_RB_6000_FOB_CurrentWeek_Avg_USD    6.972281e+03
Coal_India_5500_CFR_London_Close_USD    3.192492e+03
Price_WTI                               4.245831e+02
Price_Brent_Oil                         4.294504e+02
Price_Dubai_Brent_Oil                   4.204883e+02
Price_ExxonMobil                        7.712787e+02
Price_Shenhua                           3.765394e+01
Price_All_Share                         6.286333e+07
Price_Mining                            4.952807e+07
Price_LNG_Japan_Korea_Marker_PLATTS     1.617135e+02
Price_ZAR_USD                           3.681609e-05
Price_Natural_Gas                       3.690917e+00
Price_ICE                               1.885030e+02
Price_Dutch_TTF                         2.843977e+03
Price_Indian_en_exg_rate                2.931929e+03
dtype: float64

Standard Deviation:
 Coal_RB_4800_FOB_London_Close_USD         47.303473
Coal_RB_5500_FOB_London_Close_USD         63.105405
Coal_RB_5700_FOB_London_Close_USD         77.472391
Coal_RB_6000_FOB_CurrentWeek_Avg_USD      83.500186
Coal_India_5500_CFR_London_Close_USD      56.502138
Price_WTI                                 20.605415
Price_Brent_Oil                           20.723186
Price_Dubai_Brent_Oil                     20.505811
Price_ExxonMobil                          27.771904
Price_Shenhua                              6.136281
Price_All_Share                         7928.639789
Price_Mining                            7037.617925
Price_LNG_Japan_Korea_Marker_PLATTS       12.716663
Price_ZAR_USD                              0.006068
Price_Natural_Gas                          1.921176
Price_ICE                                 13.729638
Price_Dutch_TTF                           53.328953
Price_Indian_en_exg_rate                  54.147294
dtype: float64
```

```
Range:
 Coal_RB_4800_FOB_London_Close_USD         274.34000
Coal_RB_5500_FOB_London_Close_USD          325.68000
Coal_RB_5700_FOB_London_Close_USD          367.66000
Coal_RB_6000_FOB_CurrentWeek_Avg_USD       383.69000
Coal_India_5500_CFR_London_Close_USD       271.08000
Price_WTI                                  161.33000
Price_Brent_Oil                            108.65000
Price_Dubai_Brent_Oil                      103.46000
Price_ExxonMobil                            90.63000
Price_Shenhua                               28.24000
Price_All_Share                          36192.66000
Price_Mining                             39927.46000
Price_LNG_Japan_Korea_Marker_PLATTS         67.96000
Price_ZAR_USD                                0.02388
Price_Natural_Gas                            8.16500
Price_ICE                                   58.44000
Price_Dutch_TTF                            335.68500
Price_Indian_en_exg_rate                   250.23000
dtype: float64
```

In [12]:
```
skewness = num_df.skew()
print("Skewness:\n", skewness)
```

```
Skewness:
 Coal_RB_4800_FOB_London_Close_USD       1.312825
Coal_RB_5500_FOB_London_Close_USD        1.260528
Coal_RB_5700_FOB_London_Close_USD        1.278061
Coal_RB_6000_FOB_CurrentWeek_Avg_USD     1.269489
Coal_India_5500_CFR_London_Close_USD     0.862272
Price_WTI                               -0.536349
Price_Brent_Oil                         -0.429634
Price_Dubai_Brent_Oil                   -0.560067
Price_ExxonMobil                        -0.207440
Price_Shenhua                            0.565776
Price_All_Share                         -0.820778
Price_Mining                             0.288418
Price_LNG_Japan_Korea_Marker_PLATTS      1.084267
Price_ZAR_USD                            0.347963
Price_Natural_Gas                        1.260015
Price_ICE                                0.222972
Price_Dutch_TTF                          1.719477
Price_Indian_en_exg_rate                 0.371277
dtype: float64
```

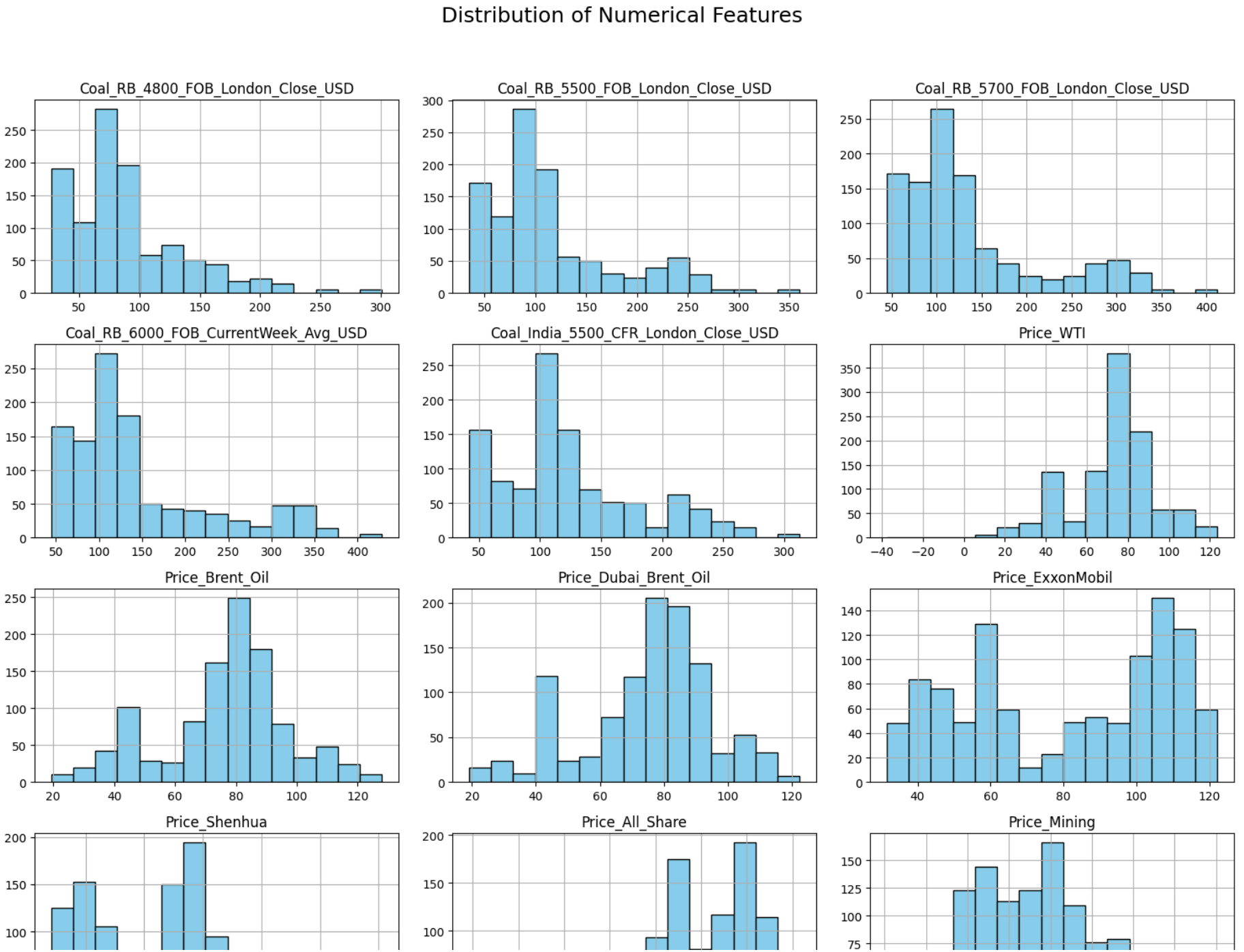
In [13]:
```
kurt = num_df.kurt()
print("Kurtosis:\n", kurt)
```

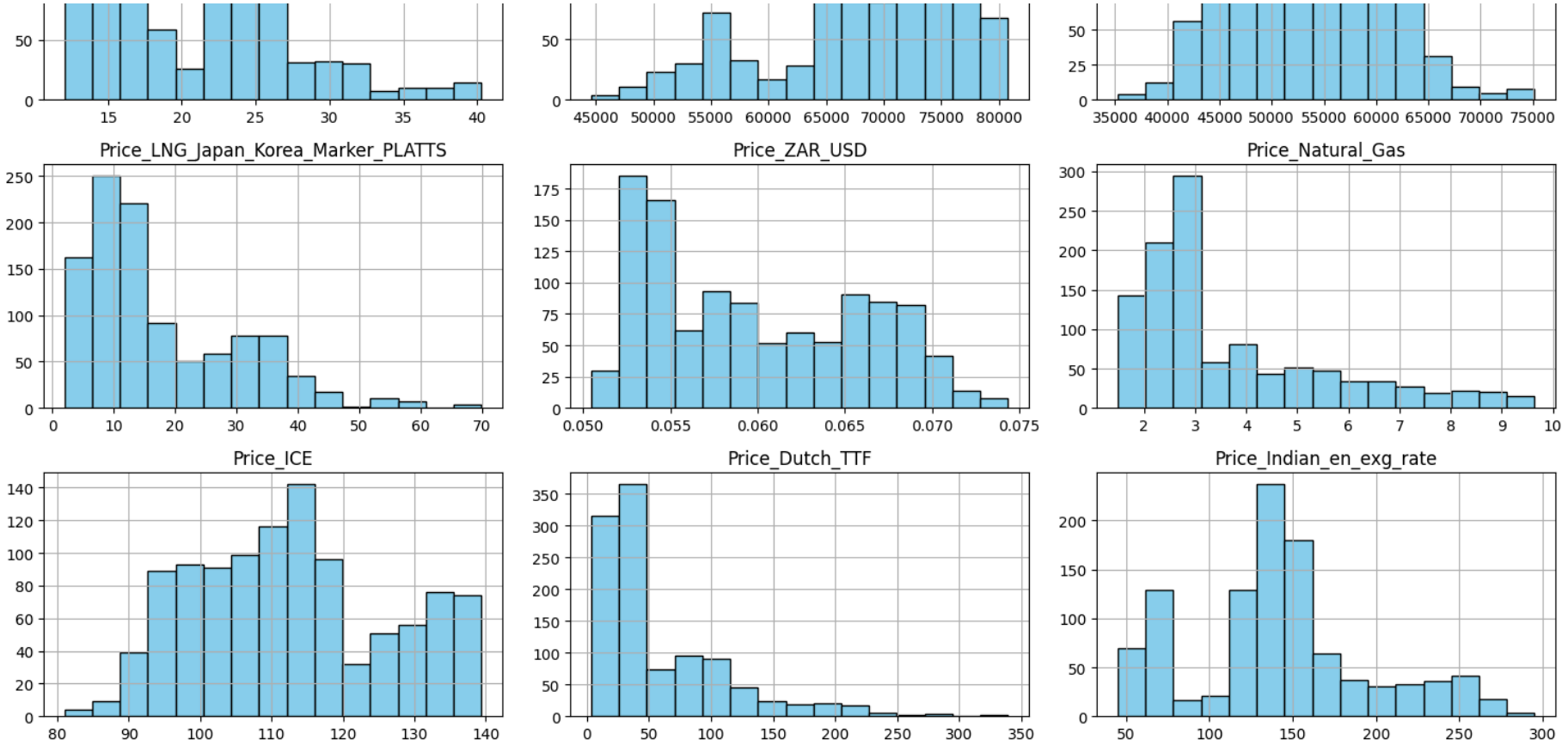
```
Kurtosis:
 Coal_RB_4800_FOB_London_Close_USD       2.116484
Coal_RB_5500_FOB_London_Close_USD        1.127323
Coal_RB_5700_FOB_London_Close_USD        0.811808
Coal_RB_6000_FOB_CurrentWeek_Avg_USD     0.717634
Coal_India_5500_CFR_London_Close_USD     0.176041
Price_WTI                                0.808288
Price_Brent_Oil                         -0.024637
Price_Dubai_Brent_Oil                   -0.038227
Price_ExxonMobil                        -1.486695
Price_Shenhua                            0.020671
Price_All_Share                         -0.119788
Price_Mining                            -0.362583
Price_LNG_Japan_Korea_Marker_PLATTS      0.789421
Price_ZAR_USD                           -1.217152
Price_Natural_Gas                        0.660356
Price_ICE                               -0.876720
Price_Dutch_TTF                          3.148565
Price_Indian_en_exg_rate                -0.091080
dtype: float64
```

In [15]:

```python
# Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math

# Load the Excel file
file_path = r"C:\Users\Prerna Pandey\Downloads\Data set (2).xlsx"
df = pd.read_excel(file_path)

# Select only numerical columns
num_df = df.select_dtypes(include=[np.number])

# Calculate number of columns
n_cols = len(num_df.columns)

# Calculate number of rows needed for subplots (3 columns per row)
n_rows = math.ceil(n_cols / 3)
```

```
# Create histograms
num_df.hist(
    bins=15,
    figsize=(15, n_rows * 3),
    layout=(n_rows, 3),
    edgecolor='black',
    color='skyblue'
)

# Add a title for the whole figure
plt.suptitle('Distribution of Numerical Features', fontsize=18, y=1.02)

# Tidy layout
plt.tight_layout()
plt.show()
```


Distribution of Numerical Features
Coal_RB_4800_FOB_London_Close_USD
Coal_RB_5500_FOB_London_Close_USD
Coal_RB_5700_FOB_London_Close_USD
Coal_RB_6000_FOB_CurrentWeek_Avg_USD
Coal_India_5500_CFR_London_Close_USD
Price_WTI
Price_Brent_Oil
Price_Dubai_Brent_Oil
Price_ExxonMobil
Price_Shenhua
Price_All_Share
Price_Mining



Exploratory Data Analysis on Excel Dataset

Project Overview

In this project, we perform **Exploratory Data Analysis (EDA)** on a dataset stored in an Excel file. The objective is to uncover patterns, detect anomalies, and summarize the main characteristics of the data using both **statistical measures** and **visualization techniques**.

This is a foundational and essential step in any data science or machine learning pipeline. Understanding your data helps guide future decisions like feature engineering or model selection.

Step-by-Step Workflow

1. **Importing Libraries**

We begin by importing the necessary Python libraries such as `pandas`, `numpy`, `matplotlib`, `seaborn`, and `scipy.stats` for statistical analysis and visualization.

2. **Loading the Dataset**

The Excel file is loaded using `pandas.read_excel()`, and we preview the data using `.head()` to understand the structure and types of features included.

3. **Data Overview**

We check:

- The shape of the dataset (rows × columns)
- Data types
- Missing values
- Summary statistics using `.describe()`

4. **Numerical Data Isolation**

We filter out only the **numerical columns** from the dataset to perform quantitative analysis.

📐 Statistical Measures

5. **Central Tendency**

We calculate:

- **Mean** – average value

- **Median** – middle value
- **Mode** – most frequent value
  These help us understand the “center” of the data.

6. **Measures of Dispersion**

We calculate:

- **Variance** – spread of the data
- **Standard Deviation** – average distance from the mean
- **Range** – difference between max and min values
  These help us understand the **variability** in the data.

7. **Skewness**

Skewness measures the **asymmetry** of the data distribution. It tells us whether the data is left-skewed, right-skewed, or symmetric.

8. **Kurtosis**

Kurtosis describes the **shape of the distribution** in terms of its "peakedness" or "flatness" compared to a normal distribution.

Visualization

9. **Histogram**

We plot histograms for each numerical feature to visually inspect their distribution.

10. **Boxplots (optional enhancement)**

Boxplots can help detect **outliers** and visualize the **spread** of each feature.

✅ Conclusion

By performing this analysis, we gain a deeper understanding of the dataset's structure, distributions, and characteristics. This is a crucial first step before applying any machine learning models or business logic.

In [6]:
```
df.to_csv(r"C:\Users\Prerna Pandey\Downloads\Data_cleaned.csv", index=False)
```

In [ ]:
```

```