This data set provides a comprehensive record of daily gold prices from January 19, 2014 to January 22, 2024. The data is provided by Nasdaq and includes key financial metrics for each trading day.

The dataset consists of the following columns:

Date: A unique date for each trading day recorded. Close: The closing price of gold on the relevant date. Volume: Gold trading volume on the relevant date. Open: The opening price of gold on the relevant date. High: The highest recorded price of gold during the trading day. Low: The lowest price recorded for gold in the trading day.

This analysis provides various insights to explore trends and patterns in gold stocks over a period of time

In [2]:
```
import pandas as pd
# Load the dataset
data = pd.read_csv('goldstock.csv')

# Display the first few rows of the dataset
data.head()
```

Out[2]:

	Unnamed: 0	Date	Close	Volume	Open	High	Low
0	0	2024-01-19	2029.3	166078.0	2027.4	2041.9	2022.2
1	1	2024-01-18	2021.6	167013.0	2009.1	2025.6	2007.7
2	2	2024-01-17	2006.5	245194.0	2031.7	2036.1	2004.6
3	3	2024-01-16	2030.2	277995.0	2053.4	2062.8	2027.6
4	4	2024-01-12	2051.6	250946.0	2033.2	2067.3	2033.1

In [9]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
# Display the number of rows and columns in the dataset
rows, columns = data.shape
print(f"Number of rows: {rows}, Number of columns: {columns}")
```

```
Number of rows: 2511, Number of columns: 7
```

In [5]:
```
# Display the data types of each column
data.dtypes
```

Out[5]:
```
Unnamed: 0      int64
Date           object
Close         float64
Volume        float64
Open          float64
High          float64
Low           float64
dtype: object
```

In [6]:
```
# Display summary statistics of the numerical columns
data.describe()
```

Out[6]:

	Unnamed: 0	Close	Volume	Open	High	Low
count	2511.000000	2511.000000	2511.000000	2511.000000	2511.000000	2511.000000
mean	1260.792911	1498.726085	185970.770609	1498.725528	1508.451454	1488.869932
std	729.262879	298.824811	97600.769382	299.118187	301.262244	296.417703
min	0.000000	1049.600000	1.000000	1051.500000	1062.700000	1045.400000
25%	630.500000	1249.850000	126693.500000	1249.500000	1257.300000	1242.350000
50%	1259.000000	1332.800000	175421.000000	1334.000000	1342.400000	1326.600000
75%	1888.500000	1805.850000	234832.000000	1805.600000	1815.450000	1793.050000
max	2532.000000	2093.100000	787217.000000	2094.400000	2098.200000	2074.600000

In [7]:

```
# Check for missing values in each column
missing_values = data.isnull().sum()

# Display the count of missing values in each column
missing_values
```

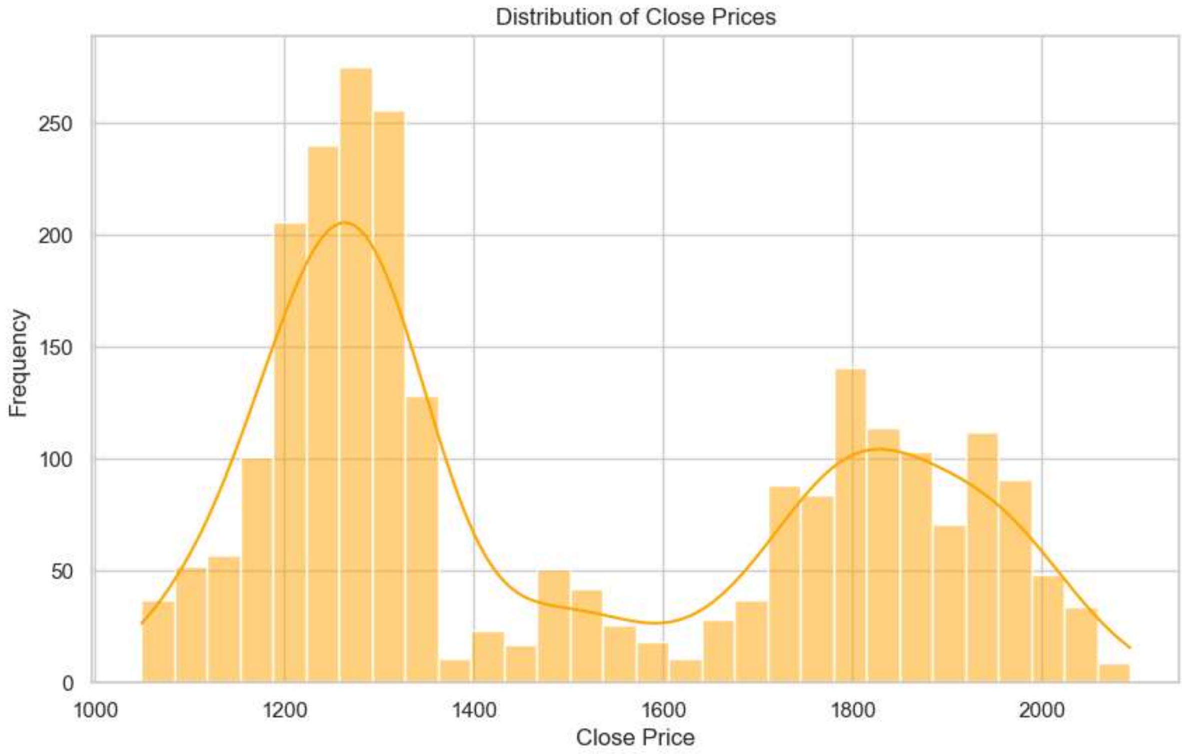
Out[7]:

```
Unnamed: 0    0
Date          0
Close         0
Volume        0
Open          0
High          0
Low           0
dtype: int64
```

Distribution of the 'Close' prices

In [25]:

```
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
sns.histplot(data['Close'], bins=30, kde=True, color='orange')
plt.title('Distribution of Close Prices')
plt.xlabel('Close Price')
plt.ylabel('Frequency')
plt.show()
```

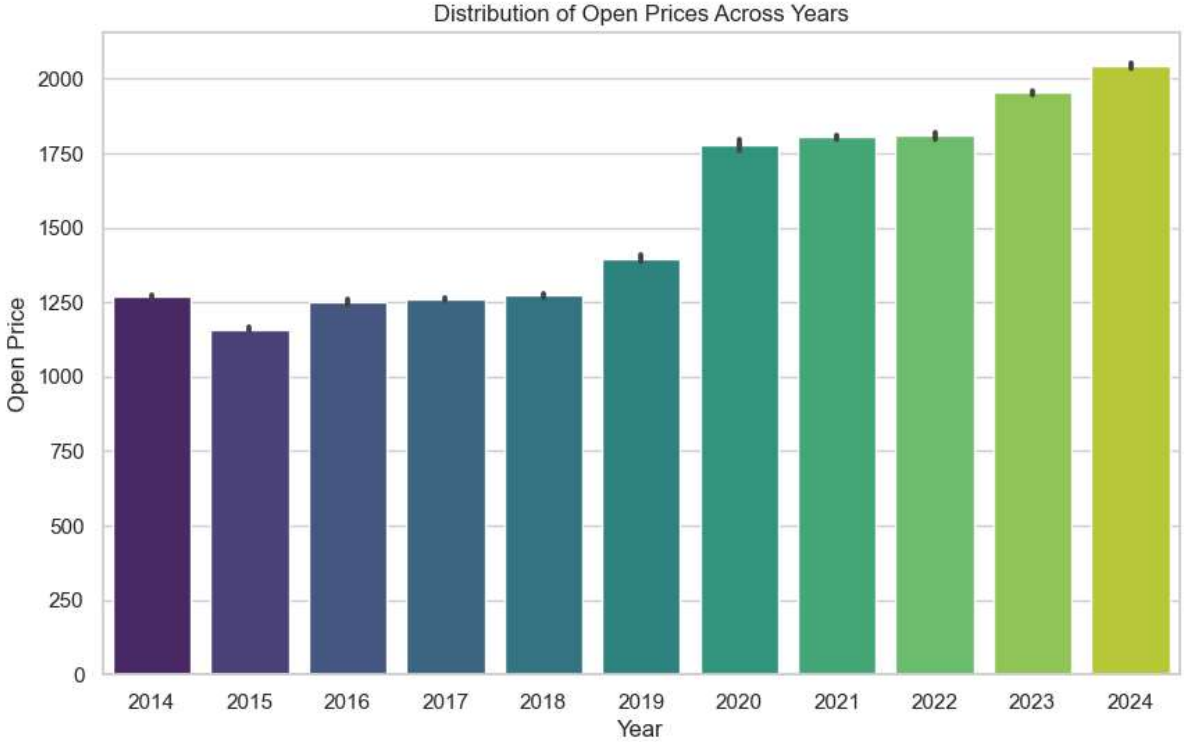


Average 'Close' price for each year

In [23]:
```
# Convert 'Date' column to datetime type
data['Date'] = pd.to_datetime(data['Date'])

# Extracting the year from the 'Date' column
data['Year'] = data['Date'].dt.year

# Calculating the average 'Close' price for each year
average_close_by_year = data.groupby('Year')['Close'].mean().reset_index()

# Plotting the bar plot for average 'Close' price by year
plt.figure(figsize=(10, 6))
sns.barplot(x='Year', y='Close', data=average_close_by_year, color='lightblue')
plt.title('Average Close Price by Year')
plt.xlabel('Year')
plt.ylabel('Average Close Price')
plt.show()
```

Distribution of Open Prices Across Years

In [24]:
```
# Convert 'Date' column to datetime type
data['Date'] = pd.to_datetime(data['Date'])

# Extracting the month from the 'Date' column
data['Year'] = data['Date'].dt.year

# Plotting the bar plot for distribution of 'Open' prices across months
plt.figure(figsize=(10, 6))
sns.barplot(x='Year', y='Open', data=data, palette='viridis')
plt.title('Distribution of Open Prices Across Years')
plt.xlabel('Year')
plt.ylabel('Open Price')
plt.show()
```

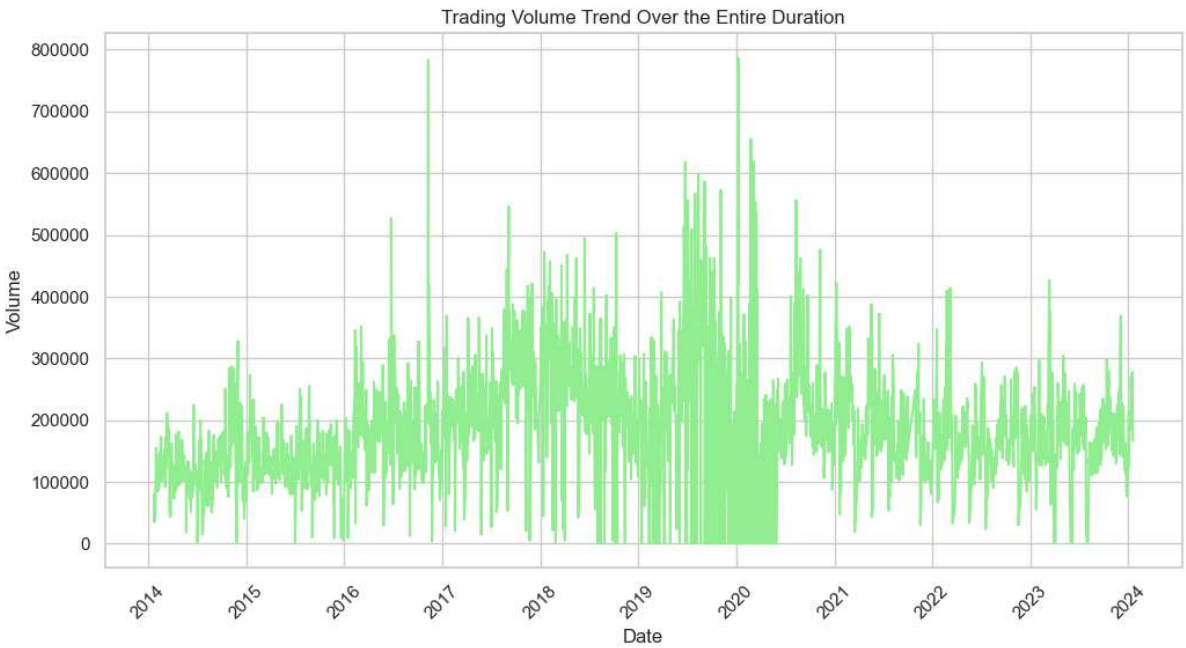


'Open' prices compare to 'Close' prices on average

In [26]:
```
# Calculating the average 'Open' and 'Close' prices
average_prices = data[['Open', 'Close']].mean()

# Plotting the bar plot for average 'Open' and 'Close' prices
plt.figure(figsize=(8, 5))
average_prices.plot(kind='bar', color=['orange', 'skyblue'])
plt.title('Average Open and Close Prices')
plt.ylabel('Average Price')
plt.show()
```

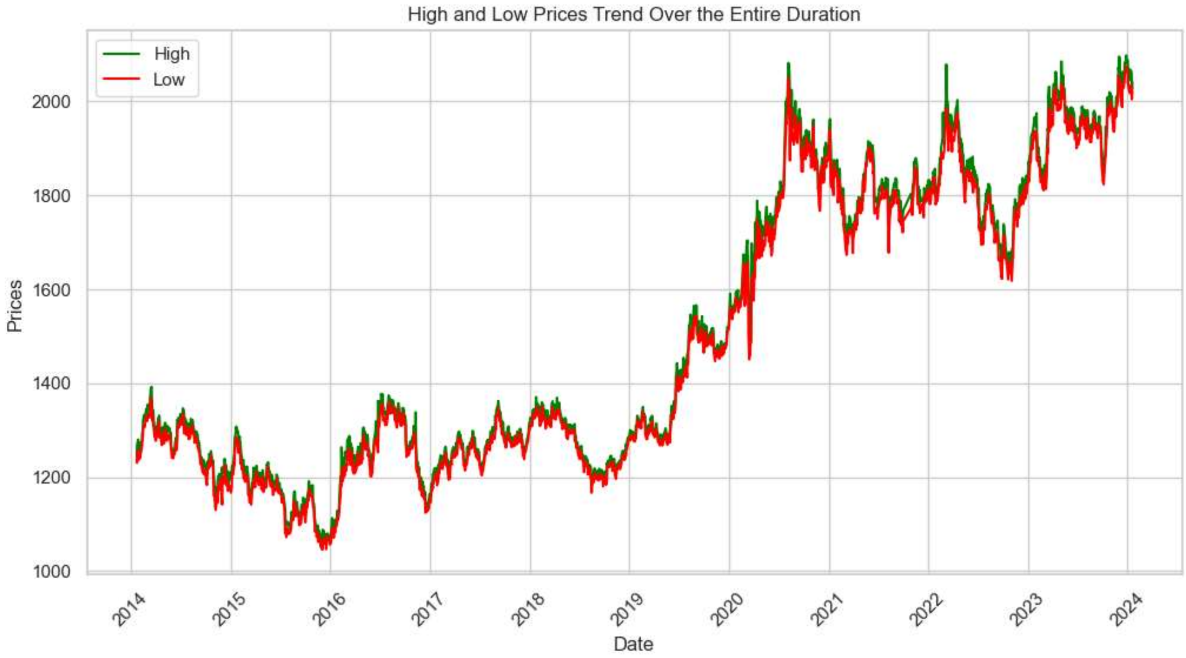
trading volume over time

In [33]:
```
# Plotting the line plot for trading volume over time
plt.figure(figsize=(12, 6))
sns.lineplot(x='Date', y='Volume', data=data, color='lightgreen')
plt.title('Trading Volume Trend Over the Entire Duration')
plt.xlabel('Date')
plt.ylabel('Volume')
plt.xticks(rotation=45)
plt.show()
```



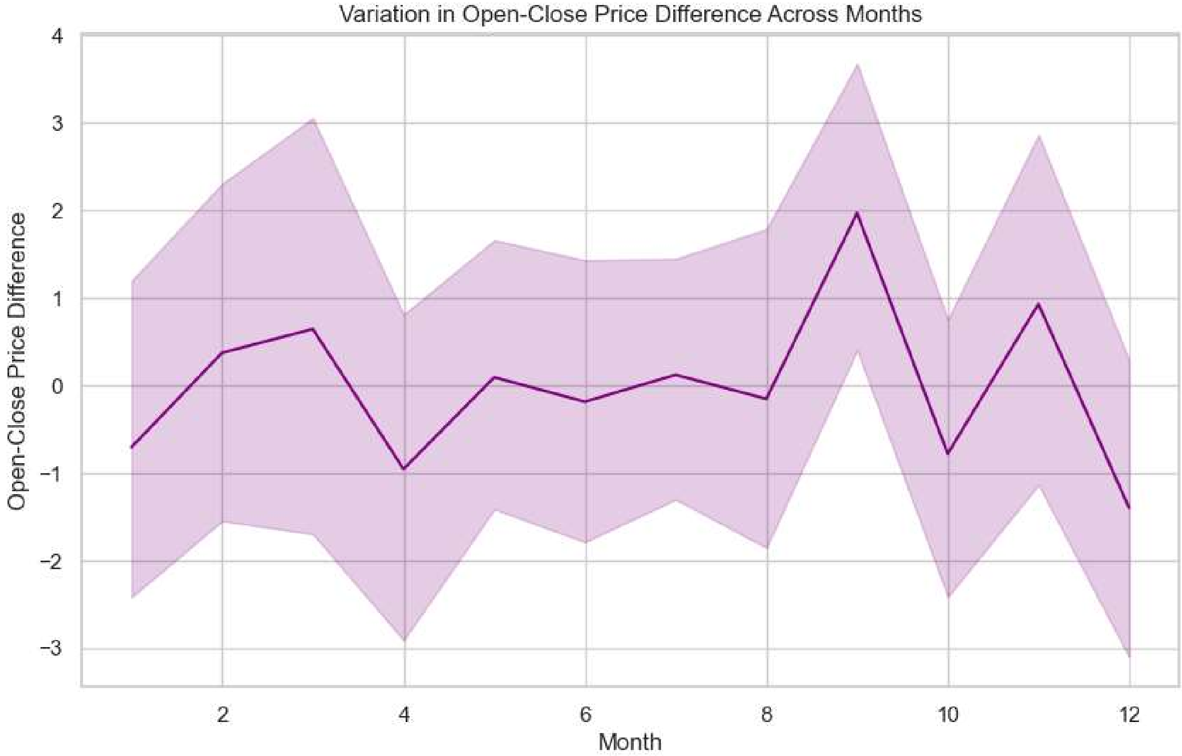
The trend in the 'High' and 'Low' prices over the entire duration

In [35]:
```
# Plotting the line plot for 'High' and 'Low' prices over time
plt.figure(figsize=(12, 6))
sns.lineplot(x='Date', y='High', data=data, label='High', color='green')
sns.lineplot(x='Date', y='Low', data=data, label='Low', color='red')
plt.title('High and Low Prices Trend Over the Entire Duration')
plt.xlabel('Date')
plt.ylabel('Prices')
plt.xticks(rotation=45)
plt.legend()
plt.show()
```



Variation in the difference between 'Open' and 'Close' prices over Months

In [39]:
```
# Calculate the difference between 'Open' and 'Close' prices
data['Open_Close_Difference'] = data['Open'] - data['Close']

# Extracting the month from the 'Date' column
data['Month'] = data['Date'].dt.month

# Plotting the line plot for the difference between 'Open' and 'Close' prices over
plt.figure(figsize=(10, 6))
sns.lineplot(x='Month', y='Open_Close_Difference', data=data, color='purple')
plt.title('Variation in Open-Close Price Difference Across Months')
plt.xlabel('Month')
plt.ylabel('Open-Close Price Difference')
plt.show()
```



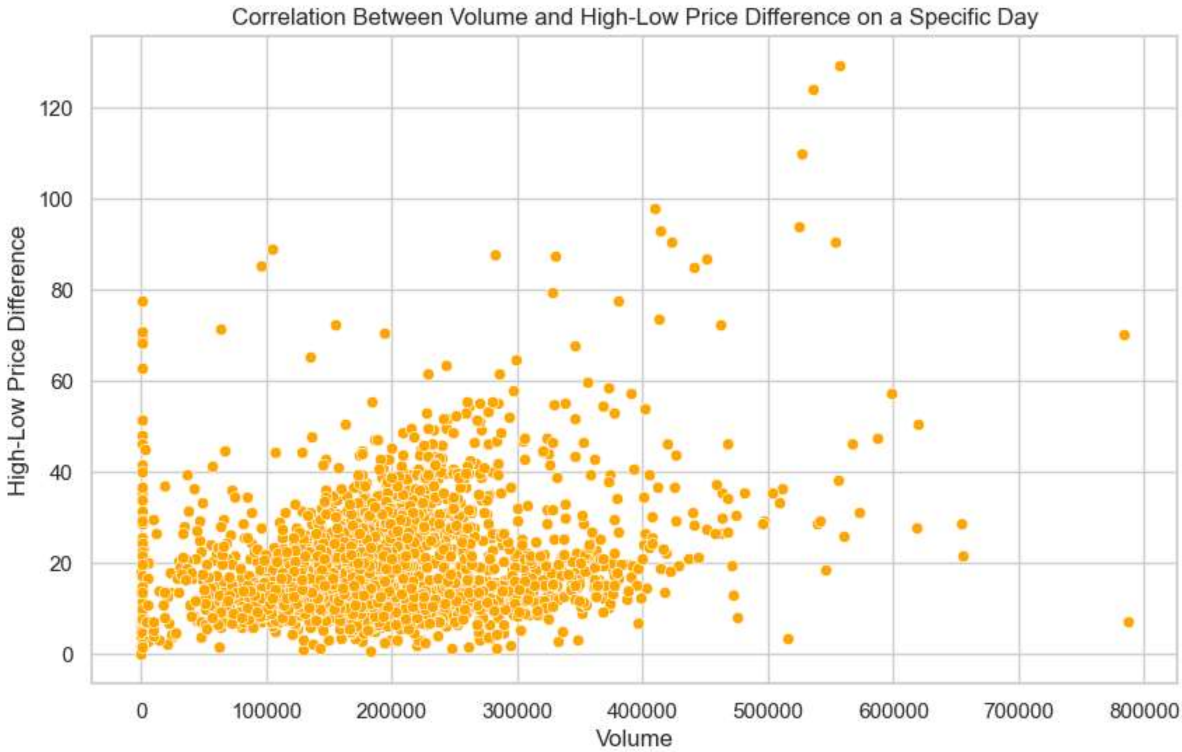
Correlation coefficient between 'High' and 'Low' prices?

In [40]:
```
from scipy.stats import pearsonr

# Calculating the correlation coefficient between 'High' and 'Low' prices
correlation_coefficient, _ = pearsonr(data['High'], data['Low'])
print(f"The correlation coefficient between 'High' and 'Low' prices: {correlation_c
```

```
The correlation coefficient between 'High' and 'Low' prices: 0.9993
```

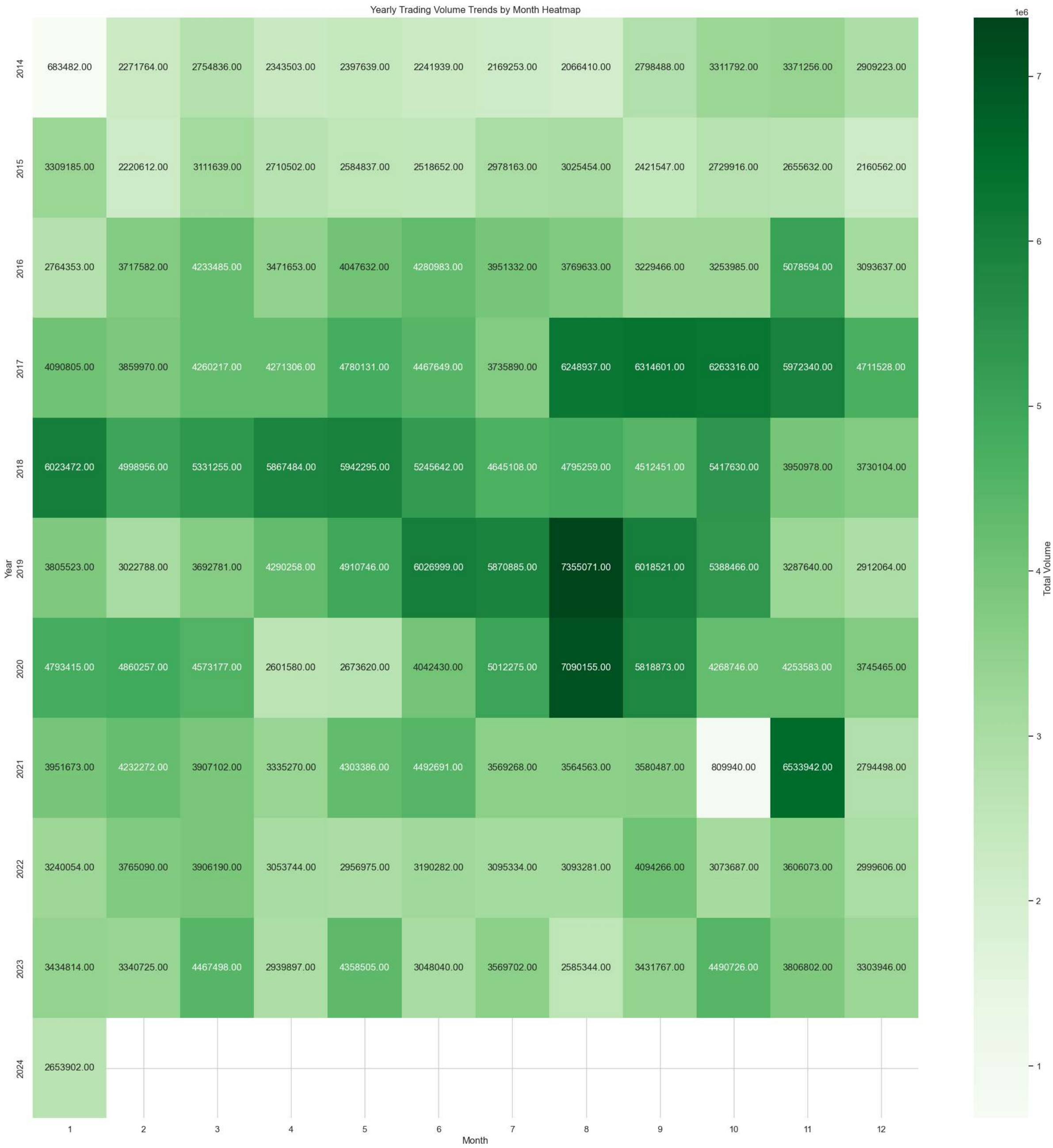
'Volume' in corelation with 'High' and 'Low' prices

In [41]:
```
# Calculate the difference between 'High' and 'Low' prices
data['High_Low_Difference'] = data['High'] - data['Low']

# Plotting the scatterplot for 'Volume' vs the difference between 'High' and 'Low'
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Volume', y='High_Low_Difference', data=data, color='orange')
plt.title('Correlation Between Volume and High-Low Price Difference on a Specific D
plt.xlabel('Volume')
plt.ylabel('High-Low Price Difference')
plt.show()
```



Heatmap of trading volume trends monthly

In [42]:
```
# Creating a pivot table for yearly trading volume trends by month
yearly_volume_trends = data.pivot_table(index='Year', columns='Month', values='Volu

# Creating a heatmap of yearly trading volume trends by month
plt.figure(figsize=(25, 25))
sns.heatmap(yearly_volume_trends, cmap='Greens', annot=True, fmt=".2f", cbar_kws={'
plt.title('Yearly Trading Volume Trends by Month Heatmap')
plt.show()
```


Yearly Trading Volume Trends by Month Heatmap
2014
683482.00
2271764.00
2754836.00
2343503.00
2397639.00
2241939.00
2169253.00
2066410.00
2798488.00
3311792.00
3371256.00
2909223.00
2015
3309185.00
2220612.00
3111639.00
2710502.00
2584837.00
2518652.00
2978163.00
3025454.00
2421547.00
2729916.00
2655632.00
2160562.00
2016
2764353.00
3717582.00
4233485.00
3471653.00
4047632.00
4280983.00
3951332.00
3769633.00
3229466.00
3253985.00
5078594.00
3093637.00
2017
4090805.00
3859970.00
4260217.00
4271306.00
4780131.00
4467649.00
3735890.00
6248937.00
6314601.00
6263316.00
5972340.00
4711528.00
2018
6023472.00
4998956.00
5331255.00
5867484.00
5942295.00
5245642.00
4645108.00
4795259.00
4512451.00
5417630.00
3950978.00
3730104.00
2019
3805523.00
3022788.00
3692781.00
4290258.00
4910746.00
6026999.00
5870885.00
7355071.00
6018521.00
5388466.00
3287640.00
2912064.00
2020
4793415.00
4860257.00
4573177.00
2601580.00
2673620.00
4042430.00
5012275.00
7090155.00
5818873.00
4268746.00
4253583.00
3745465.00
2021
3951673.00
4232272.00
3907102.00
3335270.00
4303386.00
4492691.00
3569268.00
3564563.00
3580487.00
809940.00
6533942.00
2794498.00
2022
3240054.00
3765090.00
3906190.00
3053744.00
2956975.00
3190282.00
3095334.00
3093281.00
4094266.00
3073687.00
3606073.00
2999606.00
2023
3434814.00
3340725.00
4467498.00
2939897.00
4358505.00
3048040.00
3569702.00
2585344.00
3431767.00
4490726.00
3806802.00
3303946.00
2024
2653902.00
Year
1
2
3
4
5
6
7
8
9
10
11
12
Month
1e6
7
6
5
4
3
2
1
Total Volume