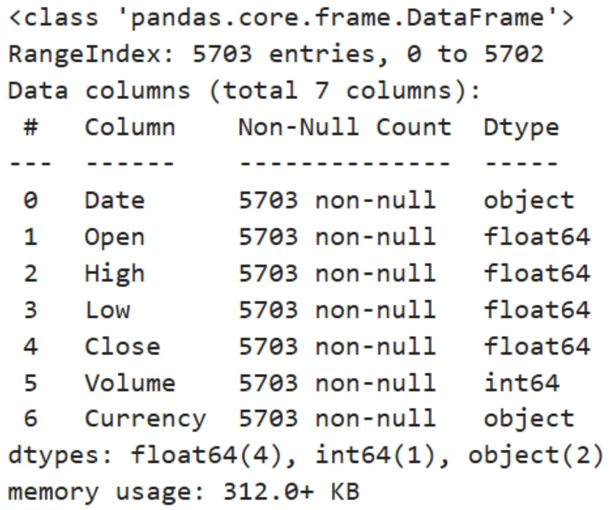
# Gold Price Data Analysis and Predicting Gold Price Using Machine Learning Regression Model

I. **Data Source:** We collected our dataset from Kaggle. The dataset contains 5703 entries with 7 features related to gold trading.

**Data Source Link:** https://www.kaggle.com/datasets/faisaljanjua0555/daily-gold-price-historical-dataset

## II. Data description:

a) **No. of features and their types:** Total number of features: 7, including both numerical (‘Open’, ‘High’, ‘Low’, ‘Close’ and ‘Volume’) and categorical (‘Date’, ‘Currency’) variables.

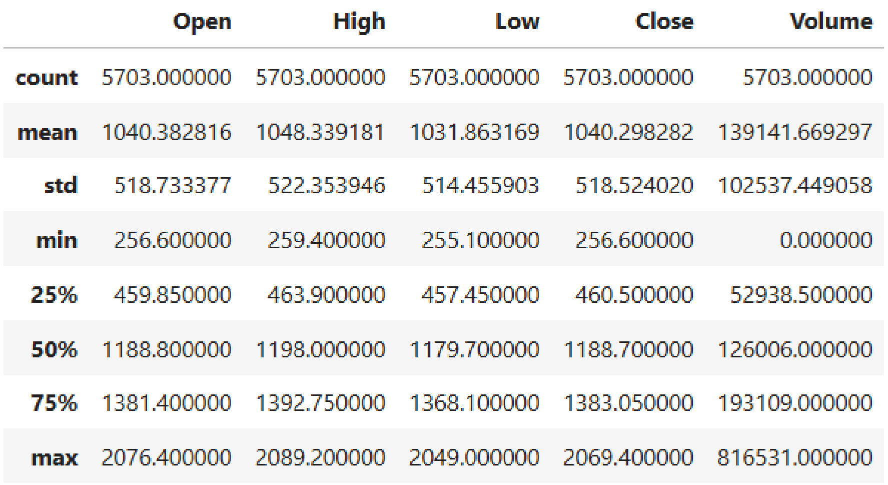


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5703 entries, 0 to 5702
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Date      5703 non-null   object 
 1   Open      5703 non-null   float64
 2   High      5703 non-null   float64
 3   Low       5703 non-null   float64
 4   Close     5703 non-null   float64
 5   Volume    5703 non-null   int64  
 6   Currency  5703 non-null   object 
dtypes: float64(4), int64(1), object(2)
memory usage: 312.0+ KB
```

b) **Response variable:** ‘Close’ price of gold, predicting using a regression approach.

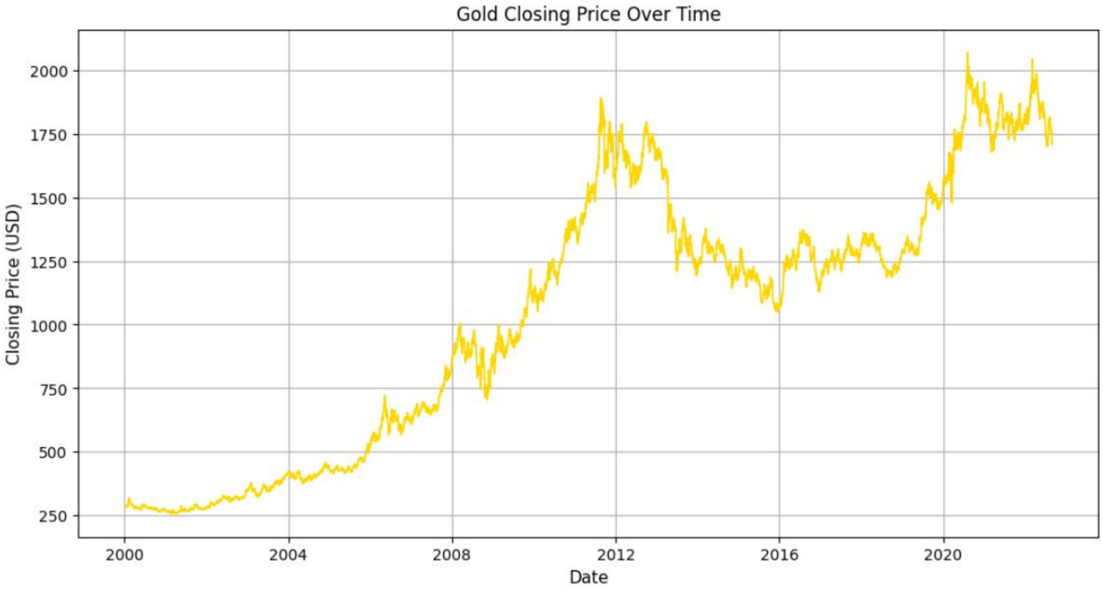
## III) Exploratory Data Analysis and Visualization:

### 1. Show basic statistics of numerical features.



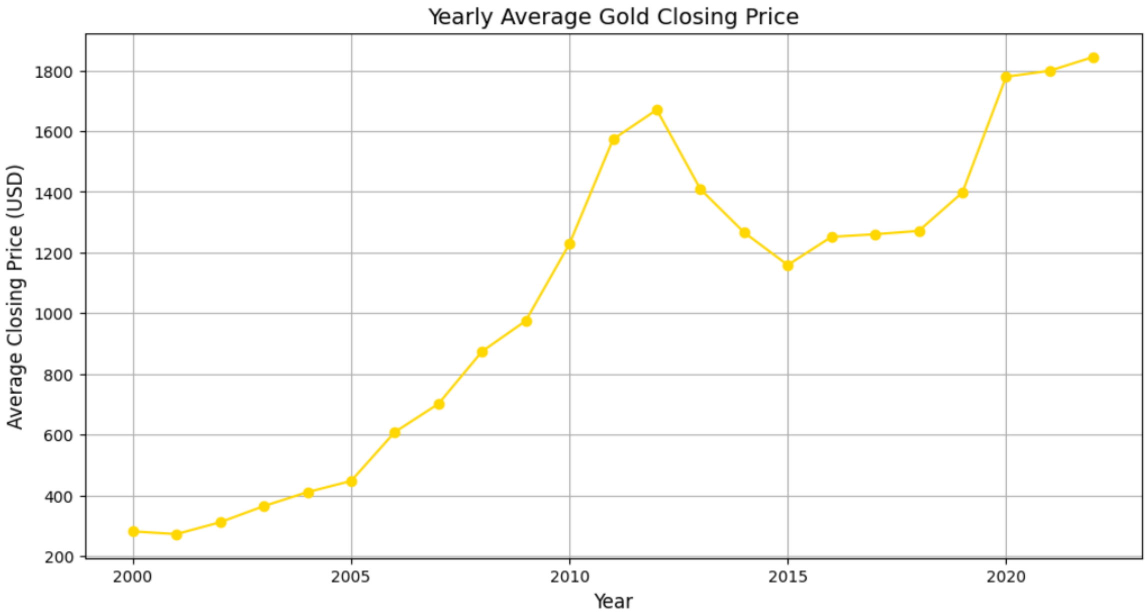
| | Open | High | Low | Close | Volume |
|---|---|---|---|---|---|
| **count** | 5703.000000 | 5703.000000 | 5703.000000 | 5703.000000 | 5703.000000 |
| **mean** | 1040.382816 | 1048.339181 | 1031.863169 | 1040.298282 | 139141.669297 |
| **std** | 518.733377 | 522.353946 | 514.455903 | 518.524020 | 102537.449058 |
| **min** | 256.600000 | 259.400000 | 255.100000 | 256.600000 | 0.000000 |
| **25%** | 459.850000 | 463.900000 | 457.450000 | 460.500000 | 52938.500000 |
| **50%** | 1188.800000 | 1198.000000 | 1179.700000 | 1188.700000 | 126006.000000 |
| **75%** | 1381.400000 | 1392.750000 | 1368.100000 | 1383.050000 | 193109.000000 |
| **max** | 2076.400000 | 2089.200000 | 2049.000000 | 2069.400000 | 816531.000000 |

2. **How has the closing price of gold changed over time? (Line plot of 'Date' vs. 'Close')**

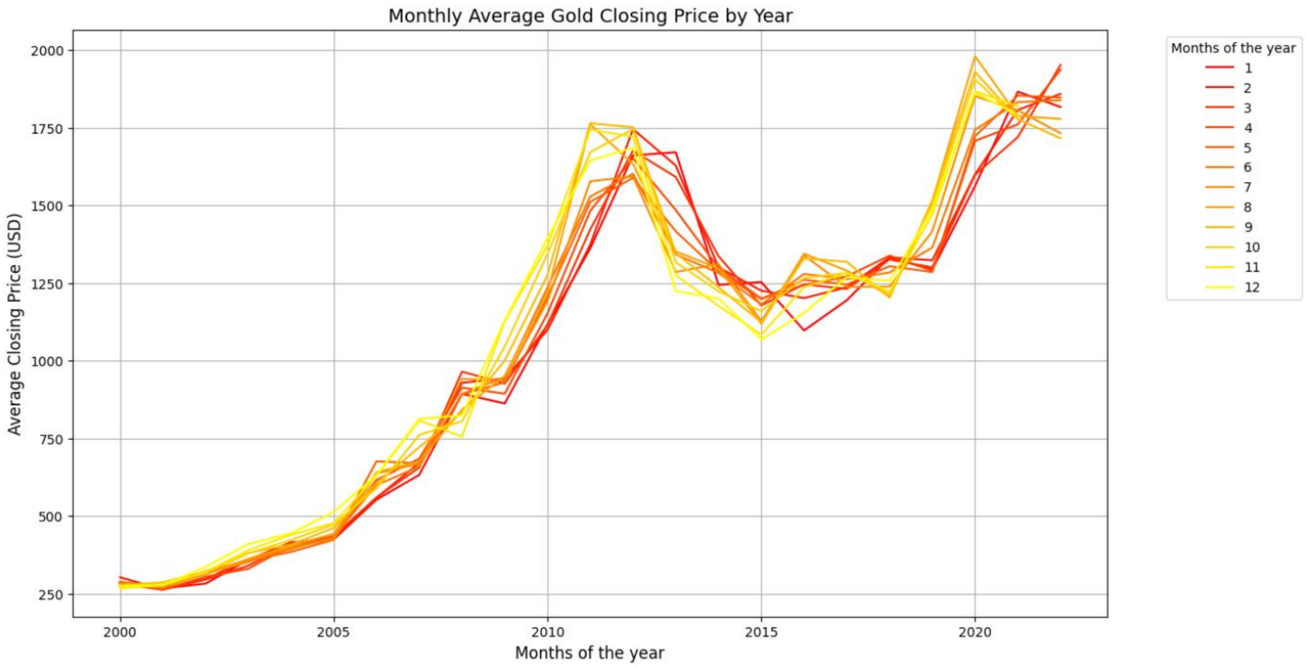


The line plot shows the closing price of gold (in USD) over time. From 2000 to 2012, there was a steady rise, with a sharp peak due to the 2008 financial crisis, as investors sought gold as a safe-haven asset. Between 2012 and 2016, the price declined as global economies recovered, reducing demand for gold. However, from 2016 onward, prices began to rise again, peaking around 2020 during the COVID-19 pandemic, driven by economic uncertainty. Post-2020, the price shows volatility, reflecting varying global economic conditions. **This line chart effectively highlights gold's long-term value as a hedge during economic crises.**

3. **What are the trends in gold prices over different periods (e.g., yearly or monthly)? (Aggregate and plot 'Close' by year/month)**

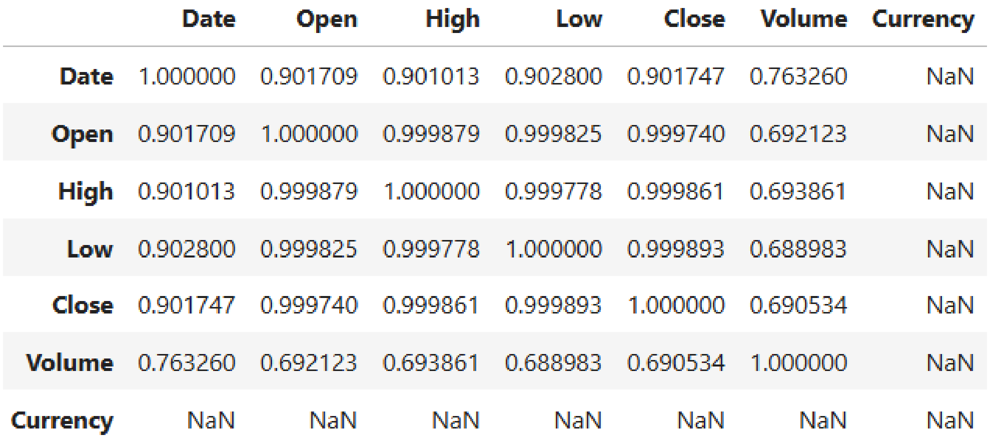


The plot shows a steady rise in gold's yearly average price from 2000 to 2012, peaking due to economic uncertainties like the 2008 financial crisis. Prices declined from 2012 to 2015 as economies recovered but surged again from 2016, reaching a peak around 2020 during the COVID-19 pandemic. **This highlights gold's value as a safe-haven asset during crises.**



This is a line chart that visualizes the monthly average closing prices of gold over multiple years. Each line represents a specific month (e.g., January, February, etc.), showing the trend for that month across the years. The chart highlights **a steady overall increase in gold prices with notable peaks around 2011 and 2020**, and it also **indicates minor seasonal variations in gold prices between different months.**

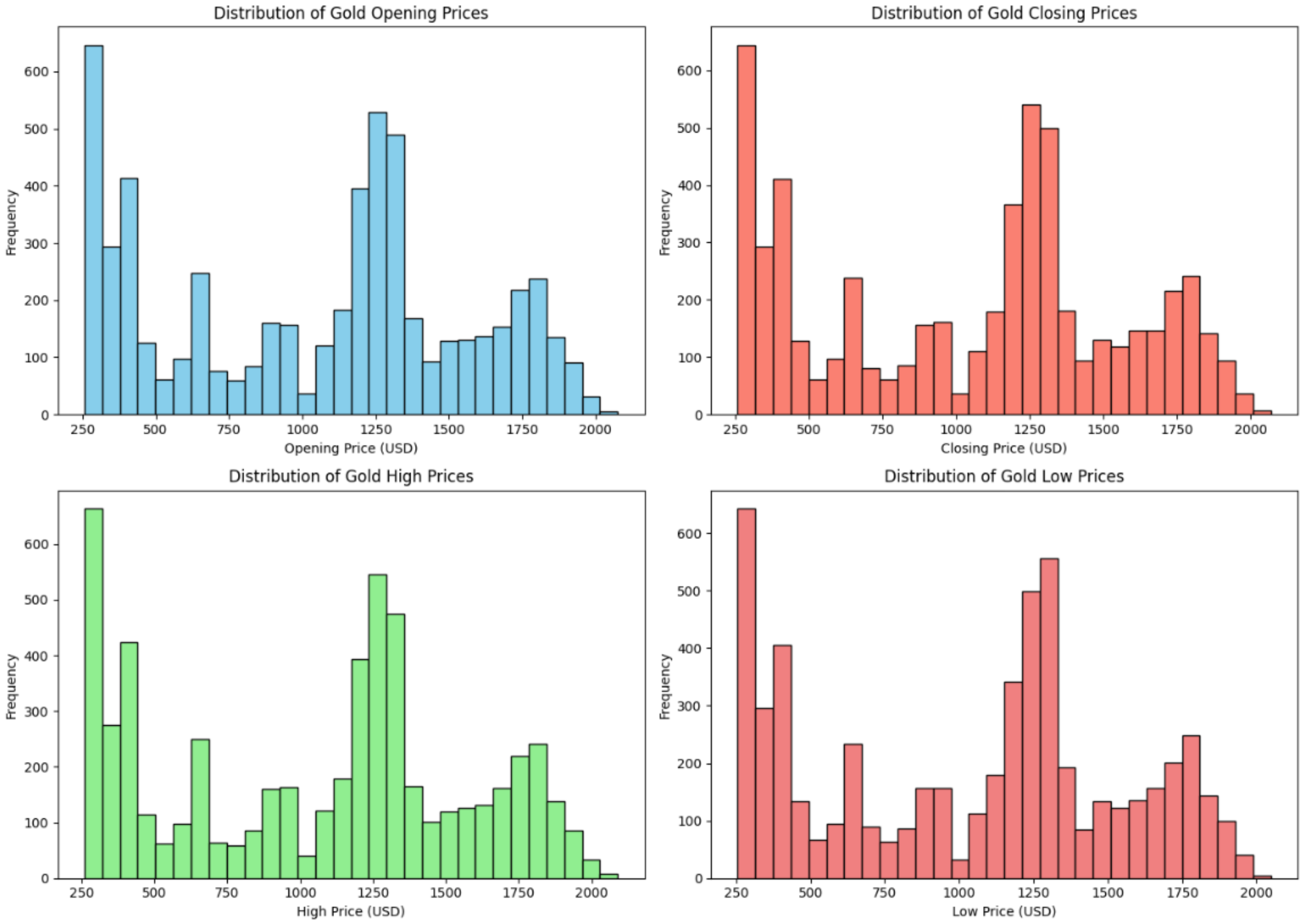
### 4. How do the open, high, low, and close prices correlate with each other?



| | Date | Open | High | Low | Close | Volume | Currency |
|---|---|---|---|---|---|---|---|
| **Date** | 1.000000 | 0.901709 | 0.901013 | 0.902800 | 0.901747 | 0.763260 | NaN |
| **Open** | 0.901709 | 1.000000 | 0.999879 | 0.999825 | 0.999740 | 0.692123 | NaN |
| **High** | 0.901013 | 0.999879 | 1.000000 | 0.999778 | 0.999861 | 0.693861 | NaN |
| **Low** | 0.902800 | 0.999825 | 0.999778 | 1.000000 | 0.999893 | 0.688983 | NaN |
| **Close** | 0.901747 | 0.999740 | 0.999861 | 0.999893 | 1.000000 | 0.690534 | NaN |
| **Volume** | 0.763260 | 0.692123 | 0.693861 | 0.688983 | 0.690534 | 1.000000 | NaN |
| **Currency** | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

This is a correlation matrix, showing the relationship between numerical variables in the dataset. The **Open, High, Low, and Close prices are highly correlated** (values near 1), indicating they move together consistently. The **Volume has a weaker correlation with price variables**, suggesting that trading volume does not strongly influence gold prices. The Currency column has ‘NaN’ values because it was likely converted to numerical form but doesn't have meaningful numerical relationships with other variables.

### 5. What is the overall distribution of gold’s opening, closing, high, and low prices? (Histogram for 'Open', 'Close', 'High', 'Low')



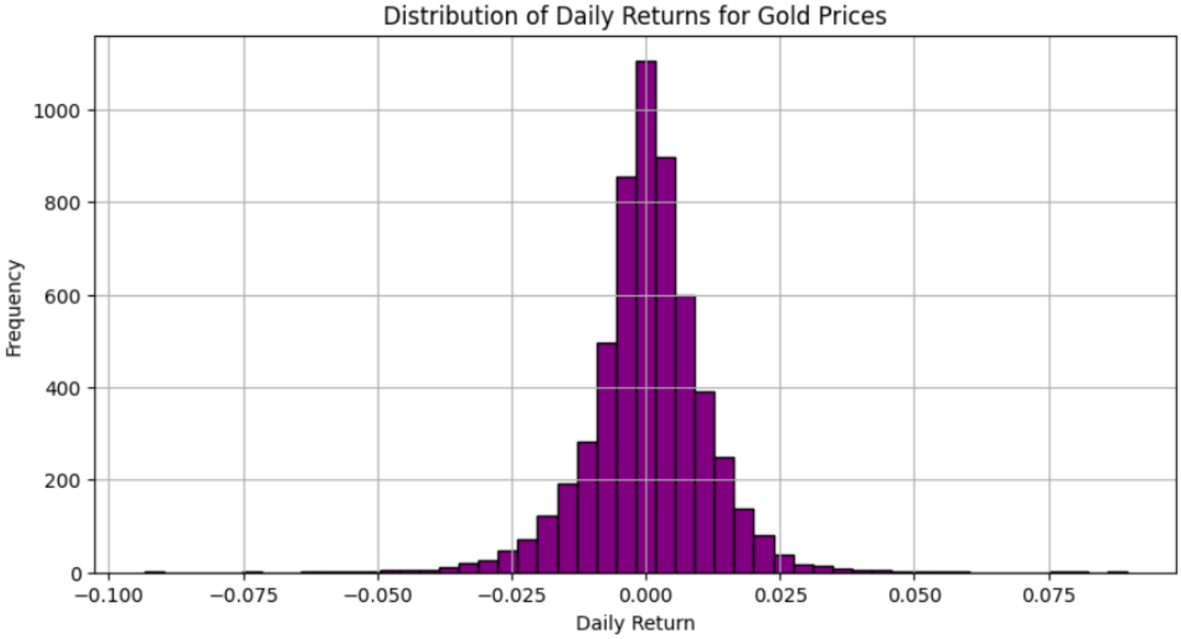
This set of histograms illustrates the distribution of gold's Open, Close, High, and Low prices:

a) **Opening and Closing Prices:** Both distributions are similar and mostly centered around the same price ranges, indicating stability between market open and close.

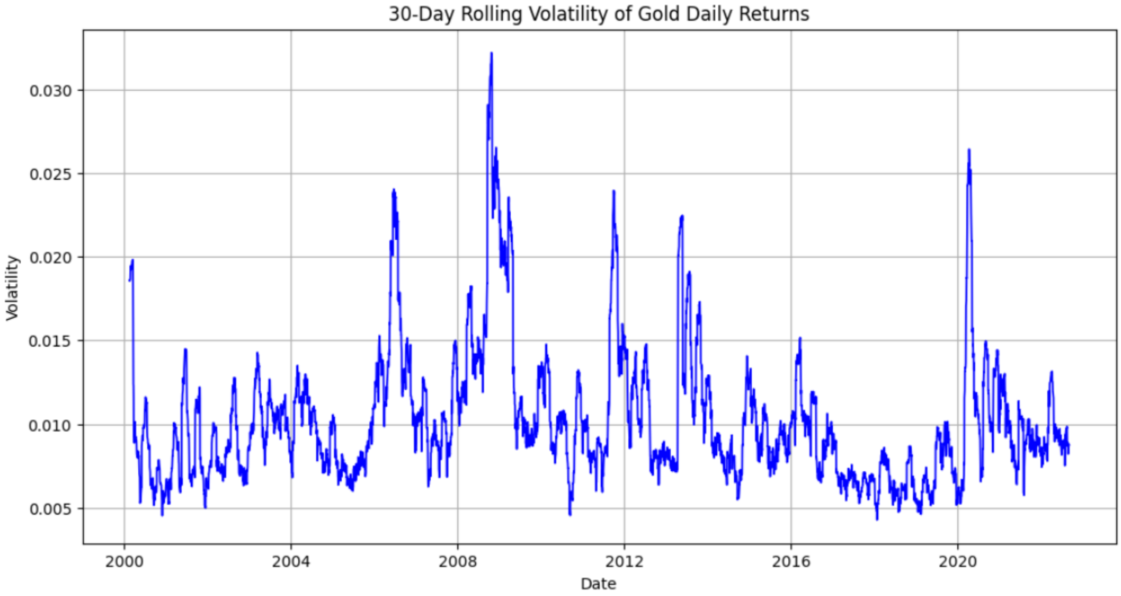
b) **High and Low Prices:** These distributions also overlap with the Opening and Closing prices, confirming that daily fluctuations (Highs and Lows) are generally within a limited range.

All distributions are **positively skewed** (with a long tail towards higher values), reflecting the gradual rise in gold prices over the years. This visualization highlights the overall pricing consistency of gold with upward trends over time.

6. **How volatile is the gold price over time? (Calculate daily returns and plot distribution or volatility over time)**

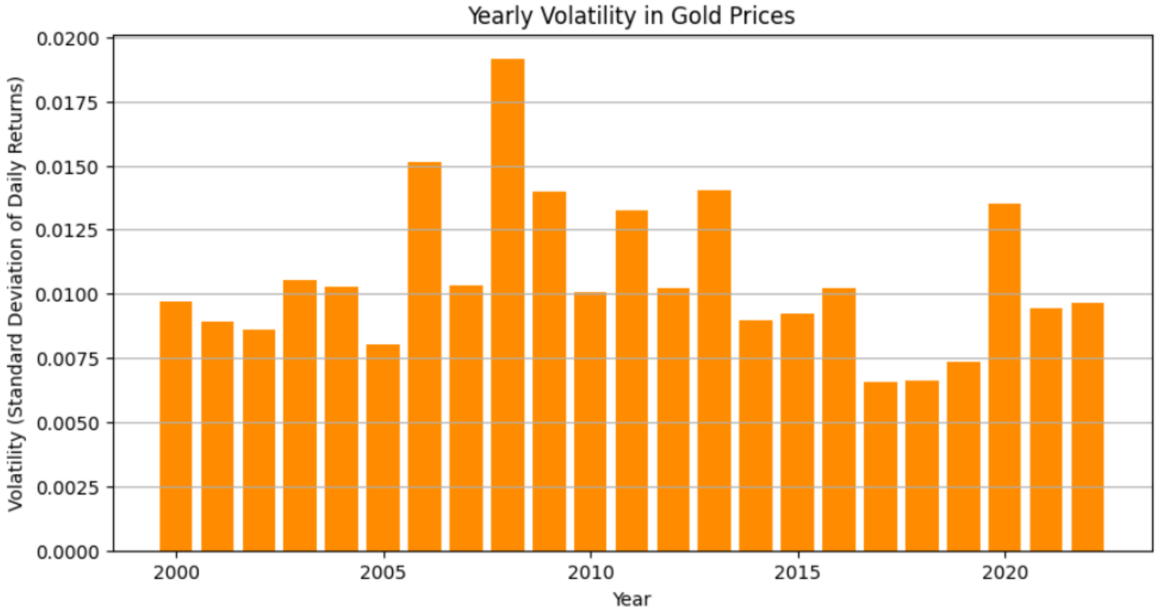


The histogram shows the frequency distribution of daily returns for gold prices. Most returns are centered around zero, with a small spread indicating relatively low daily volatility. The presence of tails suggests occasional days with higher gains or losses.

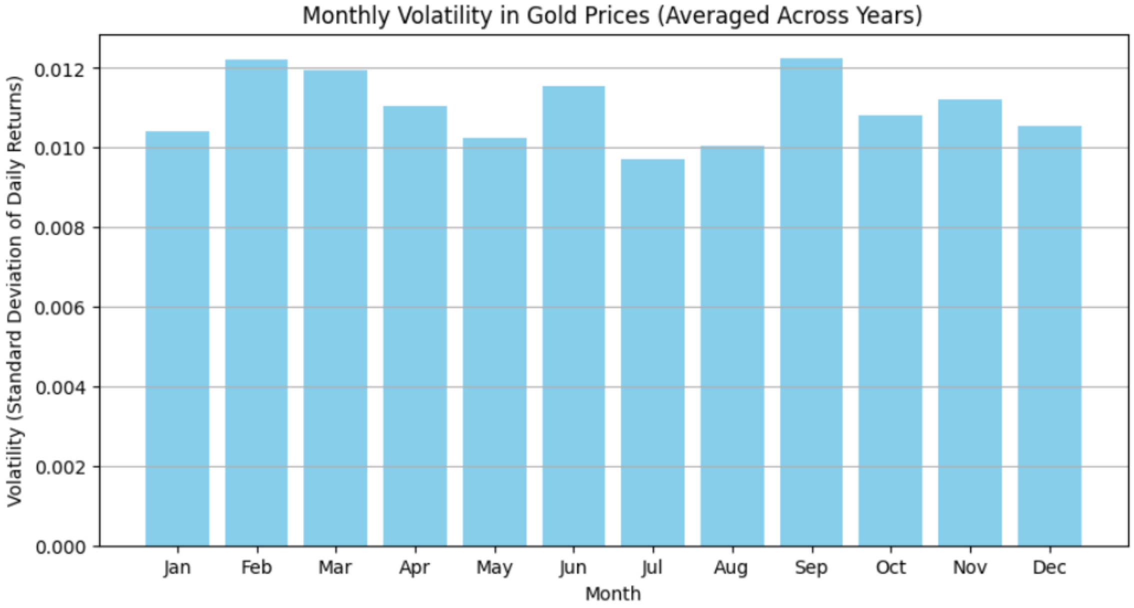


The Rolling Volatility Line Plot tracks the 30-day rolling standard deviation of daily returns, which measures price volatility over time. Periods of high volatility correspond to significant economic or geopolitical events (e.g., financial crises or global uncertainty). This visualization demonstrates how gold's stability varies over time, with spikes indicating short-term market turbulence.

### 7. Are there specific years or months where the price volatility was higher than others?

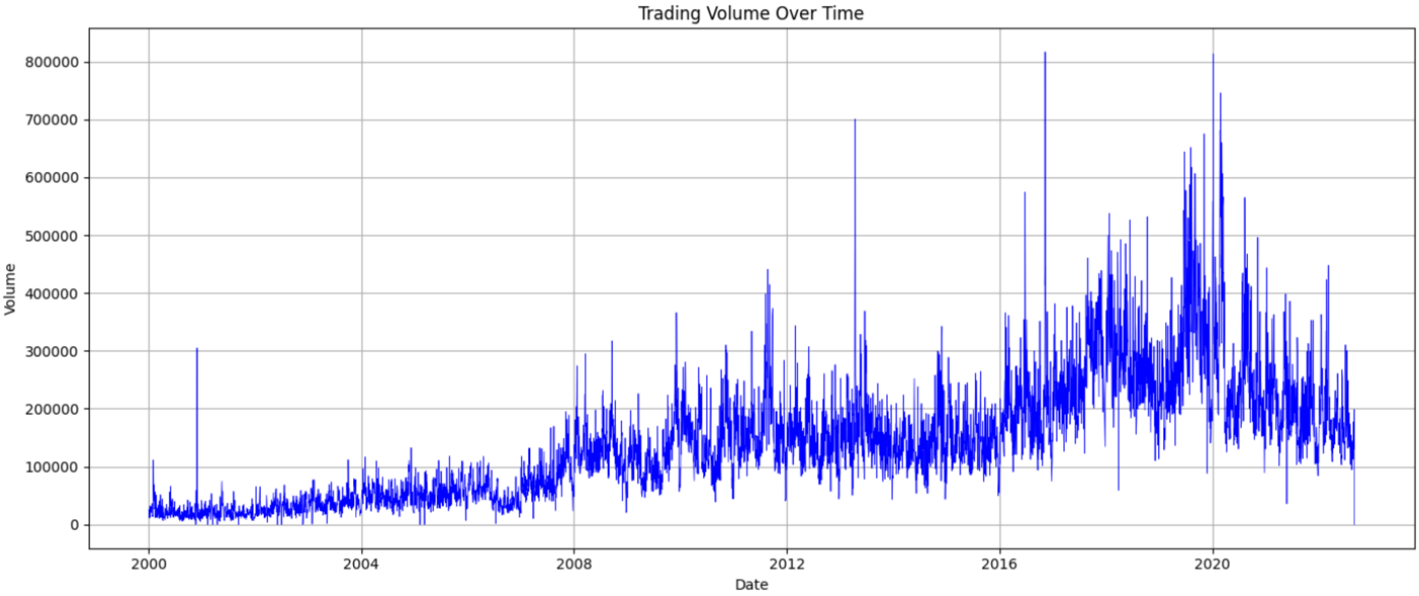


**Yearly Volatility in Gold Prices:** Gold price volatility peaked significantly during 2008-2010, coinciding with the Global Financial Crisis, reflecting economic uncertainty. After 2010, volatility gradually decreased but remained elevated until around 2015. A slight uptick is observed in 2020, likely due to the COVID-19 pandemic. **High volatility years indicate turbulent market conditions where gold is seen as a safe-haven asset, while lower volatility reflects stable economic periods.**



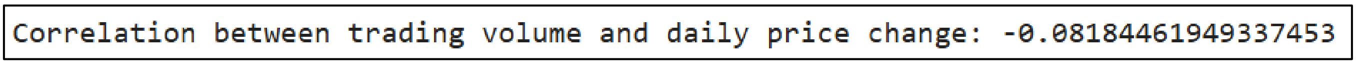
**Monthly Volatility in Gold Prices:** Volatility varies by month, with **February** and **September showing the highest average fluctuations.** September's increase could be driven by seasonal demand for gold, such as festivals and weddings, while February might reflect market rebalancing. June and July exhibit lower volatility, likely indicating a quieter trading period. **Monthly patterns highlight how seasonal and demand-driven factors impact gold price stability.**

## 8. What is the trend in trading volume over time? (Line plot of 'Date' vs. 'Volume')



The plot shows that gold trading volume was relatively low and stable in the early 2000s, with occasional spikes. It increased significantly around 2008-2012, peaking during the Global Financial Crisis as investors turned to gold amid economic uncertainty. From 2012 to 2020, trading volume remained high but fluctuated, driven by market events. Post-2020, there is a noticeable decline, likely reflecting stabilization in gold demand after the COVID-19 pandemic. Overall, **trading volume aligns closely with major global economic trends.**

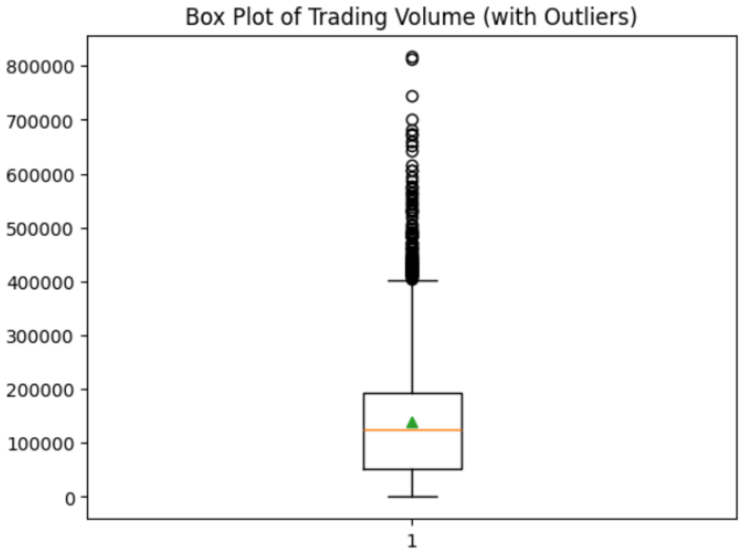
## 9. How does trading volume correlate with changes in gold prices?



```
Correlation between trading volume and daily price change: -0.0818446194933745З
```

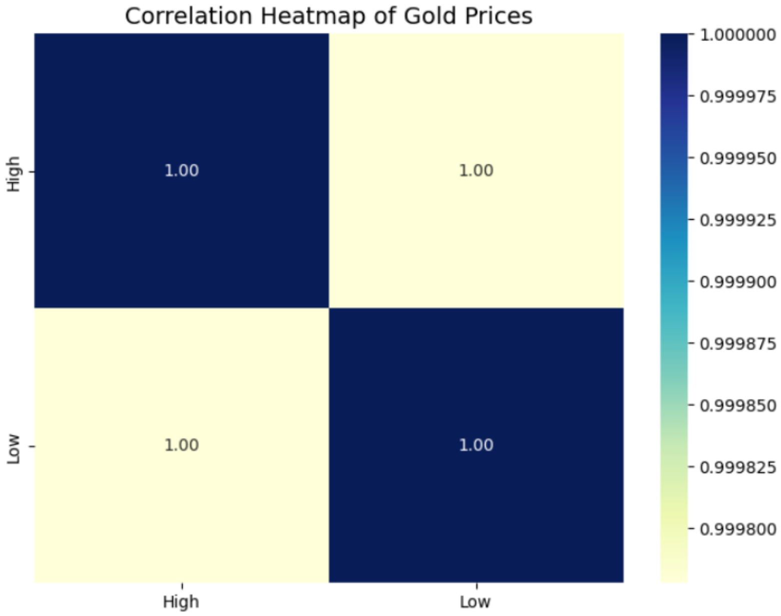
The correlation between trading volume and daily price changes is -0.0818, which is very close to zero. A negative but weak correlation suggests that as trading volume increases, daily price changes tend to decrease slightly, but the relationship is not strong. This implies that trading volume and price changes are mostly independent of each other in this dataset. In summary, **trading volume does not strongly predict gold price fluctuations on a daily basis.**

### 10. Are there any days with unusually high or low trading volumes? (Identify outliers in 'Volume')



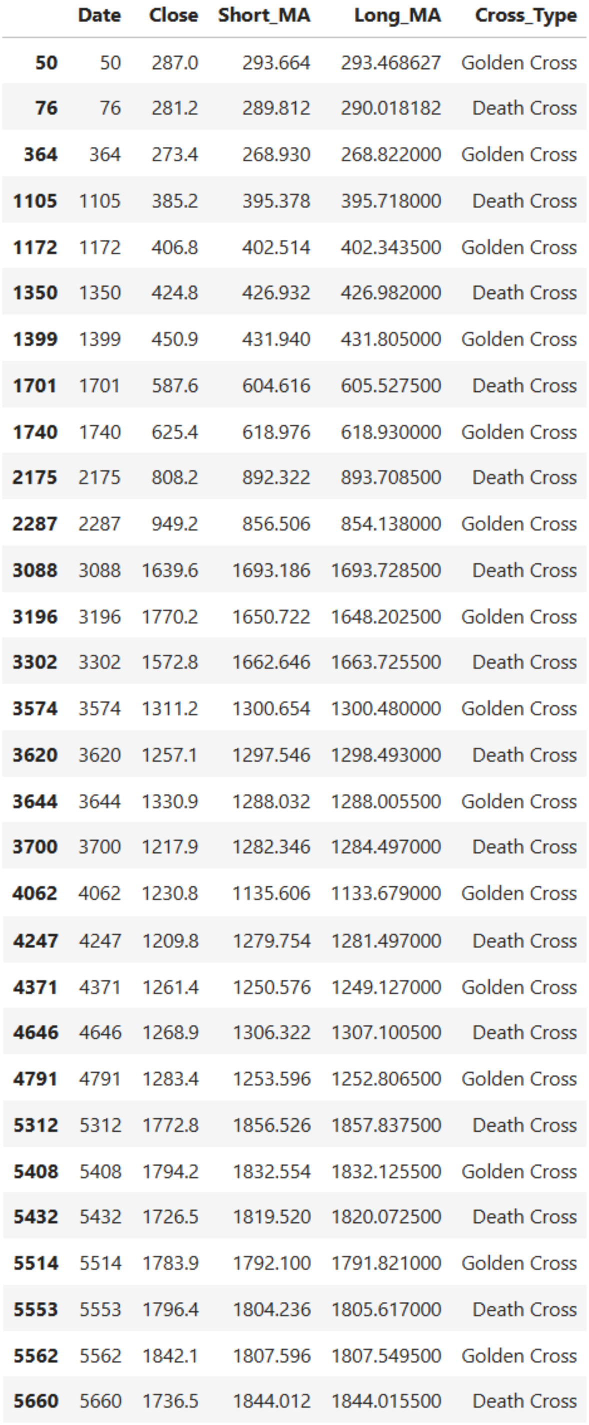
The box plot shows that most trading volumes are concentrated between 100,000 and 150,000, with the median (orange line) in this range. The mean (green triangle) is slightly higher, indicating a skew caused by numerous outliers with unusually high volumes. These outliers, above 400,000, represent days of heightened trading activity, possibly due to major market events. Overall, **trading volume is mostly stable, with occasional extreme spikes.**

### 11. How closely does the high price track with the low price? (plot using heatmap)



**The High and Low prices of gold have a very strong positive correlation (close to 1.00),** indicating they move closely together within a trading day. The heatmap visually confirms this strong relationship with a high correlation value. This is expected, as daily price ranges typically follow consistent patterns.

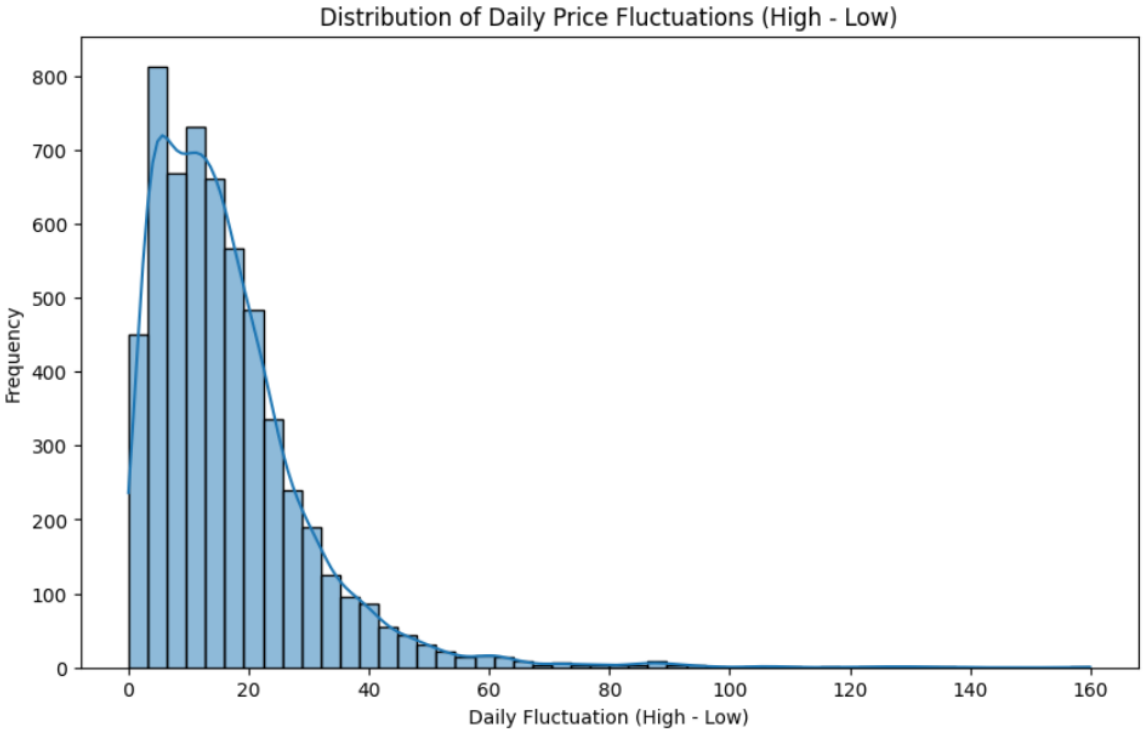
**12. When are the golden cross (short-term average crosses above long-term average) and death cross (opposite) points observed?**

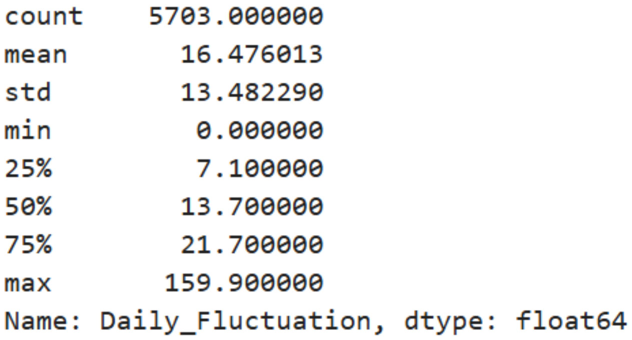


| | Date | Close | Short_MA | Long_MA | Cross_Type |
|---|---|---|---|---|---|
| **50** | 50 | 287.0 | 293.664 | 293.468627 | Golden Cross |
| **76** | 76 | 281.2 | 289.812 | 290.018182 | Death Cross |
| **364** | 364 | 273.4 | 268.930 | 268.822000 | Golden Cross |
| **1105** | 1105 | 385.2 | 395.378 | 395.718000 | Death Cross |
| **1172** | 1172 | 406.8 | 402.514 | 402.343500 | Golden Cross |
| **1350** | 1350 | 424.8 | 426.932 | 426.982000 | Death Cross |
| **1399** | 1399 | 450.9 | 431.940 | 431.805000 | Golden Cross |
| **1701** | 1701 | 587.6 | 604.616 | 605.527500 | Death Cross |
| **1740** | 1740 | 625.4 | 618.976 | 618.930000 | Golden Cross |
| **2175** | 2175 | 808.2 | 892.322 | 893.708500 | Death Cross |
| **2287** | 2287 | 949.2 | 856.506 | 854.138000 | Golden Cross |
| **3088** | 3088 | 1639.6 | 1693.186 | 1693.728500 | Death Cross |
| **3196** | 3196 | 1770.2 | 1650.722 | 1648.202500 | Golden Cross |
| **3302** | 3302 | 1572.8 | 1662.646 | 1663.725500 | Death Cross |
| **3574** | 3574 | 1311.2 | 1300.654 | 1300.480000 | Golden Cross |
| **3620** | 3620 | 1257.1 | 1297.546 | 1298.493000 | Death Cross |
| **3644** | 3644 | 1330.9 | 1288.032 | 1288.005500 | Golden Cross |
| **3700** | 3700 | 1217.9 | 1282.346 | 1284.497000 | Death Cross |
| **4062** | 4062 | 1230.8 | 1135.606 | 1133.679000 | Golden Cross |
| **4247** | 4247 | 1209.8 | 1279.754 | 1281.497000 | Death Cross |
| **4371** | 4371 | 1261.4 | 1250.576 | 1249.127000 | Golden Cross |
| **4646** | 4646 | 1268.9 | 1306.322 | 1307.100500 | Death Cross |
| **4791** | 4791 | 1283.4 | 1253.596 | 1252.806500 | Golden Cross |
| **5312** | 5312 | 1772.8 | 1856.526 | 1857.837500 | Death Cross |
| **5408** | 5408 | 1794.2 | 1832.554 | 1832.125500 | Golden Cross |
| **5432** | 5432 | 1726.5 | 1819.520 | 1820.072500 | Death Cross |
| **5514** | 5514 | 1783.9 | 1792.100 | 1791.821000 | Golden Cross |
| **5553** | 5553 | 1796.4 | 1804.236 | 1805.617000 | Death Cross |
| **5562** | 5562 | 1842.1 | 1807.596 | 1807.549500 | Golden Cross |
| **5660** | 5660 | 1736.5 | 1844.012 | 1844.015500 | Death Cross |

The table highlights Golden Cross (short-term average crosses above long-term average, bullish) and Death Cross (short-term average crosses below long-term average, bearish) events in the gold price data. Each row shows the date, closing price, and the values of the 50-day and 200-day moving averages during these crossovers. **Golden Crosses (e.g., at index 50, 364, etc.) indicate potential upward trends**, while **Death Crosses (e.g., at index 76, 1105, etc.) signal possible downward trends**. These events are commonly used to identify long-term shifts in market momentum.

### 13. How much does the price fluctuate daily (High - Low), and what is the distribution of this fluctuation?

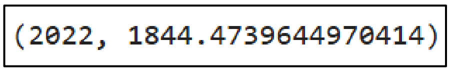




```
count     5703.000000
mean        16.476013
std         13.482290
min          0.000000
25%          7.100000
50%         13.700000
75%         21.700000
max        159.900000
Name: Daily_Fluctuation, dtype: float64
```

The histogram shows the distribution of daily price fluctuations (High - Low) in gold prices, which is right-skewed. Most fluctuations are small, concentrated between 0–20 USD, with **an average of 16.48 USD** and **a median of 13.70 USD**. Most fluctuations fall between **7.10 USD (25th percentile)** and **21.70 USD (75th percentile)**, showing a tight range for typical daily variations. Rare large fluctuations, like the maximum of **159 USD**, represent significant market volatility. **This suggests that daily gold price movements are typically moderate, with occasional spikes due to external factors.**

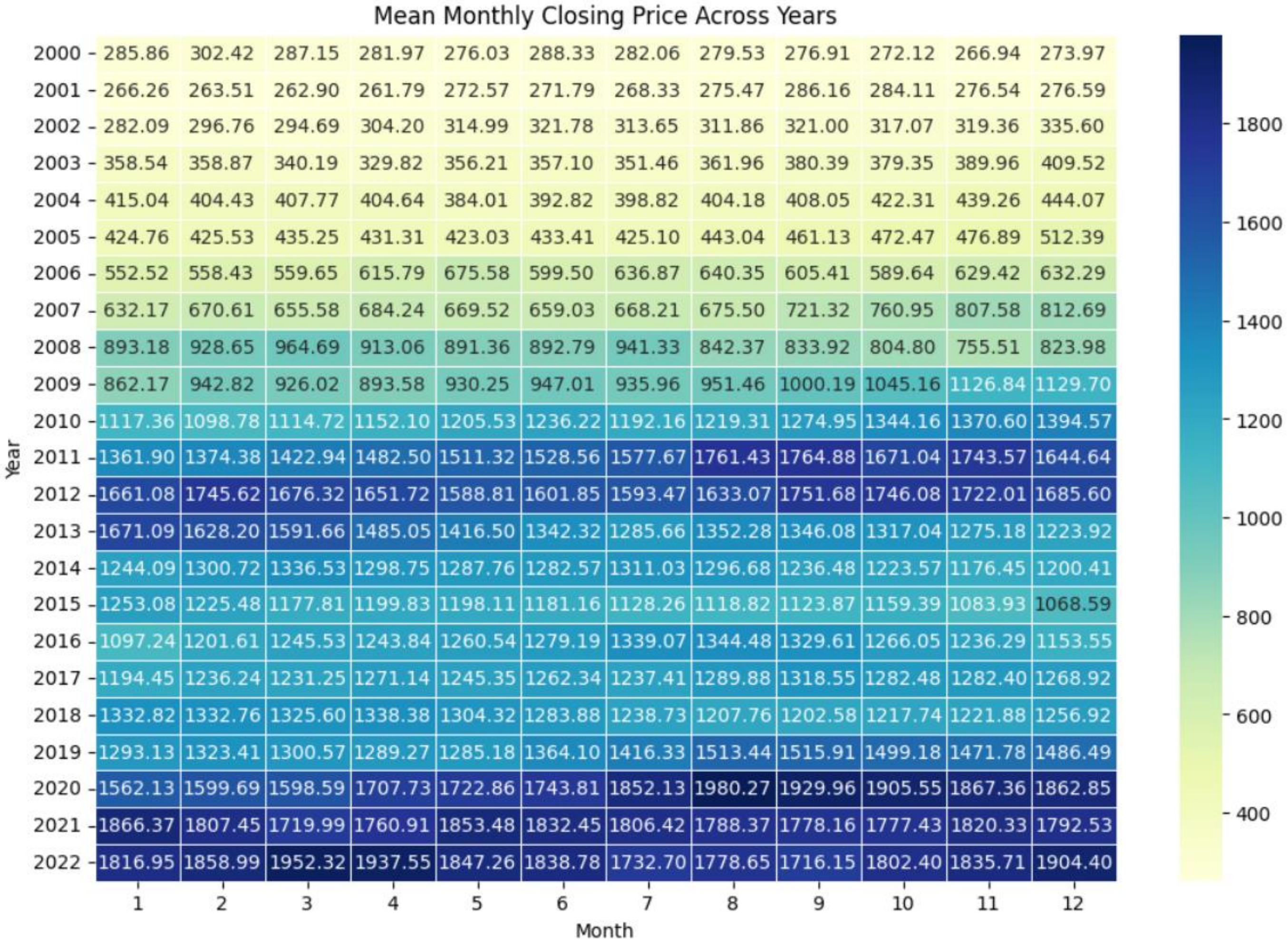
### 14. Which year had the highest average closing price and what is the highest average close value?



```
(2022, 1844.4739644970414)
```

**The year 2022 had the highest average gold closing price, with an average value of 1844.47 USD.** This indicates that **2022 was a period of high gold prices**, likely influenced by economic conditions, market trends, or global uncertainties. This result highlights how gold maintained strong demand and value during that year.

## 15. How does each month's average closing price compare across different years? (plot using Heatmap)

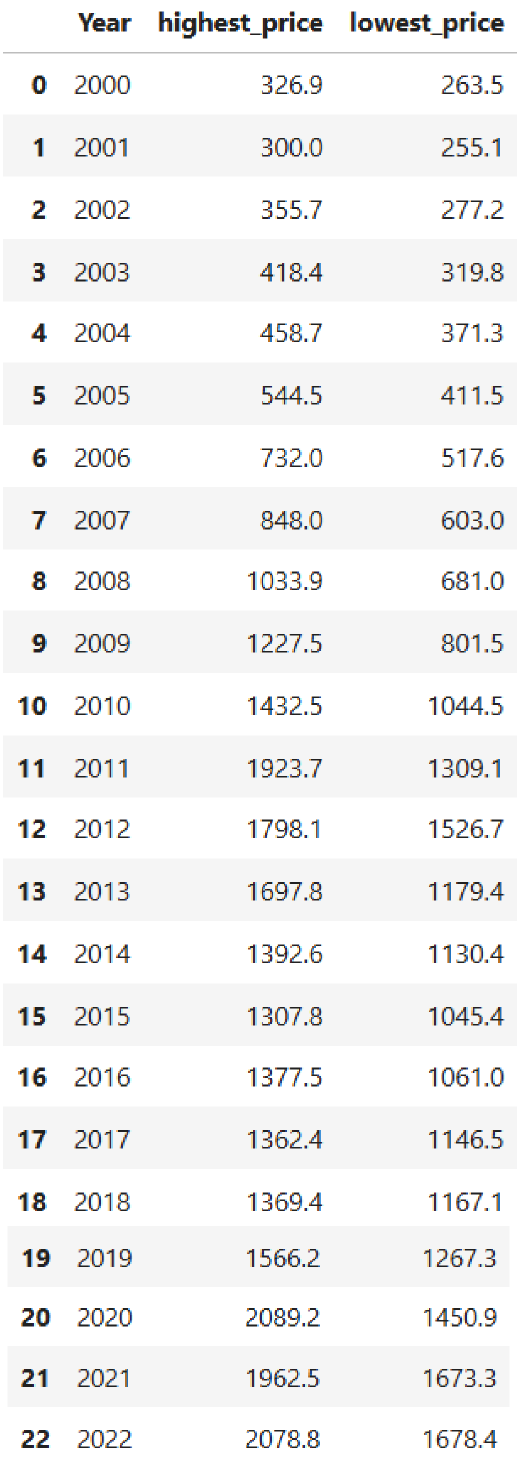


Mean Monthly Closing Price Across Years

| Year | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2000 | 285.86 | 302.42 | 287.15 | 281.97 | 276.03 | 288.33 | 282.06 | 279.53 | 276.91 | 272.12 | 266.94 | 273.97 |
| 2001 | 266.26 | 263.51 | 262.90 | 261.79 | 272.57 | 271.79 | 268.33 | 275.47 | 286.16 | 284.11 | 276.54 | 276.59 |
| 2002 | 282.09 | 296.76 | 294.69 | 304.20 | 314.99 | 321.78 | 313.65 | 311.86 | 321.00 | 317.07 | 319.36 | 335.60 |
| 2003 | 358.54 | 358.87 | 340.19 | 329.82 | 356.21 | 357.10 | 351.46 | 361.96 | 380.39 | 379.35 | 389.96 | 409.52 |
| 2004 | 415.04 | 404.43 | 407.77 | 404.64 | 384.01 | 392.82 | 398.82 | 404.18 | 408.05 | 422.31 | 439.26 | 444.07 |
| 2005 | 424.76 | 425.53 | 435.25 | 431.31 | 423.03 | 433.41 | 425.10 | 443.04 | 461.13 | 472.47 | 476.89 | 512.39 |
| 2006 | 552.52 | 558.43 | 559.65 | 615.79 | 675.58 | 599.50 | 636.87 | 640.35 | 605.41 | 589.64 | 629.42 | 632.29 |
| 2007 | 632.17 | 670.61 | 655.58 | 684.24 | 669.52 | 659.03 | 668.21 | 675.50 | 721.32 | 760.95 | 807.58 | 812.69 |
| 2008 | 893.18 | 928.65 | 964.69 | 913.06 | 891.36 | 892.79 | 941.33 | 842.37 | 833.92 | 804.80 | 755.51 | 823.98 |
| 2009 | 862.17 | 942.82 | 926.02 | 893.58 | 930.25 | 947.01 | 935.96 | 951.46 | 1000.19 | 1045.16 | 1126.84 | 1129.70 |
| 2010 | 1117.36 | 1098.78 | 1114.72 | 1152.10 | 1205.53 | 1236.22 | 1192.16 | 1219.31 | 1274.95 | 1344.16 | 1370.60 | 1394.57 |
| 2011 | 1361.90 | 1374.38 | 1422.94 | 1482.50 | 1511.32 | 1528.56 | 1577.67 | 1761.43 | 1764.88 | 1671.04 | 1743.57 | 1644.64 |
| 2012 | 1661.08 | 1745.62 | 1676.32 | 1651.72 | 1588.81 | 1601.85 | 1593.47 | 1633.07 | 1751.68 | 1746.08 | 1722.01 | 1685.60 |
| 2013 | 1671.09 | 1628.20 | 1591.66 | 1485.05 | 1416.50 | 1342.32 | 1285.66 | 1352.28 | 1346.08 | 1317.04 | 1275.18 | 1223.92 |
| 2014 | 1244.09 | 1300.72 | 1336.53 | 1298.75 | 1287.76 | 1282.57 | 1311.03 | 1296.68 | 1236.48 | 1223.57 | 1176.45 | 1200.41 |
| 2015 | 1253.08 | 1225.48 | 1177.81 | 1199.83 | 1198.11 | 1181.16 | 1128.26 | 1118.82 | 1123.87 | 1159.39 | 1083.93 | 1068.59 |
| 2016 | 1097.24 | 1201.61 | 1245.53 | 1243.84 | 1260.54 | 1279.19 | 1339.07 | 1344.48 | 1329.61 | 1266.05 | 1236.29 | 1153.55 |
| 2017 | 1194.45 | 1236.24 | 1231.25 | 1271.14 | 1245.35 | 1262.34 | 1237.41 | 1289.88 | 1318.55 | 1282.48 | 1282.40 | 1268.92 |
| 2018 | 1332.82 | 1332.76 | 1325.60 | 1338.38 | 1304.32 | 1283.88 | 1238.73 | 1207.76 | 1202.58 | 1217.74 | 1221.88 | 1256.92 |
| 2019 | 1293.13 | 1323.41 | 1300.57 | 1289.27 | 1285.18 | 1364.10 | 1416.33 | 1513.44 | 1515.91 | 1499.18 | 1471.78 | 1486.49 |
| 2020 | 1562.13 | 1599.69 | 1598.59 | 1707.73 | 1722.86 | 1743.81 | 1852.13 | 1980.27 | 1929.96 | 1905.55 | 1867.36 | 1862.85 |
| 2021 | 1866.37 | 1807.45 | 1719.99 | 1760.91 | 1853.48 | 1832.45 | 1806.42 | 1788.37 | 1778.16 | 1777.43 | 1820.33 | 1792.53 |
| 2022 | 1816.95 | 1858.99 | 1952.32 | 1937.55 | 1847.26 | 1838.78 | 1732.70 | 1778.65 | 1716.15 | 1802.40 | 1835.71 | 1904.40 |

This heatmap shows the average monthly (1 for Jan, 2 for Feb, etc.) gold closing prices across different years, where darker colours indicate higher prices. From 2000 to 2022, there is a noticeable upward trend in prices, with significant peaks in 2011, 2012, and 2022. **The data highlights seasonal fluctuations, with some months typically having higher prices than others.** This visualization helps identify long-term trends and short-term monthly variations in gold prices.

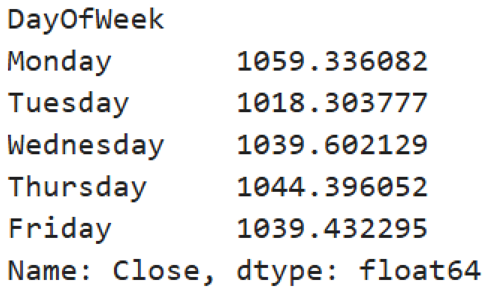
16. **What was the highest and lowest price each year? (Identify annual highs and lows)**



|  | Year | highest_price | lowest_price |
|---|---|---|---|
| **0** | 2000 | 326.9 | 263.5 |
| **1** | 2001 | 300.0 | 255.1 |
| **2** | 2002 | 355.7 | 277.2 |
| **3** | 2003 | 418.4 | 319.8 |
| **4** | 2004 | 458.7 | 371.3 |
| **5** | 2005 | 544.5 | 411.5 |
| **6** | 2006 | 732.0 | 517.6 |
| **7** | 2007 | 848.0 | 603.0 |
| **8** | 2008 | 1033.9 | 681.0 |
| **9** | 2009 | 1227.5 | 801.5 |
| **10** | 2010 | 1432.5 | 1044.5 |
| **11** | 2011 | 1923.7 | 1309.1 |
| **12** | 2012 | 1798.1 | 1526.7 |
| **13** | 2013 | 1697.8 | 1179.4 |
| **14** | 2014 | 1392.6 | 1130.4 |
| **15** | 2015 | 1307.8 | 1045.4 |
| **16** | 2016 | 1377.5 | 1061.0 |
| **17** | 2017 | 1362.4 | 1146.5 |
| **18** | 2018 | 1369.4 | 1167.1 |
| **19** | 2019 | 1566.2 | 1267.3 |
| **20** | 2020 | 2089.2 | 1450.9 |
| **21** | 2021 | 1962.5 | 1673.3 |
| **22** | 2022 | 2078.8 | 1678.4 |

This shows the annual highest and lowest gold prices from 2000 to 2022. the annual highest and lowest gold prices steadily increased, showcasing gold's growing value over time. The **highest price was $2,089.2 in 2020**, while the **lowest price was $255.1 in 2001**. Major price spikes were observed in years like **2011, 2020, and 2022**, coinciding with periods of economic uncertainty and market volatility.

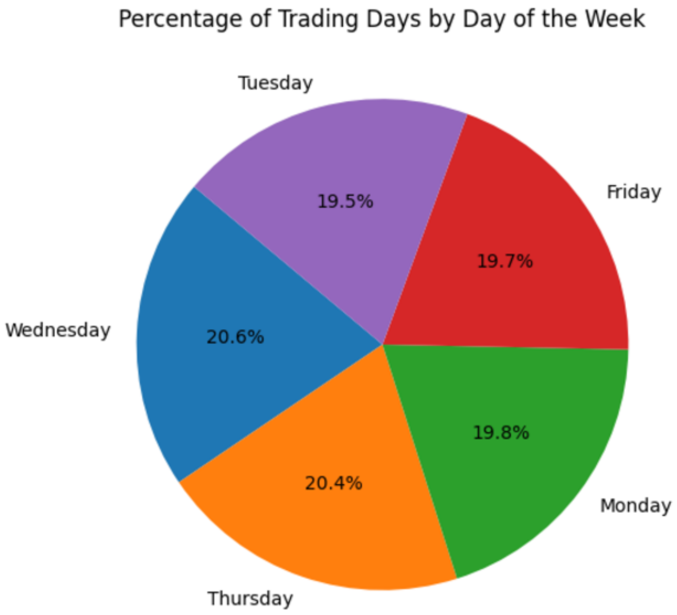
**17. Are there certain days of the week when the price tends to be higher or lower?**



```
DayOfWeek
Monday       1059.336082
Tuesday      1018.303777
Wednesday    1039.602129
Thursday     1044.396052
Friday       1039.432295
Name: Close, dtype: float64
```

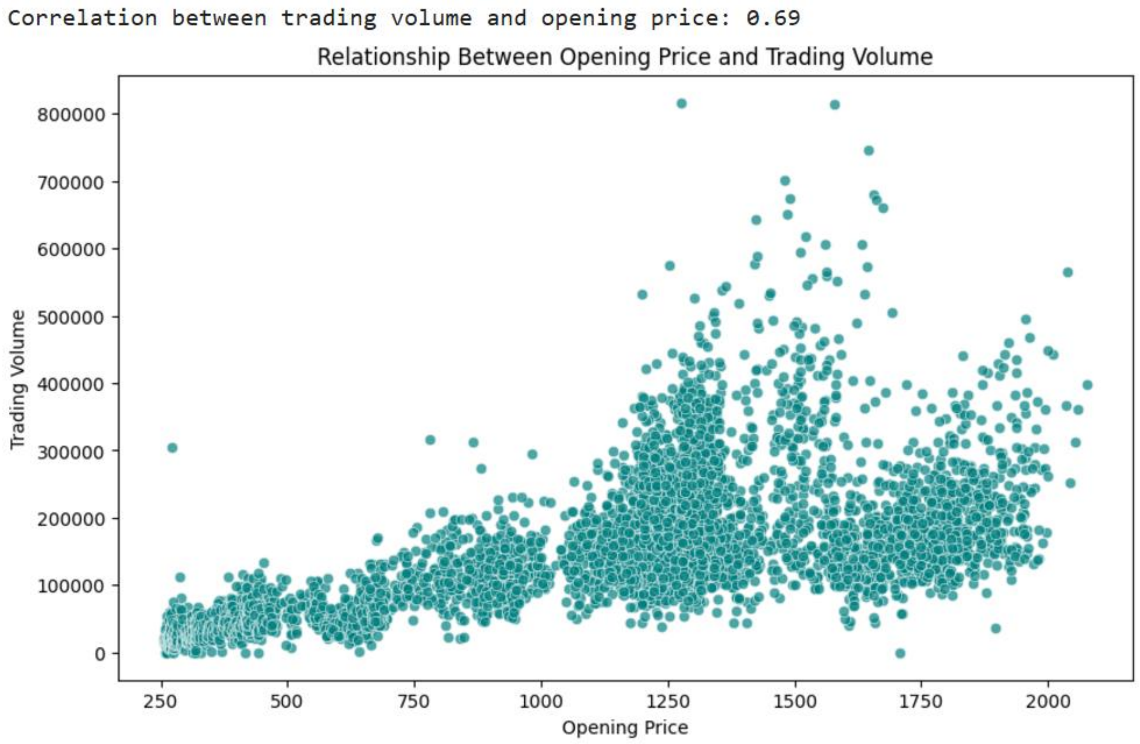
**Monday typically has the highest average closing price ($1,059.34),** while **Tuesday has the lowest ($1,018.30).** The prices fluctuate slightly during the remaining weekdays, with Thursday showing a relatively high average ($1,044.40).

**18. What is the percentage of Trading Days by Day of the Week? (Using pie chart visualization)**



The pie chart represents the percentage distribution of trading days across weekdays. **Wednesday has the highest share of trading days (20.6%),** followed by **Thursday (20.4%). Tuesday has the least share of trading days (19.5%),** while **Monday (19.8%)** and **Friday (19.7%)** are almost equally distributed. **This reflects a fairly even trading activity across the week, with slight variations.**

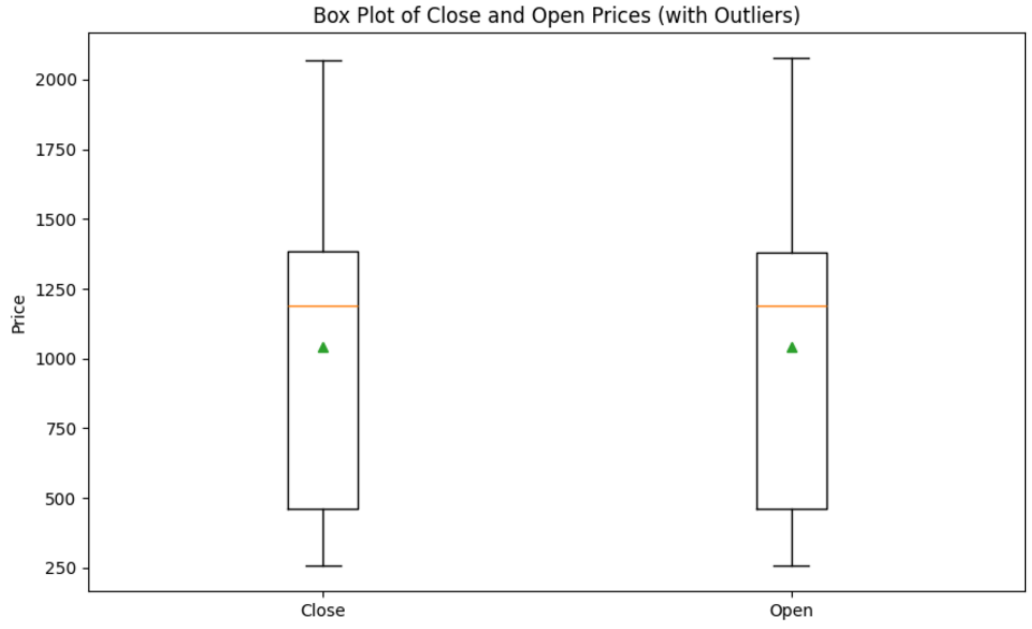
**19. What is the relationship between the daily trading volume and the opening price?**



```
Correlation between trading volume and opening price: 0.69
```

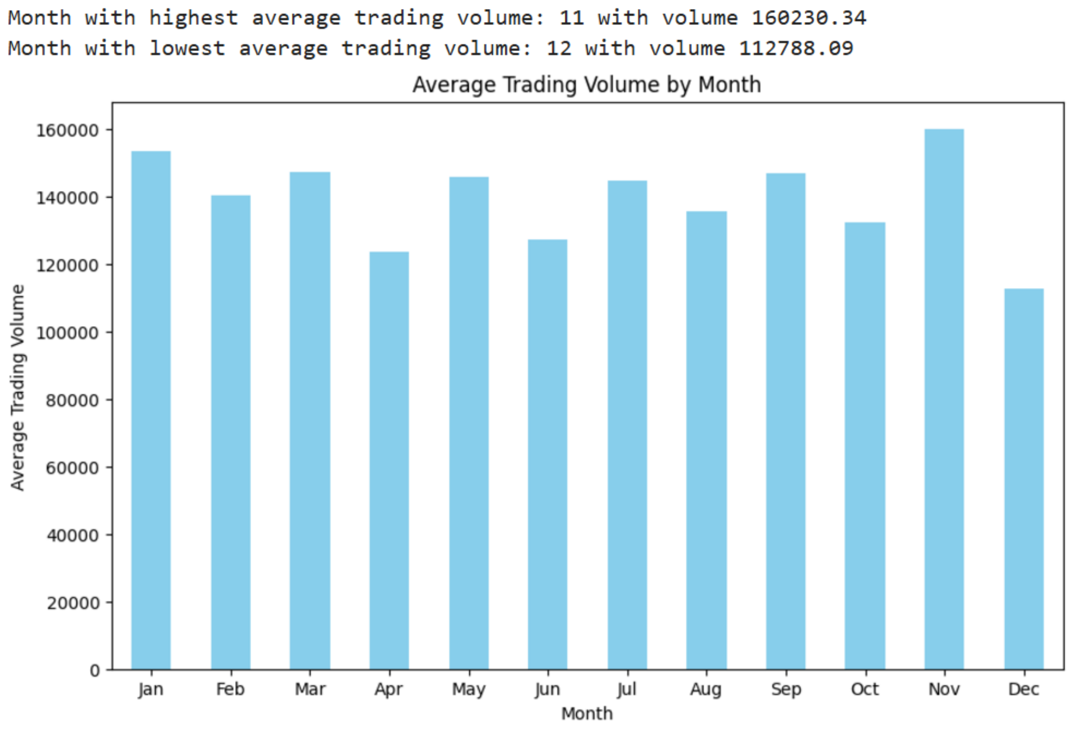
The scatter plot illustrates the relationship between daily trading volume and opening price, with a correlation coefficient of **0.69,** indicating a **moderately strong positive correlation. As the opening price increases, trading volume tends to rise as well,** though the data shows some variability. **This suggests that higher opening prices are often associated with greater market activity.**

**20. Are there any anomalies or unexpected price drops/rises? (Detect outliers in 'Close' or 'Open' prices)**



This box plot shows the distribution of gold's closing and opening prices to identify anomalies. The medians (orange lines) are similar, indicating that opening and closing prices are generally stable. The range of prices spans from around $250 to over $2000, reflecting significant variation. The green triangles represent the mean prices. **No extreme outliers are observed, suggesting that gold prices were relatively consistent with no unusual spikes or drops.**

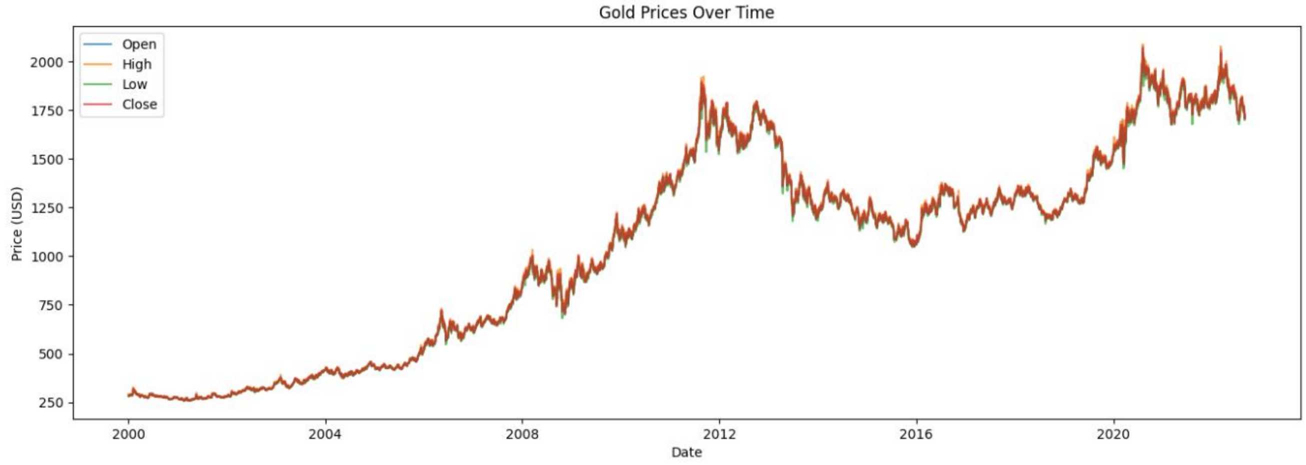
**21. Which month tends to have the highest or lowest trading volume on average?**

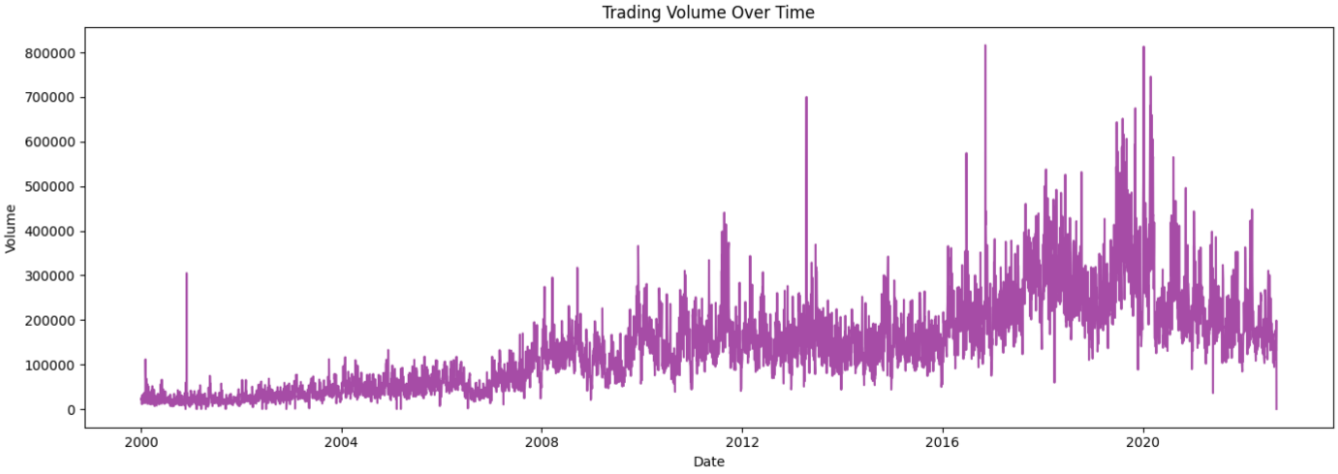


```
Month with highest average trading volume: 11 with volume 160230.34
Month with lowest average trading volume: 12 with volume 112788.09
```

The analysis shows that **November has the highest average trading volume (160,230.34), while December has the lowest (112,788.09).** The bar chart visualizes the monthly trends, with November standing out, due to increased trading activity before the year's end or due to festivals like Diwali, where gold is in high demand. Conversely, December's lower volume may reflect reduced trading activity during the holiday season. Overall, **the data highlights seasonal patterns in gold trading, influenced by cultural and market factors, which can help traders anticipate activity levels across months.**

### 22. Do time series analysis for Open, High, Low, Close, and Volume to visualize trends.



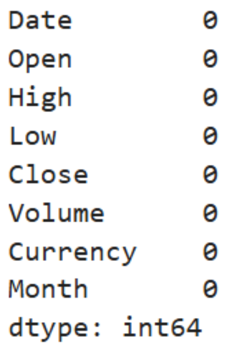


The time series plot shows the trends in gold prices and trading volume over the years:

- **Gold Prices:** From 2000 to around 2012, prices rose steadily, peaking in 2012. After a dip, there was another peak around 2020. All price metrics (open, high, low, close) moved closely, reflecting consistent market behaviour.

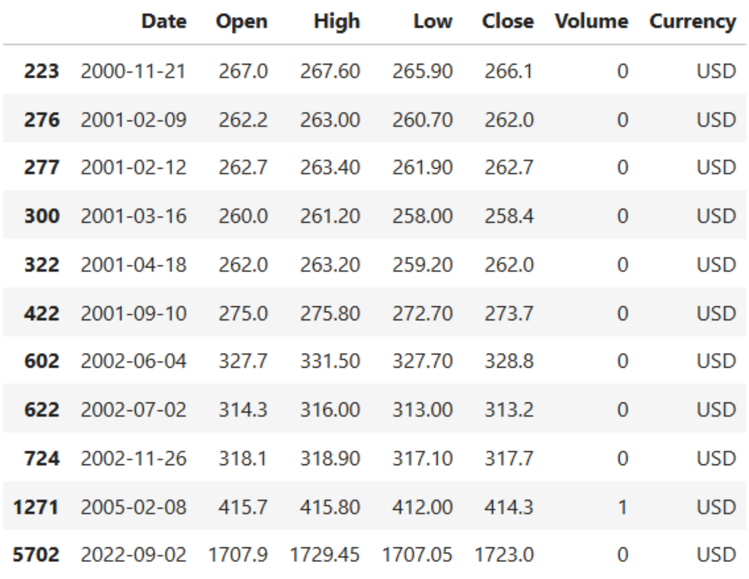
- **Trading Volume:** Volume increased significantly after 2008, peaking around 2016, likely driven by heightened market activity or volatility. It declined somewhat post-2020, showing a shift in trading behaviour.

## IV) Process null values and bad values (noises) in the dataset:



```
Date        0
Open        0
High        0
Low         0
Close       0
Volume      0
Currency    0
Month       0
dtype: int64
```

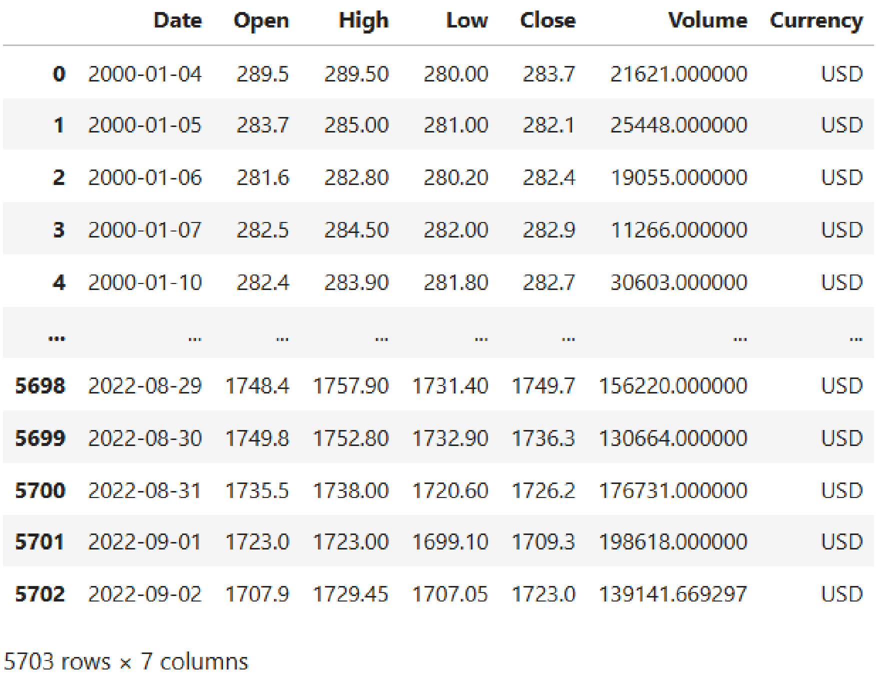
There are no null or missing values in the dataset for any column.



| | Date | Open | High | Low | Close | Volume | Currency |
|---|---|---|---|---|---|---|---|
| **223** | 2000-11-21 | 267.0 | 267.60 | 265.90 | 266.1 | 0 | USD |
| **276** | 2001-02-09 | 262.2 | 263.00 | 260.70 | 262.0 | 0 | USD |
| **277** | 2001-02-12 | 262.7 | 263.40 | 261.90 | 262.7 | 0 | USD |
| **300** | 2001-03-16 | 260.0 | 261.20 | 258.00 | 258.4 | 0 | USD |
| **322** | 2001-04-18 | 262.0 | 263.20 | 259.20 | 262.0 | 0 | USD |
| **422** | 2001-09-10 | 275.0 | 275.80 | 272.70 | 273.7 | 0 | USD |
| **602** | 2002-06-04 | 327.7 | 331.50 | 327.70 | 328.8 | 0 | USD |
| **622** | 2002-07-02 | 314.3 | 316.00 | 313.00 | 313.2 | 0 | USD |
| **724** | 2002-11-26 | 318.1 | 318.90 | 317.10 | 317.7 | 0 | USD |
| **1271** | 2005-02-08 | 415.7 | 415.80 | 412.00 | 414.3 | 1 | USD |
| **5702** | 2022-09-02 | 1707.9 | 1729.45 | 1707.05 | 1723.0 | 0 | USD |

But, there are some rows where the trading volume (Volume) is either 0 or 1, which could indicate anomalies in the dataset which may be responsible for the poor results in the machine learning model later so, we should handle this first.

To handle this, at first the average of the Volume column is calculated, which includes all rows. Then, Rows where Volume is 0 or 1 are replaced with the calculated mean value. This ensures that anomalies in the dataset don't skew further analysis and now in the resulting new DataFrame all rows where Volume was 0 or 1 (e.g., rows like 2000-11-21 and 2005-02-08) now have a new value equal to the mean of the Volume column.

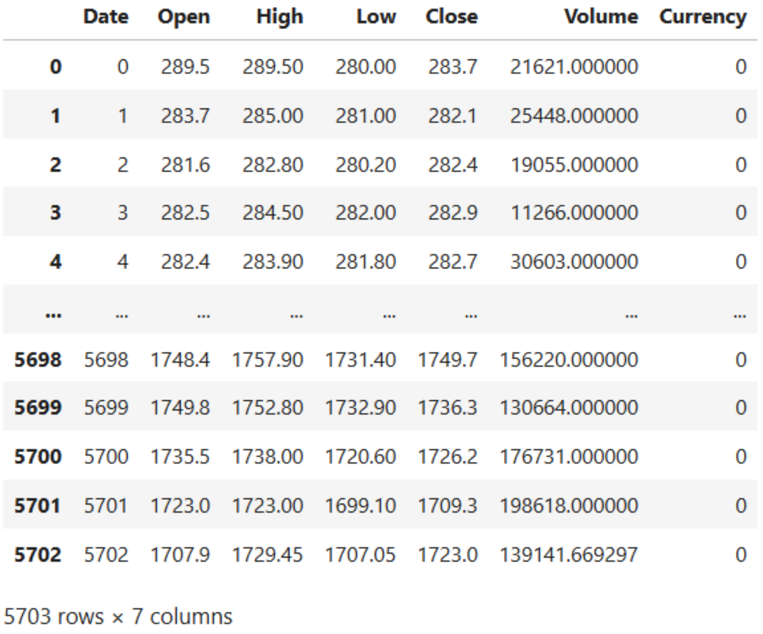


| | Date | Open | High | Low | Close | Volume | Currency |
|---|---|---|---|---|---|---|---|
| **0** | 2000-01-04 | 289.5 | 289.50 | 280.00 | 283.7 | 21621.000000 | USD |
| **1** | 2000-01-05 | 283.7 | 285.00 | 281.00 | 282.1 | 25448.000000 | USD |
| **2** | 2000-01-06 | 281.6 | 282.80 | 280.20 | 282.4 | 19055.000000 | USD |
| **3** | 2000-01-07 | 282.5 | 284.50 | 282.00 | 282.9 | 11266.000000 | USD |
| **4** | 2000-01-10 | 282.4 | 283.90 | 281.80 | 282.7 | 30603.000000 | USD |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **5698** | 2022-08-29 | 1748.4 | 1757.90 | 1731.40 | 1749.7 | 156220.000000 | USD |
| **5699** | 2022-08-30 | 1749.8 | 1752.80 | 1732.90 | 1736.3 | 130664.000000 | USD |
| **5700** | 2022-08-31 | 1735.5 | 1738.00 | 1720.60 | 1726.2 | 176731.000000 | USD |
| **5701** | 2022-09-01 | 1723.0 | 1723.00 | 1699.10 | 1709.3 | 198618.000000 | USD |
| **5702** | 2022-09-02 | 1707.9 | 1729.45 | 1707.05 | 1723.0 | 139141.669297 | USD |

5703 rows × 7 columns

For instance, the previously anomalous row from 2022-09-02 now has a corrected volume of approximately 139,141.67.

## V) Convert categorical features into numbers:

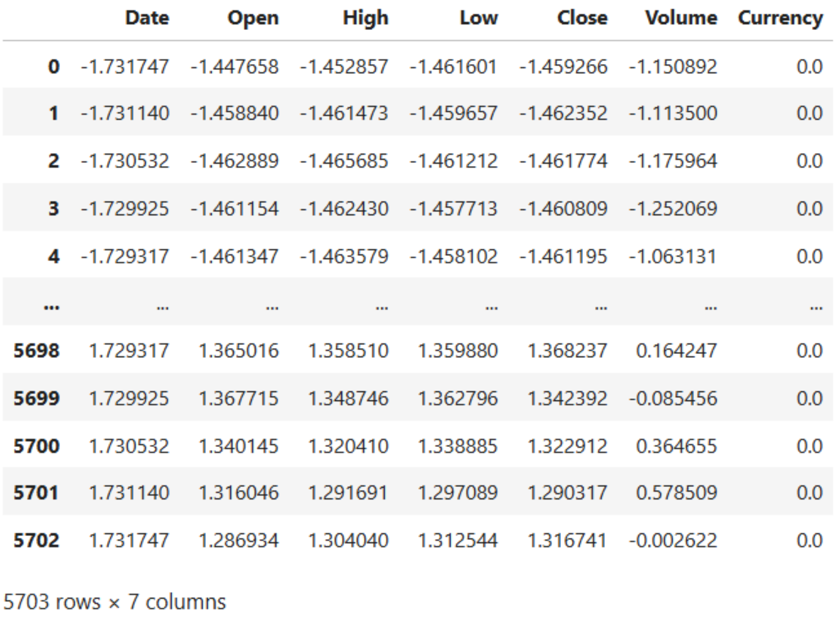


| | Date | Open | High | Low | Close | Volume | Currency |
|---|---|---|---|---|---|---|---|
| **0** | 0 | 289.5 | 289.50 | 280.00 | 283.7 | 21621.000000 | 0 |
| **1** | 1 | 283.7 | 285.00 | 281.00 | 282.1 | 25448.000000 | 0 |
| **2** | 2 | 281.6 | 282.80 | 280.20 | 282.4 | 19055.000000 | 0 |
| **3** | 3 | 282.5 | 284.50 | 282.00 | 282.9 | 11266.000000 | 0 |
| **4** | 4 | 282.4 | 283.90 | 281.80 | 282.7 | 30603.000000 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **5698** | 5698 | 1748.4 | 1757.90 | 1731.40 | 1749.7 | 156220.000000 | 0 |
| **5699** | 5699 | 1749.8 | 1752.80 | 1732.90 | 1736.3 | 130664.000000 | 0 |
| **5700** | 5700 | 1735.5 | 1738.00 | 1720.60 | 1726.2 | 176731.000000 | 0 |
| **5701** | 5701 | 1723.0 | 1723.00 | 1699.10 | 1709.3 | 198618.000000 | 0 |
| **5702** | 5702 | 1707.9 | 1729.45 | 1707.05 | 1723.0 | 139141.669297 | 0 |

5703 rows × 7 columns

Now, we convert all categorical features in the new DataFrame into numeric values using **LabelEncoder.** For each column with object data type (strings or categories), it assigns unique integer labels to each unique category. In the output, columns like Date and Currency, which were categorical, are now represented as numeric values (e.g., 0, 1, 2, etc.). The numerical columns remain unchanged. **This transformation is necessary for machine learning models that require numerical inputs.**

## VI) Scale numeric features:



| | Date | Open | High | Low | Close | Volume | Currency |
|---|---|---|---|---|---|---|---|
| **0** | -1.731747 | -1.447658 | -1.452857 | -1.461601 | -1.459266 | -1.150892 | 0.0 |
| **1** | -1.731140 | -1.458840 | -1.461473 | -1.459657 | -1.462352 | -1.113500 | 0.0 |
| **2** | -1.730532 | -1.462889 | -1.465685 | -1.461212 | -1.461774 | -1.175964 | 0.0 |
| **3** | -1.729925 | -1.461154 | -1.462430 | -1.457713 | -1.460809 | -1.252069 | 0.0 |
| **4** | -1.729317 | -1.461347 | -1.463579 | -1.458102 | -1.461195 | -1.063131 | 0.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **5698** | 1.729317 | 1.365016 | 1.358510 | 1.359880 | 1.368237 | 0.164247 | 0.0 |
| **5699** | 1.729925 | 1.367715 | 1.348746 | 1.362796 | 1.342392 | -0.085456 | 0.0 |
| **5700** | 1.730532 | 1.340145 | 1.320410 | 1.338885 | 1.322912 | 0.364655 | 0.0 |
| **5701** | 1.731140 | 1.316046 | 1.291691 | 1.297089 | 1.290317 | 0.578509 | 0.0 |
| **5702** | 1.731747 | 1.286934 | 1.304040 | 1.312544 | 1.316741 | -0.002622 | 0.0 |

5703 rows × 7 columns

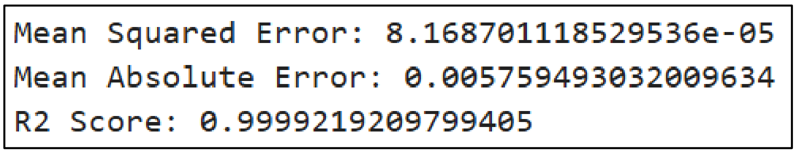
Now, we standardize all numeric columns in the new DataFrame using **StandardScaler.** This process transforms the values to have a mean of 0 and a standard deviation of 1.

As a result, all numeric features are scaled to a standard range, **making them suitable for machine learning algorithms sensitive to feature scales.** The output shows the transformed values, where each numeric column has been rescaled while preserving its relative differences. The Currency column remains constant (0.0) because it likely had a single unique value.

## VII) Apply Machine Learning Regression Model:

1) **Simple linear regression model:** At first, a simple linear regression model was applied to predict gold's closing price (Close) using other features such as Date, Open, High, Low, Volume, and Currency as input variables. The dataset was split into training (80%) and testing (20%) sets, and the model was trained on the training data.

The evaluation metrics reveal excellent model performance:



```
Mean Squared Error: 8.168701118529536e-05
Mean Absolute Error: 0.005759493032009634
R2 Score: 0.9999219209799405
```

**Mean Squared Error (MSE): 8.17e-05,** indicating minimal average squared prediction error.

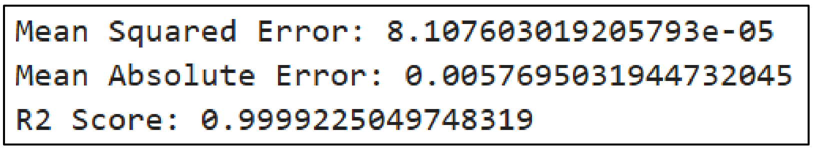
**Mean Absolute Error (MAE): 0.00576,** showing small absolute errors.

**R2 Score: 0.9999,** demonstrating the model explains nearly all the variance in the closing price.

- **These results suggest the features are highly predictive of the target variable (Close).**

2) **Polynomial regression model:** A polynomial regression model with a degree of 2 (quadratic) was applied to predict gold's closing price (Close) using features such as Date, Open, High, Low, Volume, and Currency. The dataset was split into training (80%) and testing (20%) sets, and polynomial features were generated to capture non-linear relationships. The model was trained on the transformed training data.

The evaluation metrics indicate exceptional performance:



```
Mean Squared Error: 8.107603019205793e-05
Mean Absolute Error: 0.0057695031944732045
R2 Score: 0.9999225049748319
```

**Mean Squared Error (MSE): 8.11e-05,** reflecting very low average squared prediction error.

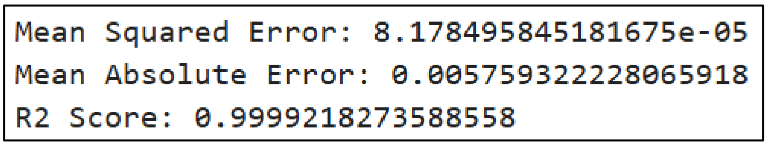
**Mean Absolute Error (MAE): 0.00577,** indicating minimal absolute errors.

**R2 Score: 0.99992,** demonstrating the model explains nearly all the variance in the closing price.

- **These results confirm that the quadratic model captures complex patterns and provides highly accurate predictions for the target variable (Close).**

3) **Multivariate regression model:** A multivariate regression model was used to predict gold's closing price (Close) based on numerical features like Open, High, Low, and Volume (excluding Date and Currency). The dataset was split into training (80%) and testing (20%) sets. A linear regression model was trained on the training data to assess the relationship between the predictors and the target variable.

The evaluation metrics indicate strong model performance:



```
Mean Squared Error: 8.178495845181675e-05
Mean Absolute Error: 0.005759322228065918
R2 Score: 0.9999218273588558
```

**Mean Squared Error (MSE): 8.18e-05,** showing very low average squared prediction error.

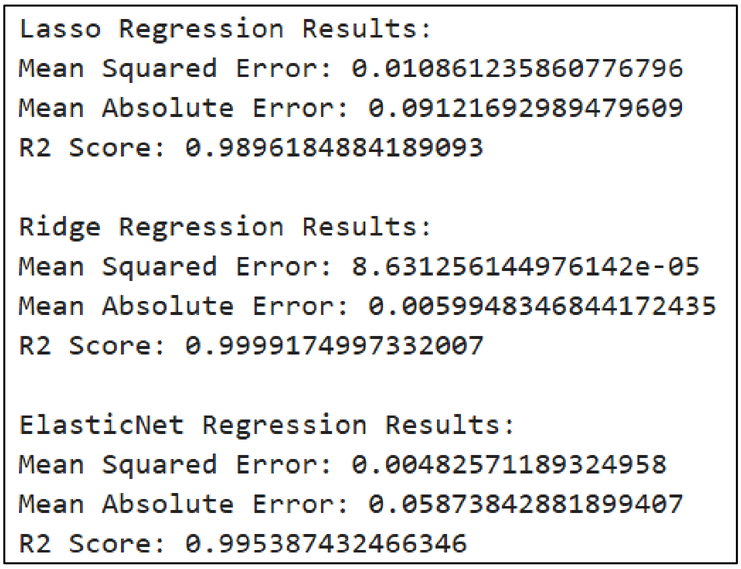
**Mean Absolute Error (MAE): 0.00576,** reflecting minimal absolute errors.

**R2 Score: 0.99992,** signifying that the model explains nearly all the variance in the closing price.

- **These results highlight the effectiveness of the multivariate approach in accurately predicting the closing price using key numerical features.**

4) **Lasso, Ridge, ElasticNet regression model:** Lasso, Ridge, and ElasticNet regression models were applied to predict gold's closing price (Close) using all features. The dataset was split into training (80%) and testing (20%) sets, and regularized regression models were trained on the training data to handle multicollinearity and improve generalization.

**Results for each model:**

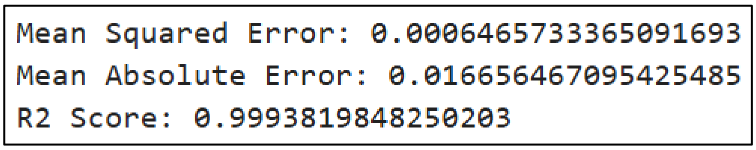


```
Lasso Regression Results:
Mean Squared Error: 0.010861235860776796
Mean Absolute Error: 0.09121692989479609
R2 Score: 0.9896184884189093

Ridge Regression Results:
Mean Squared Error: 8.631256144976142e-05
Mean Absolute Error: 0.0059948346844172435
R2 Score: 0.9999174997332007

ElasticNet Regression Results:
Mean Squared Error: 0.00482571189324958
Mean Absolute Error: 0.05873842881899407
R2 Score: 0.995387432466346
```

- **The Ridge Regression model performed the best with a near-perfect R2 Score (0.9999),** indicating it effectively captures the relationship between the features and the target variable while controlling for overfitting.

5) **KNN regression model:** The K-Nearest Neighbors (KNN) regression model was applied to predict the gold closing price (Close) based on the input features. The dataset was divided into training (80%) and testing (20%) sets, and a KNN regressor was initialized with n_neighbors=5. This means the prediction for each test data point was based on the average of its 5 nearest neighbors in the feature space.

**Results:**



```
Mean Squared Error: 0.0006465733365091693
Mean Absolute Error: 0.016656467095425485
R2 Score: 0.9993819848250203
```

**Mean Squared Error (MSE): 0.00065,** indicating low average squared error in predictions.

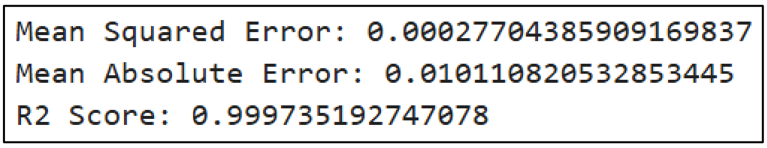
**Mean Absolute Error (MAE): 0.0167,** suggesting small absolute prediction errors.

**R2 Score: 0.9994,** showing that the model explains nearly all the variance in the target variable.

- **The results indicate that KNN regression effectively models the relationship between features and the closing price, achieving high accuracy and minimal prediction errors.**

6) **Decision Tree regression model:** The Decision Tree Regressor was applied to predict the gold closing price (Close) using the input features. The data was split into 80% training and 20% testing, ensuring the model was trained on a larger portion of the data. A DecisionTreeRegressor was initialized with a random_state=42 to ensure reproducible results. Decision trees split the data into subsets based on feature values, learning patterns that minimize error within each split.

**Results:**



```
Mean Squared Error: 0.00027704385909169837
Mean Absolute Error: 0.010110820532853445
R2 Score: 0.999735192747078
```

**Mean Squared Error (MSE): 0.00028,** indicating a very small average squared error.

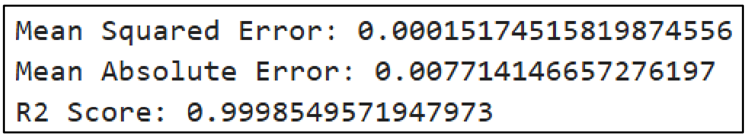
**Mean Absolute Error (MAE): 0.0101,** highlighting minimal absolute prediction errors.

**R2 Score: 0.9997,** showcasing the model's ability to explain almost all the variance in the target variable.

- **The Decision Tree Regressor performed exceptionally well, achieving a strong fit with minimal error and high accuracy.**

7) **Random Forest regression model:** The Random Forest Regressor was utilized to predict the gold closing price (Close) using various input features. The dataset was split into 80% training and 20% testing, ensuring a larger portion of the data was used to train the model. A RandomForestRegressor was initialized with 100 trees (n_estimators=100) and random_state=42 for stable predictions ensures reproducibility by controlling the randomness of the tree splits.

**Results:**



```
Mean Squared Error: 0.00015174515819874556
Mean Absolute Error: 0.007714146657276197
R2 Score: 0.9998549571947973
```

**Mean Squared Error (MSE): 0.00015,** indicating a very low average squared error.

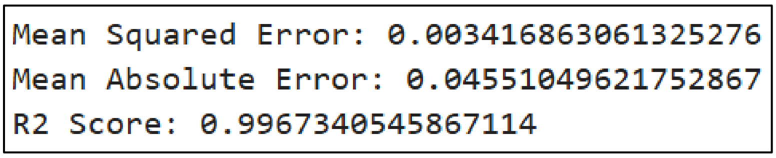
**Mean Absolute Error (MAE): 0.0077,** highlighting minimal absolute prediction errors.

**R2 Score: 0.9998,** showcasing that the model explains nearly all the variance in the target variable.

- The Random Forest Regressor **achieved excellent results with minimal error and high accuracy.** Its ensemble nature makes it robust against overfitting and suitable for this predictive task. Further tuning of hyperparameters like n_estimators or max_depth can optimize performance further.

8) **AdaBoost Regression model:** The AdaBoost Regressor was implemented to predict the gold closing price (Close) using the given input features. The dataset was divided into 80% training and 20% testing to train the model on a larger dataset portion while evaluating its performance on unseen data. The AdaBoostRegressor was initialized with n_estimators=50, random_state=42. AdaBoost builds an ensemble of weak learners (typically decision trees) by iteratively focusing on harder-to-predict samples, thereby improving overall accuracy.

**Results:**



```
Mean Squared Error: 0.003416863061325276
Mean Absolute Error: 0.04551049621752867
R2 Score: 0.9967340545867114
```

**Mean Squared Error (MSE): 0.0034,** indicating low average squared prediction error.

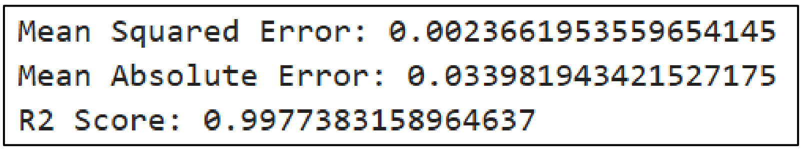
**Mean Absolute Error (MAE): 0.0455,** highlighting minimal absolute prediction deviations.

**R2 Score: 0.9967,** demonstrating the model explains almost all variance in the target variable.

- The AdaBoost Regressor achieved excellent performance, effectively learning the underlying relationships in the dataset. However, its reliance on iterative training makes it sensitive to noise and outliers. The model can benefit from further tuning of hyperparameters like n_estimators and learning_rate to balance accuracy and generalization.

9) **Support Vector Regression model:** The Support Vector Regressor (SVR) was utilized to predict the gold closing price (Close) using the provided input features. The dataset was divided into 80% training and 20% testing, ensuring the model was trained on a significant portion of the data. SVR, with its RBF kernel, effectively captured non-linear relationships in the data, optimizing predictions within a defined margin of error.

**Results:**



```
Mean Squared Error: 0.0023661953559654145
Mean Absolute Error: 0.03398194342 1527175
R2 Score: 0.9977383158964637
```

**Mean Squared Error (MSE): 0.0023,** indicating a very small average squared prediction error.

**Mean Absolute Error (MAE): 0.0339,** reflecting minimal absolute prediction deviations.

**R2 Score: 0.9977,** demonstrating the model's ability to explain nearly all the variance in the target variable.

- **The SVR model exhibited outstanding performance, achieving a strong fit with low error metrics and high accuracy.** Its ability to handle non-linear patterns through the RBF kernel made it particularly suitable for this dataset. For further optimization, experimenting with different kernels or fine-tuning hyperparameters like C and epsilon could enhance its predictive capabilities.

- **Comparison of Regression Models:**

| Model | Mean Squared Error | Mean Absolute Error | R2 Score |
|---|---|---|---|
| **Simple Linear Regression** | 8.168701118529536e-05 | 0.005759493032009634 | 0.99992 |
| **Polynomial Regression** | 8.107603019205793e-05 | 0.0057695031944732045 | 0.99992 |
| **Multivariate Regression** | 8.178495845181675e-05 | 0.005759322228065918 | 0.99992 |
| **Lasso Regression** | 0.010861235860776796 | 0.09121692989479609 | 0.98961 |
| **Ridge Regression** | 8.631256144976142e-05 | 0.0059948346844172435 | 0.99991 |
| **ElasticNet Regression** | 0.00482571189324958 | 0.05873842881899407 | 0.99538 |
| **KNN Regression** | 0.0006465733365091693 | 0.016656467095425485 | 0.99938 |
| **Decision Tree Regression** | 0.0002770438590916983 | 0.010110820532853445 | 0.99973 |
| **Random Forest Regression** | 0.00015174515819874556 | 0.007714146657276197 | 0.99985 |
| **AdaBoost Regression** | 0.003416863061325276 | 0.04551049621752867 | 0.99673 |
| **Support Vector Regression** | 0.0023661953559654145 | 0.033981943421527175 | 0.99773 |

- After analysing the results, it is clear that **Polynomial Regression has the lowest MSE (8.11e-05) and highest R2 Score (0.99992),** indicating **it provides the best overall performance.** Simple Linear Regression and Multivariate Regression also show comparable results but are slightly behind.

## ❖ Conclusion:

- ✓ This project aimed to predict gold prices using various machine learning regression models. Among the models tested, **Polynomial Regression** demonstrated the best performance with the lowest prediction error **(MSE: 8.11e-05)** and the highest ability to explain variance (**R2 Score: 0.99992**).

- ✓ Other models, such as Random Forest Regression and Support Vector Regression, also delivered strong performances, showing that advanced regression techniques effectively handle the complexities of gold price prediction.

- ✓ In conclusion, **Polynomial Regression** is the most suitable model for this dataset, but further enhancements, such as hyperparameter tuning or ensembling methods, could refine predictions further.

- **Group members:** Subhayan Chatterjee, Avradeep Sanyal