```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
pip install opendatasets
```

```
Collecting opendatasets
  Downloading opendatasets-0.1.22-py3-none-any.whl.metadata (9.2 kB)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from opendatasets) (4.67.2)
Requirement already satisfied: kaggle in /usr/local/lib/python3.12/dist-packages (from opendatasets) (1.7.4.5)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from opendatasets) (8.3.1)
Requirement already satisfied: bleach in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (6.3.0)
Requirement already satisfied: certifi>=14.05.14 in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (2026.1.
Requirement already satisfied: charset-normalizer in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (3.4.4)
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (3.11)
Requirement already satisfied: protobuf in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (5.29.5)
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (2.
Requirement already satisfied: python-slugify in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (8.0.4)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (2.32.4)
Requirement already satisfied: setuptools>=21.0.0 in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (75.2.0
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (1.17.0)
Requirement already satisfied: text-unidecode in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (1.3)
Requirement already satisfied: urllib3>=1.15.1 in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (2.5.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (0.5.1)
Downloading opendatasets-0.1.22-py3-none-any.whl (15 kB)
Installing collected packages: opendatasets
Successfully installed opendatasets-0.1.22
```

```
#{"username":"osamaemad18112022","key":"077431ab0445a802e9c7b71e749190e2"}
```

```
import opendatasets as od
od.download('https://www.kaggle.com/datasets/vishardmehta/gold-price-forecasting-dataset')
```

```
Please provide your Kaggle credentials to download this dataset. Learn more: http://bit.ly/kaggle-creds
Your Kaggle username: osamaemad18112022
Your Kaggle Key: ··········
Dataset URL: https://www.kaggle.com/datasets/vishardmehta/gold-price-forecasting-dataset
Downloading gold-price-forecasting-dataset.zip to ./gold-price-forecasting-dataset
100%|██████████| 135k/135k [00:00<00:00, 255MB/s]
```

```
df=pd.read_csv('/content/gold-price-forecasting-dataset/gold_price_forecasting_dataset.csv')
```

```
df.sample()
```

	date	adj close	close	high	low	open	volume	ma_7	ma_30	ma_90	daily_ret
761	2024-06-21	2316.399902	2316.399902	2331.199951	2316.399902	2331.199951	76	2325.799944	2348.02998	2256.554434	-0.015

exploring data

extract months and years as new features

```
#convert date to dtype
df['date']=pd.to_datetime(df['date'])
```

```
df['year']=df['date'].dt.year
df['month']=df['date'].dt.month
```

```
df.info()
```

Show hidden output

```
df.describe()
```

	date	adj close	close	high	low	open	volume	ma_7	ma_30
count	1167	1167.000000	1167.000000	1167.000000	1167.000000	1167.000000	1167.000000	1167.000000	1167.000000
mean	2023-10-05 16:37:01.079691520	2363.105823	2363.105823	2375.595715	2349.989545	2362.322109	4170.995716	2354.938862	2327.283295
min	2021-06-11 00:00:00	1623.300049	1623.300049	1623.300049	1615.099976	1620.400024	0.000000	1643.671422	1659.240002
25%	2022-08-08 12:00:00	1827.299988	1827.299988	1831.600037	1816.800049	1825.250000	88.000000	1827.585711	1830.560004
50%	2023-10-05 00:00:00	1985.900024	1985.900024	1996.199951	1979.800049	1989.000000	297.000000	1986.785697	1978.813334
75%	2024-12-02 12:00:00	2666.099976	2666.099976	2675.449951	2654.599976	2662.400024	867.500000	2654.771449	2659.524996
max	2026-01-30 00:00:00	5318.399902	5318.399902	5586.200195	5301.600098	5415.700195	209783.000000	5070.914272	4620.783317
std	NaN	749.127289	749.127289	757.244374	741.042344	749.389242	21394.527815	735.998718	697.354115

```
df.head(10)
```

Show hidden output

explore each year

```
df_2021=df[df['date'].dt.year==2021]
df_2022=df[df['date'].dt.year==2022]
df_2023=df[df['date'].dt.year==2023]
df_2024=df[df['date'].dt.year==2024]
df_2025=df[df['date'].dt.year==2025]
df_2026=df[df['date'].dt.year==2026]
```

compresion between years

```
df.sample()
```

	date	adj close	close	high	low	open	volume	ma_7	ma_30	ma_90	daily_return	volatility_7	vo
800	2024-08-16	2498.600098	2498.600098	2508.0	2451.399902	2453.5	197	2453.5	2415.7	2366.58555	0.018548	0.009696	

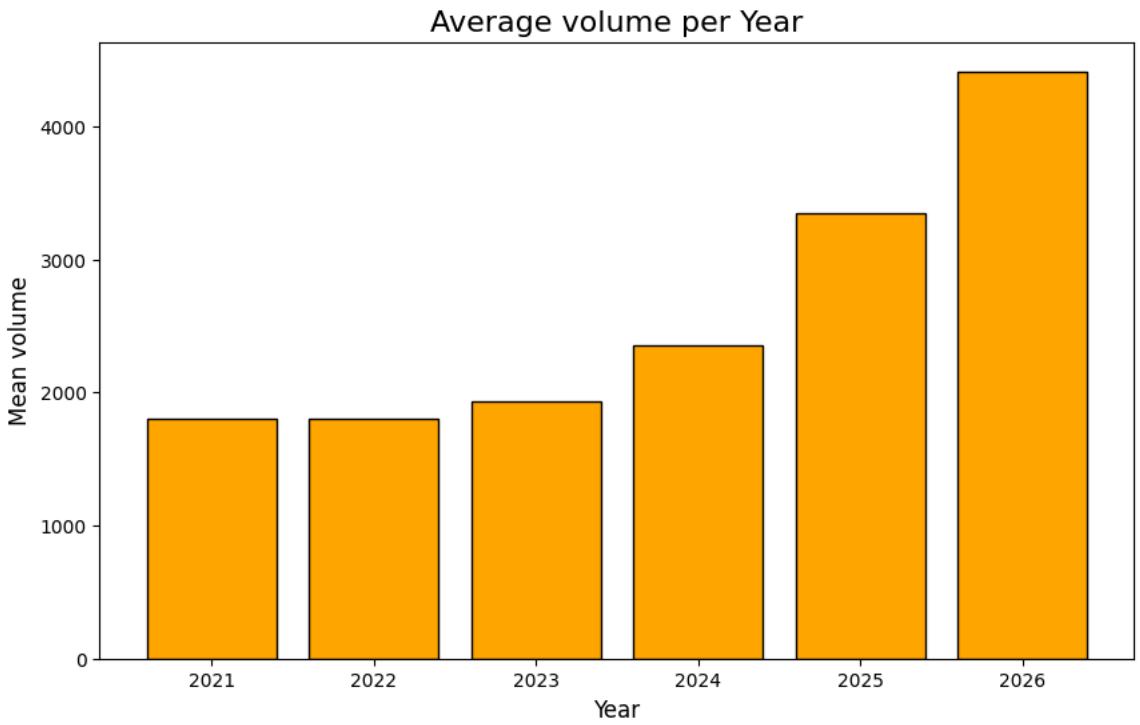
```
mean_ma_30_per_year = df.groupby(df['date'].dt.year)['ma_30'].mean().reset_index()

# إعادة تسمية الأعمدة للتوضيح
mean_ma_30_per_year.columns = ['Year', 'mean_ma_30']

# عرض الجدول
print(mean_ma_30_per_year)

#"From 2021 to 2022, the gold market was stable at ~$1800, then it started to hold better in 2023, and made a rapid increase ir
```

```
   Year   mean_ma_30
0  2021  1801.698540
1  2022  1800.043677
2  2023  1928.904105
3  2024  2355.186938
4  2025  3350.309624
5  2026  4413.814500
```

```
mean_ma_30_per_year
```

	Year	mean_ma_30
0	2021	1801.698540
1	2022	1800.043677
2	2023	1928.904105
3	2024	2355.186938
4	2025	3350.309624
5	2026	4413.814500

```
# رسم الأعمدة
plt.figure(figsize=(10,6))
plt.bar(mean_ma_30_per_year['Year'], mean_ma_30_per_year['mean_ma_30'], color='orange', edgecolor='black')

# إضافة العنوان والمحاور
plt.title('Average volume per Year', fontsize=16)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Mean volume', fontsize=12)
plt.xticks(mean_ma_30_per_year['Year'])

plt.show()

# التعاملات ذادت ف 2026 و احنا لسه ف اول شهر
```



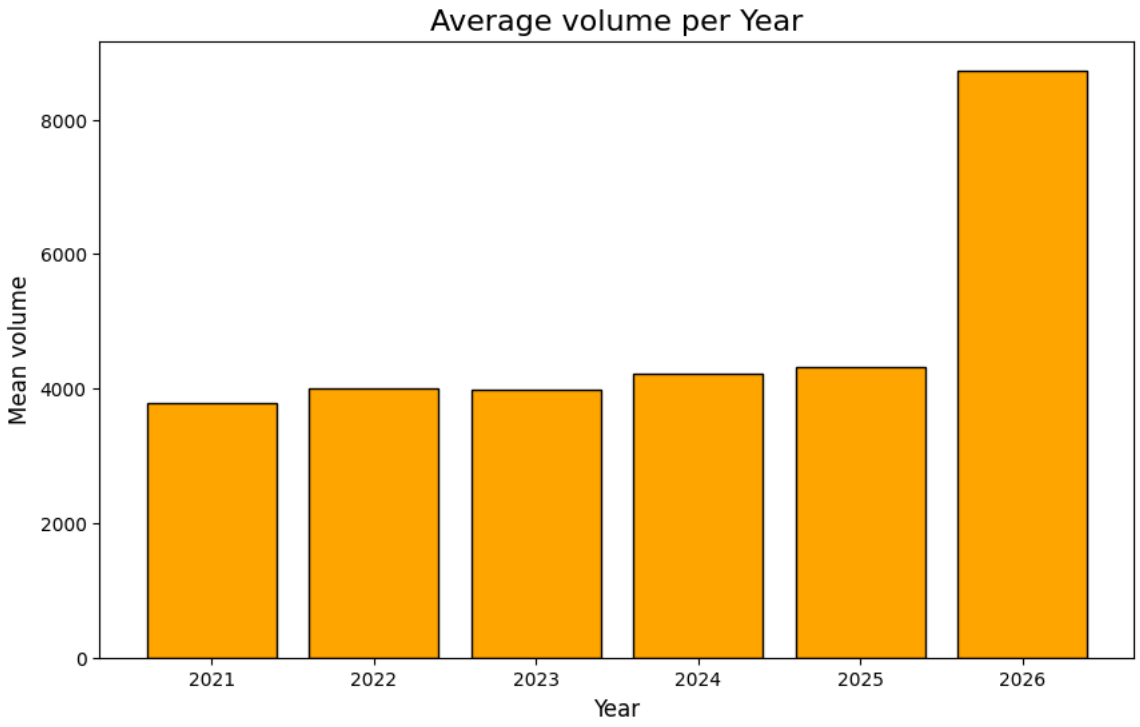

```
mean_close_per_year = df.groupby(df['date'].dt.year)['volume'].mean().reset_index()

# إعادة تسمية الأعمدة للتوضيح
mean_close_per_year.columns = ['Year', 'mean_volume']

# عرض الجدول
print(mean_close_per_year)
```

```
   Year  mean_volume
0  2021  3779.528169
1  2022  4002.637450
2  2023  3985.948000
3  2024  4229.702381
4  2025  4322.214286
5  2026  8731.350000
```

```
mean_close_per_year
```

	Year	Mean_Close
0	2021	1792.989431
1	2022	1800.068127
2	2023	1942.769197
3	2024	2390.006343
4	2025	3447.348409
5	2026	4730.859961

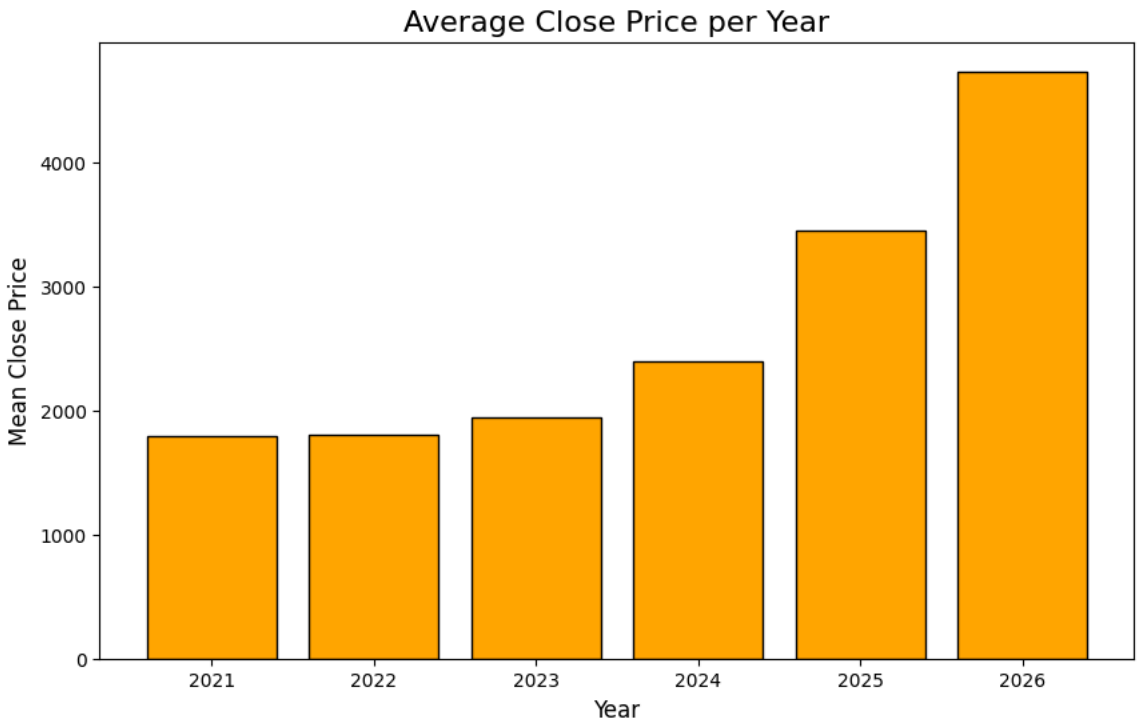
```
# رسم الأعمدة
plt.figure(figsize=(10,6))
plt.bar(mean_close_per_year['Year'], mean_close_per_year['mean_volume'], color='orange', edgecolor='black')

# إضافة العنوان والمحاور
plt.title('Average volume per Year', fontsize=16)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Mean volume', fontsize=12)
plt.xticks(mean_close_per_year['Year'])

plt.show()

# التعاملات زادت ف 2026 و احنا لسه ف اول شهر
```




```
# أولاً لو لسه ما حولتش التاريخ
df['date'] = pd.to_datetime(df['date'])

# احسب المتوسط لكل سنة
mean_close_per_year = df.groupby(df['date'].dt.year)['close'].mean().reset_index()

# إعادة تسمية الأعمدة للتوضيح
mean_close_per_year.columns = ['Year', 'Mean_Close']

# عرض الجدول
print(mean_close_per_year)
```

Show hidden output

```
# رسم الأعمدة
plt.figure(figsize=(10,6))
plt.bar(mean_close_per_year['Year'], mean_close_per_year['Mean_Close'], color='orange', edgecolor='black')

# إضافة العنوان والمحاور
plt.title('Average Close Price per Year', fontsize=16)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Mean Close Price', fontsize=12)
plt.xticks(mean_close_per_year['Year'])

plt.show()

# الدهب رايح ف داهيه
```



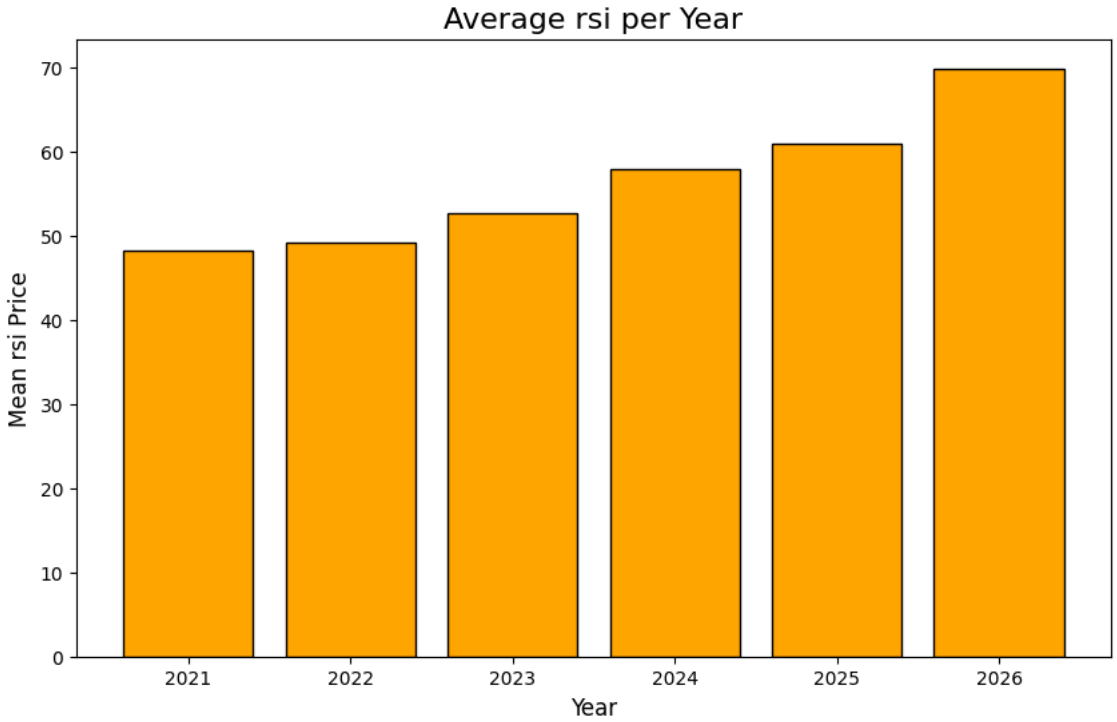

```
# احسب المتوسط لكل سنة
mean_rsi_per_year = df.groupby(df['date'].dt.year)['rsi'].mean().reset_index()

# إعادة تسمية الأعمدة للتوضيح
mean_rsi_per_year.columns = ['Year', 'Mean_rsi']

# عرض الجدول
print(mean_rsi_per_year)
```

```
   Year   Mean_rsi
0  2021  48.234382
1  2022  49.216922
2  2023  52.708372
3  2024  57.864660
4  2025  60.938999
5  2026  69.886982
```

```
mean_rsi_per_year
```

	Year	Mean_rsi
0	2021	48.234382
1	2022	49.216922
2	2023	52.708372
3	2024	57.864660
4	2025	60.938999
5	2026	69.886982

```
# رسم الأعمدة
plt.figure(figsize=(10,6))
plt.bar(mean_rsi_per_year['Year'], mean_rsi_per_year['Mean_rsi'], color='orange', edgecolor='black')

# إضافة العنوان والمحاور
plt.title('Average rsi per Year', fontsize=16)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Mean rsi Price', fontsize=12)
plt.xticks(mean_close_per_year['Year'])

plt.show()

# الدهب رايح ف داهيه
```




```
df.sample()
```

	date	adj close	close	high	low	open	volume	ma_7	ma_30	ma_90	daily_re
1117	2025-11-18	4061.300049	4061.300049	4063.399902	4035.600098	4037.399902	1375	4118.157087	4074.739974	3704.981098	-0.00

We can then say that the gold market was very calm and average until 2025 and 2026

compration over monthes

```
df_2021.sample()
```

	date	adj close	close	high	low	open	volume	ma_7	ma_30	ma_90	daily_return	volatility_7	vo
20	2021-07-12	1805.5	1805.5	1805.5	1799.5	1802.599976	147	1795.514282	1827.393339	1791.834448	-0.002486	0.003326	

```
# ma_30 of each monthin 2021
df_2021.groupby('month')['ma_30'].mean()
```

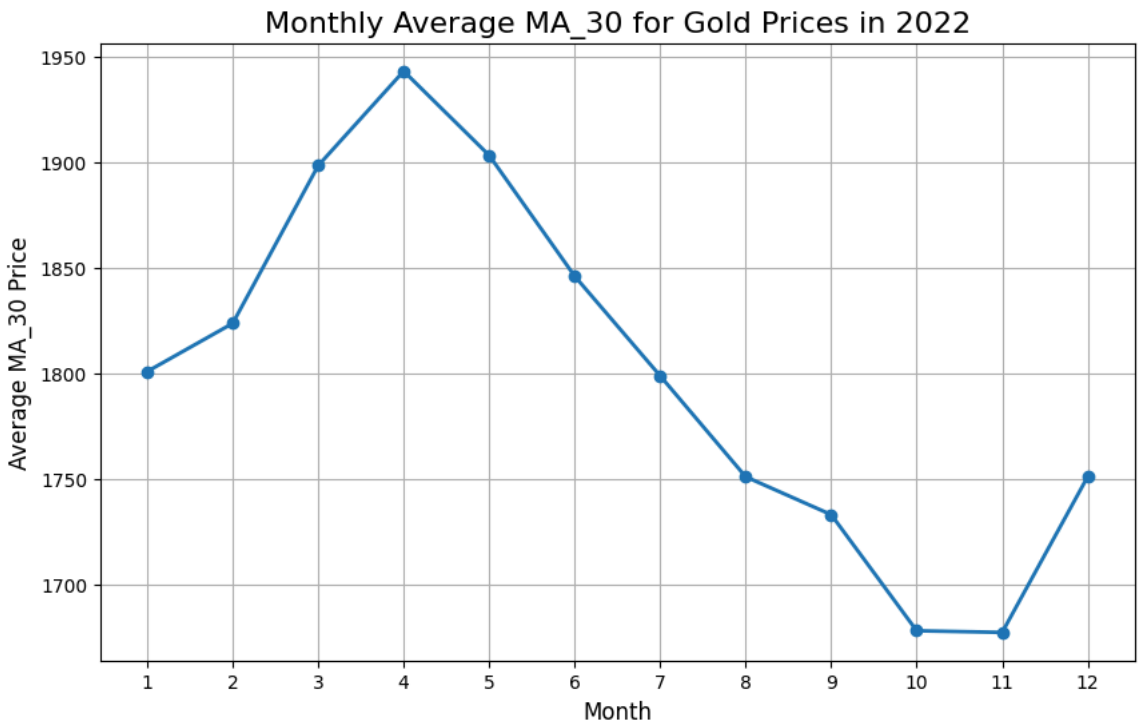
Show hidden output

```
# ma_30 of each monthin 2022
df_2022.groupby('month')['ma_30'].mean()
```

month	ma_30
1	1800.883991
2	1823.691571
3	1898.415498
4	1943.159992
5	1903.246512
6	1846.045877
7	1798.817667
8	1751.032026
9	1733.090001
10	1678.098254
11	1677.320161
12	1751.070485

dtype: float64

```
# حساب المتوسط الشهري
monthly_ma30_2022 = df_2022.groupby('month')['ma_30'].mean()

# الرسم
plt.figure(figsize=(10,6))
plt.plot(
    monthly_ma30_2022.index,
    monthly_ma30_2022.values,
    marker='o',
    linewidth=2
)

# العناوين
plt.title('Monthly Average MA_30 for Gold Prices in 2022', fontsize=16)
plt.xlabel('Month', fontsize=12)
plt.ylabel('Average MA_30 Price', fontsize=12)
plt.xticks(range(1,13))

# شبكة خفيفة
plt.grid(True)

plt.show()
```




```
# ma_30 of each monthin 2023
df_2023.groupby('month')['ma_30'].mean()
```

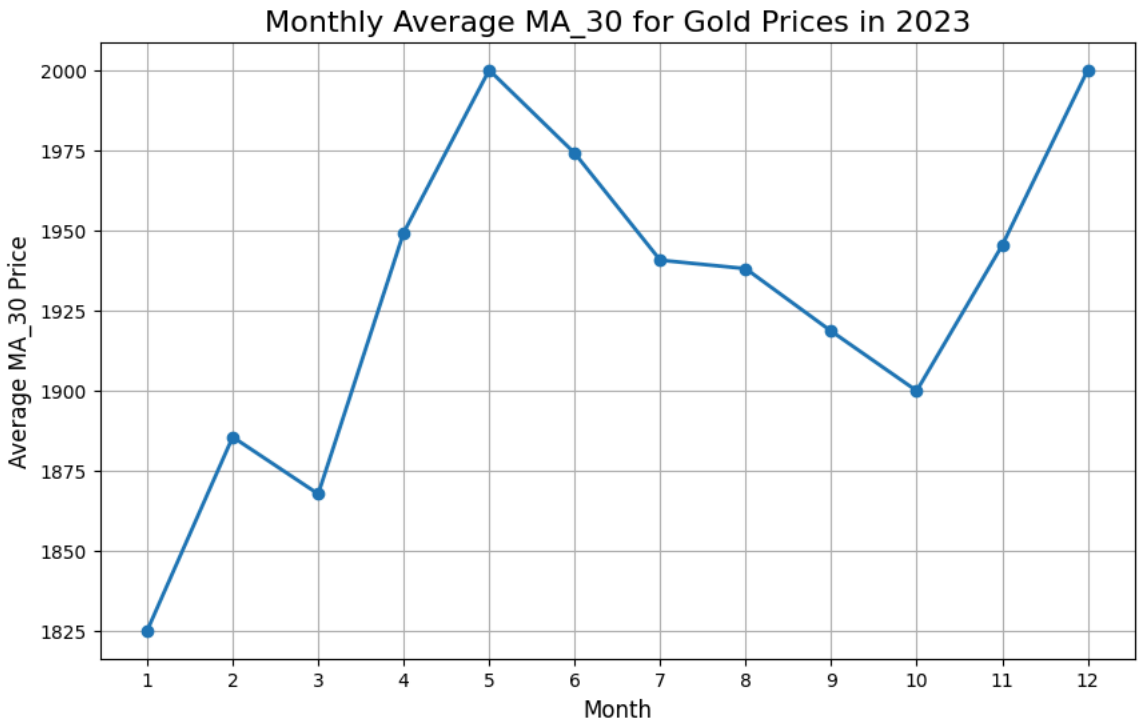
month	ma_30
1	1825.137832
2	1885.610524
3	1867.817826
4	1949.490353
5	2000.271968
6	1974.276508
7	1940.863994
8	1938.215506
9	1918.682999
10	1900.041209
11	1945.492060
12	2000.230831

dtype: float64

Gold prices in 2022 peaked during Q2, followed by a prolonged downtrend until Q4, with early signs of recovery toward the end of the year.

```
# حساب المتوسط الشهري
monthly_ma30_2023 = df_2023.groupby('month')['ma_30'].mean()

# الرسم
plt.figure(figsize=(10,6))
plt.plot(
    monthly_ma30_2023.index,
    monthly_ma30_2023.values,
```

```
    marker='o',
    linewidth=2
)

# العناوين
plt.title('Monthly Average MA_30 for Gold Prices in 2023', fontsize=16)
plt.xlabel('Month', fontsize=12)
plt.ylabel('Average MA_30 Price', fontsize=12)
plt.xticks(range(1,13))

# شبكة خفيفة
plt.grid(True)

plt.show()
```



in 2023 Gold maintained a stable long-term upward trend throughout the year, with only a mild mid-year correction and strong recovery toward year-end.

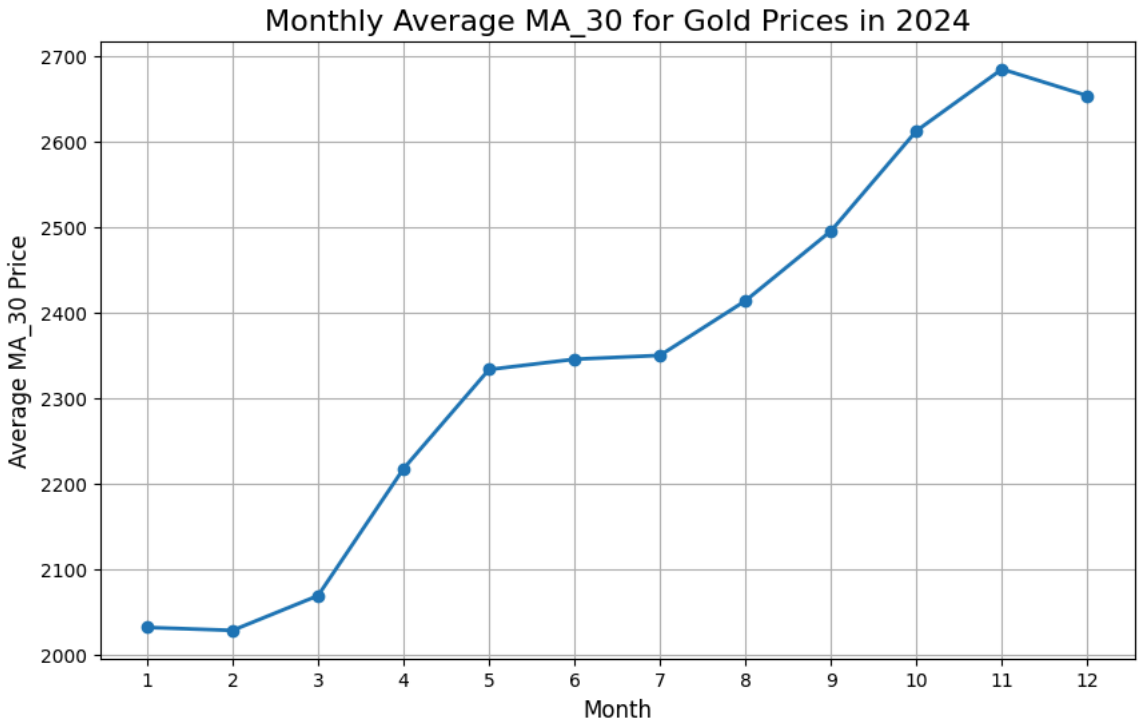
```
# ma_30 of each monthin 2024
df_2024.groupby('month')['ma_30'].mean()
```

Show hidden output

In 2024, gold showed a strong and sustained upward trend in its 30-day moving average, indicating consistent price growth throughout the year, with only a slight slowdown toward the year-end.

```
# حساب المتوسط الشهري
monthly_ma30_2024 = df_2024.groupby('month')['ma_30'].mean()

# الرسم
plt.figure(figsize=(10,6))
plt.plot(
    monthly_ma30_2024.index,
    monthly_ma30_2024.values,
    marker='o',
    linewidth=2
)

# العناوين
plt.title('Monthly Average MA_30 for Gold Prices in 2024', fontsize=16)
plt.xlabel('Month', fontsize=12)
```

```
plt.ylabel('Average MA_30 Price', fontsize=12)
plt.xticks(range(1,13))

# شبكة خفيفة
plt.grid(True)

plt.show()
```



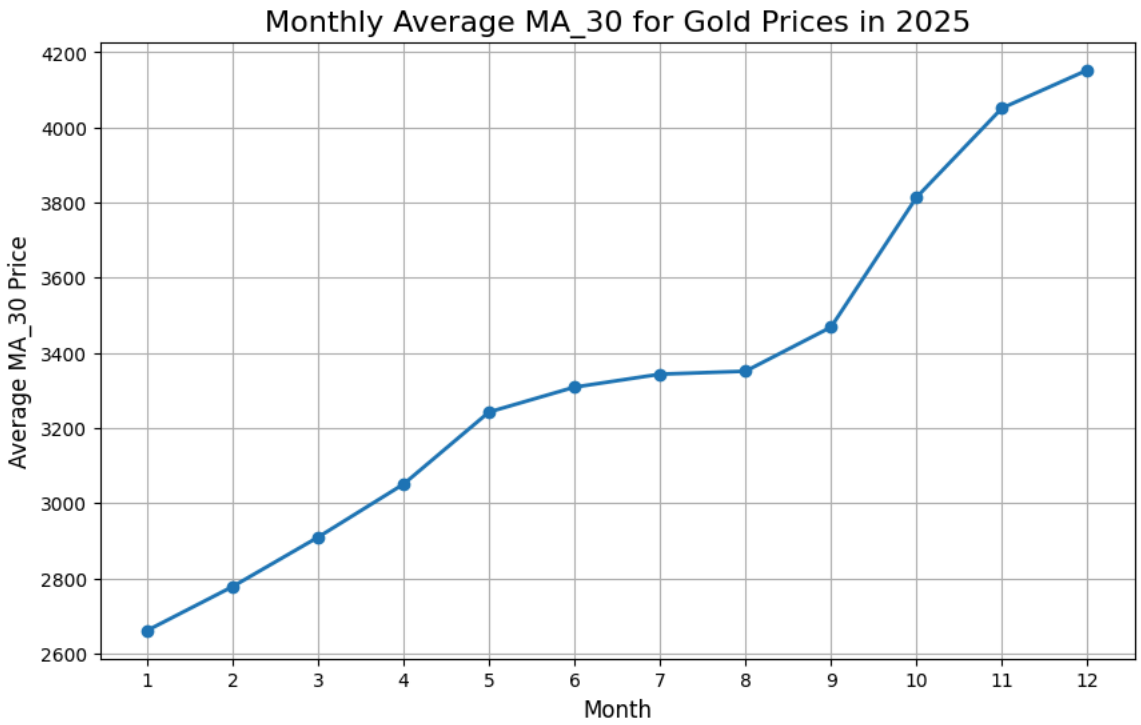

```
# ma_30 of each monthin 2025
df_2025.groupby('month')['ma_30'].mean()
```

month	ma_30
1	2660.988089
2	2777.796679
3	2909.384775
4	3050.629198
5	3242.640623
6	3308.903168
7	3343.369130
8	3351.304595
9	3468.603477
10	3814.122609
11	4051.563136
12	4152.965889

dtype: float64

```
# حساب المتوسط الشهري
monthly_ma30_2025 = df_2025.groupby('month')['ma_30'].mean()

# الرسم
plt.figure(figsize=(10,6))
plt.plot(
    monthly_ma30_2025.index,
```

```
    monthly_ma30_2025.values,
    marker='o',
    linewidth=2
)

# العناوين
plt.title('Monthly Average MA_30 for Gold Prices in 2025', fontsize=16)
plt.xlabel('Month', fontsize=12)
plt.ylabel('Average MA_30 Price', fontsize=12)
plt.xticks(range(1,13))

# شبكة خفيفة
plt.grid(True)

plt.show()
```



2025 is same like 2024

comprition based on rsi over years

```
df_2021.groupby('month')['rsi'].mean()
#The average Relative Strength Index (RSI) began to decline in June (36.9), then unexpectedly improved to reach a high of 55.6
```

month	rsi
6	36.882765
7	48.099289
8	47.145341
9	45.668086
10	51.998041
11	55.611281
12	48.491607

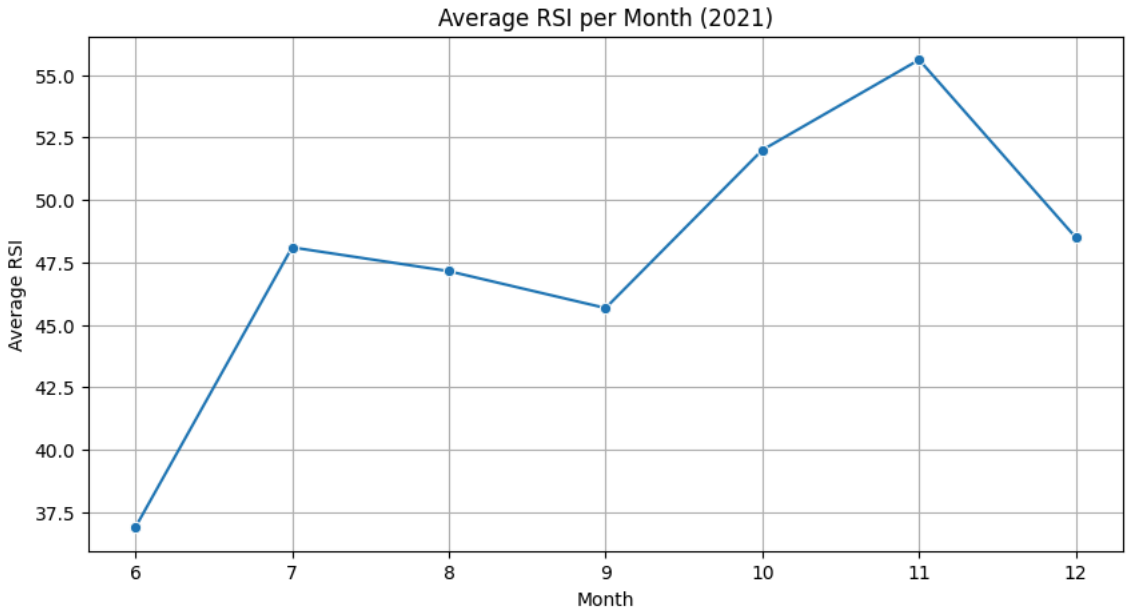
dtype: float64

```
def rsi_calc(d_f,year1):
  monthly_rsi=d_f.groupby('month',as_index=False)['rsi'].mean()
  plt.figure(figsize=(10, 5))
  sns.lineplot(data=monthly_rsi, x='month', y='rsi', marker='o')
  plt.title(f'Average RSI per Month ({year1})')
  plt.xlabel('Month')
  plt.ylabel('Average RSI')
  plt.grid(True)
  plt.show()
  print('hi')
```

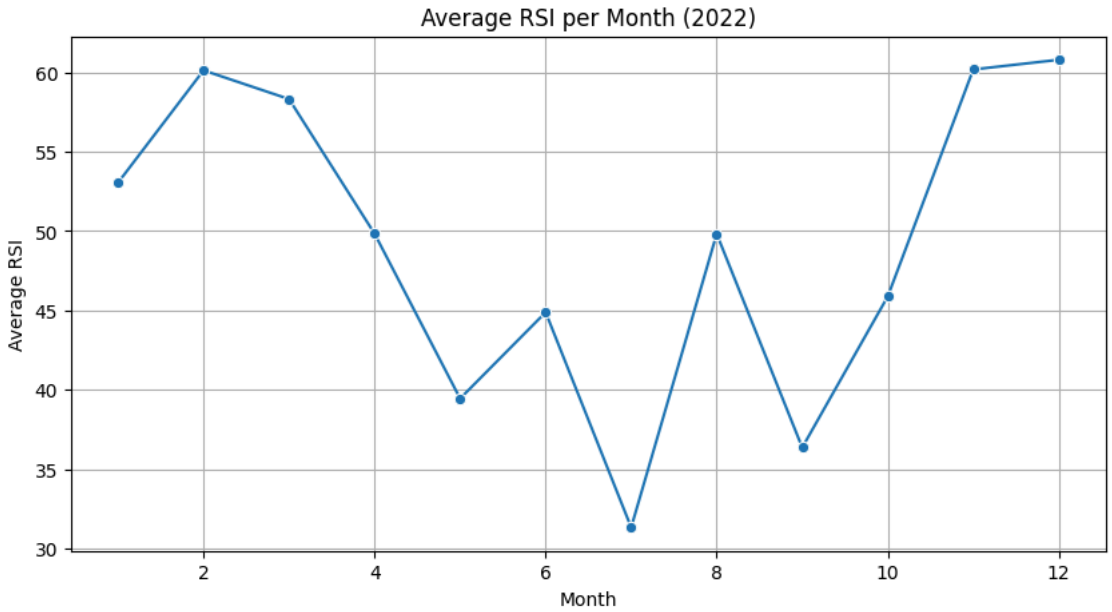
```
df_2022.groupby('month',as_index=False)['rsi'].mean()
```

	month	rsi
0	1	53.047869
1	2	60.118233
2	3	58.310410
3	4	49.836042
4	5	39.464520
5	6	44.893110
6	7	31.350484
7	8	49.826766
8	9	36.379845
9	10	45.889354
10	11	60.176128
11	12	60.785519

```
# حساب المتوسط
monthly_rsi = df_2021.groupby('month', as_index=False)['rsi'].mean()

# الرسم
plt.figure(figsize=(10, 5))
sns.lineplot(data=monthly_rsi, x='month', y='rsi', marker='o')

plt.title('Average RSI per Month (2021)')
plt.xlabel('Month')
plt.ylabel('Average RSI')
plt.grid(True)
plt.show()
```




```
rsi_calc(df_2022,'2022')
```

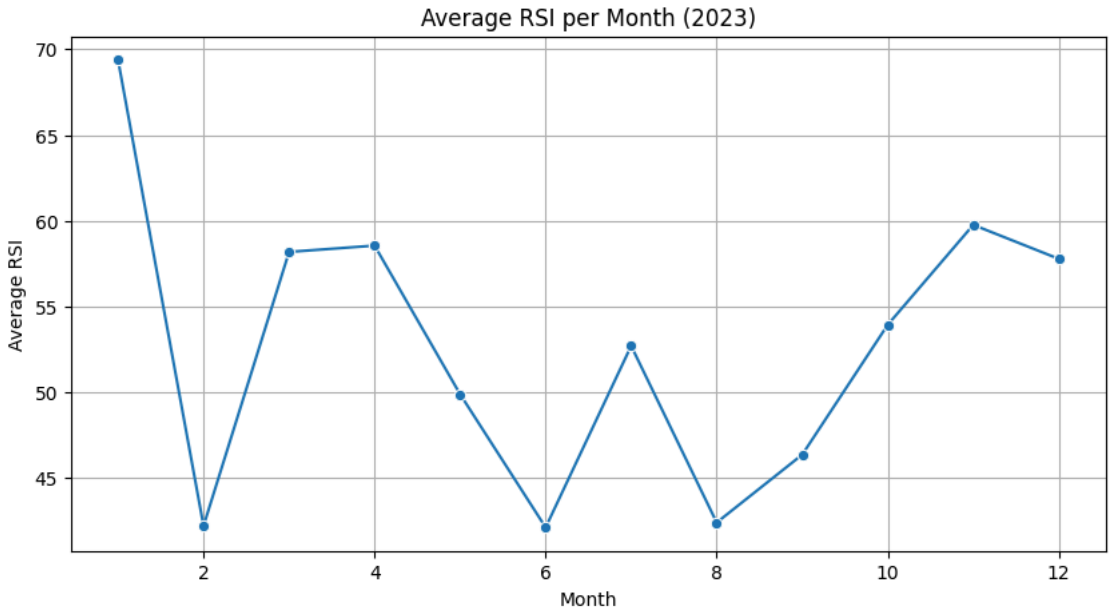



```
hi
```

```
df_2023.groupby('month')['rsi'].mean()
```

Show hidden output

```
rsi_calc(df_2023,'2023')
```



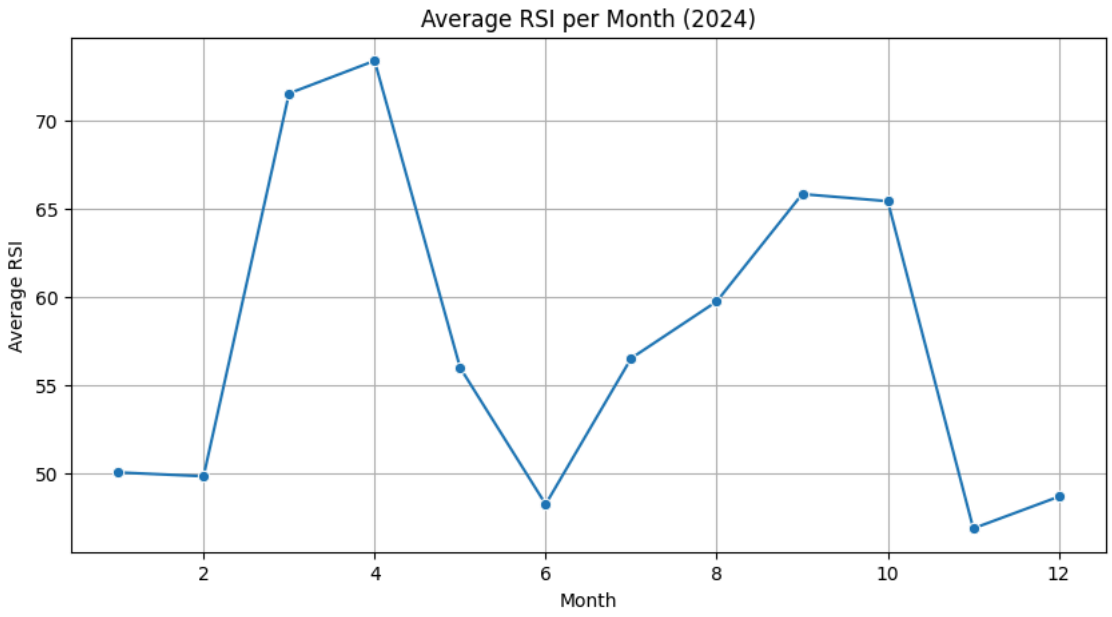

```
hi
```

In 2023, the same fluctuation occurred, but the year began with higher than normal buying momentum, which was temporary and decreased completely in February. The fluctuation continued from May to August.

```
df_2024.groupby('month')['rsi'].mean()
```

Show hidden output

```
rsi_calc(df_2024,'2024')
```



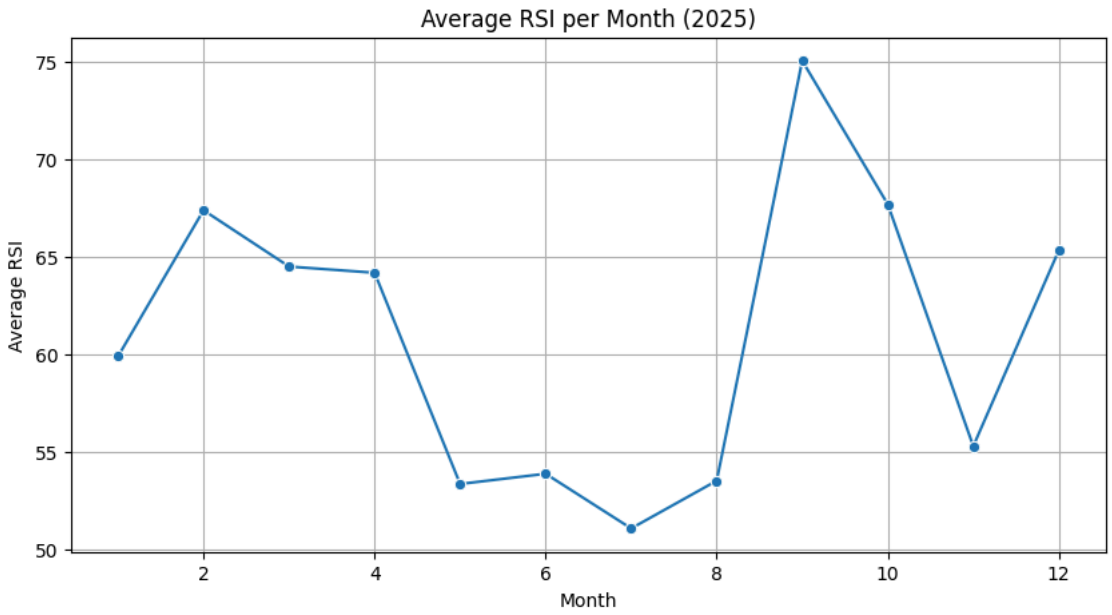

```
hi
```

```
df_2025.groupby('month')['rsi'].mean()
```

month	rsi
1	59.891163
2	67.370428
3	64.489079
4	64.171725
5	53.361369
6	53.883495
7	51.085692
8	53.527126
9	75.021653
10	67.679947
11	55.296903
12	65.316564

dtype: float64

```
rsi_calc(df_2025,'2025')
```




```
hi
```

"over the years From February to March, there is a sharp and quick increase in RSI, but it soon returns to normal levels. Then, during July, August, and September, RSI experiences a gradual rise."

volume rate over monthes in each year

```
df.sample()
```

	date	adj close	close	high	low	open	volume	ma_7	ma_30	ma_90	daily_return
1055	2025-08-21	3336.899902	3336.899902	3349.399902	3326.5	3349.399902	142	3336.471366	3354.176652	3329.60666	-0.001944

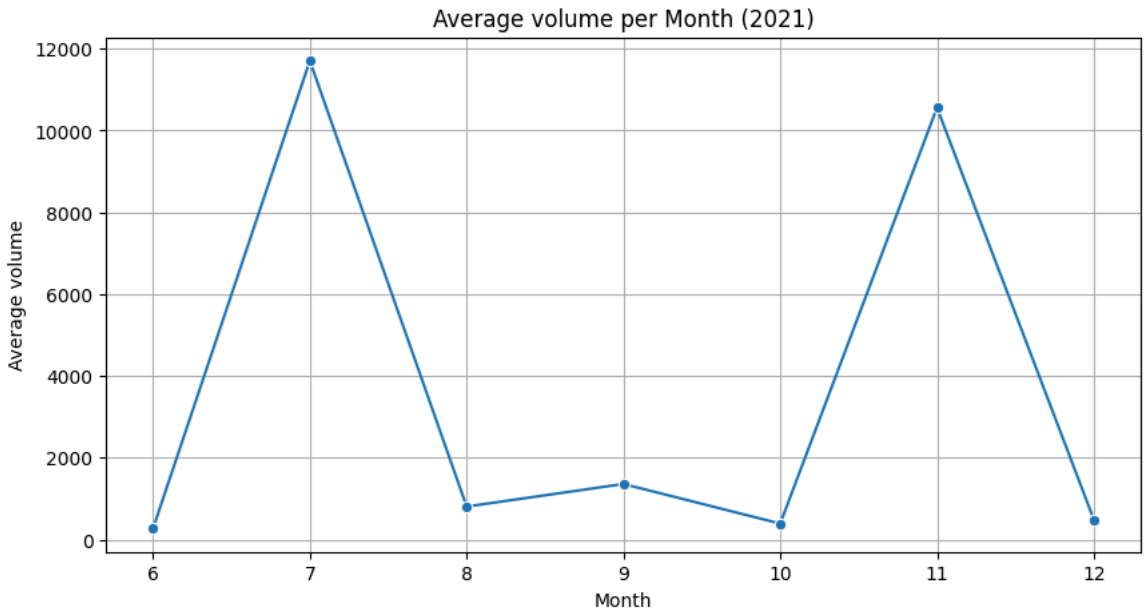
```
def vol_calc(d_f,year1):
  monthly_vol=d_f.groupby('month',as_index=False)['volume'].mean()
  plt.figure(figsize=(10, 5))
  sns.lineplot(data=monthly_vol, x='month', y='volume', marker='o')
  plt.title(f'Average volume per Month ({year1})')
  plt.xlabel('Month')
  plt.ylabel('Average volume')
  plt.grid(True)
  plt.show()
  print('hi')
```

```
df_2021.groupby('month')['volume'].mean()
```

month	volume
6	272.857143
7	11705.428571
8	807.136364
9	1359.190476
10	390.857143
11	10559.428571
12	491.045455

dtype: float64

```
vol_calc(df_2021,'2021')
```

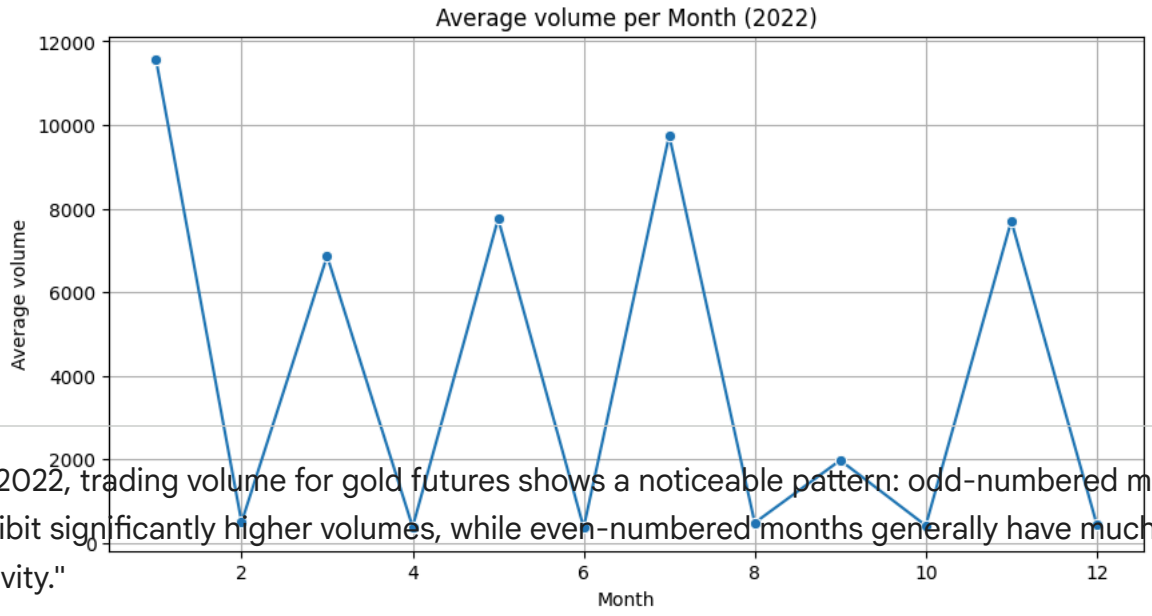



```
hi
```

```
df_2022.groupby('month')['volume'].mean()
```

Show hidden output

```
vol_calc(df_2022,'2022')
```



"In 2022, trading volume for gold futures shows a noticeable pattern: odd-numbered months exhibit significantly higher volumes, while even-numbered months generally have much lower activity."

hi

```
df_2023.groupby('month')['volume'].mean()
```

Show hidden output

```
vol_calc(df_2023,'2023')
```

Show hidden output

```
df_2024.groupby('month')['volume'].mean()
```

Show hidden output

```
df_2025.groupby('month')['volume'].mean()
```

Show hidden output

```
vol_calc(df_2024,'2024')
```