```
import logging
import pandas as pd
import numpy as np
from numpy import random
import nltk
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
import re
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
import numpy as np
import scipy.stats as spstats
import seaborn as sns


%matplotlib inline
mpl.style.reload_library()
mpl.style.use('classic')
mpl.rcParams['figure.facecolor'] = (1, 1, 1, 0)
mpl.rcParams['figure.figsize'] = [6.0, 4.0]
mpl.rcParams['figure.dpi'] = 40


! pip install seaborn
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
Requirement already satisfied: seaborn in /usr/local/lib/python3.7/dist-packages (0.11.
Requirement already satisfied: matplotlib>=2.2 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: pandas>=0.23 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: scipy>=1.0 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/l
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from
```

Gold Dataset

Load Glod Dataset and set Date as Index.

```
#Gold price data
goldDF = pd.read_csv("/content/drive/MyDrive/Data Sets/VargoldCommodity/lbma_gold_am_usd_1967
goldDF = goldDF.set_index('Date')
goldDF.head()
```

Date	GoldPrice
2022-03-31	1924.10
2022-03-30	1917.80
2022-03-29	1911.05
2022-03-28	1927.00
2022-03-25	1956.65

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mou
```

Check for nulls

```
goldDF.isnull().sum() ## missing values
```

```
GoldPrice    0
dtype: int64
```

```
goldDF.info()
```

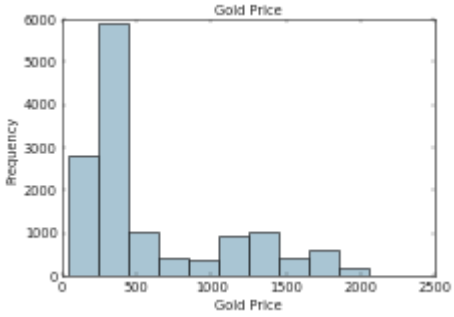
```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 13712 entries, 2022-03-31 to 1968-02-01
Data columns (total 1 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   GoldPrice  13712 non-null  float64
dtypes: float64(1)
memory usage: 214.2 KB
```

```
goldDF.columns
```

```
Index(['GoldPrice'], dtype='object')
```

```
fig, ax = plt.subplots()
goldDF['GoldPrice'].hist(color='#A9C5D3', edgecolor='black',
                         grid=False)
ax.set_title('Gold Price', fontsize=12)
ax.set_xlabel('Gold Price', fontsize=12)
ax.set_ylabel('Frequency', fontsize=12)
```

```
Text(0, 0.5, 'Frequency')
```



Create Gold Price Bins - divide price into five quantiles (0, 25%, 50%, 75%, and 100%)

```
goldDF['GoldPrice_bin_round'] = np.array(np.floor(
                                np.array(goldDF['GoldPrice']) / 10.))
goldDF[['GoldPrice', 'GoldPrice_bin_round']].iloc[1071:2076]
```

Date	GoldPrice	GoldPrice_bin_round
2018-08-01	1318.80	131.0
2018-05-01	1317.90	131.0
2018-04-01	1313.70	131.0
2018-03-01	1314.60	131.0
2018-02-01	1312.80	131.0
2017-12-29	1296.50	129.0
2017-12-28	1291.60	129.0
2017-12-27	1285.40	128.0
2017-12-22	1268.05	126.0
2017-12-21	1265.85	126.0
2017-12-20	1265.95	126.0
2017-12-19	1263.10	126.0
2017-12-18	1258.65	125.0
2017-12-15	1257.25	125.0
2017-12-14	1255.60	125.0
2017-12-13	1241.60	124.0
2017-12-12	1243.40	124.0
2017-11-12	1251.40	125.0
2017-08-12	1245.85	124.0
2017-07-12	1256.80	125.0
2017-06-12	1268.55	126.0
2017-05-12	1275.90	127.0
2017-04-12	1279.10	127.0
2017-01-12	1277.25	127.0
2017-11-30	1282.15	128.0
2017-11-29	1294.85	129.0
2017-11-28	1293.90	129.0
2017-11-27	1294.70	129.0
2017-11-24	1289.15	128.0

2017-11-23	1290.15	129.0
2017-11-22	1283.95	128.0
2017-11-21	1280.00	128.0
2017-11-20	1292.35	129.0
2017-11-17	1283.85	128.0
2017-11-16	1277.70	127.0
2017-11-15	1285.70	128.0
2017-11-14	1273.70	127.0
2017-11-13	1278.40	127.0
2017-10-11	1284.45	128.0
2017-09-11	1284.00	128.0
2017-08-11	1282.25	128.0
2017-07-11	1276.35	127.0
2017-06-11	1271.60	127.0
2017-03-11	1275.30	127.0
2017-02-11	1276.40	127.0
2017-01-11	1279.25	127.0
2017-10-31	1274.40	127.0
2017-10-30	1272.75	127.0
2017-10-27	1267.80	126.0
2017-10-26	1278.00	127.0
2017-10-25	1273.00	127.0
2017-10-24	1278.30	127.0
2017-10-23	1275.25	127.0
2017-10-20	1280.25	128.0
2017-10-19	1283.40	128.0
2017-10-18	1280.65	128.0
2017-10-17	1289.70	128.0
2017-10-16	1305.15	130.0
2017-10-13	1293.90	129.0
2017-12-10	1294.45	129.0
2017-11-10	1290.20	129.0

2017-10-10	1289.60	128.0
2017-09-10	1282.15	128.0
2017-06-10	1268.20	126.0
2017-05-10	1278.40	127.0
2017-04-10	1275.55	127.0
2017-03-10	1270.70	127.0
2017-02-10	1273.10	127.0
2017-09-29	1286.95	128.0
2017-09-28	1284.30	128.0
2017-09-27	1291.30	129.0
2017-09-26	1306.90	130.0
2017-09-25	1295.50	129.0
2017-09-22	1297.00	129.0
2017-09-21	1297.35	129.0
2017-09-20	1314.90	131.0
2017-09-19	1308.45	130.0
2017-09-18	1314.40	131.0
2017-09-15	1325.00	132.0
2017-09-14	1323.00	132.0
2017-09-13	1332.25	133.0
2017-12-09	1326.25	132.0
2017-11-09	1338.75	133.0
2017-08-09	1350.90	135.0
2017-07-09	1340.45	134.0
2017-06-09	1340.15	134.0
2017-05-09	1331.15	133.0
2017-04-09	1334.60	133.0
2017-01-09	1318.40	131.0
2017-08-31	1305.80	130.0
2017-08-30	1310.60	131.0
2017-08-29	1323.40	132.0
2017-08-25	1287.05	128.0

2017-08-24	1285.90	128.0
2017-08-23	1286.45	128.0
2017-08-22	1285.10	128.0
2017-08-21	1287.60	128.0
2017-08-18	1295.25	129.0
2017-08-17	1285.90	128.0
2017-08-16	1270.15	127.0
2017-08-15	1274.60	127.0
2017-08-14	1281.10	128.0
2017-11-08	1288.30	128.0
2017-10-08	1278.90	127.0
2017-09-08	1267.95	126.0
2017-08-08	1261.45	126.0
2017-07-08	1257.55	125.0
2017-04-08	1269.30	126.0
2017-03-08	1261.80	126.0
2017-02-08	1266.65	126.0
2017-01-08	1267.05	126.0
2017-07-31	1266.35	126.0
2017-07-28	1259.60	125.0
2017-07-27	1262.05	126.0
2017-07-26	1245.40	124.0
2017-07-25	1252.00	125.0
2017-07-24	1255.85	125.0
2017-07-21	1247.25	124.0
2017-07-20	1236.55	123.0
2017-07-19	1239.85	123.0
2017-07-18	1237.10	123.0
2017-07-17	1229.85	122.0
2017-07-14	1218.95	121.0
2017-07-13	1221.40	122.0

2017-12-07	1219.40	121.0
2017-11-07	1211.90	121.0
2017-10-07	1207.55	120.0
2017-07-07	1220.40	122.0
2017-06-07	1224.30	122.0
2017-05-07	1221.90	122.0
2017-04-07	1224.25	122.0
2017-03-07	1235.20	123.0
2017-06-30	1243.25	124.0
2017-06-29	1246.60	124.0
2017-06-28	1251.60	125.0
2017-06-27	1250.40	125.0
2017-06-26	1240.85	124.0
2017-06-23	1256.30	125.0
2017-06-22	1251.40	125.0
2017-06-21	1247.05	124.0
2017-06-20	1246.50	124.0
2017-06-19	1251.10	125.0
2017-06-16	1256.60	125.0
2017-06-15	1260.25	126.0
2017-06-14	1268.25	126.0
2017-06-13	1261.30	126.0
2017-12-06	1269.25	126.0
2017-09-06	1274.25	127.0
2017-08-06	1284.80	128.0
2017-07-06	1292.70	129.0
2017-06-06	1287.85	128.0
2017-05-06	1280.70	128.0
2017-02-06	1260.95	126.0
2017-01-06	1266.15	126.0
2017-05-31	1263.80	126.0
2017-05-30	1262.80	126.0

2017-05-26	1265.00	126.0
2017-05-25	1257.10	125.0
2017-05-24	1251.35	125.0
2017-05-23	1259.90	125.0
2017-05-22	1255.25	125.0
2017-05-19	1251.85	125.0
2017-05-18	1261.35	126.0
2017-05-17	1244.60	124.0
2017-05-16	1234.05	123.0
2017-05-15	1231.50	123.0
2017-12-05	1227.90	122.0
2017-11-05	1221.00	122.0
2017-10-05	1222.95	122.0
2017-09-05	1225.15	122.0
2017-08-05	1229.70	122.0
2017-05-05	1239.40	123.0
2017-04-05	1235.85	123.0
2017-03-05	1253.95	125.0
2017-02-05	1255.80	125.0
2017-04-28	1265.55	126.0
2017-04-27	1264.30	126.0
2017-04-26	1264.95	126.0
2017-04-25	1270.50	127.0
2017-04-24	1271.80	127.0
2017-04-21	1281.50	128.0
2017-04-20	1279.90	127.0
2017-04-19	1282.05	128.0
2017-04-18	1285.00	128.0
2017-04-13	1286.10	128.0
2017-12-04	1272.30	127.0
2017-11-04	1255.70	125.0
2017-10-04	1253.60	125.0

2017-10-04	1253.60	125.0
2017-07-04	1264.30	126.0
2017-06-04	1253.75	125.0
2017-05-04	1252.50	125.0
2017-04-04	1258.65	125.0
2017-03-04	1246.25	124.0
2017-03-31	1241.70	124.0
2017-03-30	1250.90	125.0
2017-03-29	1252.90	125.0
2017-03-28	1253.65	125.0
2017-03-27	1256.90	125.0
2017-03-24	1244.00	124.0
2017-03-23	1247.90	124.0
2017-03-22	1246.10	124.0
2017-03-21	1232.05	123.0
2017-03-20	1233.00	123.0
2017-03-17	1228.75	122.0
2017-03-16	1225.60	122.0
2017-03-15	1202.25	120.0
2017-03-14	1203.55	120.0
2017-03-13	1207.80	120.0
2017-10-03	1196.55	119.0
2017-09-03	1204.60	120.0
2017-08-03	1213.30	121.0
2017-07-03	1223.70	122.0
2017-06-03	1231.15	123.0
2017-03-03	1228.75	122.0
2017-02-03	1243.30	124.0
2017-01-03	1246.05	124.0
2017-02-28	1251.90	125.0
2017-02-27	1256.25	125.0
2017-02-24	1255.35	125.0

2017-02-23	1237.35	123.0
2017-02-22	1237.50	123.0
2017-02-21	1228.70	122.0
2017-02-20	1235.35	123.0
2017-02-17	1241.40	124.0
2017-02-16	1236.75	123.0
2017-02-15	1225.15	122.0
2017-02-14	1229.65	122.0
2017-02-13	1229.40	122.0
2017-10-02	1225.75	122.0
2017-09-02	1241.75	124.0
2017-08-02	1235.60	123.0
2017-07-02	1231.00	123.0
2017-06-02	1221.85	122.0
2017-03-02	1213.05	121.0
2017-02-02	1224.05	122.0
2017-01-02	1210.00	121.0
2017-01-31	1198.80	119.0
2017-01-30	1189.85	118.0
2017-01-27	1184.20	118.0
2017-01-26	1191.55	119.0
2017-01-25	1203.50	120.0
2017-01-24	1213.30	121.0
2017-01-23	1213.75	121.0
2017-01-20	1199.10	119.0
2017-01-19	1203.35	120.0
2017-01-18	1212.50	121.0
2017-01-17	1217.50	121.0
2017-01-16	1202.75	120.0
2017-01-13	1196.35	119.0
2017-12-01	1206.65	120.0
2017-11-01	1187.55	118.0

2017-10-01	1183.20	118.0
2017-09-01	1176.10	117.0
2017-06-01	1178.00	117.0
2017-05-01	1173.05	117.0
2017-04-01	1165.90	116.0
2017-03-01	1148.65	114.0
2016-12-30	1159.10	115.0
2016-12-29	1146.80	114.0
2016-12-28	1139.75	113.0
2016-12-23	1131.00	113.0
2016-12-22	1130.55	113.0
2016-12-21	1134.40	113.0
2016-12-20	1132.75	113.0
2016-12-19	1137.60	113.0
2016-12-16	1134.85	113.0
2016-12-15	1132.45	113.0
2016-12-14	1160.95	116.0
2016-12-13	1157.35	115.0
2016-12-12	1154.40	115.0
2016-09-12	1168.90	116.0
2016-08-12	1174.75	117.0
2016-07-12	1171.25	117.0
2016-06-12	1171.15	117.0
2016-05-12	1164.90	116.0
2016-02-12	1171.65	117.0
2016-01-12	1168.75	116.0
2016-11-30	1187.40	118.0
2016-11-29	1187.30	118.0
2016-11-28	1189.10	118.0
2016-11-25	1187.50	118.0
2016-11-24	1187.25	118.0

2016-11-23	1213.25	121.0
2016-11-22	1217.55	121.0
2016-11-21	1214.95	121.0
2016-11-18	1206.10	120.0
2016-11-17	1232.00	123.0
2016-11-16	1225.70	122.0
2016-11-15	1228.90	122.0
2016-11-14	1222.60	122.0
2016-11-11	1255.65	125.0
2016-10-11	1280.90	128.0
2016-09-11	1304.55	130.0
2016-08-11	1284.00	128.0
2016-07-11	1286.80	128.0
2016-04-11	1301.70	130.0
2016-03-11	1293.00	129.0
2016-02-11	1295.85	129.0
2016-01-11	1284.40	128.0
2016-10-31	1274.20	127.0
2016-10-28	1265.90	126.0
2016-10-27	1269.30	126.0
2016-10-26	1273.90	127.0
2016-10-25	1269.30	126.0
2016-10-24	1267.00	126.0
2016-10-21	1263.95	126.0
2016-10-20	1269.20	126.0
2016-10-19	1269.75	126.0
2016-10-18	1261.65	126.0
2016-10-17	1252.70	125.0
2016-10-14	1256.15	125.0
2016-10-13	1258.00	125.0
2016-12-10	1255.70	125.0
2016-11-10	1256.40	125.0

2016-10-10	1262.10	126.0
2016-07-10	1255.00	125.0
2016-06-10	1265.50	126.0
2016-05-10	1274.00	127.0
2016-04-10	1309.15	130.0
2016-03-10	1318.65	131.0
2016-09-30	1327.90	132.0
2016-09-29	1320.85	132.0
2016-09-28	1324.80	132.0
2016-09-27	1335.85	133.0
2016-09-26	1336.30	133.0
2016-09-23	1335.90	133.0
2016-09-22	1332.45	133.0
2016-09-21	1319.60	131.0
2016-09-20	1315.40	131.0
2016-09-19	1315.05	131.0
2016-09-16	1314.25	131.0
2016-09-15	1320.10	132.0
2016-09-14	1323.20	132.0
2016-09-13	1328.50	132.0
2016-12-09	1327.50	132.0
2016-09-09	1335.65	133.0
2016-08-09	1348.00	134.0
2016-07-09	1348.75	134.0
2016-06-09	1330.05	133.0
2016-05-09	1328.30	132.0
2016-02-09	1311.50	131.0
2016-01-09	1305.70	130.0
2016-08-31	1314.45	131.0
2016-08-30	1318.85	131.0
2016-08-26	1324.90	132.0
2016-08-25	1324.50	132.0

2016-08-25	1324.50	132.0
2016-08-24	1337.30	133.0
2016-08-23	1338.50	133.0
2016-08-22	1334.30	133.0
2016-08-19	1346.85	134.0
2016-08-18	1347.10	134.0
2016-08-17	1342.75	134.0
2016-08-16	1349.10	134.0
2016-08-15	1339.20	133.0
2016-12-08	1336.70	133.0
2016-11-08	1344.55	134.0
2016-10-08	1351.85	135.0
2016-09-08	1332.90	133.0
2016-08-08	1330.00	133.0
2016-05-08	1362.60	136.0
2016-04-08	1351.15	135.0
2016-03-08	1364.40	136.0
2016-02-08	1358.15	135.0
2016-01-08	1348.85	134.0
2016-07-29	1332.50	133.0
2016-07-28	1341.30	134.0
2016-07-27	1320.80	132.0
2016-07-26	1321.25	132.0
2016-07-25	1315.00	131.0
2016-07-22	1323.20	132.0
2016-07-21	1322.00	132.0
2016-07-20	1325.60	132.0
2016-07-19	1332.20	133.0
2016-07-18	1326.15	132.0
2016-07-15	1330.50	133.0
2016-07-14	1325.70	132.0
2016-07-13	1340.25	134.0

2016-12-07	1352.85	135.0
2016-11-07	1358.25	135.0
2016-08-07	1356.10	135.0
2016-07-07	1367.10	136.0
2016-06-07	1370.00	137.0
2016-05-07	1344.75	134.0
2016-04-07	1348.75	134.0
2016-01-07	1331.75	133.0
2016-06-30	1317.00	131.0
2016-06-29	1317.75	131.0
2016-06-28	1312.00	131.0
2016-06-27	1324.60	132.0
2016-06-24	1313.85	131.0
2016-06-23	1265.75	126.0
2016-06-22	1265.00	126.0
2016-06-21	1280.80	128.0
2016-06-20	1283.25	128.0
2016-06-17	1284.50	128.0
2016-06-16	1307.00	130.0
2016-06-15	1282.00	128.0
2016-06-14	1279.40	127.0
2016-06-13	1284.10	128.0
2016-10-06	1266.60	126.0
2016-09-06	1258.35	125.0
2016-08-06	1252.40	125.0
2016-07-06	1241.10	124.0
2016-06-06	1240.55	124.0
2016-03-06	1211.00	121.0
2016-02-06	1215.50	121.0
2016-01-06	1216.25	121.0
2016-05-31	1210.50	121.0
2016-05-27	1221.25	122.0

2016-05-26	1226.65	122.0
2016-05-25	1220.75	122.0
2016-05-24	1242.65	124.0
2016-05-23	1250.40	125.0
2016-05-20	1256.50	125.0
2016-05-19	1253.75	125.0
2016-05-18	1270.90	127.0
2016-05-17	1270.10	127.0
2016-05-16	1281.00	128.0
2016-05-13	1275.15	127.0
2016-12-05	1268.30	126.0
2016-11-05	1271.80	127.0
2016-10-05	1264.85	126.0
2016-09-05	1277.75	127.0
2016-06-05	1280.25	128.0
2016-05-05	1275.75	127.0
2016-04-05	1280.30	128.0
2016-03-05	1296.50	129.0
2016-04-29	1274.50	127.0
2016-04-28	1256.60	125.0
2016-04-27	1244.75	124.0
2016-04-26	1234.50	123.0
2016-04-25	1230.85	123.0
2016-04-22	1245.40	124.0
2016-04-21	1257.65	125.0
2016-04-20	1247.75	124.0
2016-04-19	1241.70	124.0
2016-04-18	1237.70	123.0
2016-04-15	1229.75	122.0
2016-04-14	1240.30	124.0
2016-04-13	1245.75	124.0

2016-12-04	1259.20	125.0
2016-11-04	1247.25	124.0
2016-08-04	1235.00	123.0
2016-07-04	1237.50	123.0
2016-06-04	1225.75	122.0
2016-05-04	1231.50	123.0
2016-04-04	1215.00	121.0
2016-01-04	1232.10	123.0
2016-03-31	1233.60	123.0
2016-03-30	1238.20	123.0
2016-03-29	1216.45	121.0
2016-03-24	1216.45	121.0
2016-03-23	1232.20	123.0
2016-03-22	1251.80	125.0
2016-03-21	1244.25	124.0
2016-03-18	1254.50	125.0
2016-03-17	1269.60	126.0
2016-03-16	1233.10	123.0
2016-03-15	1233.60	123.0
2016-03-14	1256.55	125.0
2016-11-03	1262.25	126.0
2016-10-03	1247.25	124.0
2016-09-03	1258.25	125.0
2016-08-03	1274.10	127.0
2016-07-03	1267.60	126.0
2016-04-03	1271.50	127.0
2016-03-03	1241.95	124.0
2016-02-03	1229.35	122.0
2016-01-03	1240.00	124.0
2016-02-29	1234.15	123.0
2016-02-26	1231.00	123.0
2016-02-25	1235.40	123.0

2016-02-24	1232.25	123.0
2016-02-23	1218.75	121.0
2016-02-22	1203.65	120.0
2016-02-19	1221.50	122.0
2016-02-18	1204.40	120.0
2016-02-17	1202.40	120.0
2016-02-16	1212.00	121.0
2016-02-15	1208.45	120.0
2016-12-02	1239.50	123.0
2016-11-02	1223.25	122.0
2016-10-02	1183.40	118.0
2016-09-02	1188.90	118.0
2016-08-02	1173.40	117.0
2016-05-02	1158.50	115.0
2016-04-02	1146.25	114.0
2016-03-02	1130.00	113.0
2016-02-02	1123.60	112.0
2016-01-02	1122.00	112.0
2016-01-29	1112.90	111.0
2016-01-28	1119.00	111.0
2016-01-27	1116.50	111.0
2016-01-26	1114.70	111.0
2016-01-25	1103.70	110.0
2016-01-22	1097.65	109.0
2016-01-21	1096.80	109.0
2016-01-20	1093.20	109.0
2016-01-19	1087.00	108.0
2016-01-18	1090.45	109.0
2016-01-15	1081.10	108.0
2016-01-14	1090.75	109.0
2016-01-13	1081.80	108.0
2016-12-01	1094.85	109.0

2016-11-01	1104.70	110.0
2016-08-01	1097.45	109.0
2016-07-01	1096.00	109.0
2016-06-01	1083.85	108.0
2016-05-01	1078.00	107.0
2016-04-01	1072.70	107.0
2015-12-31	1062.25	106.0
2015-12-30	1067.80	106.0
2015-12-29	1067.25	106.0
2015-12-24	1071.90	107.0
2015-12-23	1071.50	107.0
2015-12-22	1077.00	107.0
2015-12-21	1071.15	107.0
2015-12-18	1055.25	105.0
2015-12-17	1065.85	106.0
2015-12-16	1065.75	106.0
2015-12-15	1069.15	106.0
2015-12-14	1068.00	106.0
2015-11-12	1067.20	106.0
2015-10-12	1072.00	107.0
2015-09-12	1078.40	107.0
2015-08-12	1071.75	107.0
2015-07-12	1082.70	108.0
2015-04-12	1063.00	106.0
2015-03-12	1050.60	105.0
2015-02-12	1066.90	106.0
2015-01-12	1069.25	106.0
2015-11-30	1055.65	105.0
2015-11-27	1064.65	106.0
2015-11-26	1070.50	107.0
2015-11-25	1072.20	107.0

2015-11-24	1073.00	107.0
2015-11-23	1068.35	106.0
2015-11-20	1085.15	108.0
2015-11-19	1073.10	107.0
2015-11-18	1070.50	107.0
2015-11-17	1080.80	108.0
2015-11-16	1094.50	109.0
2015-11-13	1083.75	108.0
2015-12-11	1087.60	108.0
2015-11-11	1088.60	108.0
2015-10-11	1092.50	109.0
2015-09-11	1095.60	109.0
2015-06-11	1107.70	110.0
2015-05-11	1107.30	110.0
2015-04-11	1118.00	111.0
2015-03-11	1130.90	113.0
2015-02-11	1135.80	113.0
2015-10-30	1147.75	114.0
2015-10-29	1159.00	115.0
2015-10-28	1171.50	117.0
2015-10-27	1165.75	116.0
2015-10-26	1168.50	116.0
2015-10-23	1171.55	117.0
2015-10-22	1166.45	116.0
2015-10-21	1174.40	117.0
2015-10-20	1173.70	117.0
2015-10-19	1171.65	117.0
2015-10-16	1176.35	117.0
2015-10-15	1183.35	118.0
2015-10-14	1173.70	117.0
2015-10-13	1154.40	115.0
2015-12-10	1164.20	116.0

2015-09-10	1151.50	115.0
2015-08-10	1143.30	114.0
2015-07-10	1147.90	114.0
2015-06-10	1136.90	113.0
2015-05-10	1134.35	113.0
2015-02-10	1106.30	110.0
2015-01-10	1114.20	111.0
2015-09-30	1122.50	112.0
2015-09-29	1124.60	112.0
2015-09-28	1137.50	113.0
2015-09-25	1145.50	114.0
2015-09-24	1134.45	113.0
2015-09-23	1124.60	112.0
2015-09-22	1129.30	112.0
2015-09-21	1136.85	113.0
2015-09-18	1136.00	113.0
2015-09-17	1118.15	111.0
2015-09-16	1109.75	110.0
2015-09-15	1105.50	110.0
2015-09-14	1108.00	110.0
2015-11-09	1106.35	110.0
2015-10-09	1107.75	110.0
2015-09-09	1122.30	112.0
2015-08-09	1120.85	112.0
2015-07-09	1121.00	112.0
2015-04-09	1125.00	112.0
2015-03-09	1130.05	113.0
2015-02-09	1140.00	114.0
2015-01-09	1141.90	114.0
2015-08-28	1125.50	112.0
2015-08-27	1128.50	112.0

2015-08-26	1134.40	113.0
2015-08-25	1154.25	115.0
2015-08-24	1153.50	115.0
2015-08-21	1149.35	114.0
2015-08-20	1137.95	113.0
2015-08-19	1123.20	112.0
2015-08-18	1119.15	111.0
2015-08-17	1117.30	111.0
2015-08-14	1116.75	111.0
2015-08-13	1117.35	111.0
2015-12-08	1116.80	111.0
2015-11-08	1113.25	111.0
2015-10-08	1094.80	109.0
2015-07-08	1091.35	109.0
2015-06-08	1085.00	108.0
2015-05-08	1086.50	108.0
2015-04-08	1092.60	109.0
2015-03-08	1093.00	109.0
2015-07-31	1080.05	108.0
2015-07-30	1085.65	108.0
2015-07-29	1096.75	109.0
2015-07-28	1095.60	109.0
2015-07-27	1098.60	109.0
2015-07-24	1083.75	108.0
2015-07-23	1101.65	110.0
2015-07-22	1096.80	109.0
2015-07-21	1108.00	110.0
2015-07-20	1115.00	111.0
2015-07-17	1143.00	114.0
2015-07-16	1145.10	114.0
2015-07-15	1154.75	115.0
2015-07-14	1153.20	115.0

2015-07-13	1154.95	115.0
2015-10-07	1162.40	116.0
2015-09-07	1162.10	116.0
2015-08-07	1154.25	115.0
2015-07-07	1166.25	116.0
2015-06-07	1164.25	116.0
2015-03-07	1168.25	116.0
2015-02-07	1164.30	116.0
2015-01-07	1171.70	117.0
2015-06-30	1175.00	117.0
2015-06-29	1176.50	117.0
2015-06-26	1174.40	117.0
2015-06-25	1174.60	117.0
2015-06-24	1175.75	117.0
2015-06-23	1183.35	118.0
2015-06-22	1193.70	119.0
2015-06-19	1198.15	119.0
2015-06-18	1198.00	119.0
2015-06-17	1178.50	117.0
2015-06-16	1182.10	118.0
2015-06-15	1178.25	117.0
2015-12-06	1179.25	117.0
2015-11-06	1180.50	118.0
2015-10-06	1186.00	118.0
2015-09-06	1181.00	118.0
2015-08-06	1173.40	117.0
2015-05-06	1175.90	117.0
2015-04-06	1182.45	118.0
2015-03-06	1186.60	118.0
2015-02-06	1188.75	118.0
2015-01-06	1187.30	118.0
2015-05-29	1190.40	119.0

2015-05-28	1189.45	118.0
2015-05-27	1187.85	118.0
2015-05-26	1194.00	119.0
2015-05-22	1211.00	121.0
2015-05-21	1209.60	120.0
2015-05-20	1206.75	120.0
2015-05-19	1219.65	121.0
2015-05-18	1228.15	122.0
2015-05-15	1216.30	121.0
2015-05-14	1214.75	121.0
2015-05-13	1193.00	119.0
2015-12-05	1184.45	118.0
2015-11-05	1184.75	118.0
2015-08-05	1185.25	118.0
2015-07-05	1183.00	118.0
2015-06-05	1191.25	119.0
2015-05-05	1187.40	118.0
2015-01-05	1179.00	117.0
2015-04-30	1204.30	120.0
2015-04-29	1204.80	120.0
2015-04-28	1201.40	120.0
2015-04-27	1182.75	118.0
2015-04-24	1192.15	119.0
2015-04-23	1187.75	118.0
2015-04-22	1202.40	120.0
2015-04-21	1197.70	119.0
2015-04-20	1203.25	120.0
2015-04-17	1204.55	120.0
2015-04-16	1204.60	120.0
2015-04-15	1189.85	118.0
2015-04-14	1191.45	119.0

2015-04-13	1197.85	119.0
2015-10-04	1201.90	120.0
2015-09-04	1196.00	119.0
2015-08-04	1211.10	121.0
2015-07-04	1208.50	120.0
2015-02-04	1201.50	120.0
2015-01-04	1181.25	118.0
2015-03-31	1179.25	117.0
2015-03-30	1187.40	118.0
2015-03-27	1198.00	119.0
2015-03-26	1209.40	120.0
2015-03-25	1192.55	119.0
2015-03-24	1193.25	119.0
2015-03-23	1181.40	118.0
2015-03-20	1171.75	117.0
2015-03-19	1164.00	116.0
2015-03-18	1149.00	114.0
2015-03-17	1154.75	115.0
2015-03-16	1157.00	115.0
2015-03-13	1156.50	115.0
2015-12-03	1161.25	116.0
2015-11-03	1158.75	115.0
2015-10-03	1161.00	116.0
2015-09-03	1173.75	117.0
2015-06-03	1196.50	119.0
2015-05-03	1199.75	119.0
2015-04-03	1204.25	120.0
2015-03-03	1207.75	120.0
2015-02-03	1216.75	121.0
2015-02-27	1205.00	120.0
2015-02-26	1220.00	122.0
2015-02-25	1206.50	120.0

2015-02-24	1195.50	119.0
2015-02-23	1193.50	119.0
2015-02-20	1203.50	120.0
2015-02-19	1217.75	121.0
2015-02-18	1206.50	120.0
2015-02-17	1221.75	122.0
2015-02-16	1233.50	123.0
2015-02-13	1225.75	122.0
2015-12-02	1225.25	122.0
2015-11-02	1235.50	123.0
2015-10-02	1237.50	123.0
2015-09-02	1242.25	124.0
2015-06-02	1264.00	126.0
2015-05-02	1263.75	126.0
2015-04-02	1269.25	126.0
2015-03-02	1281.00	128.0
2015-02-02	1274.25	127.0
2015-01-30	1263.50	126.0
2015-01-29	1275.50	127.0
2015-01-28	1287.00	128.0
2015-01-27	1279.00	127.0
2015-01-26	1282.75	128.0
2015-01-23	1293.50	129.0
2015-01-22	1287.00	128.0
2015-01-21	1298.00	129.0
2015-01-20	1292.25	129.0
2015-01-19	1275.50	127.0
2015-01-16	1258.25	125.0
2015-01-15	1235.25	123.0
2015-01-14	1228.75	122.0
2015-01-13	1239.00	123.0
2015-12-01	1222.00	122.0

2015-12-01	1222.00	122.0
2015-09-01	1211.25	121.0
2015-08-01	1206.50	120.0
2015-07-01	1213.75	121.0
2015-06-01	1211.00	121.0
2015-05-01	1192.00	119.0
2015-02-01	1184.25	118.0
2014-12-31	1199.25	119.0
2014-12-30	1186.50	118.0
2014-12-29	1194.00	119.0
2014-12-24	1177.00	117.0
2014-12-23	1179.50	117.0
2014-12-22	1195.25	119.0
2014-12-19	1197.50	119.0
2014-12-18	1210.75	121.0
2014-12-17	1199.00	119.0
2014-12-16	1199.25	119.0
2014-12-15	1210.75	121.0
2014-12-12	1223.50	122.0
2014-11-12	1219.50	121.0
2014-10-12	1228.25	122.0
2014-09-12	1206.50	120.0
2014-08-12	1195.25	119.0
2014-05-12	1204.50	120.0
2014-04-12	1204.00	120.0
2014-03-12	1203.25	120.0
2014-02-12	1197.00	119.0
2014-01-12	1178.75	117.0
2014-11-28	1184.50	118.0
2014-11-27	1196.50	119.0
2014-11-26	1195.75	119.0
2014-11-25	1202.25	120.0

2014-11-24	1196.00	119.0
2014-11-21	1193.25	119.0
2014-11-20	1194.00	119.0
2014-11-19	1200.75	120.0
2014-11-18	1202.00	120.0
2014-11-17	1187.00	118.0
2014-11-14	1154.00	115.0
2014-11-13	1161.00	116.0
2014-12-11	1163.25	116.0
2014-11-11	1151.25	115.0
2014-10-11	1172.00	117.0
2014-07-11	1145.00	114.0
2014-06-11	1144.50	114.0
2014-05-11	1145.25	114.0
2014-04-11	1169.25	116.0
2014-03-11	1170.75	117.0
2014-10-31	1173.25	117.0
2014-10-30	1205.75	120.0
2014-10-29	1228.00	122.0
2014-10-28	1228.25	122.0
2014-10-27	1230.50	123.0
2014-10-24	1231.75	123.0
2014-10-23	1240.50	124.0
2014-10-22	1246.75	124.0
2014-10-21	1251.75	125.0
2014-10-20	1241.00	124.0
2014-10-17	1238.00	123.0
2014-10-16	1241.00	124.0
2014-10-15	1223.50	122.0
2014-10-14	1233.00	123.0
2014-10-13	1228.00	122.0
2014-10-10	1222.25	122.0

2014-09-10	1227.50	122.0
2014-08-10	1220.00	122.0
2014-07-10	1207.50	120.0
2014-06-10	1193.25	119.0
2014-03-10	1207.50	120.0
2014-02-10	1214.50	121.0
2014-01-10	1208.50	120.0
2014-09-30	1210.00	121.0
2014-09-29	1217.75	121.0
2014-09-26	1222.25	122.0
2014-09-25	1210.50	121.0
2014-09-24	1224.00	122.0
2014-09-23	1225.00	122.0
2014-09-22	1214.00	121.0
2014-09-19	1222.50	122.0
2014-09-18	1223.00	122.0
2014-09-17	1236.50	123.0
2014-09-16	1238.75	123.0
2014-09-15	1234.75	123.0
2014-12-09	1237.25	123.0
2014-11-09	1247.00	124.0
2014-10-09	1254.25	125.0
2014-09-09	1256.00	125.0
2014-08-09	1267.25	126.0
2014-05-09	1264.00	126.0
2014-04-09	1271.00	127.0
2014-03-09	1268.50	126.0
2014-02-09	1277.75	127.0
2014-01-09	1287.25	128.0
2014-08-29	1285.75	128.0
2014-08-28	1288.00	128.0

2014-08-27	1285.00	128.0
2014-08-26	1286.50	128.0
2014-08-22	1281.00	128.0
2014-08-21	1280.50	128.0
2014-08-20	1294.50	129.0
2014-08-19	1300.25	130.0
2014-08-18	1302.75	130.0
2014-08-15	1313.60	131.0
2014-08-14	1315.00	131.0
2014-08-13	1309.25	130.0
2014-12-08	1311.00	131.0
2014-11-08	1308.25	130.0
2014-08-08	1317.50	131.0
2014-07-08	1302.00	130.0
2014-06-08	1288.50	128.0
2014-05-08	1292.75	129.0
2014-04-08	1293.50	129.0
2014-01-08	1284.50	128.0
2014-07-31	1295.00	129.0
2014-07-30	1297.50	129.0
2014-07-29	1307.50	130.0
2014-07-28	1305.00	130.0
2014-07-25	1292.50	129.0
2014-07-24	1300.00	130.0
2014-07-23	1307.50	130.0
2014-07-22	1307.00	130.0
2014-07-21	1312.75	131.0
2014-07-18	1310.25	131.0
2014-07-17	1302.75	130.0
2014-07-16	1297.50	129.0
2014-07-15	1312.00	131.0
2014-07-14	1321.25	132.0

2014-11-07	1336.50	133.0
2014-10-07	1343.25	134.0
2014-09-07	1322.50	132.0
2014-08-07	1318.25	131.0
2014-07-07	1313.25	131.0
2014-04-07	1321.50	132.0
2014-03-07	1322.50	132.0
2014-02-07	1326.75	132.0
2014-01-07	1325.75	132.0
2014-06-30	1313.00	131.0
2014-06-27	1315.25	131.0
2014-06-26	1311.50	131.0
2014-06-25	1312.00	131.0
2014-06-24	1323.00	132.0
2014-06-23	1313.50	131.0
2014-06-20	1310.00	131.0
2014-06-19	1282.00	128.0
2014-06-18	1269.00	126.0
2014-06-17	1264.50	126.0
2014-06-16	1281.75	128.0
2014-06-13	1273.00	127.0
2014-12-06	1261.75	126.0
2014-11-06	1262.50	126.0
2014-10-06	1253.50	125.0
2014-09-06	1255.00	125.0
2014-06-06	1254.00	125.0
2014-05-06	1244.75	124.0
2014-04-06	1246.00	124.0
2014-03-06	1244.25	124.0
2014-02-06	1244.75	124.0
2014-05-30	1254.00	125.0
2014-05-29	1254.00	125.0

2014-05-29	1254.00	125.0
2014-05-28	1265.25	126.0
2014-05-27	1283.00	128.0
2014-05-23	1292.00	129.0
2014-05-22	1294.50	129.0
2014-05-21	1292.00	129.0
2014-05-20	1291.50	129.0
2014-05-19	1301.00	130.0
2014-05-16	1293.75	129.0
2014-05-15	1303.75	130.0
2014-05-14	1300.25	130.0
2014-05-13	1292.75	129.0
2014-12-05	1292.75	129.0
2014-09-05	1289.00	128.0
2014-08-05	1291.25	129.0
2014-07-05	1311.00	131.0
2014-06-05	1308.50	130.0
2014-02-05	1285.00	128.0
2014-01-05	1283.00	128.0
2014-04-30	1292.00	129.0
2014-04-29	1289.75	128.0
2014-04-28	1302.00	130.0
2014-04-25	1294.25	129.0
2014-04-24	1283.50	128.0
2014-04-23	1283.50	128.0
2014-04-22	1290.75	129.0
2014-04-17	1299.25	129.0
2014-04-16	1299.00	129.0
2014-04-15	1311.50	131.0
2014-04-14	1324.50	132.0
2014-11-04	1317.25	131.0
2014-10-04	1321.50	132.0

2014-09-04	1309.75	130.0
2014-08-04	1314.75	131.0
2014-07-04	1299.00	129.0
2014-04-04	1293.50	129.0
2014-03-04	1287.25	128.0
2014-02-04	1284.00	128.0
2014-01-04	1286.50	128.0
2014-03-31	1294.00	129.0
2014-03-28	1295.75	129.0
2014-03-27	1295.00	129.0
2014-03-26	1314.50	131.0
2014-03-25	1314.75	131.0
2014-03-24	1322.00	132.0
2014-03-21	1338.50	133.0
2014-03-20	1327.00	132.0
2014-03-19	1346.00	134.0
2014-03-18	1362.50	136.0
2014-03-17	1379.00	137.0
2014-03-14	1370.00	137.0
2014-03-13	1371.00	137.0
2014-12-03	1355.75	135.0
2014-11-03	1348.00	134.0
2014-10-03	1334.25	133.0
2014-07-03	1348.25	134.0
2014-06-03	1334.25	133.0
2014-05-03	1333.50	133.0
2014-04-03	1339.50	133.0
2014-03-03	1344.25	134.0
2014-02-28	1327.75	132.0
2014-02-27	1331.00	133.0
2014-02-26	1340.00	134.0
2014-02-25	1332.75	133.0

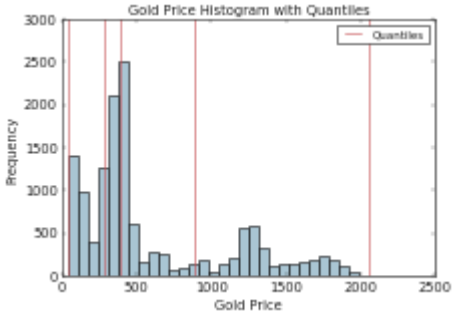
2014-02-24	1333.00	133.0
2014-02-21	1320.75	132.0
2014-02-20	1313.75	131.0
2014-02-19	1318.75	131.0
2014-02-18	1314.00	131.0
2014-02-17	1326.00	132.0
2014-02-14	1308.50	130.0
2014-02-13	1290.25	129.0
2014-12-02	1286.50	128.0
2014-11-02	1282.75	128.0
2014-10-02	1273.50	127.0
2014-07-02	1260.00	126.0
2014-06-02	1258.50	125.0
2014-05-02	1257.00	125.0
2014-04-02	1253.00	125.0
2014-03-02	1246.50	124.0
2014-01-31	1246.50	124.0
2014-01-30	1254.00	125.0
2014-01-29	1254.75	125.0
2014-01-28	1253.50	125.0
2014-01-27	1270.00	127.0
2014-01-24	1259.25	125.0
2014-01-23	1244.25	124.0
2014-01-22	1239.50	123.0

```
quantile_list = [0, .25, .5, .75, 1.]
quantiles = goldDF['GoldPrice'].quantile(quantile_list)
quantiles
```

```
0.00      34.780
0.25     281.775
0.50     384.000
0.75     892.250
1.00    2061.500
Name: GoldPrice, dtype: float64
```

```
fig, ax = plt.subplots()
goldDF['GoldPrice'].hist(bins=30, color='#A9C5D3',
                         edgecolor='black', grid=False)
for quantile in quantiles:
    qvl = plt.axvline(quantile, color='r')
ax.legend([qvl], ['Quantiles'], fontsize=10)
ax.set_title('Gold Price Histogram with Quantiles',
             fontsize=12)
ax.set_xlabel('Gold Price', fontsize=12)
ax.set_ylabel('Frequency', fontsize=12)
```

```
Text(0, 0.5, 'Frequency')
```

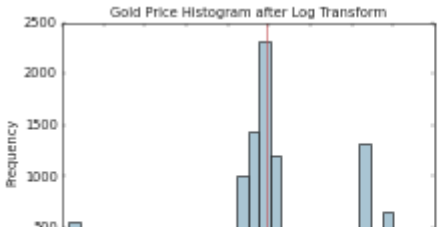



```
goldDF['GoldPrice_log'] = np.log((1+ goldDF['GoldPrice']))
goldDF[[ 'GoldPrice', 'GoldPrice_log']].iloc[4:9]
```

Date	GoldPrice	GoldPrice_log
2022-03-25	1956.65	7.579500
2022-03-24	1945.90	7.573994
2022-03-23	1932.15	7.566906
2022-03-22	1929.35	7.565457
2022-03-21	1925.05	7.563227

```
goldprice_log_mean = np.round(np.mean(goldDF['GoldPrice_log']), 2)
fig, ax = plt.subplots()
goldDF['GoldPrice_log'].hist(bins=30, color='#A9C5D3',
                             edgecolor='black', grid=False)
plt.axvline(goldprice_log_mean, color='r')
ax.set_title('Gold Price Histogram after Log Transform',
             fontsize=12)
ax.set_xlabel('Gold Price (log scale)', fontsize=12)
ax.set_ylabel('Frequency', fontsize=12)
ax.text(11.5, 450, r'$\mu$='+str(goldprice_log_mean), fontsize=10)
```

```
Text(11.5, 450, '$\\mu$=5.96')
```




```
goldprice = np.array(goldDF['GoldPrice'])
goldprice_clean = goldprice[~np.isnan(goldprice)]
l, opt_lambda = spstats.boxcox(goldprice_clean)
print('Optimal lambda value:', opt_lambda)
```

```
Optimal lambda value: 0.22789603130062897
```

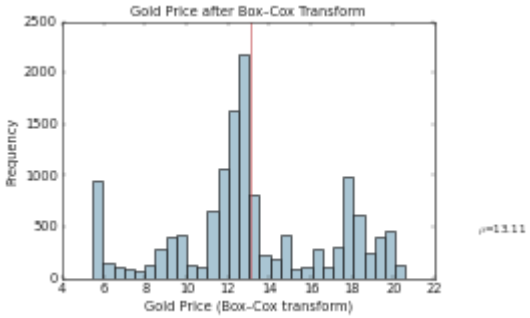
```
goldDF['goldprice_boxcox_lambda_0'] = spstats.boxcox(
                                        (1+goldDF['GoldPrice']),
                                          lmbda=0)
goldDF['goldprice_boxcox_lambda_opt'] = spstats.boxcox(
                                            goldDF['GoldPrice'],
                                              lmbda=opt_lambda)

goldDF[['GoldPrice', 'GoldPrice_log',
               'goldprice_boxcox_lambda_0',
               'goldprice_boxcox_lambda_opt']].iloc[4:9]
```

Date	GoldPrice	GoldPrice_log	goldprice_boxcox_lambda_0	goldprice_boxcox_lambda_opt
2022-03-25	1956.65	7.579500	7.579500	20.294357
2022-03-24	1945.90	7.573994	7.573994	20.263387
2022-03-23	1932.15	7.566906	7.566906	20.223581

```
goldprice_boxcox_mean = np.round(
                      np.mean(
                       goldDF['goldprice_boxcox_lambda_opt']),2)
fig, ax = plt.subplots()
goldDF['goldprice_boxcox_lambda_opt'].hist(bins=30,
                     color='#A9C5D3', edgecolor='black', grid=False)
plt.axvline(goldprice_boxcox_mean, color='r')
ax.set_title('Gold Price after Box-Cox Transform',
             fontsize=12)
ax.set_xlabel('Gold Price (Box-Cox transform)', fontsize=12)
ax.set_ylabel('Frequency', fontsize=12)
ax.text(24, 450, r'$\mu$='+str(goldprice_boxcox_mean), fontsize=10)
```

```
Text(24, 450, '$\\mu$=13.11')
```




```
#Silver price data
silverDF = pd.read_csv("/content/drive/MyDrive/Data Sets/VargoldCommodity/lbma_silver_am_usd_
silverDF = silverDF.set_index('Date')
silverDF.head()
```

Date	SilverPrice
2022-03-31	24.815
2022-03-30	24.755
2022-03-29	24.640
2022-03-28	24.905
2022-03-25	25.620

```
silverDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 13723 entries, 2022-03-31 to 1968-02-01
Data columns (total 1 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   SilverPrice  13704 non-null  float64
dtypes: float64(1)
memory usage: 214.4 KB
```

```
silverDF.isnull().sum() ## missing values
```

```
SilverPrice    19
dtype: int64
```

```
mean_imputation_silverDF = silverDF.copy()
mean_imputation_silverDF['SilverPrice_Mean_Filled'] = mean_imputation_silverDF['SilverPrice']
```

```
silverDF.SilverPrice.describe().T
```

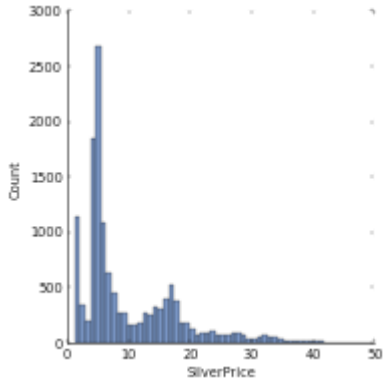
```
count    13704.000000
mean         9.852661
std          7.993719
min          1.272000
25%          4.615000
50%          5.985500
75%         14.820000
max         49.450000
Name: SilverPrice, dtype: float64
```

```
mean_imputation_silverDF.SilverPrice.describe().T
```

```
count    13704.000000
mean         9.852661
std          7.993719
min          1.272000
25%          4.615000
50%          5.985500
75%         14.820000
max         49.450000
Name: SilverPrice, dtype: float64
```

```
sns.displot(silverDF, x="SilverPrice")
```

```
<seaborn.axisgrid.FacetGrid at 0x7fdb286bda10>
```




```
sns.displot(mean_imputation_silverDF, x="SilverPrice")
```

```
<seaborn.axisgrid.FacetGrid at 0x7fdb3946c610>
```

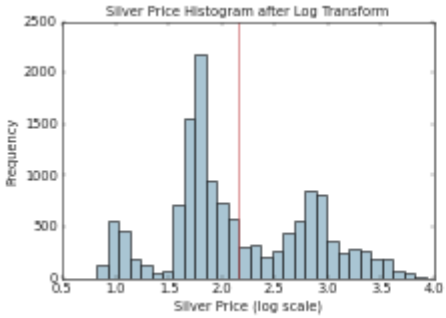


```
silverDF['SilverPrice_log'] = np.log((1+ silverDF['SilverPrice']))
silverDF[[ 'SilverPrice', 'SilverPrice_log']].iloc[4:9]
```

Date	SilverPrice	SilverPrice_log
2022-03-25	25.620	3.281663
2022-03-24	25.315	3.270139
2022-03-23	25.015	3.258673
2022-03-22	25.085	3.261360
2022-03-21	25.035	3.259442

```
silverprice_log_mean = np.round(np.mean(silverDF['SilverPrice_log']), 2)
fig, ax = plt.subplots()
silverDF['SilverPrice_log'].hist(bins=30, color='#A9C5D3',
                                 edgecolor='black', grid=False)
plt.axvline(silverprice_log_mean, color='r')
ax.set_title('Silver Price Histogram after Log Transform',
             fontsize=12)
ax.set_xlabel('Silver Price (log scale)', fontsize=12)
ax.set_ylabel('Frequency', fontsize=12)
ax.text(11.5, 450, r'$\mu$='+str(silverprice_log_mean), fontsize=10)
```

```
Text(11.5, 450, '$\\mu$=2.15')
```




```
! pip install impyute
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
Requirement already satisfied: impyute in /usr/local/lib/python3.7/dist-packages (0.0.8
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from im
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from im
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (
```

```
# silverDF


# USD Index
# Trade Weighted U.S. Dollar Index: Broad, Goods and Services (DTWEXBGS)
# https://fred.stlouisfed.org/series/DTWEXBGS
usdFedFundsDF = pd.read_csv("/content/drive/MyDrive/Data Sets/VargoldCommodity/FEDFUNDS_2022-


usdFedFundsDF = usdFedFundsDF.set_index('Date')

usdFedFundsDF.head()
```

Date	FEDFUNDS
1954-01-07	0.80
1954-01-08	1.22
1954-01-09	1.07
1954-01-10	0.85
1954-01-11	0.83

```
usdFedFundsDF.tail(10)
```

Date	FEDFUNDS
2021-01-06	0.08
2021-01-07	0.10
2021-01-08	0.09
2021-01-09	0.08
2021-01-10	0.08
2021-01-11	0.08
2021-01-12	0.08
2022-01-01	0.08
2022-01-02	0.08
2022-01-03	0.20

```
usdFedFundsDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 813 entries, 1954-01-07 to 2022-01-03
Data columns (total 1 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   FEDFUNDS  813 non-null    float64
dtypes: float64(1)
memory usage: 12.7 KB
```

```
usdFedFundsDF.isnull().sum() ## missing values
```

```
FEDFUNDS    0
dtype: int64
```

```
# Interest
# 10-Year Treasury Constant Maturity Rate (DGS10)
# https://fred.stlouisfed.org/series/DGS10

### https://fred.stlouisfed.org/series/FEDFUNDS
interestRateDF = pd.read_csv("/content/drive/MyDrive/Data Sets/VargoldCommodity/DGS10.csv",pa
interestRateDF = interestRateDF.rename(columns={"DATE":'Date'})

interestRateDF = interestRateDF.set_index('Date')

interestRateDF.head()
```

Date	DGS10
1962-01-02	4.06
1962-01-03	4.03
1962-01-04	3.99
1962-01-05	4.02
1962-01-08	4.03

```
interestRateDF.tail(10)
```

Date	DGS10
2022-04-06	2.61
2022-04-07	2.66
2022-04-08	2.72
2022-04-11	2.79
2022-04-12	2.72

```
interestRateDF['DGS10'] = interestRateDF['DGS10'].replace('.',np.nan)
```

```
interestRateDF.isnull().sum() ## missing values
```

```
DGS10    671
dtype: int64
```

```
interestRateDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 15731 entries, 1962-01-02 to 2022-04-19
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   DGS10   15060 non-null  object
dtypes: object(1)
memory usage: 245.8+ KB
```

```
interestRateDF.isnull().sum() ## missing values
```

```
DGS10    671
dtype: int64
```

```
# S&P Index
# https://finance.yahoo.com/quote/%5EGSPC/history?period1=-1325635200&period2=1611360000&inte
# https://finance.yahoo.com/quote/%5EGSPC/history?period1=-628819200&period2=1650412800&inter
spindex500Index = pd.read_csv("/content/drive/MyDrive/Data Sets/VargoldCommodity/SPIndex_2022
```

```
spindex500Index = spindex500Index.set_index('Date')
```

```
spindex500Index.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 14100 entries, 2022-04-19 to 2066-04-14
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
```

```
 0   Open         14100 non-null  object
 1   High         14100 non-null  object
 2   Low          14100 non-null  object
 3   Close*       14100 non-null  object
 4   Adj Close**  14100 non-null  object
 5   Volume       14100 non-null  object
dtypes: object(6)
memory usage: 771.1+ KB
```

```
spindex500Index.isnull().sum() ## missing values
```

```
Open          0
High          0
Low           0
Close*        0
Adj Close**   0
Volume        0
dtype: int64
```

```
spindex500Index.columns
```

```
Index(['Open', 'High', 'Low', 'Close*', 'Adj Close**', 'Volume'], dtype='object')
```

```
spindex500Index = spindex500Index.drop('Open', 1)
spindex500Index = spindex500Index.drop('High', 1)
spindex500Index = spindex500Index.drop('Low', 1)
spindex500Index = spindex500Index.drop('Close*', 1)
spindex500Index = spindex500Index.drop('Volume', 1)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: In a fut
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: In a fut

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: FutureWarning: In a fut
  This is separate from the ipykernel package so we can avoid doing imports until
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: FutureWarning: In a fut
  after removing the cwd from sys.path.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: FutureWarning: In a fut
  """
```

```
spindex500Index= spindex500Index.rename(columns={'Adj Close**':'AdjClose'})
```

```
spindex500Index['AdjClose'] = spindex500Index['AdjClose'].replace(',', '')
```

```
spindex500Index.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 14100 entries, 2022-04-19 to 2066-04-14
Data columns (total 1 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   AdjClose  14100 non-null  object
dtypes: object(1)
memory usage: 220.3+ KB
```

```
#spindex500Index
```

```
spindex500Index=spindex500Index.loc['19720101':'20220419']
#spindex500Index
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Value ba
  """Entry point for launching an IPython kernel.
```

```
# Crude Oil Prices: West Texas Intermediate (WTI) - Cushing, Oklahoma
# https://fred.stlouisfed.org/series/DCOILWTICO
oilPricesDF = pd.read_csv("/content/drive/MyDrive/Data Sets/VargoldCommodity/DCOILWTICO_2022-
oilPricesDF = oilPricesDF.set_index('Date')
oilPricesDF.head()
```

Date	CrudeOilPrices(WTI)
1986-02-01	25.56
1986-03-01	26
1986-06-01	26.53
1986-07-01	25.85
1986-08-01	25.87

```
oilPricesDF['CrudeOilPrices(WTI)'] = oilPricesDF['CrudeOilPrices(WTI)'].replace('.',np.nan)
```

```
oilPricesDF.isnull().sum() ## missing values
```

```
CrudeOilPrices(WTI)    323
dtype: int64
```

```
oilPricesDF['CrudeOilPrices(WTI)']
```

```
2021-10-14     81.43
2021-10-15     82.39
```

```
2021-10-18    82.62
2021-10-19    83.19
2021-10-20     84.4
2021-10-21    82.64
2021-10-22    84.53
2021-10-25    84.64
2021-10-26    85.64
2021-10-27    82.66
2021-10-28    82.78
2021-10-29     83.5
2021-01-11    84.08
2021-02-11    83.91
2021-03-11    80.82
2021-04-11    78.88
2021-05-11    81.25

2021-08-11    81.96
2021-09-11    84.12
2021-10-11    81.23
2021-11-11    81.47
2021-12-11    80.87
2021-11-15    80.85
2021-11-16    80.76
2021-11-17    78.32
2021-11-18    78.92
2021-11-19    76.11
2021-11-22    76.74
2021-11-23    78.32
2021-11-24    78.32
2021-11-25      NaN
2021-11-26      NaN
2021-11-29    69.88
2021-11-30    66.14
2021-01-12    65.44
2021-02-12     66.6
2021-03-12    66.39
2021-06-12    69.62
2021-07-12    71.94
2021-08-12    72.43
2021-09-12    70.87
2021-10-12    71.71
2021-12-13    71.19
2021-12-14    70.57
2021-12-15    70.89
2021-12-16    72.34
2021-12-17    70.93
2021-12-20    68.69
2021-12-21     71.1
2021-12-22    72.82
2021-12-23    73.89
2021-12-24      NaN
2021-12-27    75.49
2021-12-28    76.01
2021-12-29    76.58
2021-12-30    76.83
2021-12-31    75.33
2022-03-01    75.99
```

Exploratory analysis:

Let's load the data and do some analysis with visualization to know insights of the data. Exploratory data analysis is quite extensive in multivariate time series. I will cover some areas here to get insights of the data. However, it is advisable to conduct all statistical tests to ensure our clear understanding on data distribution.

```
oilPricesDF.columns
### https://stackoverflow.com/questions/46834732/convert-pandas-datetime-column-yyyy-mm-dd-tc
## https://datatofish.com/strings-to-datetime-pandas/
```

```
Index(['CrudeOilPrices(WTI)'], dtype='object')
```

```
pd.set_option('display.max_rows', None)
dataset = pd.concat([goldDF,silverDF,oilPricesDF,usdFedFundsDF,interestRateDF,spindex500Index
print('Number of colums in Dataframe : ', len(dataset.columns))
print('Number of rows in Dataframe : ', len(dataset.index))
print(dataset)
```

```
Number of colums in Dataframe :  11
Number of rows in Dataframe :  17460
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 17460 entries, 1954-01-07 to 2022-12-04
Data columns (total 11 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   GoldPrice                   13712 non-null  float64
 1   GoldPrice_bin_round         13712 non-null  float64
 2   GoldPrice_log               13712 non-null  float64
 3   goldprice_boxcox_lambda_0   13712 non-null  float64
 4   goldprice_boxcox_lambda_opt 13712 non-null  float64
 5   SilverPrice                 13704 non-null  float64
 6   SilverPrice_log             13704 non-null  float64
 7   CrudeOilPrices(WTI)         9145 non-null   object 
 8   FEDFUNDS                    813 non-null    float64
```

```
 9   DGS10                      15060 non-null  object
 10  AdjClose                   12685 non-null  object
dtypes: float64(8), object(3)
memory usage: 1.6+ MB
```

```
dataset.drop(['GoldPrice_bin_round','GoldPrice_log','goldprice_boxcox_lambda_0','goldprice_bc
```

Let's fix the dates for all the series.

```
# dataset
```

```
dataset.tail()
```

Date	GoldPrice	SilverPrice	CrudeOilPrices(WTI)	FEDFUNDS	DGS10	AdjClose
2022-11-02	1826.25	22.895	93.1	NaN	NaN	NaN
2022-11-03	1991.45	25.655	109.31	NaN	NaN	NaN
2022-11-04	NaN	NaN	94.22	NaN	NaN	NaN
2022-12-01	1816.40	22.745	82.51	NaN	NaN	NaN
2022-12-04	NaN	NaN	100.52	NaN	NaN	NaN

```
### dataset=dataset.loc['20100101':'20200824']
dataset=dataset.loc['20000101':'20220301']
dataset.head()
```

Date	GoldPrice	SilverPrice	CrudeOilPrices(WTI)	FEDFUNDS	DGS10	AdjClose
2000-01-01	NaN	NaN	NaN	5.45	NaN	NaN
2000-01-02	283.65	5.243	28.28	5.73	NaN	NaN
2000-01-03	293.75	5.125	31.71	5.85	6.58	1,455.22
2000-01-04	NaN	NaN	NaN	6.02	6.49	1,399.42
2000-01-05	NaN	NaN	25.84	6.27	6.62	1,402.11

```
dataset.tail()
```

Date	GoldPrice	SilverPrice	CrudeOilPrices(WTI)	FEDFUNDS	DGS10	AdjClose
2022-02-23	1895.70	24.105	92.14	NaN	1.99	4,225.50
2022-02-24	1968.35	25.315	92.77	NaN	1.96	4,288.70
2022-02-25	1912.15	24.210	91.68	NaN	1.97	4,384.65

```
dataset.isnull().sum() ## missing values
```

```
GoldPrice              848
SilverPrice            848
CrudeOilPrices(WTI)    887
FEDFUNDS              6170
DGS10                  893
AdjClose               861
dtype: int64
```

```
# https://stackoverflow.com/questions/31170550/pandas-df-fillnamethod-pad-not-working-on-2800

dataset=dataset.fillna(method='pad')
dataset = dataset.fillna(method = 'bfill')
```

```
dataset.isnull().sum() ## missing values
```

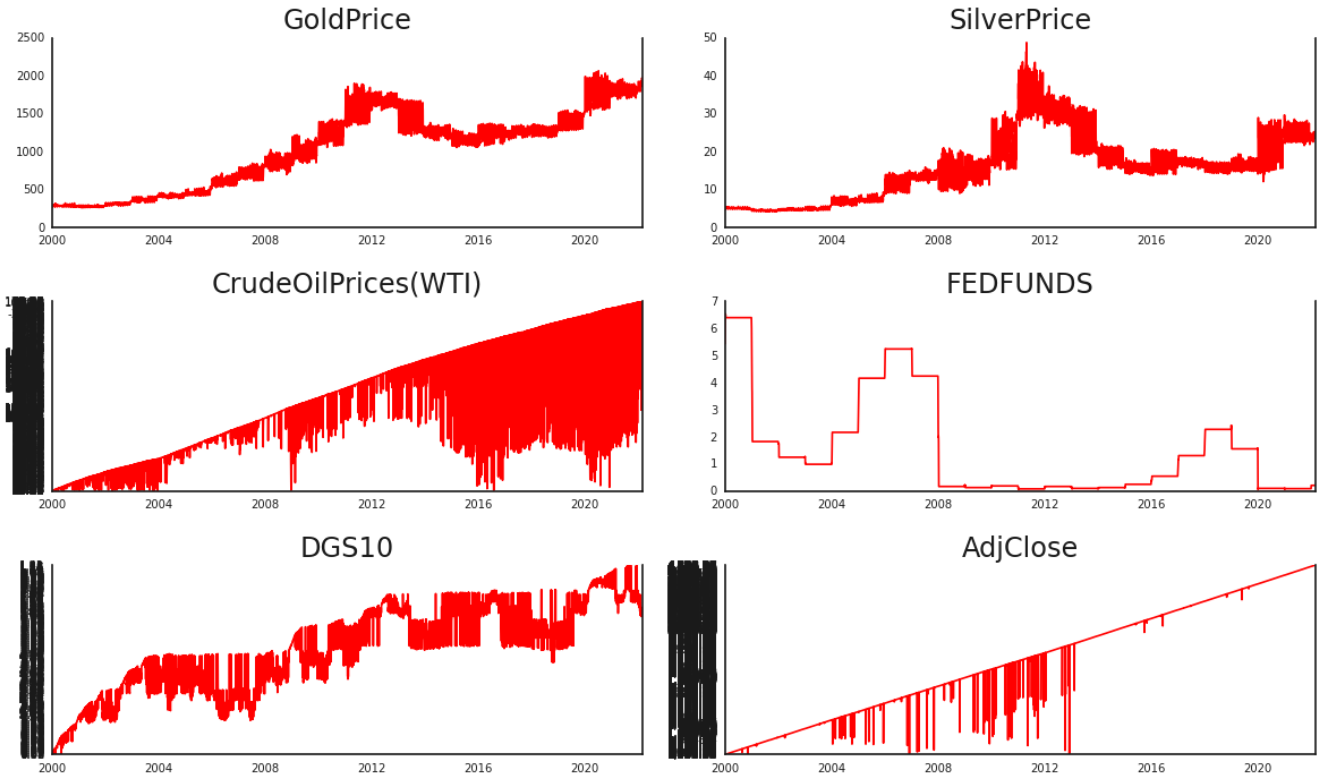
```
GoldPrice             0
SilverPrice           0
CrudeOilPrices(WTI)   0
FEDFUNDS              0
DGS10                 0
AdjClose              0
dtype: int64
```

```
dataset=dataset.dropna()
dataset.head()
```

Date	GoldPrice	SilverPrice	CrudeOilPrices(WTI)	FEDFUNDS	DGS10	AdjClose
2000-01-01	283.65	5.243	28.28	5.45	6.58	1,455.22
2000-01-02	283.65	5.243	28.28	5.73	6.58	1,455.22
2000-01-03	293.75	5.125	31.71	5.85	6.58	1,455.22
2000-01-04	293.75	5.125	31.71	6.02	6.49	1,399.42
2000-01-05	293.75	5.125	25.84	6.27	6.62	1,402.11

```
#print(dataset)


# Plot
fig, axes = plt.subplots(nrows=3, ncols=2, dpi=120, figsize=(10,6))
for i, ax in enumerate(axes.flatten()):
 data = dataset[dataset.columns[i]]
 ax.plot(data, color='red', linewidth=1)
 ax.set_title(dataset.columns[i])
 ax.xaxis.set_ticks_position('none')
 ax.yaxis.set_ticks_position('none')
 ax.spines["top"].set_alpha(0)
 ax.tick_params(labelsize=6)
plt.tight_layout();
```



From the above plots, we can visible conclude that, all the series contain unit root with stochastic trend showing a systematic pattern that is unpredictable.

Normality Test

To extract maximum information from our data, it is important to have a ***normal or Gaussian distribution*** of the data. To check for that, we have done a normality test based on the Null and Alternate Hypothesis intuition.

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 6437 entries, 2000-01-01 to 2022-03-01
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   GoldPrice            6437 non-null   float64
 1   SilverPrice          6437 non-null   float64
 2   CrudeOilPrices(WTI)  6437 non-null   object 
 3   FEDFUNDS             6437 non-null   float64
 4   DGS10                6437 non-null   object 
 5   AdjClose             6437 non-null   object 
dtypes: float64(3), object(3)
memory usage: 352.0+ KB
```

```
itemcost='1,455.22'
itemcostProc=itemcost.replace(',','')
print(itemcostProc)
print(float(itemcostProc))
```

```
1455.22
1455.22
```

```
dataset['AdjClose']= dataset['AdjClose'].apply(lambda x: x.replace(',',''))
```

```
dataset['AdjClose']
```

```
2021-10-09    4391.34
2021-10-11    4361.19
2021-10-12    4350.65
2021-10-13    4363.80
2021-10-14    4438.26
2021-10-15    4471.37
2021-10-18    4486.46
2021-10-19    4519.63
2021-10-20    4536.19
2021-10-21    4549.78
2021-10-22    4544.90
2021-10-25    4566.48
2021-10-26    4574.79
2021-10-27    4551.68
2021-10-28    4596.42
2021-10-29    4605.38
2021-11-01    4613.67
```

```
2021-11-02    4630.65
2021-11-03    4660.57
2021-11-04    4680.06
2021-11-05    4697.53
2021-11-06    4697.53
2021-11-08    4701.70
2021-11-09    4685.25
2021-11-10    4646.71
2021-11-11    4649.27
2021-11-12    4682.85
2021-11-15    4682.80
2021-11-16    4700.90
2021-11-17    4688.67
2021-11-18    4704.54
2021-11-19    4697.96
2021-11-22    4682.94
2021-11-23    4690.70
2021-11-24    4701.46
2021-11-25    4701.46
2021-11-26    4594.62
2021-11-29    4655.27
2021-11-30    4567.00
2021-12-01    4513.04
2021-12-02    4577.10
2021-12-03    4538.43
2021-12-04    4538.43
2021-12-05    4538.43
2021-12-06    4591.67
2021-12-07    4686.75
2021-12-08    4701.21
2021-12-09    4667.45
2021-12-10    4712.02
2021-12-11    4712.02
2021-12-13    4668.97
2021-12-14    4634.09
2021-12-15    4709.85
2021-12-16    4668.67
2021-12-17    4620.64
2021-12-20    4568.02
2021-12-21    4649.23
2021-12-22    4696.56
2021-12-23    4725.79
```

```
dataset['CrudeOilPrices(WTI)'] = dataset['CrudeOilPrices(WTI)'].astype('float64')
dataset['CrudeOilPrices(WTI)'] = dataset['CrudeOilPrices(WTI)'].astype('float64')
dataset['DGS10'] = dataset['DGS10'].astype('float64')
dataset['AdjClose'] = dataset['AdjClose'].astype('float64')
dataset.dtypes
```

```
GoldPrice             float64
SilverPrice           float64
CrudeOilPrices(WTI)   float64
FEDFUNDS              float64
DGS10                 float64
AdjClose              float64
dtype: object
```

```
dataset['CrudeOilPrices(WTI)']
```

```
2021-10-09    69.82
2021-10-11    81.23
2021-10-12    71.71
2021-10-13    80.67
2021-10-14    81.43
2021-10-15    82.39
2021-10-18    82.62
2021-10-19    83.19
2021-10-20    84.40
2021-10-21    82.64
2021-10-22    84.53
2021-10-25    84.64
2021-10-26    85.64
2021-10-27    82.66
2021-10-28    82.78
2021-10-29    83.50
2021-11-01    52.15
2021-11-02    58.22
2021-11-03    66.02
2021-11-04    66.02
2021-11-05    65.31
2021-11-06    71.00
2021-11-08    69.30
2021-11-09    69.30
2021-11-10    80.64
2021-11-11    81.47
2021-11-12    81.47
2021-11-15    80.85
2021-11-16    80.76
2021-11-17    78.32
2021-11-18    78.92
2021-11-19    76.11
2021-11-22    76.74
2021-11-23    78.32
2021-11-24    78.32
2021-11-25    78.32
2021-11-26    78.32
2021-11-29    69.88
2021-11-30    66.14
2021-12-01    53.08
2021-12-02    59.50
2021-12-03    65.59
2021-12-04    59.70
2021-12-05    65.96
2021-12-06    65.96
2021-12-07    74.21
2021-12-08    69.12
2021-12-09    69.12
2021-12-10    80.75
2021-12-11    80.87
2021-12-13    71.19
2021-12-14    70.57
2021-12-15    70.89
```

```
2021-12-16     72.34
2021-12-17     70.93
2021-12-20     68.69
2021-12-21     71.10
2021-12-22     72.82
2021-12-23     73.89
```

```
from scipy import stats

GoldPrice=dataset.GoldPrice.values
print(GoldPrice)


stat,p = stats.normaltest(GoldPrice)

print("GoldPrice Statistics = %.3f, p=%.3f" % (stat,p))
alpha = 0.05
if p> alpha:
    print('Data looks Gaussian (fail to reject null hypothesis)')
else:
    print('Data looks non-Gaussian (reject null hypothesis)')
```

```
 [ 283.65  283.65  293.75 ... 1912.15 1903.3  1903.3 ]
 GoldPrice Statistics = 10532.947, p=0.000
 Data looks non-Gaussian (reject null hypothesis)
```

```
from scipy import stats

SilverPrice=dataset.SilverPrice.values
print(SilverPrice)


stat,p = stats.normaltest(SilverPrice)

print("SilverPrice Statistics = %.3f, p=%.3f" % (stat,p))
alpha = 0.05
if p> alpha:
    print('Data looks Gaussian (fail to reject null hypothesis)')
else:
    print('Data looks non-Gaussian (reject null hypothesis)')
```

```
 [ 5.243  5.243  5.125 ... 24.21  24.35  24.35 ]
 SilverPrice Statistics = 370.994, p=0.000
 Data looks non-Gaussian (reject null hypothesis)
```

```
from scipy import stats

OilPrice=dataset['CrudeOilPrices(WTI)'].values
```

```
print(OilPrice)


stat,p = stats.normaltest(OilPrice)

print("OilPrice Statistics = %.3f, p=%.3f" % (stat,p))
alpha = 0.05
if p> alpha:
    print('Data looks Gaussian (fail to reject null hypothesis)')
else:
    print('Data looks non-Gaussian (reject null hypothesis)')
```

```
 [28.28 28.28 31.71 ... 91.68 96.13 75.99]
 OilPrice Statistics = 375.902, p=0.000
 Data looks non-Gaussian (reject null hypothesis)
```

```
from scipy import stats

FEDFUNDS=dataset.FEDFUNDS.values
print(FEDFUNDS)


stat,p = stats.normaltest(FEDFUNDS)

print("FEDFUNDS Statistics = %.3f, p=%.3f" % (stat,p))
alpha = 0.05
if p> alpha:
    print('Data looks Gaussian (fail to reject null hypothesis)')
else:
    print('Data looks non-Gaussian (reject null hypothesis)')
```

```
 [5.45 5.73 5.85 ... 0.2  0.2  0.2 ]
 FEDFUNDS Statistics = 1222.581, p=0.000
 Data looks non-Gaussian (reject null hypothesis)
```

```
from scipy import stats

InterestRate=dataset.DGS10.values
print(InterestRate)


stat,p = stats.normaltest(InterestRate)

print("InterestRate Statistics = %.3f, p=%.3f" % (stat,p))
alpha = 0.05
if p> alpha:
    print('Data looks Gaussian (fail to reject null hypothesis)')
else:
    print('Data looks non-Gaussian (reject null hypothesis)')
```

```
 [6.58 6.58 6.58 ... 1.97 1.83 1.72]
```