# EX.NO: 02 31/01/2025

AIM:

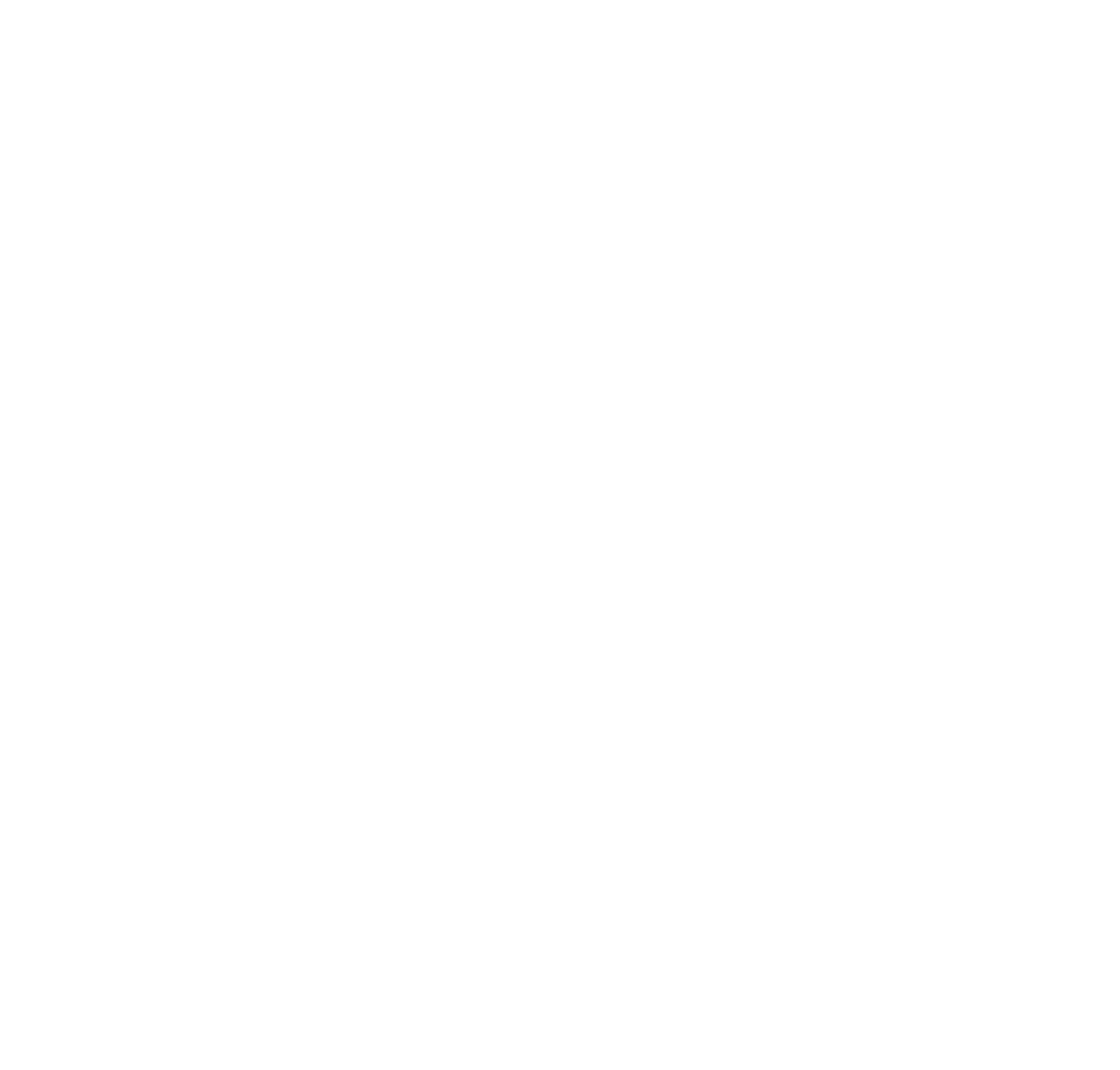
# VISUALIZATION OF TIMESERIES DATA

221501019

To write a python program for visualizing the time series data.

# ALGORITHM:

1. Import required library files
2. Prepare the data
3. Visualize the data using different methods CODE:

import pandas as pd import numpy as np

data = pd.read\_csv("Gold\_Price.csv") import matplotlib.pyplot as plt

# Convert 'Date' column to datetime objects if it's not already data['Date'] = pd.to\_datetime(data['Date'])

# Set 'Date' as the index for easier time series plotting data = data.set\_index('Date')

# Create the plot

plt.figure(1,figsize=(12, 6)) # Adjust figure size as needed plt.plot(data['Price'], label='Price') plt.plot(data['Open'], label='Open')

plt.plot(data['High'], label='High') plt.plot(data['Low'], label='Low')

# Customize the plot

plt.title('Time Series Analysis of Price, Open, High, and Low') plt.xlabel('Date')

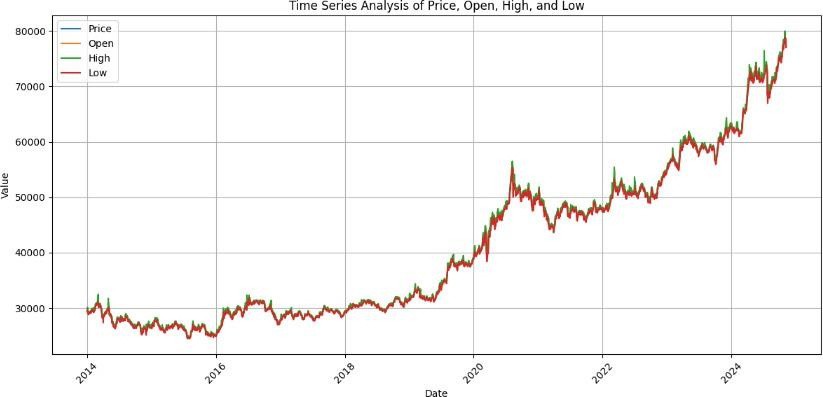
plt.ylabel('Value')

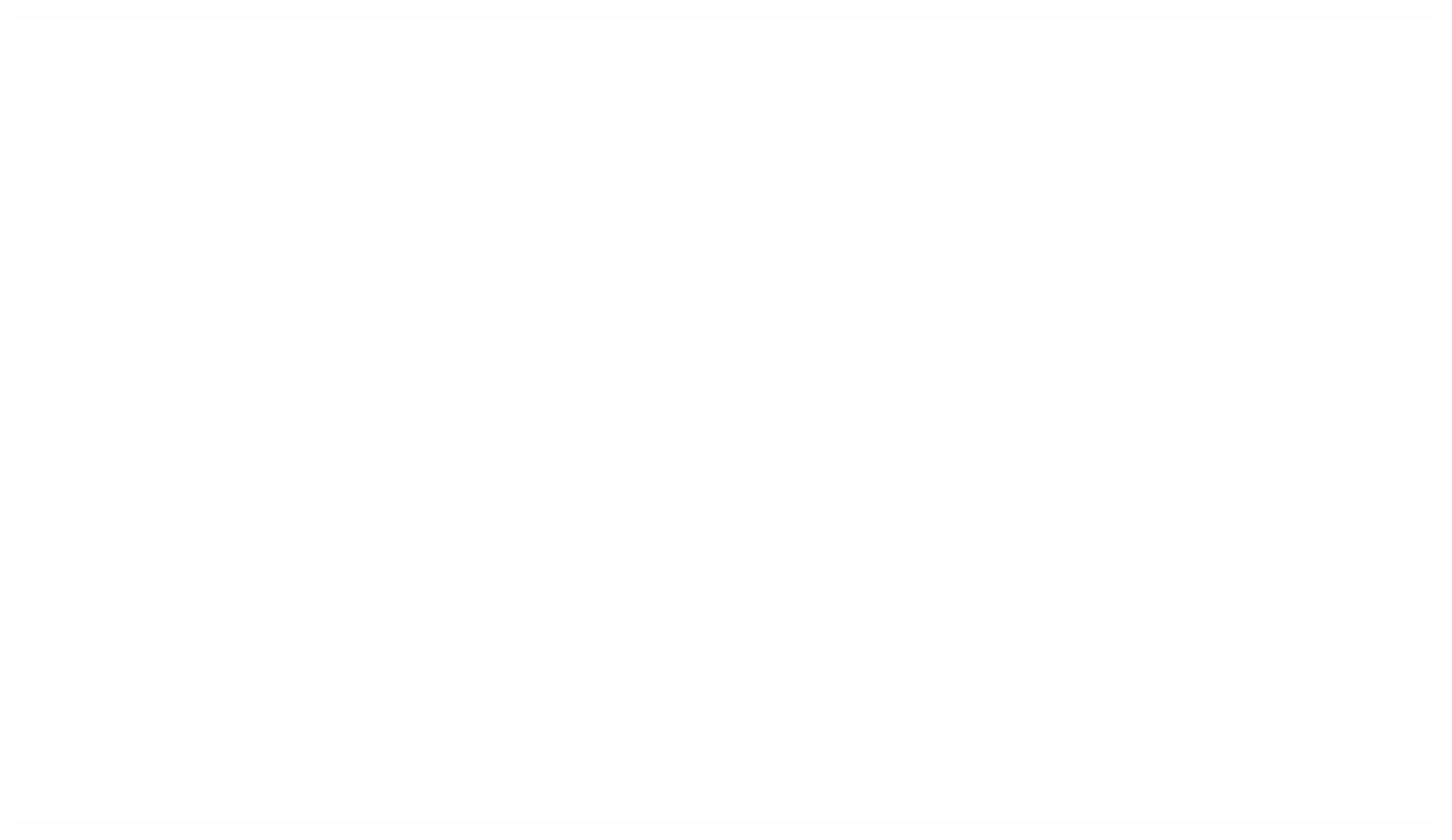
plt.legend() # Display the legend

plt.grid(True) # Add gridlines for better readability plt.xticks(rotation=45) # Rotate x-axis labels for better readability

# Show the plot

plt.tight\_layout() # Adjust plot layout to prevent labels from overlapping plt.show()





plt.figure(figsize=(12,6))

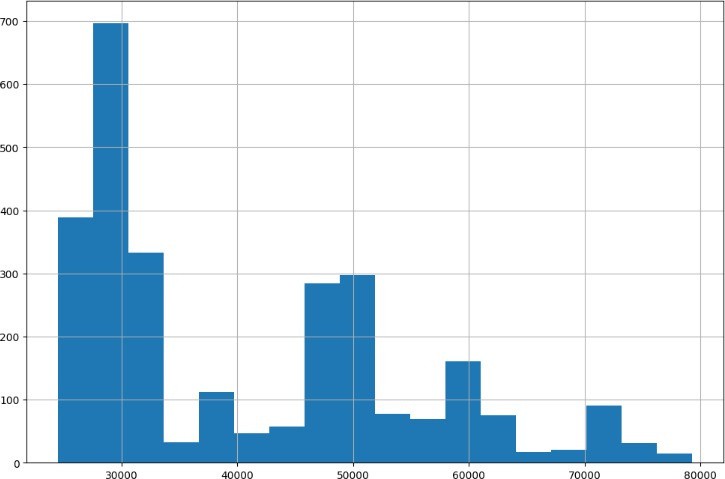
plt.plot(data['Price'],label = 'price') plt.plot(data['Open'],label = 'open') plt.plot(data['High'],label = 'high') plt.plot(data['Low'],label = 'low') plt.grid(True)

plt.legend() plt.tight\_layout() plt.show()

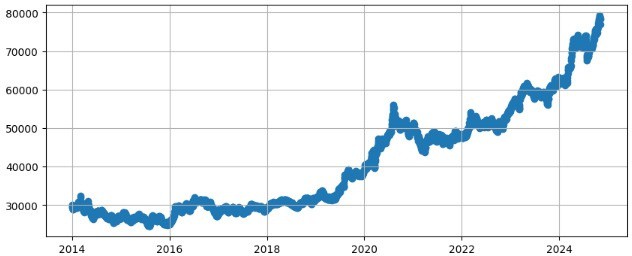
import matplotlib.pyplot as plt

plt.figure(2,figsize= (12,8)) plt.hist(data['Price'],bins ='auto') plt.grid()

plt.show()



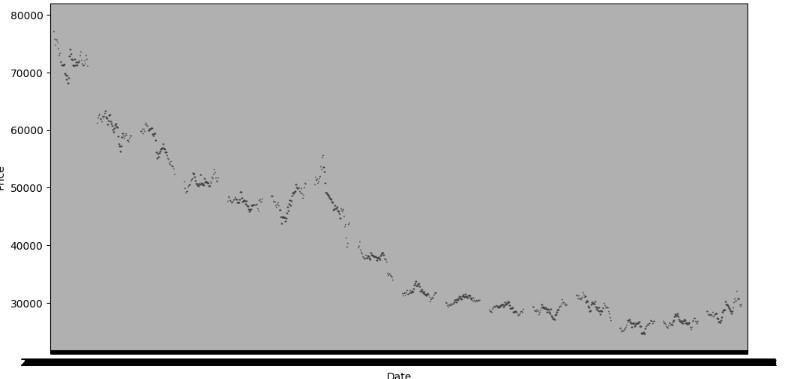
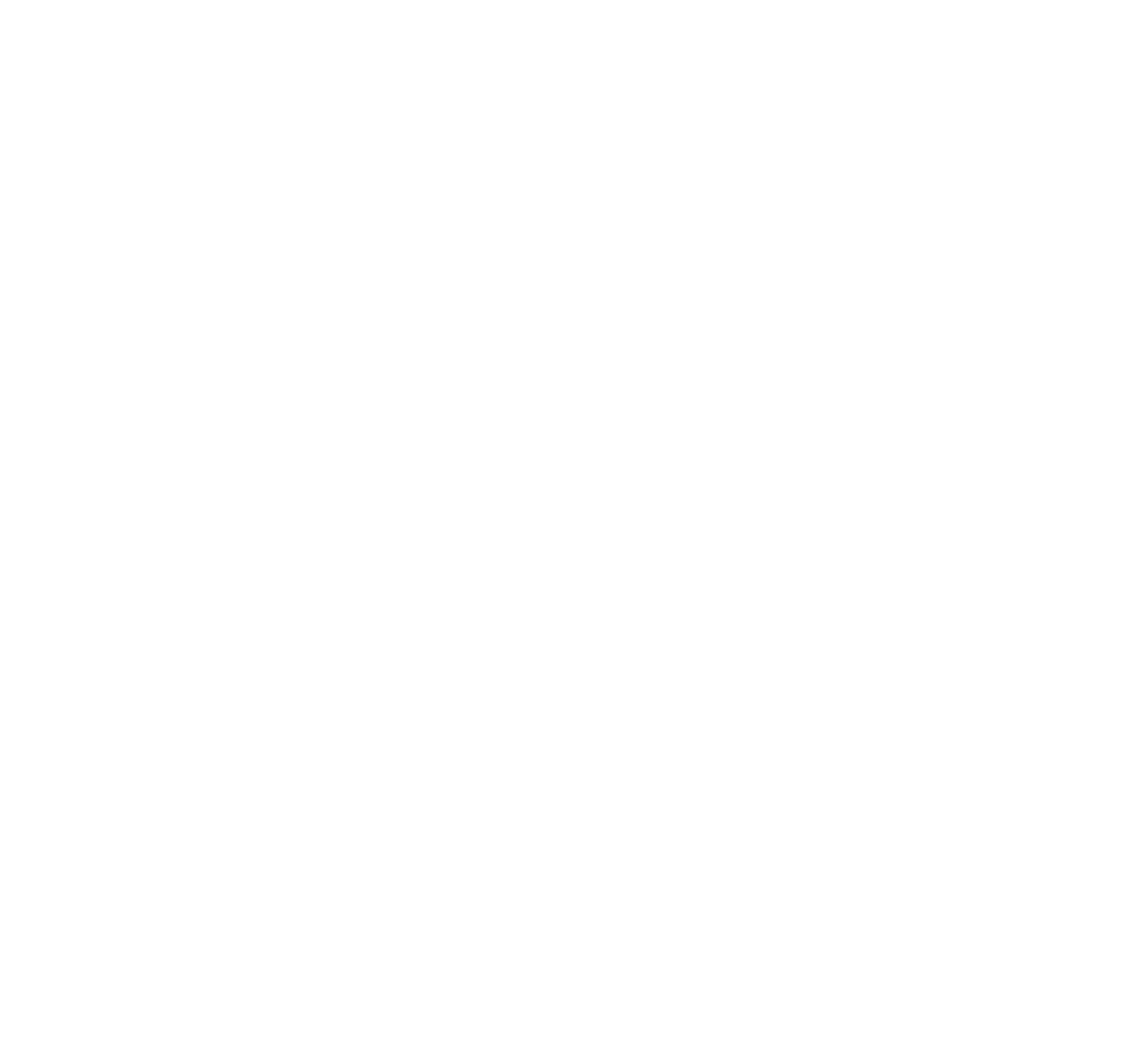
plt.figure(2,figsize=(10,4)) plt.scatter(data.index,data['Price'])

plt.grid() plt.show()



import seaborn as sns plt.figure(3,figsize=(12,6))

sns.boxplot(x=data.index,y=data['Price'],data = data) plt.grid()



plt.show()

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.graphics.tsaplots import plot\_acf, plot\_pacf

plt.figure(figsize=(10, 6))

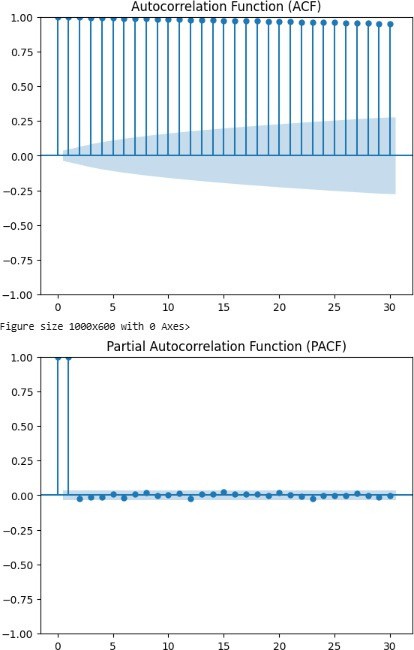
plot\_acf(data['Price'], lags=30) # Adjust the number of lags as needed plt.title('Autocorrelation Function (ACF)')

plt.show()

plt.figure(figsize=(10, 6))

plot\_pacf(data['Price'], lags=30) # Adjust the number of lags as needed plt.title('Partial Autocorrelation Function (PACF)')

plt.show()



# RESULT:

The Visualization of timeseries data has been created .