Analysis and Visualization of Stock Market Data

July 24, 2024

[1]:
```
pip install statsmodels
```

```
Collecting statsmodelsNote: you may need to restart the kernel to use updated
packages.

  Downloading statsmodels-0.14.2-cp312-cp312-win_amd64.whl.metadata (9.5 kB)
Requirement already satisfied: numpy>=1.22.3 in c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-packages (from
statsmodels) (1.26.3)
Requirement already satisfied: scipy!=1.9.2,>=1.8 in c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-packages (from
statsmodels) (1.11.4)
Requirement already satisfied: pandas!=2.1.0,>=1.4 in c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-packages (from
statsmodels) (2.1.4)
Collecting patsy>=0.5.6 (from statsmodels)
  Downloading patsy-0.5.6-py2.py3-none-any.whl.metadata (3.5 kB)
Requirement already satisfied: packaging>=21.3 in c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-packages (from
statsmodels) (23.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-packages (from
pandas!=2.1.0,>=1.4->statsmodels) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-packages (from
pandas!=2.1.0,>=1.4->statsmodels) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-packages (from
pandas!=2.1.0,>=1.4->statsmodels) (2023.4)
Requirement already satisfied: six in c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-packages (from
patsy>=0.5.6->statsmodels) (1.16.0)
Downloading statsmodels-0.14.2-cp312-cp312-win_amd64.whl (9.8 MB)
   ---------------------------------------- 0.0/9.8 MB ? eta -:--:--
   - -------------------------------------- 0.4/9.8 MB 11.2 MB/s eta 0:00:01
   --- ------------------------------------ 0.9/9.8 MB 11.2 MB/s eta 0:00:01
   ----- ---------------------------------- 1.4/9.8 MB 11.3 MB/s eta 0:00:01
   -------- ------------------------------- 2.0/9.8 MB 11.4 MB/s eta 0:00:01
```

```
   ---------- ---------------------------- 2.5/9.8 MB 11.4 MB/s eta 0:00:01
   ----------- ---------------------------- 3.0/9.8 MB 11.2 MB/s eta 0:00:01
   ------------- ------------------------- 3.5/9.8 MB 11.3 MB/s eta 0:00:01
   --------------- ----------------------- 4.1/9.8 MB 11.3 MB/s eta 0:00:01
   ----------------- --------------------- 4.6/9.8 MB 11.4 MB/s eta 0:00:01
   -------------------- ------------------ 5.2/9.8 MB 11.4 MB/s eta 0:00:01
   --------------------- ----------------- 5.5/9.8 MB 11.7 MB/s eta 0:00:01
   ------------------------ -------------- 6.2/9.8 MB 11.3 MB/s eta 0:00:01
   -------------------------- ------------ 6.7/9.8 MB 11.3 MB/s eta 0:00:01
   ---------------------------- ---------- 7.2/9.8 MB 11.3 MB/s eta 0:00:01
   ----------------------------- --------- 7.5/9.8 MB 11.4 MB/s eta 0:00:01
   ------------------------------- ------- 8.0/9.8 MB 10.9 MB/s eta 0:00:01
   -------------------------------- ----- 8.6/9.8 MB 11.0 MB/s eta 0:00:01
   ---------------------------------- -- 9.1/9.8 MB 11.0 MB/s eta 0:00:01
   ---------------------------------------  9.6/9.8 MB 11.2 MB/s eta 0:00:01
   ---------------------------------------- 9.8/9.8 MB 10.8 MB/s eta 0:00:00
Downloading patsy-0.5.6-py2.py3-none-any.whl (233 kB)
   ---------------------------------------- 0.0/233.9 kB ? eta -:--:--
   ---------------------------------------- 233.9/233.9 kB 14.0 MB/s eta 0:00:00
Installing collected packages: patsy, statsmodels
Successfully installed patsy-0.5.6 statsmodels-0.14.2

DEPRECATION: Loading egg at c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-
packages\spylon_kernel-0+unknown-py3.12.egg is deprecated. pip 24.3 will enforce
this behaviour change. A possible replacement is to use pip for package
installation. Discussion can be found at
https://github.com/pypa/pip/issues/12330

[notice] A new release of pip is available: 24.1.1 -> 24.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

[7]:
```
pip install plotly
```

```
Requirement already satisfied: plotly in c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-packages (5.22.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-packages (from plotly)
(8.5.0)
Requirement already satisfied: packaging in c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-packages (from plotly)
(23.2)
Note: you may need to restart the kernel to use updated packages.

DEPRECATION: Loading egg at c:\users\mubashir
khan\appdata\local\programs\python\python312\lib\site-
packages\spylon_kernel-0+unknown-py3.12.egg is deprecated. pip 24.3 will enforce
this behaviour change. A possible replacement is to use pip for package
installation. Discussion can be found at
```

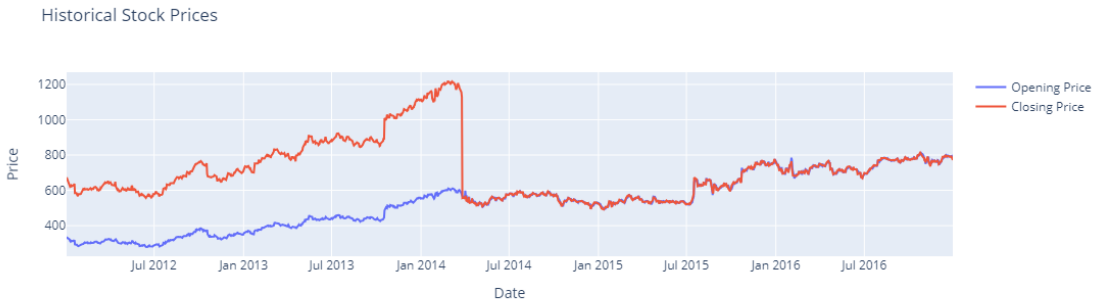
```
https://github.com/pypa/pip/issues/12330

[notice] A new release of pip is available: 24.1.1 -> 24.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

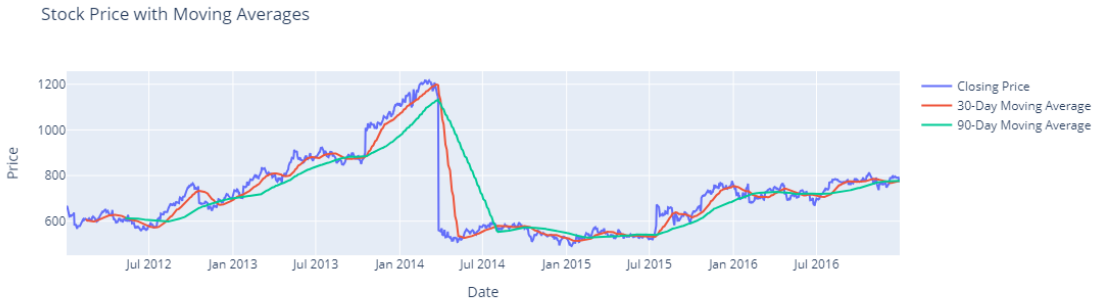
[10]:
```python
import pandas as pd
import plotly.graph_objects as go
import plotly.express as px
from statsmodels.tsa.arima.model import ARIMA

# Load the datasets
train_file_path = r'C:\Users\MUBASHIR␣
 ↪KHAN\Desktop\jupyter\DMV\Google_Stock_Price_Train.csv'
test_file_path = r'C:\Users\MUBASHIR␣
 ↪KHAN\Desktop\jupyter\DMV\Google_Stock_Price_Test.csv'

train_df = pd.read_csv(train_file_path)
test_df = pd.read_csv(test_file_path)

# Ensure 'Date' column is in datetime format
train_df['Date'] = pd.to_datetime(train_df['Date'])
test_df['Date'] = pd.to_datetime(test_df['Date'])

# Set 'Date' as the index
train_df.set_index('Date', inplace=True)
test_df.set_index('Date', inplace=True)

# Check the data type of the 'Close' column
print(train_df['Close'].dtype)
print(test_df['Close'].dtype)

# Remove commas and convert 'Close' column to numeric if it's not already␣
 ↪numeric
if train_df['Close'].dtype == 'object':
    train_df['Close'] = train_df['Close'].str.replace(',', '').astype(float)

if test_df['Close'].dtype == 'object':
    test_df['Close'] = test_df['Close'].str.replace(',', '').astype(float)

# Plot historical stock price trends using Plotly
fig = go.Figure()
fig.add_trace(go.Scatter(x=train_df.index, y=train_df['Open'], mode='lines',␣
 ↪name='Opening Price'))
fig.add_trace(go.Scatter(x=train_df.index, y=train_df['Close'], mode='lines',␣
 ↪name='Closing Price'))
fig.update_layout(title='Historical Stock Prices', xaxis_title='Date',␣
 ↪yaxis_title='Price')
```

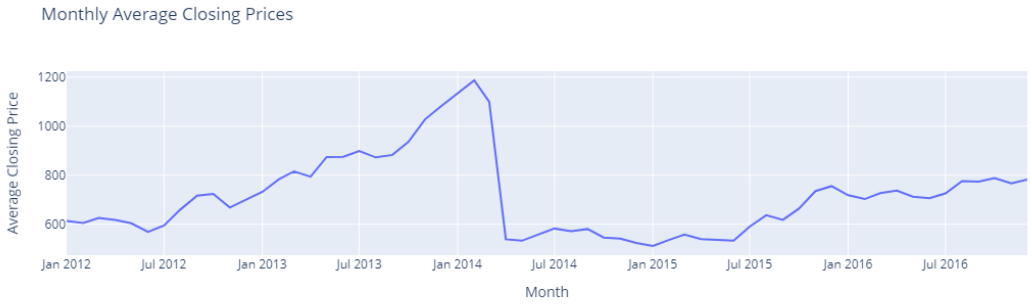
```
fig.show()

# Calculate and plot moving averages
train_df['30_Day_MA'] = train_df['Close'].rolling(window=30).mean()
train_df['90_Day_MA'] = train_df['Close'].rolling(window=90).mean()

fig = go.Figure()
fig.add_trace(go.Scatter(x=train_df.index, y=train_df['Close'], mode='lines',␣
 ↪name='Closing Price'))
fig.add_trace(go.Scatter(x=train_df.index, y=train_df['30_Day_MA'],␣
 ↪mode='lines', name='30-Day Moving Average'))
fig.add_trace(go.Scatter(x=train_df.index, y=train_df['90_Day_MA'],␣
 ↪mode='lines', name='90-Day Moving Average'))
fig.update_layout(title='Stock Price with Moving Averages', xaxis_title='Date',␣
 ↪yaxis_title='Price')
fig.show()

# Perform seasonality analysis
train_df['Month'] = train_df.index.to_period('M')
monthly_avg = train_df.groupby('Month')['Close'].mean().reset_index()
monthly_avg['Month'] = monthly_avg['Month'].astype(str)

fig = px.line(monthly_avg, x='Month', y='Close', title='Monthly Average Closing␣
 ↪Prices')
fig.update_layout(xaxis_title='Month', yaxis_title='Average Closing Price')
fig.show()

# Analyze correlation with other variables
fig = go.Figure()
fig.add_trace(go.Scatter(x=train_df.index, y=train_df['Volume'], mode='lines',␣
 ↪name='Volume'))
fig.update_layout(title='Stock Volume Over Time', xaxis_title='Date',␣
 ↪yaxis_title='Volume')
fig.show()

# Forecast future stock prices using ARIMA
# Prepare data for ARIMA
train_df = train_df.asfreq('D')
model = ARIMA(train_df['Close'], order=(5, 1, 0))
model_fit = model.fit()

# Forecast
forecast = model_fit.get_forecast(steps=len(test_df))
forecast_index = test_df.index
forecast_mean = forecast.predicted_mean
forecast_conf_int = forecast.conf_int()
```

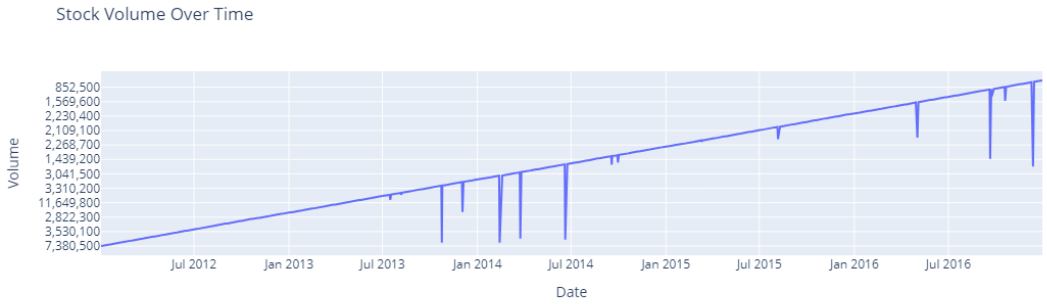
```
# Plot forecast
fig = go.Figure()
fig.add_trace(go.Scatter(x=train_df.index, y=train_df['Close'], mode='lines',␣
 ↪name='Historical'))
fig.add_trace(go.Scatter(x=forecast_index, y=forecast_mean, mode='lines',␣
 ↪name='Forecast'))
fig.add_trace(go.Scatter(x=forecast_index, y=forecast_conf_int.iloc[:, 0],␣
 ↪mode='lines', name='Lower Bound', line=dict(dash='dash')))
fig.add_trace(go.Scatter(x=forecast_index, y=forecast_conf_int.iloc[:, 1],␣
 ↪mode='lines', name='Upper Bound', line=dict(dash='dash')))
fig.update_layout(title='Stock Price Forecast', xaxis_title='Date',␣
 ↪yaxis_title='Price')
fig.show()
```

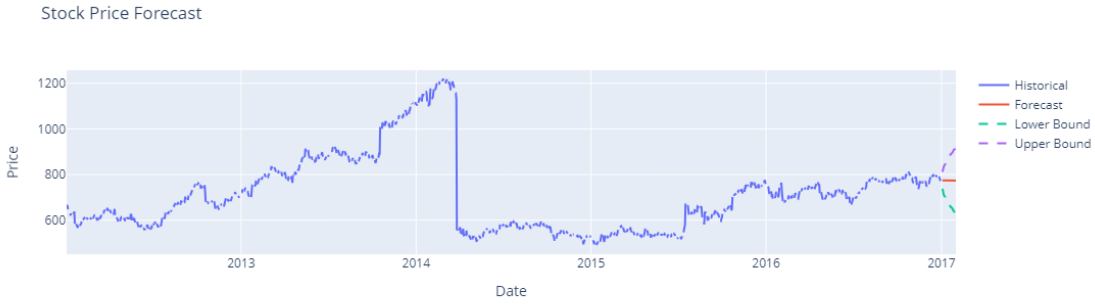
```
object
float64
```






Monthly Average Closing Prices
Average Closing Price
1200
1000
800
600
Jan 2012
Jul 2012
Jan 2013
Jul 2013
Jan 2014
Jul 2014
Jan 2015
Jul 2015
Jan 2016
Jul 2016
Month


Stock Volume Over Time
Volume
852,500
1,569,600
2,230,400
2,109,100
2,268,700
1,439,200
3,041,500
3,310,200
11,649,800
2,822,300
3,530,100
7,380,500
Jul 2012
Jan 2013
Jul 2013
Jan 2014
Jul 2014
Jan 2015
Jul 2015
Jan 2016
Jul 2016
Date


Stock Price Forecast
Price
1200
1000
800
600
2013
2014
2015
2016
2017
Date
Historical
Forecast
Lower Bound
Upper Bound

[ ]: