Assignment No-7

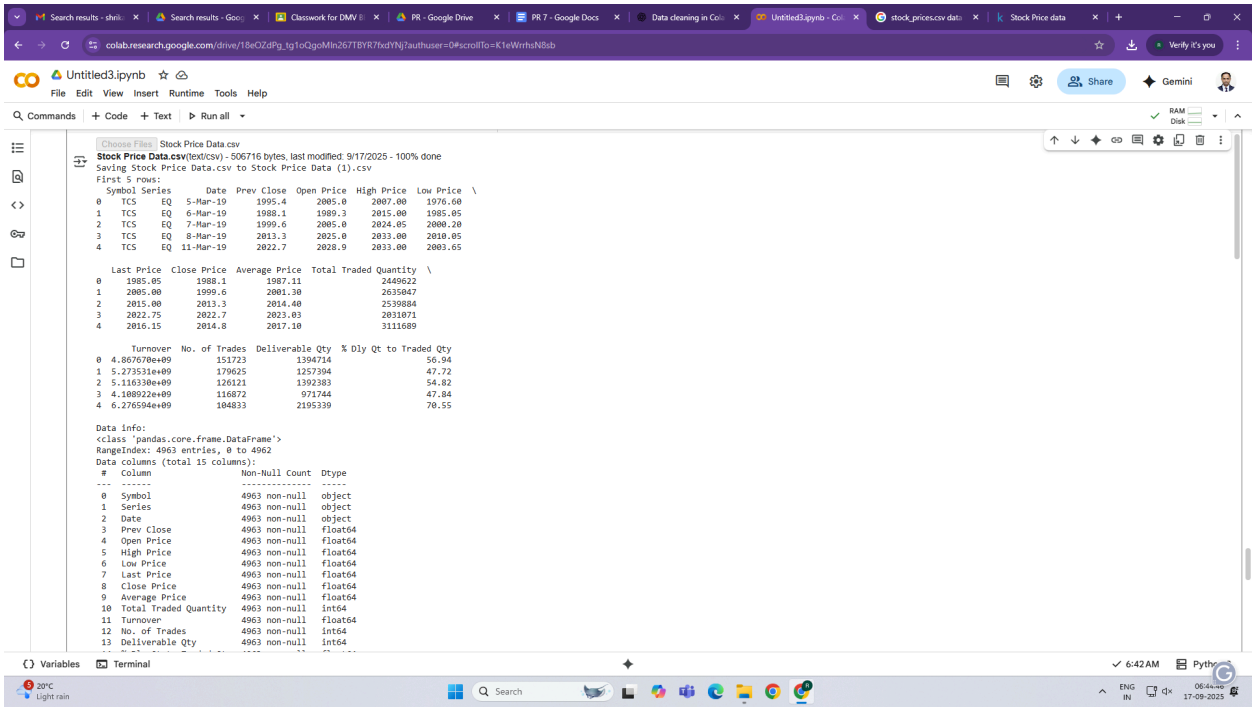
CODE:-

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from google.colab import files

sns.set(style="whitegrid")

# Upload and import dataset
uploaded = files.upload()
filename = next(iter(uploaded))
df = pd.read_csv(filename)

# Explore dataset
print("First 5 rows:")
print(df.head())

print("\nData info:")
print(df.info())

print("\nColumns:")
print(df.columns.tolist())

# Convert 'Date' column to datetime and sort
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')  # coerce errors
just in case
df = df.sort_values('Date')
df.set_index('Date', inplace=True)

# Check for missing dates or NaT after conversion
if df.index.hasnans:
    print("Warning: Some dates could not be parsed and are missing.")
```

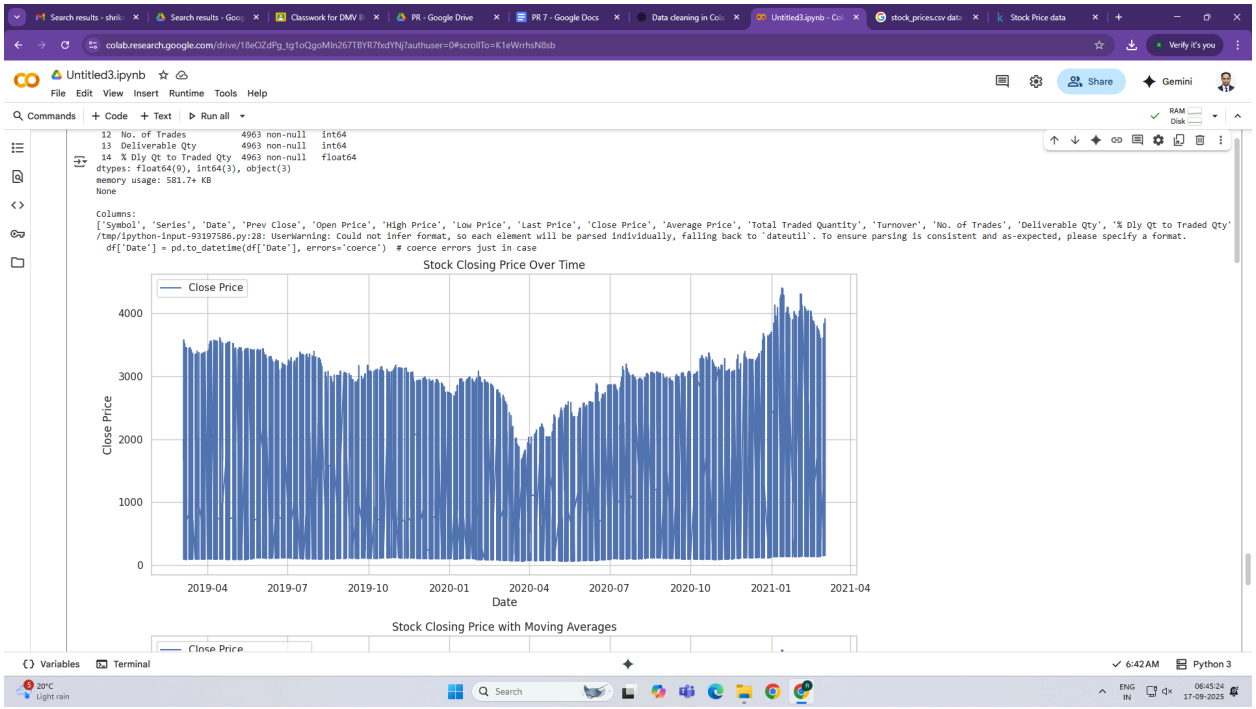
```
# Plot historical 'Close Price' trend
plt.figure(figsize=(14,6))
plt.plot(df['Close Price'], label='Close Price')
plt.title('Stock Closing Price Over Time')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.legend()
plt.show()

# Calculate and plot moving averages (20-day and 50-day)
df['MA20'] = df['Close Price'].rolling(window=20).mean()
df['MA50'] = df['Close Price'].rolling(window=50).mean()

plt.figure(figsize=(14,6))
plt.plot(df['Close Price'], label='Close Price')
plt.plot(df['MA20'], label='20-Day Moving Average')
plt.plot(df['MA50'], label='50-Day Moving Average')
plt.title('Stock Closing Price with Moving Averages')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

# Seasonal decomposition (monthly approx., period=30)
result = seasonal_decompose(df['Close Price'], model='additive',
period=30)

plt.figure(figsize=(14,10))
result.plot()
plt.suptitle('Seasonal Decomposition of Closing Prices')
plt.show()

# Correlation with 'Total Traded Quantity' (volume)
if 'Total Traded Quantity' in df.columns:
    plt.figure(figsize=(10,6))
    sns.scatterplot(x='Total Traded Quantity', y='Close Price', data=df)
    plt.title('Scatter Plot of Close Price vs Total Traded Quantity')
    plt.xlabel('Total Traded Quantity')
    plt.ylabel('Close Price')
```

```
    plt.show()
    corr = df['Close Price'].corr(df['Total Traded Quantity'])
    print(f"Correlation between Close Price and Total Traded Quantity:
{corr:.4f}")
else:
    print("No 'Total Traded Quantity' column found for correlation
analysis.")
# Forecasting using ARIMA
arima_model = ARIMA(df['Close Price'], order=(5,1,0))
arima_result = arima_model.fit()
print(arima_result.summary())

# Forecast next 30 days
arima_forecast = arima_result.forecast(steps=30)

plt.figure(figsize=(14,6))
plt.plot(df['Close Price'], label='Historical Close Price')
plt.plot(arima_forecast.index, arima_forecast, label='ARIMA Forecast',
color='red')
plt.title('ARIMA Forecast of Closing Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

# Holt-Winters Exponential Smoothing forecast
hw_model = ExponentialSmoothing(df['Close Price'], trend='add',
seasonal=None, initialization_method='estimated')
hw_fit = hw_model.fit()
hw_forecast = hw_fit.forecast(30)

plt.figure(figsize=(14,6))
plt.plot(df['Close Price'], label='Historical Close Price')
plt.plot(hw_forecast.index, hw_forecast, label='Holt-Winters Forecast',
color='green')
plt.title('Holt-Winters Forecast of Closing Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```

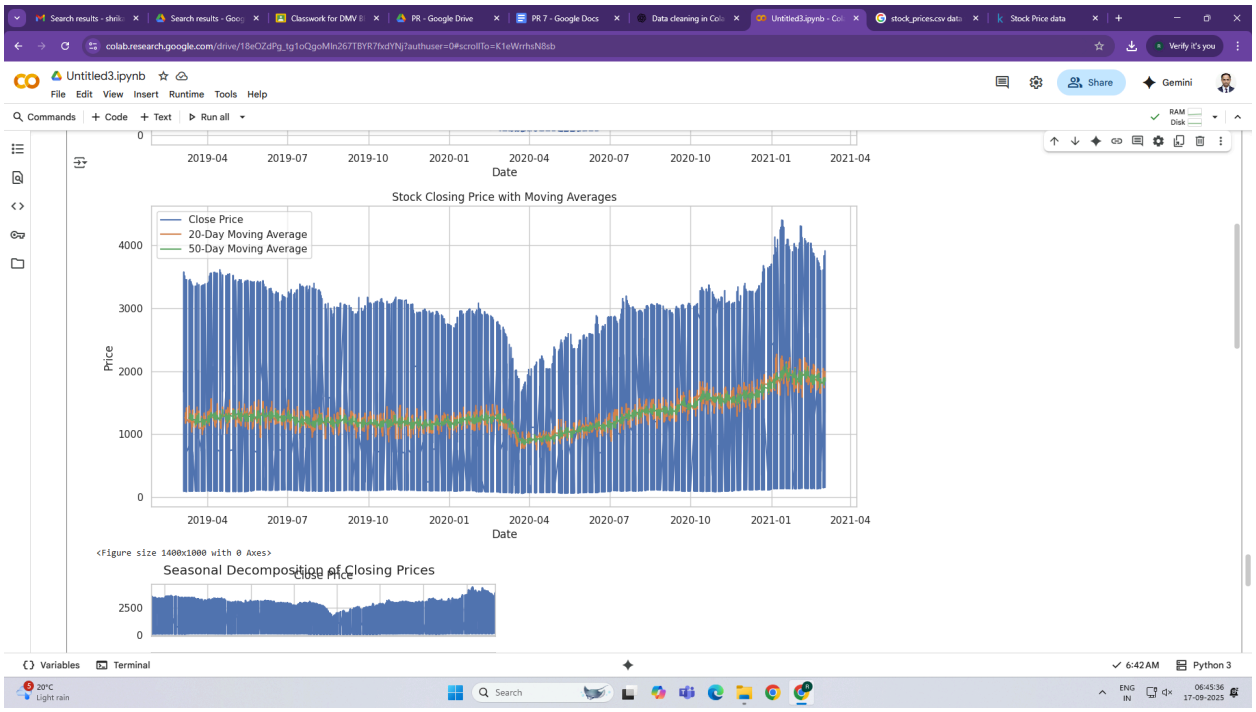
OUTPUT : –

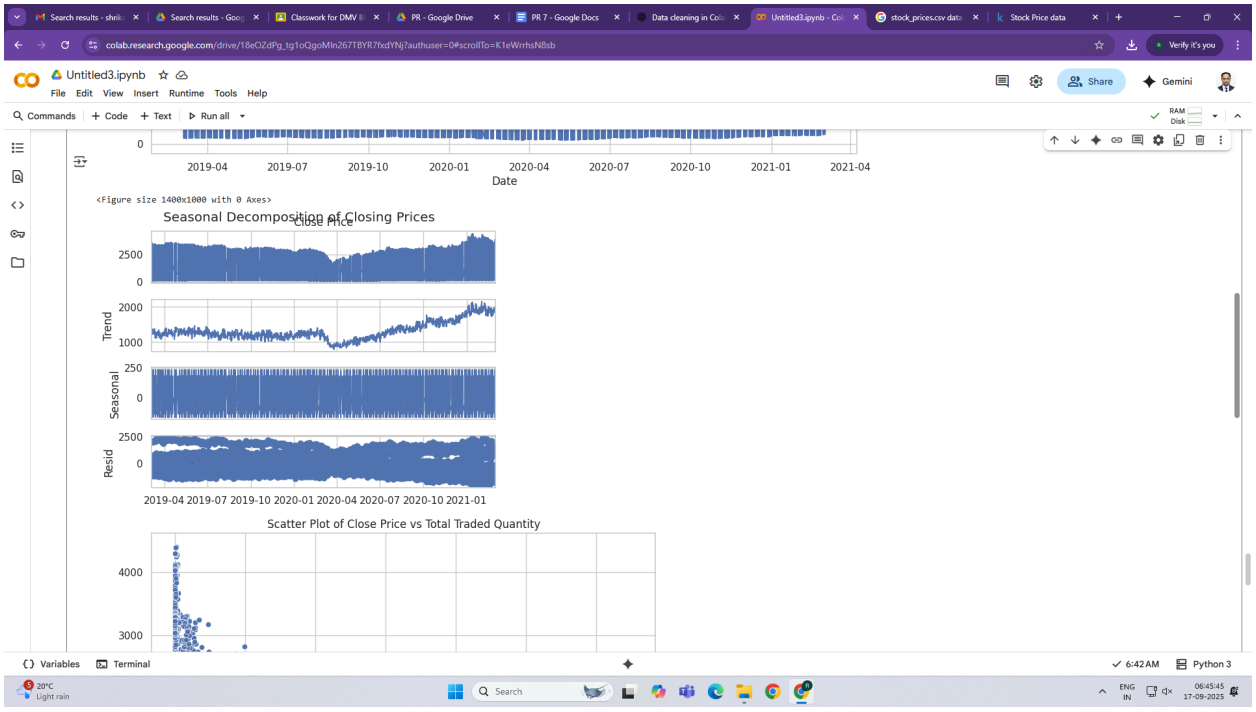


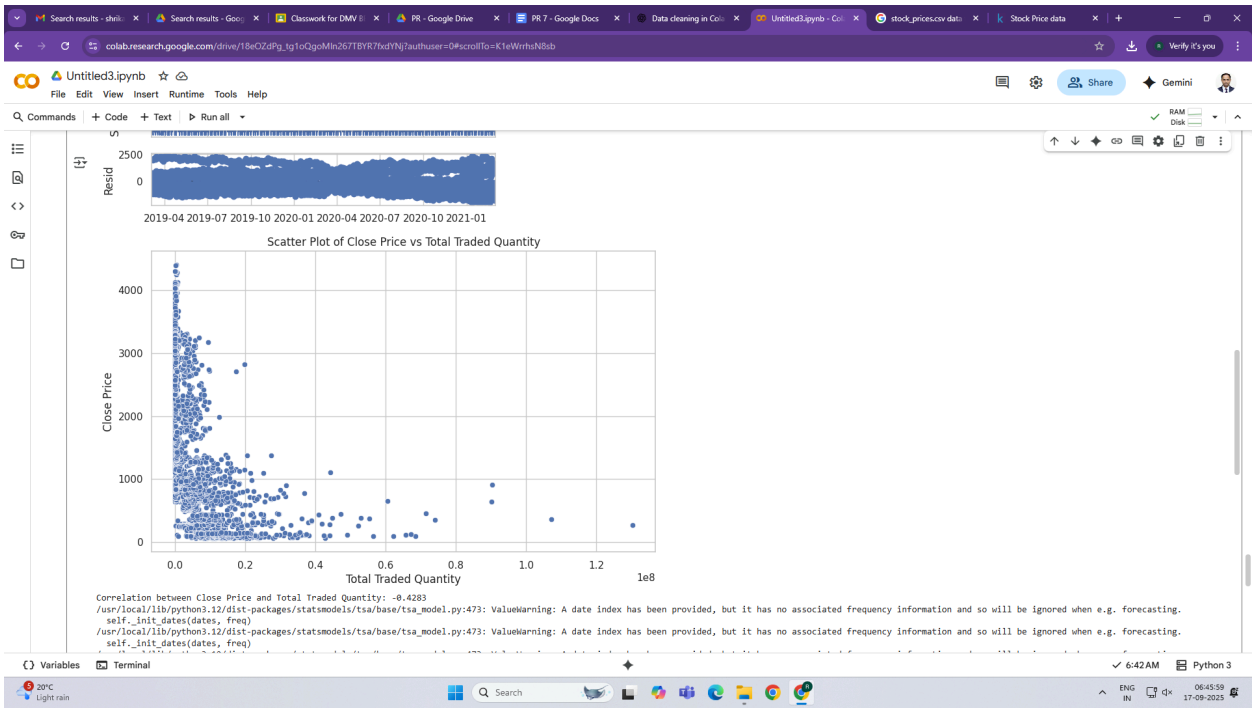
```
Choose Files  Stock Price Data.csv
Stock Price Data.csv(text/csv) - 506716 bytes, last modified: 9/17/2025 - 100% done
Saving Stock Price Data.csv to Stock Price Data (1).csv
First 5 rows:
  Symbol Series        Date  Prev Close  Open Price  High Price  Low Price  \
0    TCS     EQ   5-Mar-19      1995.4      2005.0      2007.00     1976.60
1    TCS     EQ   6-Mar-19      1988.1      1989.3      2015.00     1985.05
2    TCS     EQ   7-Mar-19      1999.6      2005.0      2024.05     2000.20
3    TCS     EQ   8-Mar-19      2013.3      2025.0      2033.00     2010.05
4    TCS     EQ  11-Mar-19      2022.7      2028.9      2033.00     2003.65

   Last Price  Close Price  Average Price  Total Traded Quantity  \
0     1985.05       1988.1        1987.11                2449622
1     2005.00       1999.6        2001.30                2635047
2     2015.00       2013.3        2014.40                2539884
3     2022.75       2022.7        2023.03                2031071
4     2016.15       2014.8        2017.10                3111689

       Turnover  No. of Trades  Deliverable Qty  % Dly Qt to Traded Qty
0  4.867670e+09         151723          1394714                   56.94
1  5.273531e+09         179625          1257304                   47.72
2  5.116330e+09         126121          1392383                   54.82
3  4.108922e+09         116872           971744                   47.84
4  6.276594e+09         104833          2195339                   70.55

Data info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4963 entries, 0 to 4962
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Symbol                4963 non-null   object
 1   Series                4963 non-null   object
 2   Date                  4963 non-null   object
 3   Prev Close            4963 non-null   float64
 4   Open Price            4963 non-null   float64
 5   High Price            4963 non-null   float64
 6   Low Price             4963 non-null   float64
 7   Last Price            4963 non-null   float64
 8   Close Price           4963 non-null   float64
 9   Average Price         4963 non-null   float64
 10  Total Traded Quantity 4963 non-null   int64
 11  Turnover              4963 non-null   float64
 12  No. of Trades         4963 non-null   int64
 13  Deliverable Qty       4963 non-null   int64
```

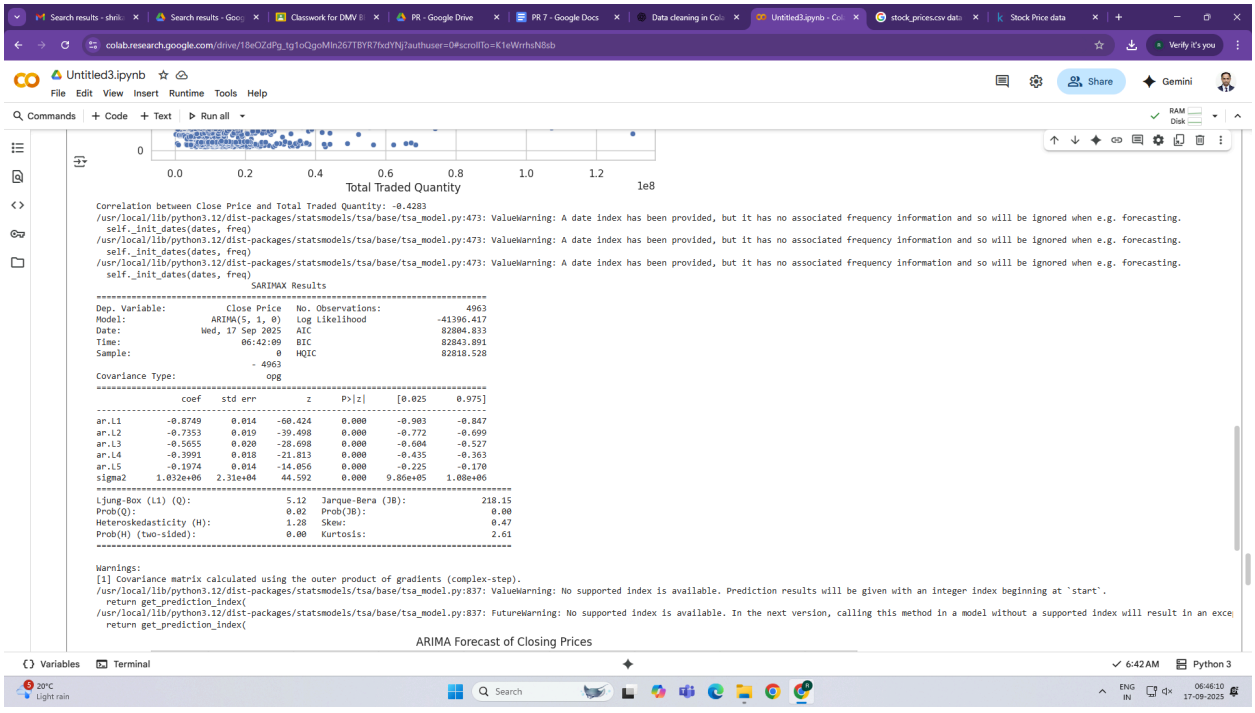


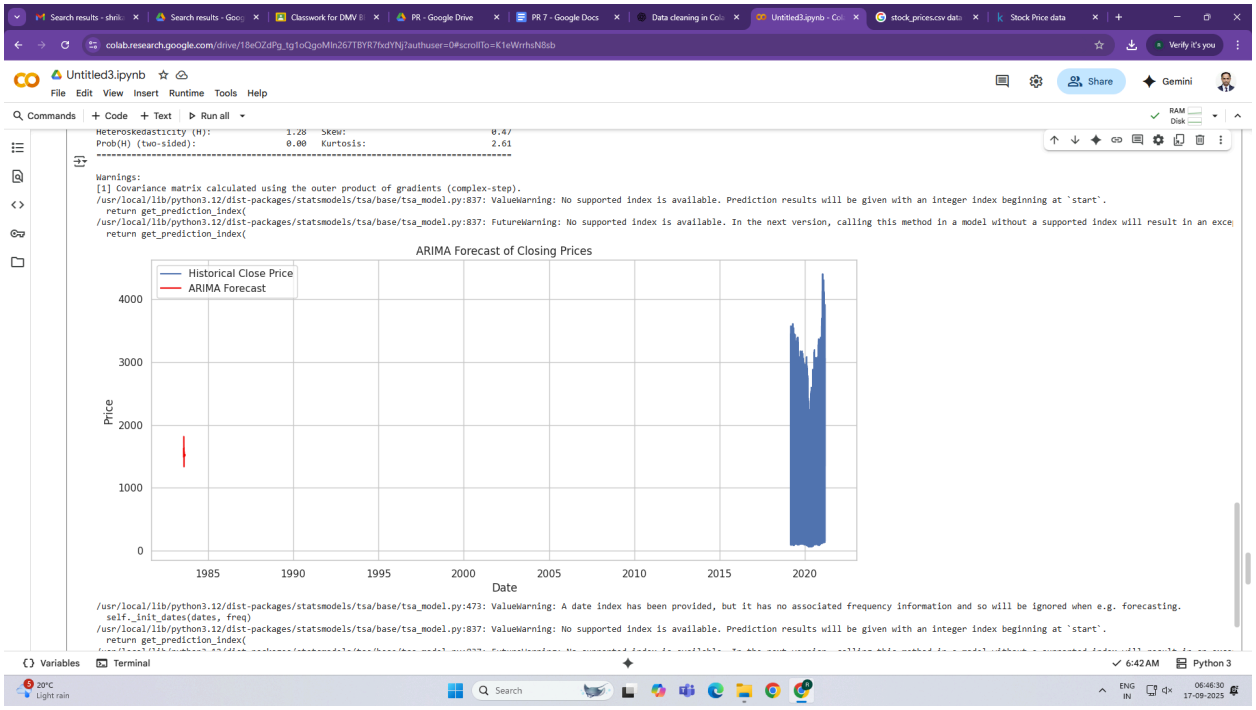
```
 12  No. of Trades         4963 non-null   int64
 13  Deliverable Qty       4963 non-null   int64
 14  % Dly Qt to Traded Qty  4963 non-null   float64
dtypes: float64(9), int64(3), object(3)
memory usage: 581.7+ KB
None

Columns:
['Symbol', 'Series', 'Date', 'Prev Close', 'Open Price', 'High Price', 'Low Price', 'Last Price', 'Close Price', 'Average Price', 'Total Traded Quantity', 'Turnover', 'No. of Trades', 'Deliverable Qty', '% Dly Qt to Traded Qty']
/tmp/ipython-input-93197586.py:28: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df['Date'] = pd.to_datetime(df['Date'], errors='coerce')  # coerce errors just in case
```

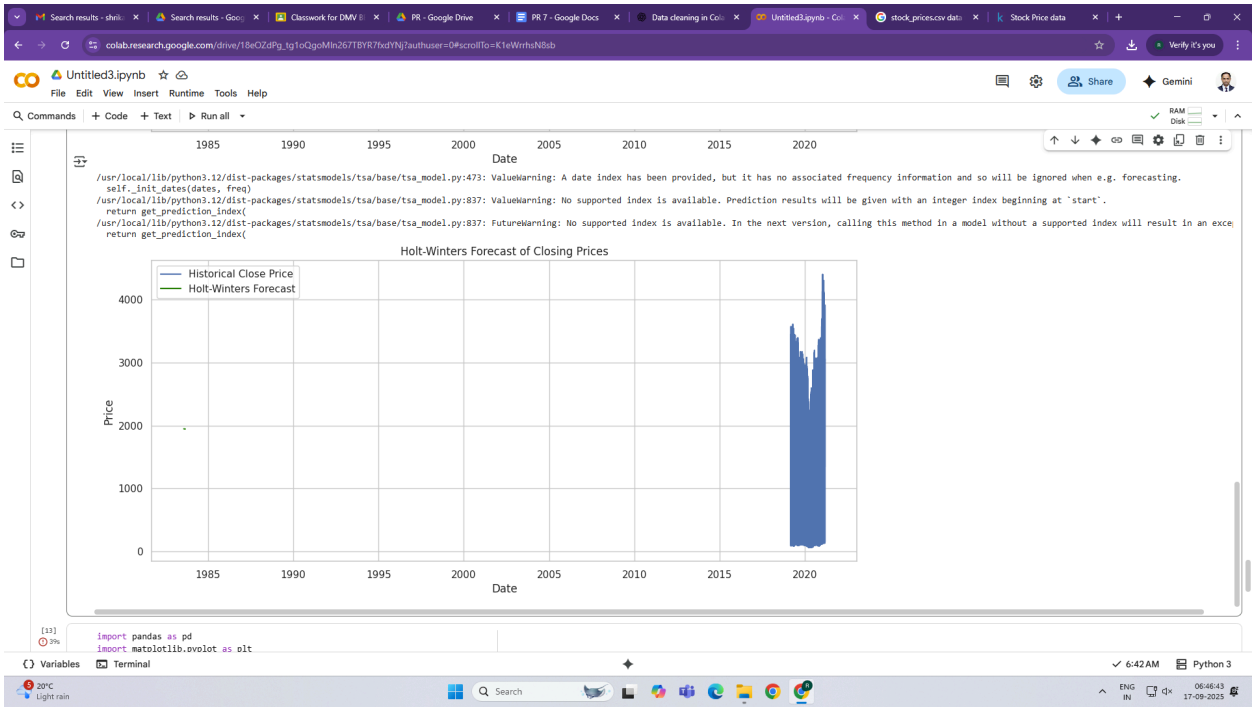

Untitled3.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
Date
Stock Closing Price with Moving Averages
Close Price
20-Day Moving Average
50-Day Moving Average
Price
Date
<Figure size 1400x1000 with 0 Axes>
Seasonal Decomposition of Closing Prices
Close Price
Variables
Terminal
6:42 AM
Python 3


Untitled3.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
Date
<Figure size 1400x1000 with 0 Axes>
Seasonal Decomposition of Closing Prices
Close Price
Trend
Seasonal
Resid
Scatter Plot of Close Price vs Total Traded Quantity
Variables
Terminal
6:42 AM
Python 3


Untitled3.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
Resid
2500
0
2019-04 2019-07 2019-10 2020-01 2020-04 2020-07 2020-10 2021-01
Scatter Plot of Close Price vs Total Traded Quantity
Close Price
4000
3000
2000
1000
0
0.0 0.2 0.4 0.6 0.8 1.0 1.2
Total Traded Quantity
1e8
Correlation between Close Price and Total Traded Quantity: -0.4283
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it has no associated frequency information and so will be ignored when e.g. forecasting.
self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it has no associated frequency information and so will be ignored when e.g. forecasting.
self._init_dates(dates, freq)
Variables Terminal
6:42 AM Python 3


Untitled3.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
0
0.0 0.2 0.4 0.6 0.8 1.0 1.2
Total Traded Quantity
1e8
Correlation between Close Price and Total Traded Quantity: -0.4283
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it has no associated frequency information and so will be ignored when e.g. forecasting.
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it has no associated frequency information and so will be ignored when e.g. forecasting.
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it has no associated frequency information and so will be ignored when e.g. forecasting.
SARIMAX Results
Dep. Variable: Close Price No. Observations: 4963
Model: ARIMA(5, 1, 0) Log Likelihood -41396.417
Date: Wed, 17 Sep 2025 AIC 82804.833
Time: 06:42:09 BIC 82843.891
Sample: 0 HQIC 82818.528
- 4963
Covariance Type: opg
coef std err z P>|z| [0.025 0.975]
ar.L1 -0.8749 0.014 -60.424 0.000 -0.903 -0.847
ar.L2 -0.7353 0.019 -39.498 0.000 -0.772 -0.699
ar.L3 -0.5655 0.020 -28.698 0.000 -0.604 -0.527
ar.L4 -0.3991 0.018 -21.813 0.000 -0.435 -0.363
ar.L5 -0.1974 0.014 -14.056 0.000 -0.225 -0.170
sigma2 1.032e+06 2.31e+04 44.592 0.000 9.86e+05 1.08e+06
Ljung-Box (L1) (Q): 5.12 Jarque-Bera (JB): 218.15
Prob(Q): 0.02 Prob(JB): 0.00
Heteroskedasticity (H): 1.28 Skew: 0.47
Prob(H) (two-sided): 0.00 Kurtosis: 2.61
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Prediction results will be given with an integer index beginning at `start`.
return get_prediction_index(
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. In the next version, calling this method in a model without a supported index will result in an exce
return get_prediction_index(
ARIMA Forecast of Closing Prices
Variables Terminal
6:42 AM Python 3


Untitled3.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
Heteroskedasticity (H): 1.28 Skew: 0.47
Prob(H) (two-sided): 0.00 Kurtosis: 2.61
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Prediction results will be given with an integer index beginning at `start`.
return get_prediction_index(
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. In the next version, calling this method in a model without a supported index will result in an exce
return get_prediction_index(
ARIMA Forecast of Closing Prices
Historical Close Price
ARIMA Forecast
Price
Date
0
1000
2000
3000
4000
1985
1990
1995
2000
2005
2010
2015
2020
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it has no associated frequency information and so will be ignored when e.g. forecasting.
self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Prediction results will be given with an integer index beginning at `start`.
return get_prediction_index(
Variables Terminal
6:42 AM Python 3
20°C Light rain
Search
ENG IN
06:46:30 17-09-2025


Untitled3.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
Date
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it has no associated frequency information and so will be ignored when e.g. forecasting.
self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Prediction results will be given with an integer index beginning at `start`.
return get_prediction_index(
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. In the next version, calling this method in a model without a supported index will result in an exce
return get_prediction_index(
Holt-Winters Forecast of Closing Prices
Historical Close Price
Holt-Winters Forecast
Price
Date
0
1000
2000
3000
4000
1985
1990
1995
2000
2005
2010
2015
2020
[33]
import pandas as pd
import matplotlib.pyplot as plt
Variables Terminal
6:42 AM Python 3
20°C Light rain
Search
ENG IN
06:46:43 17-09-2025