# Pakistan Stock Exchange Analysis Report

**Author: M Huzaifa Khan**

Date: 17.11.2024

# Contents:

**Executive Summary**

**Introduction**

**Data Overview**

**Exploratory Data Analysis**

- **Close Price Analysis**
- **Trading Volume Analysis**
- **Distribution Analysis**
- **Correlations Analysis**

**Technical Indicator Analysis**

- **Moving Average Analysis**
- **RSI Analysis**
- **Rolling Standard Deviation**
- **Average True Range (ATR)**

**Risk Analysis**

- **Value at Risk (VaR)**
- **Sharpe Ratio**

**Market Sentiment Analysis**

**Backtesting Strategy**

- **Preparing the Data for Visualization**
- **Portfolio Value over Time**

**Methodology**

**Conclusion**

**Recommendations**

**Reference**

**Index**

## Executive Summary:

This report provides an in-depth analysis of the Pakistan Stock Exchange (KSE) from 2008 to 2020, based on time series analysis, volatility analysis, risk measures, and a backtesting trading strategy. The findings highlight key market trends, volatility periods, and investment opportunities. A trading strategy based on RSI, moving averages, and volume spikes was tested with a positive outcome, indicating profitable trading opportunities in specific market conditions.

# Introduction:

The purpose of this project is to analyze the historical stock price data of the Pakistan Stock Exchange (KSE) over the period from 2008 to 2020. The analysis covers various aspects including stock price trends, trading volume, risk assessment, volatility measures, and backtesting a trading strategy.

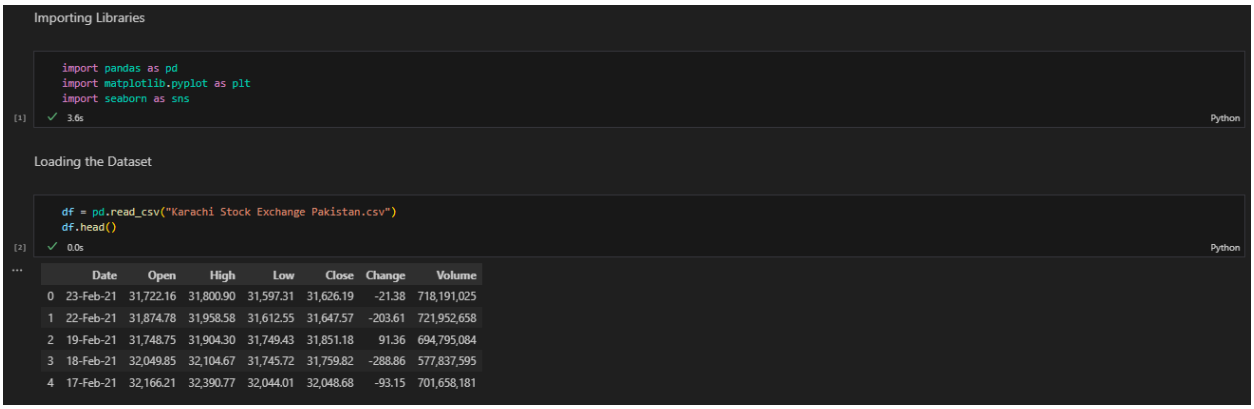
# Data Overview:

The dataset was collected from the Kaggle Datasets. The dataset consists of historical stock data from the Pakistan Stock Exchange, including the following columns:

- Date
- Open
- High
- Low
- Close
- Change
- Volume

The dataset was downloaded into a desktop folder in CSV format and later loaded into the pandas dataframe for analysis.

- **Importing Data in the Pandas Dataframe:**



Importing Libraries

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

[1] ✓ 3.6s Python

Loading the Dataset

```
df = pd.read_csv("Karachi Stock Exchange Pakistan.csv")
df.head()
```

[2] ✓ 0.0s Python

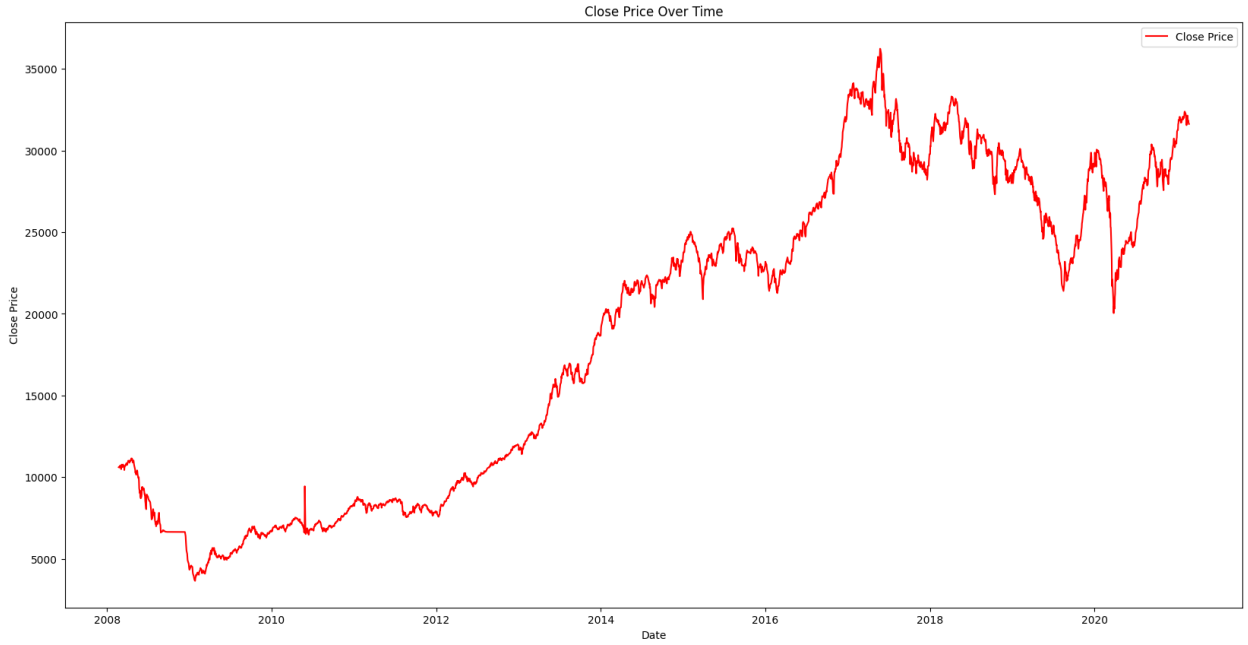
| | Date | Open | High | Low | Close | Change | Volume |
|---|---|---|---|---|---|---|---|
| 0 | 23-Feb-21 | 31,722.16 | 31,800.90 | 31,597.31 | 31,626.19 | -21.38 | 718,191,025 |
| 1 | 22-Feb-21 | 31,874.78 | 31,958.58 | 31,612.55 | 31,647.57 | -203.61 | 721,952,658 |
| 2 | 19-Feb-21 | 31,748.75 | 31,904.30 | 31,749.43 | 31,851.18 | 91.36 | 694,795,084 |
| 3 | 18-Feb-21 | 32,049.85 | 32,104.67 | 31,745.72 | 31,759.82 | -288.86 | 577,837,595 |
| 4 | 17-Feb-21 | 32,166.21 | 32,390.77 | 32,044.01 | 32,048.68 | -93.15 | 701,658,181 |

# Exploratory Data Analysis (EDA):

- **Close Price Time Series Analysis:**

```
# Time Series Analysis

plt.figure(figsize=(20, 10))
plt.plot(df['Date'], df['Close'], label="Close Price", color="red")
plt.title("Close Price Over Time")
plt.xlabel("Date")
plt.ylabel("Close Price")
plt.legend()
plt.show()
```

[9] ✓ 0.5s Python

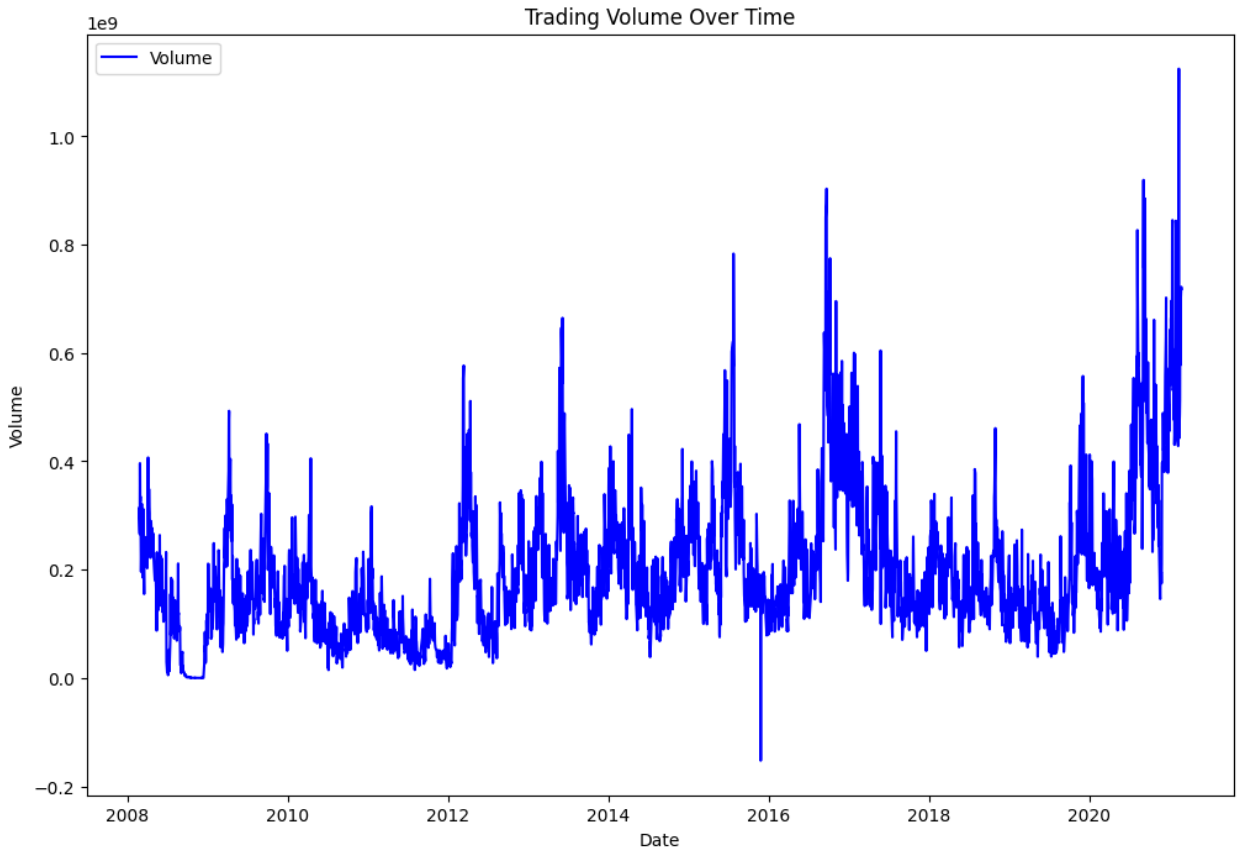



The stock price of the Pakistan Stock Exchange began at around Rs 10,000 in 2008, experienced a significant drop in 2009, and gradually increased, peaking at over Rs 35,000 in 2017. From 2017 to 2020, the price steadily declined back to around Rs 20,000.

- **Trading Volume Time Series Analysis:**

```
plt.figure(figsize=(25, 10))
plt.plot(df['Date'], df['Volume'], label="Volume", color="blue")
plt.title("Trading Volume Over Time")
plt.xlabel("Date")
plt.ylabel("Volume")
plt.legend()
plt.show()
```

[10] ✓ 0.3s Python




There were notable spikes in trading volume in the following years:

- 2012
- 2016
- 2020

Periods of low trading volume were observed during:

- 2008-2009
- 2012
- 2017

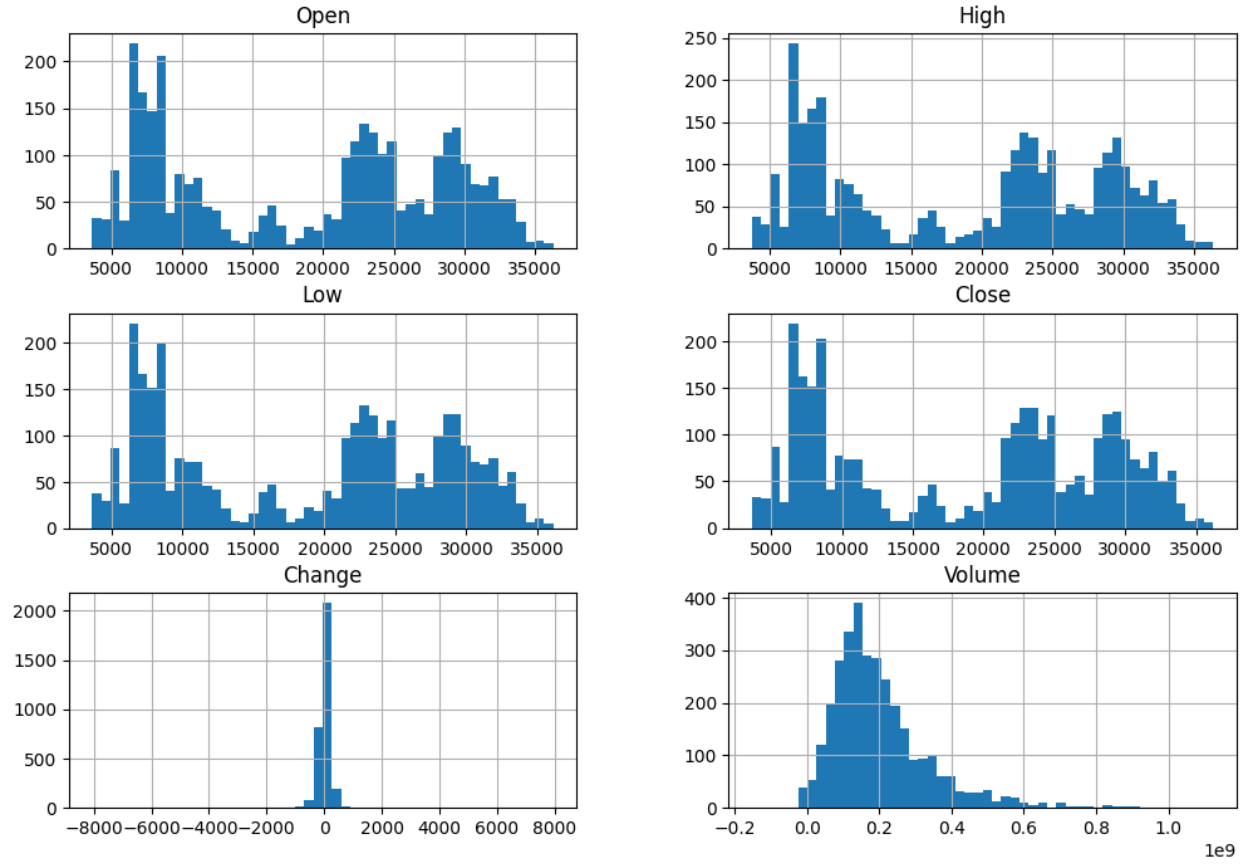
There were notable spikes in trading volume during the years 2012, 2016, and 2020, indicating periods of heightened market activity. Conversely, trading volume was relatively low during certain periods, particularly from 2008 to 2009, 2012, and 2017. These fluctuations in volume suggest that the market experienced varying levels of investor interest and trading activity, with

certain years seeing significant surges in volume, while other years were characterized by quieter market conditions.

- **Distribution Analysis:**

```
# Distribution Analysis

df_numeric = df.drop(columns=['Date'])
df_numeric.hist(bins=50, figsize=(12, 8))
```

[11] ✓ 1.3s Python




The distribution of **High**, **Low**, **Open**, and **Close** prices shows two major peaks: one around Rs 7,000 and another around Rs 23,000, this suggests that there are two points where the prices are more concentrated. The prices are also skewed to the right, this means that there are more prices below the mean than above the mean.

The **Change** histogram shows a rightward skew, indicating more days of price decrease than increase. The highest bar lies on **0,** which shows there no major changes in the price either not positively or negatively.

The **Volume** histogram is also skewed right, suggesting more days with low trading volume than high.

- **Correlation Analysis:**

Correlation Analysis

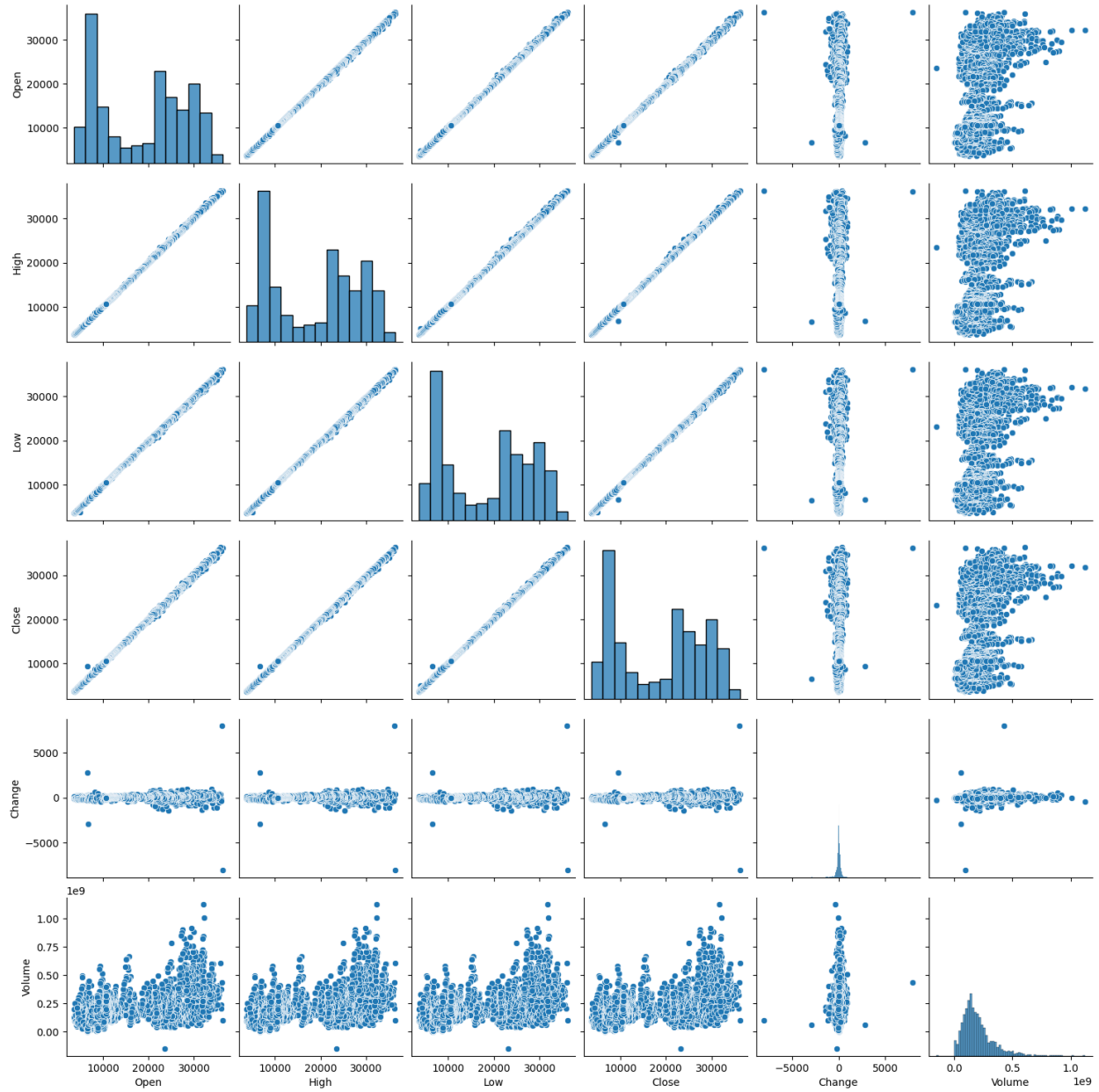
```
correlation_matrix = df.corr()
print(correlation_matrix)
```

[13] ✓ 0.0s Python

```
            Date      Open      High       Low     Close    Change    Volume
Date    1.000000  0.911607  0.912197  0.911230  0.911787  0.008685  0.384619
Open    0.911607  1.000000  0.999933  0.999891  0.999800 -0.000276  0.428193
High    0.912197  0.999933  1.000000  0.999889  0.999901  0.005177  0.430356
Low     0.911230  0.999891  0.999889  1.000000  0.999914  0.007641  0.429827
Close   0.911787  0.999800  0.999901  0.999914  1.000000  0.013363  0.430720
Change  0.008685 -0.000276  0.005177  0.007641  0.013363  1.000000  0.125037
Volume  0.384619  0.428193  0.430356  0.429827  0.430720  0.125037  1.000000
```

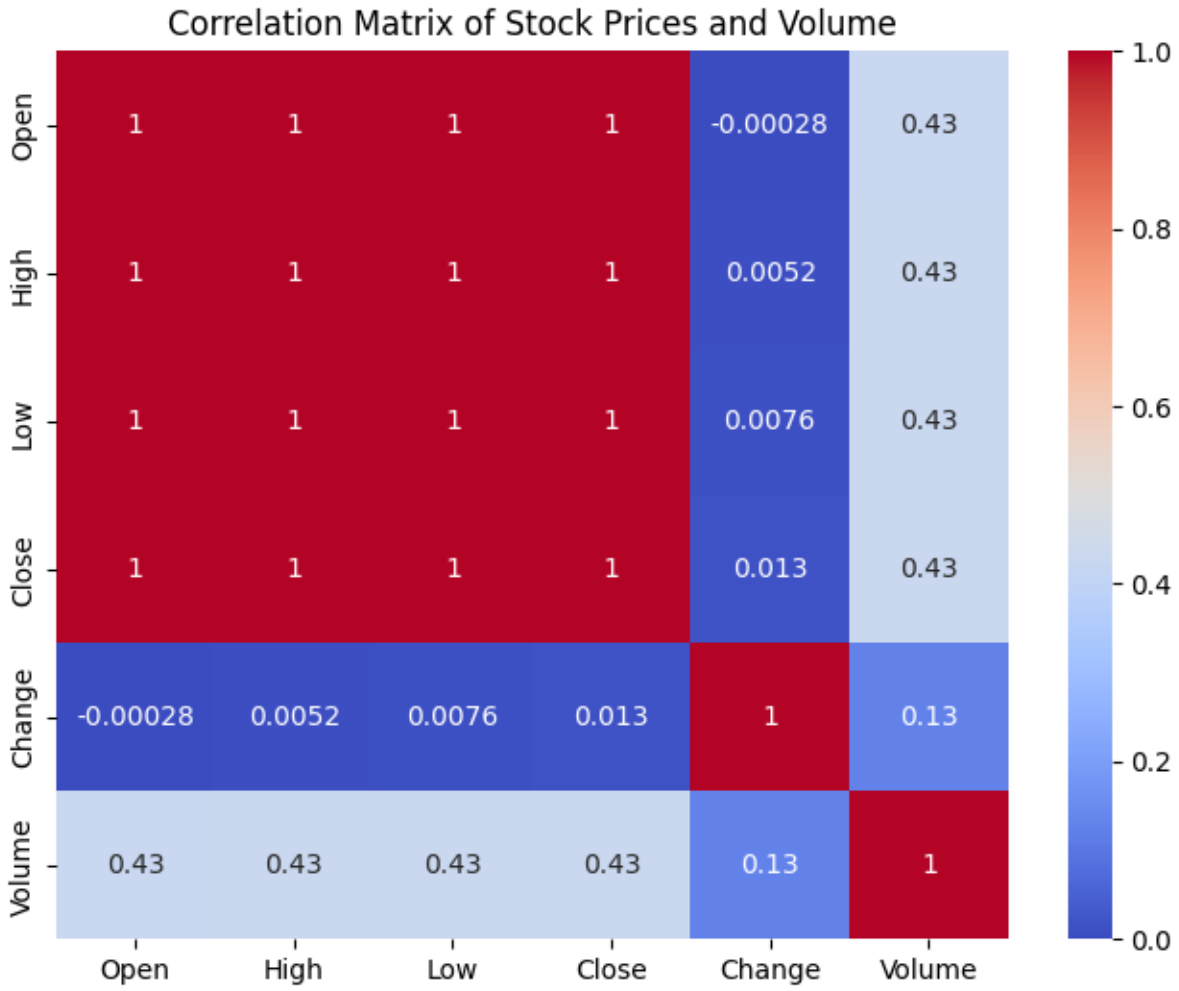
```
sns.pairplot(df)
```

[14] ✓ 12.6s Python



```
plt.figure(figsize=(12, 8))
sns.heatmap(df[['Open', 'High', 'Low', 'Close', 'Change', 'Volume']].corr(), annot=True, cmap="coolwarm")
plt.title("Correlation matrix of stock Price and Volume")
plt.show()
```

[15] ✓ 0.3s Python




1. **Strongly Positive Correlation:** The OPEN, HIGH, LOW, and CLOSE prices are highly positively correlated. This is expected as those represent different points in the same trading day.
2. **Weak Positive Correlation**: **Volume** is slightly positively correlated with the other price columns. This suggests that when a stock is traded more frequently, it tends to have a higher Value.
3. **No Significant Correlation**: The **Change** in price does not show any strong correlation with other variables. This indicates that the daily change in price is not strongly influenced by OPENING, HIGH, LOW, CLOSE price, or VOLUME.

# Technical Indicator Analysis:

- **Moving Average Analysis:**

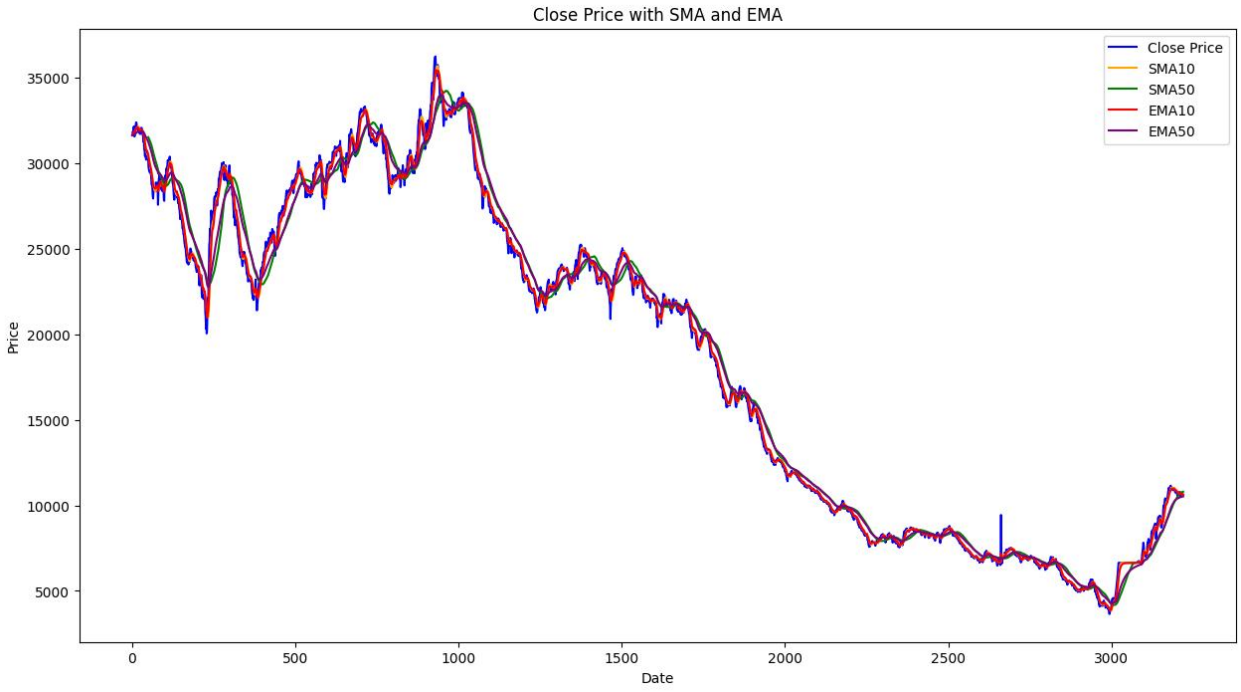
Moving Average Analysis

```
short_window = 10
long_window = 50

df['Short_SMA'] = df['Close'].rolling(window=short_window).mean()
df['Long_SMA'] = df['Close'].rolling(window=long_window).mean()

df['Short_EMA'] = df['Close'].ewm(span=short_window, adjust=False).mean()
df['Long_EMA'] = df['Close'].ewm(span=long_window, adjust=False).mean()
```

[16] ✓ 0.0s Python

```
# Ploting Close Price with SMA and EMA

plt.figure(figsize=(15, 8))
plt.plot(df['Close'], label="Close Price", color="blue")
plt.plot(df['Short_SMA'], label=f'SMA{short_window}', color="orange")
plt.plot(df['Long_SMA'], label=f'SMA{long_window}', color="green")
plt.plot(df['Short_EMA'], label=f'EMA{short_window}', color="red")
plt.plot(df['Long_EMA'], label=f'EMA{long_window}', color="purple")


plt.title("Close Price with SMA and EMA")
plt.xlabel("Date")
plt.ylabel("Price")
plt.legend()
plt.show()
```

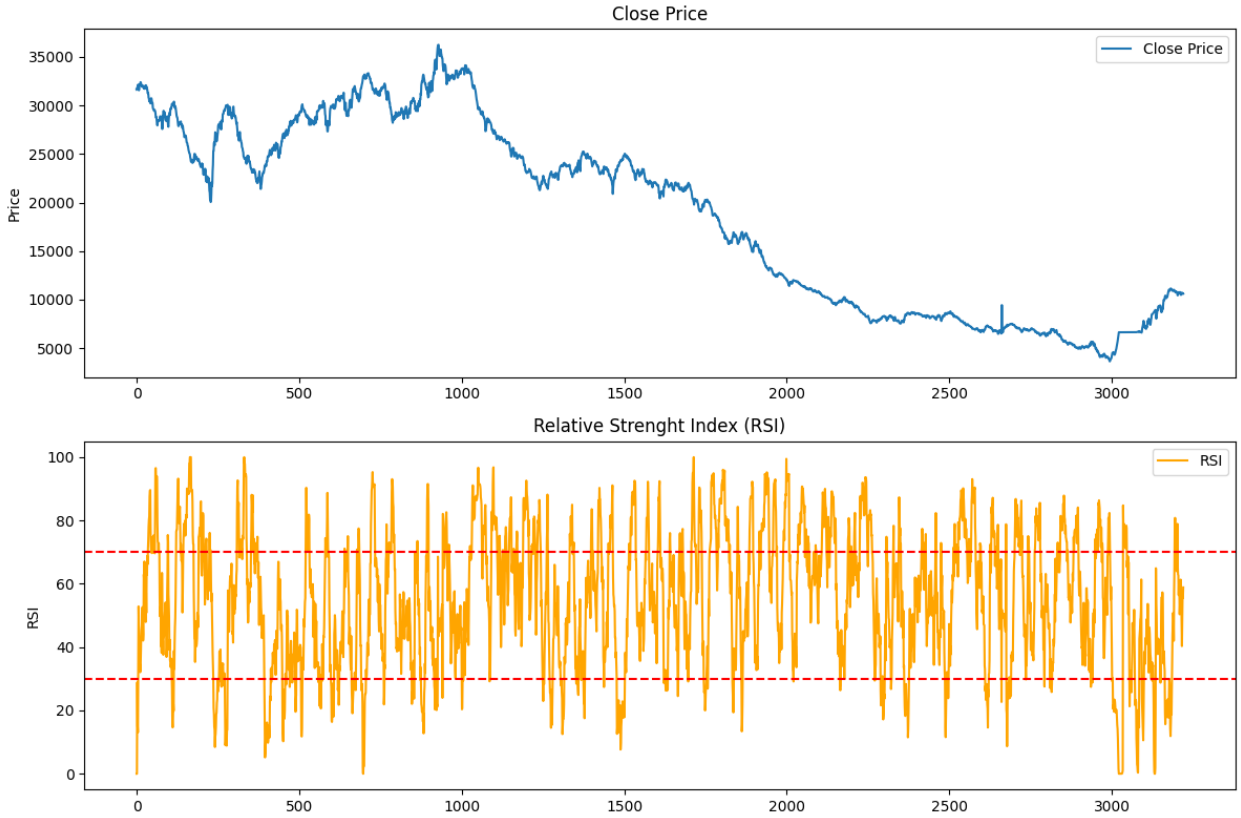
[17] ✓ 0.2s Python




The moving average of the stock price a general downward trend over time, as the price has declined from earlier highs to much lower levels. This downtrend might suggest a caution approach, where buying signals are only strong if the short-term moving average stay consistently above the long-term average.

- **Relative Strength Index (RSI):**

```
Relative Strenght Index (RSI)

# calculating the Gain and Loss

df['Gain'] = df['Change'].apply(lambda x: x if x > 0 else 0)
df['Loss'] = df['Change'].apply(lambda x: -x if x < 0 else 0)

# calculating the average Gain and the average Loss

window_lenght = 14
df['Avg_Gain'] = df['Gain'].rolling(window=window_lenght, min_periods=1).mean()
df['Avg_Loss'] = df['Loss'].rolling(window=window_lenght, min_periods=1).mean()

# Calculating the RS and the RSI

df['RS'] = df['Avg_Gain'] / df['Avg_Loss']
df['RSI'] = 100 - (100 / (1 + df['RS']))
[20] ✓ 0.0s Python

fig, (axis1, axis2) = plt.subplots(2, 1, figsize=(12, 8))

# Plotting the subplot of Close price
axis1.plot(df['Close'], label = "Close Price")
axis1.set_title('Close Price')
axis1.set_ylabel("Price")
axis1.legend()

# Plotting the subplot of RSI

axis2.plot(df['RSI'], label='RSI', color='orange')
axis2.axhline(70, color='red', linestyle='--')
axis2.axhline(30, color='red', linestyle='--')
axis2.set_title("Relative Strenght Index (RSI)")
axis2.set_ylabel("RSI")
axis2.legend()

plt.tight_layout()
plt.show()
[21] ✓ 0.5s Python
```



The Relative Strength Index (RSI) fluctuated mostly within the 30-70 range, indicating a sideways or range-bound market. It shows:

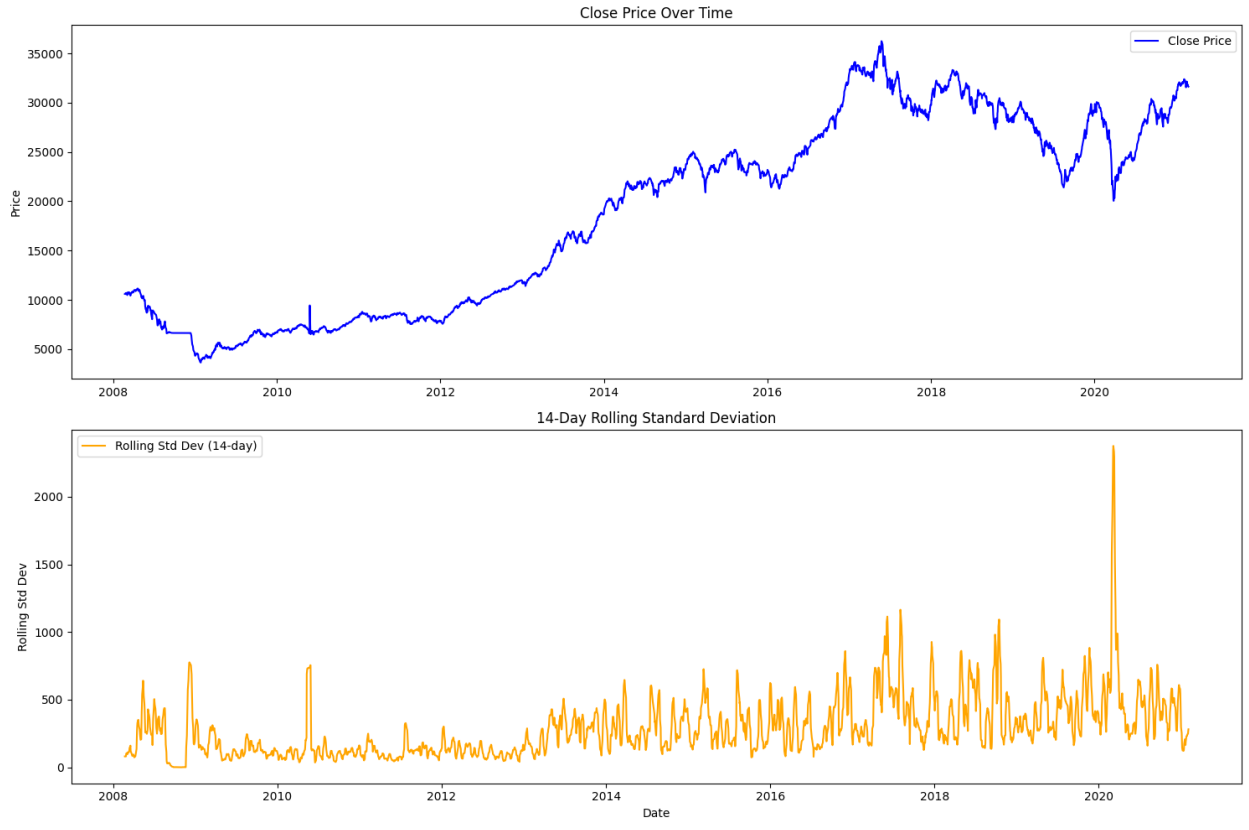
1. **No strong momentum**: The market lacked strong buying or selling pressure. The RSI within the 30-70 range suggests that the asset isn't experiencing strong buying or selling pressure. It indicates that buyers and sellers are relatively balanced, leading to smaller fluctuations around a median price rather than strong uptrends or downtrends.
2. **Stable market conditions**: Price changes were gradual without extreme volatility.
3. **Short-term trading opportunities**: Suitable for range trading, buying near 30 and selling near 70.

- **Rolling Standard Deviation:**

Rolling Standard Deviation

```
window_lenght = 14
df['Rolling_Std_Dev'] = df['Close'].rolling(window=window_lenght).std()
```

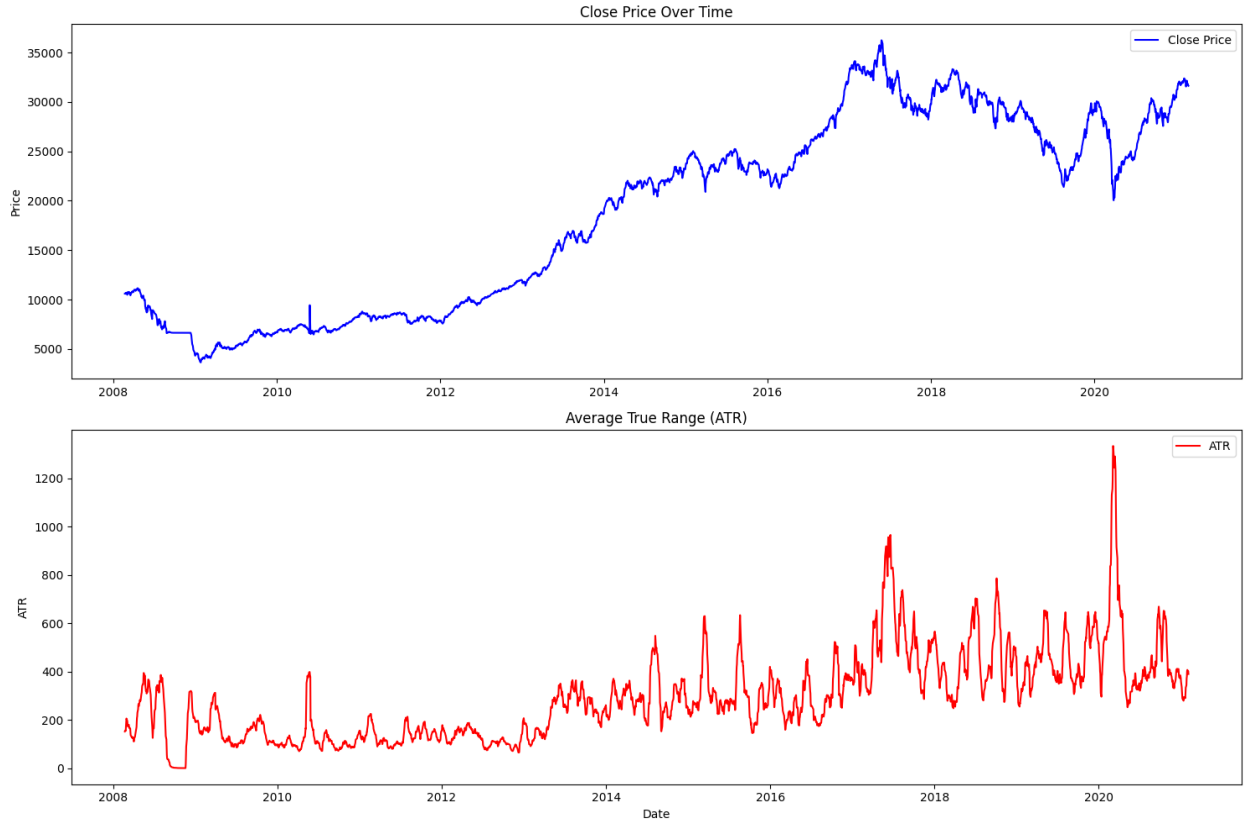
```
# Plot Close Price and Rolling Standard Deviation

fig, (axis1, axis2) = plt.subplots(2, 1, figsize=(15, 10))

# Plot Close Price

axis1.plot(df['Date'], df['Close'], color='blue', label='Close Price')
axis1.set_title('Close Price Over Time')
axis1.set_ylabel('Price')
axis1.legend()

# Plot Rolling Standard Deviation

axis2.plot(df['Date'], df['Rolling_Std_Dev'], color='orange', label=f'Rolling Std Dev ({14}-day)')
axis2.set_title(f'{14}-Day Rolling Standard Deviation')
axis2.set_ylabel('Rolling Std Dev')
axis2.set_xlabel('Date')
axis2.legend()

plt.tight_layout()
plt.show()
```



The top graph shows the closing price of a stock over time. We can see that the stock price has generally been trending upwards since 2008. The bottom graph shows the rolling standard deviation of the closing price. The spike in the rolling standard deviation around 2020 indicates that the closing price was very volatile during that period. This is likely due to the COVID-19 pandemic, which had a major impact on the stock market. Overall, these graphs show that the stock has been relatively stable, with a period of high volatility around the time of the COVID-19 pandemic.

- **Average True Range (ATR):**

```
Average True Range (ATR)

# Calculating the TR

df['High-Low'] = df['High'] - df['Low']
df['High-PrevClose'] = abs(df['High'] - df['Close'].shift(1))
df['Low-PrevClose'] = abs(df['Low'] - df['Close'].shift(1))
df['TR'] = df[['High-Low', 'High-PrevClose', 'Low-PrevClose']].max(axis=1)

# Calculating the ATR

window_lenght = 14
df['ATR'] = df['TR'].rolling(window=window_lenght).mean()
[24] ✓ 0.0s                                                                    Python

fig, (axis1, axis2) = plt.subplots(2, 1, figsize=(15, 10))

# Plot Close Price

axis1.plot(df['Date'], df['Close'], color='blue', label='Close Price')
axis1.set_title("Close Price Over Time")
axis1.set_ylabel("Price")
axis1.legend()

# Plot ATR

axis2.plot(df['Date'], df['ATR'], color='red', label='ATR')
axis2.set_title("Average True Range (ATR)")
axis2.set_ylabel("ATR")
axis2.set_xlabel("Date")
axis2.legend()

plt.tight_layout()
plt.show()
[25] ✓ 0.8s                                                                    Python
```

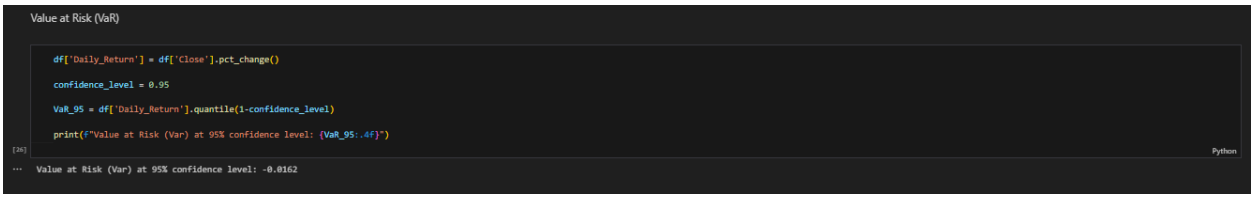


The average true range reveals that the stock price trend is generally upward from 2008 to 2020, with some periods of downward volatility. The ATR is relatively low for most of the period, but there are several spikes in volatility. The most significant spike occurred in 2020, which corresponds to the market crash caused by the COVID-19 pandemic.

This suggest that the stock has been relatively stable over time, with periods of heightened volatility. The high ATR spike in 2020 shows the impact of major market events on the stock's price movement.

# Risk Analysis:

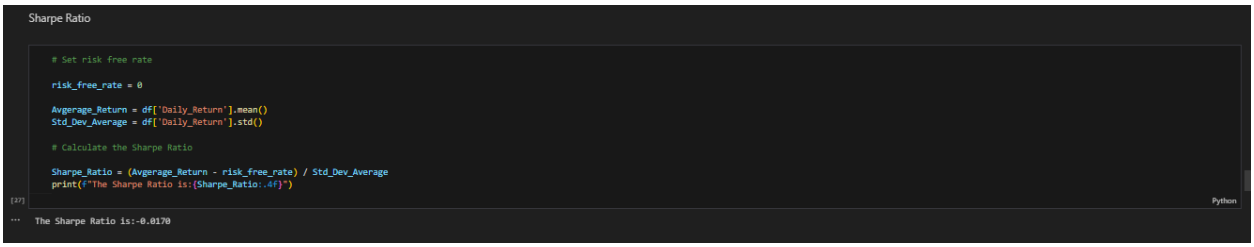
- **Value at Risk (VaR):**




***Result:*** *Value at Risk (VaR) at 95% confidence level: -0.0162*

Accordingly case, a VaR of -0.0162 at a 95% confidence level means that there is a 95% probability that the loss on the investment or portfolio will not exceed 1.62% over the specified period (in our case, one day). Conversely, there is a 5% chance that the loss could exceed this amount.

- **Sharpe Ratio:**




***Result:*** *The Sharpe Ratio is: -0.0170*

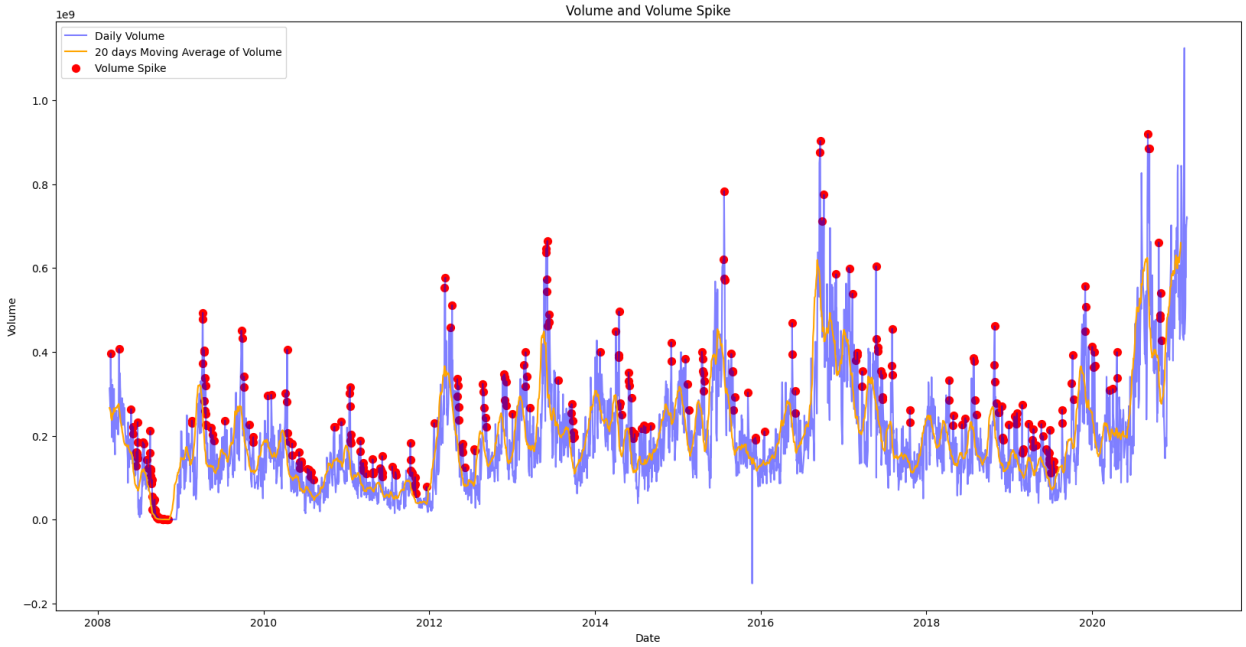
A Sharpe Ratio of -0.0170 indicates that the investment has underperformed the risk-free rate when adjusted for risk. In other words, the investment returns are insufficient to compensate for the risk taken. A negative Sharpe Ratio suggests that the investment is not providing a good risk-adjusted return. Investors typically seek a positive Sharpe Ratio, as it indicates that the investment generates returns that exceed the risk-free rate when considering the volatility of those returns. Since the Sharpe Ratio is negative, it implies that the average return of the investment is less than the risk-free rate. This could mean that the investment is failing to provide excess returns and potentially losing value when compared to a virtually risk-free investment. A Sharpe Ratio close to zero or negative may lead investors to reconsider their investment strategy. It may suggest that the investment is not worth the risk compared to safer alternatives, such as government bonds or other low-risk assets.

## Market Sentiment Analysis:

Market Sentiment Analysis

```
# Calculate the Moving Average of Volume

df['Volume_MA'] = df['Volume'].rolling(window=20).mean()

threshold = 1.5
df['Volume_Spike'] = df['Volume'] > (threshold * df['Volume_MA'])

# Plotting Volume and highlighting the spikes

plt.figure(figsize=(20, 10))
plt.plot(df['Date'], df['Volume'], label='Daily Volume',  color='blue', alpha=0.5)
plt.plot(df['Date'], df['Volume_MA'], label='20 days Moving Average of Volume', color='orange')

plt.scatter(df[df['Volume_Spike']]['Date'], df[df['Volume_Spike']]['Volume'], label='Volume Spike', color='red', marker='o', s=50)
plt.title("Volume and Volume Spike")
plt.xlabel("Date")
plt.ylabel("Volume")
plt.legend()
plt.show()
```

[28] Python



The graph highlights the daily trading volume and its 20-day moving average, with red dots marking significant "volume spikes" where trading activity surpasses the moving average. The overall trend shows a gradual decrease in volume over time, with intermittent increases during specific periods. Notably, higher trading activity is observed in late 2016 and early 2017, indicating potential seasonality. Volume spikes occur throughout the timeframe, reflecting periods of heightened market activity, and are positively correlated with the moving average, which rises during and after these spikes.

# Backtesting Strategy:

Backtesting Strategy Analysis

```
# Define buy condition based on RSI, Moving Average Crossover, and Volume Spike
df['Buy_Signal'] = (
    (df['RSI'] < 30) |  # Buy when RSI indicates oversold
    ((df['Short_SMA'] > df['Long_SMA']) & (df['Short_SMA'].shift(1) <= df['Long_SMA'].shift(1))) |  # Buy on SMA crossover
    (df['Volume_Spike'])  # Volume spike confirms interest
)

# Define sell condition
df['Sell_Signal'] = (
    (df['RSI'] > 70) |  # Sell when RSI indicates overbought
    ((df['Short_SMA'] < df['Long_SMA']) & (df['Short_SMA'].shift(1) >= df['Long_SMA'].shift(1)))  # Sell on SMA cross down
)
```

Python

```
# Initialize a few variables
initial_cash = 100000  # starting cash in PKR
cash = initial_cash
position = 0  # No shares held at start
trade_log = []

for index, row in df.iterrows():
    if row['Buy_Signal'] and cash > 0:
        # Buy action
        position = cash / row['Close']  # number of shares
        cash = 0  # all cash is invested
        trade_log.append(('Buy', index, row['Date'], row['Close'], cash, position))

    elif row['Sell_Signal'] and position > 0:
        # Sell action
        cash = position * row['Close']  # liquidate the position
        position = 0
        trade_log.append(('Sell', index, row['Date'], row['Close'], cash, position))

# Final portfolio value
final_value = cash + (position * df['Close'].iloc[-1])  # account for any unsold position
return_percentage = ((final_value - initial_cash) / initial_cash) * 100

print(f"Backtesting completed. Final portfolio value: Rs{final_value:.2f}")
print(f"Total return: {return_percentage:.2f}%")
```

Python

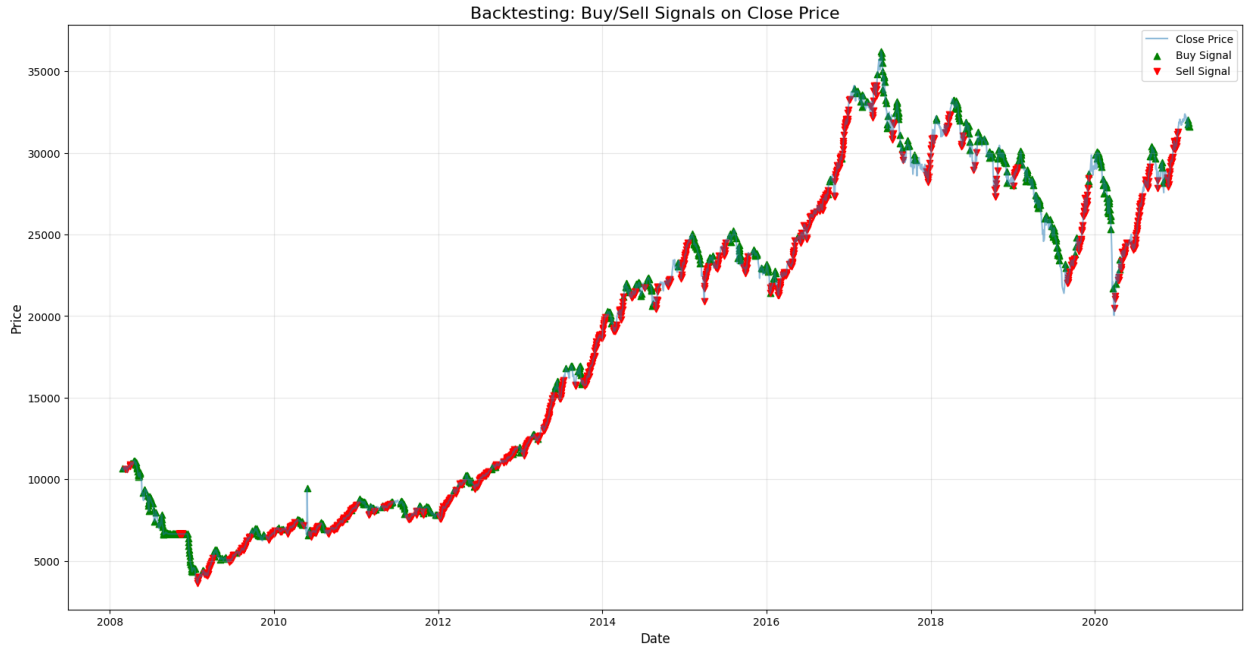
***Result:*** *Backtesting completed. Final portfolio value: Rs313506.64*

*Total return: 213.51%*

- **Preparing the Data for Visualization:**

```
# Prepare the data for visualization

df['Portfolio_Value'] = cash + (position * df['Close'])  # Portfolio value at each step
buy_signals = df[df['Buy_Signal']]  # Filter rows where Buy_Signal is True
sell_signals = df[df['Sell_Signal']]  # Filter rows where Sell_Signal is True

# Plot the Close prices

plt.figure(figsize=(20, 10))
plt.plot(df['Date'], df['Close'], label='Close Price', alpha=0.5)

# Plot the buy signals

plt.scatter(buy_signals['Date'], buy_signals['Close'], label='Buy Signal', marker='^', color='green', alpha=1)

# Plot the sell signals

plt.scatter(sell_signals['Date'], sell_signals['Close'], label='Sell Signal', marker='v', color='red', alpha=1)

plt.title('Backtesting: Buy/Sell Signals on Close Price', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Price', fontsize=12)
plt.legend()
plt.grid(alpha=0.3)

plt.show()
```

Python

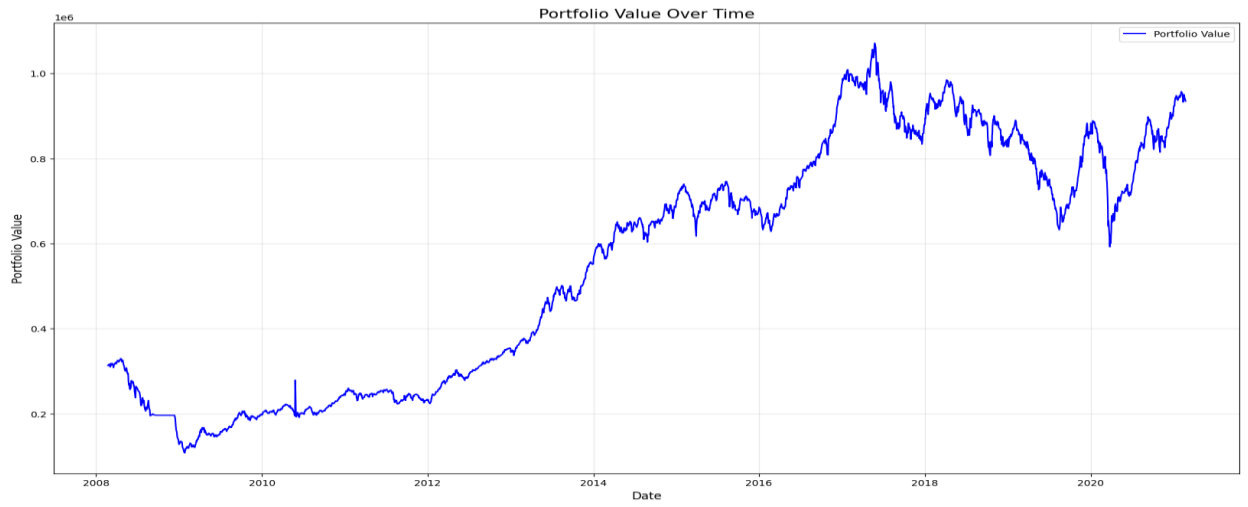



Conducted a backtesting analysis using a trading strategy that combined **RSI**, **Moving Average Crossovers**, and **Volume Spikes** to generate Buy and Sell signals. The strategy involved buying when RSI indicated oversold conditions (RSI < 30), short SMA crossed above long SMA, or a volume spike occurred, and selling when RSI indicated overbought conditions (RSI > 70) or short SMA crossed below long SMA. Starting with an initial capital of Rs 100,000, the backtesting resulted in a final portfolio value of Rs 313,506.64, achieving a **213.51% return**. This indicates strong profitability over the tested period, with trades logged systematically and visualized alongside closing price data.

- **Portfolio Value over Time:**

```
# Plot the portfolio value over time

plt.figure(figsize=(20, 10))
plt.plot(df['Date'], df['Portfolio_Value'], label='Portfolio Value', color='blue')

plt.title('Portfolio Value Over Time', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Portfolio Value', fontsize=12)
plt.legend()
plt.grid(alpha=0.3)

plt.show()
```

0.2s Python



The graph depicts a portfolio that has grown significantly over time, exhibiting periods of volatility and resilience. It suggests a portfolio that has navigated market fluctuations well and delivered strong returns.

## Methodology:

The methodology used in this project includes:

1. **Data Collection**: The data was sourced from Kaggle Datasets, which includes stock price data (Open, High, Low, and Close), trading volume, and changes in stock price.
2. **Exploratory Data Analysis (EDA)**: This step involved visualizing trends, distributions, and correlations in the data using various charts like line graphs, histograms, and heatmaps.
3. **Technical Indicator Calculation**: Key technical indicators such as Moving Averages, RSI, ATR, and standard deviation were calculated to analyze price trends and market volatility.
4. **Risk Analysis**: Value at Risk (VaR) and the Sharpe Ratio were calculated to assess the investment's risk and returns.
5. **Market Sentiment Analysis**: Volume spikes were identified, and their relationship with price movements was analyzed to gauge market sentiment.
6. **Backtesting**: A backtesting strategy was employed using RSI, Moving Average Crossovers, and Volume Spikes to generate buy and sell signals, simulating trading activity and evaluating performance.

## Conclusion:

The analysis of Pakistan Stock Exchange data from 2008 to 2020 highlights periods of both stability and volatility, uncovering key trends in stock prices and trading volumes. The backtested trading strategy showed strong profitability, with the portfolio value more than tripling. However, the negative Sharpe Ratio indicates that the returns did not fully justify the risk involved, suggesting the need for caution. Overall, this report offers a clear view of market behavior and evaluates how well technical indicators can guide stock trading decisions.

# Recommendation:

Based on the findings of the project, here are some recommendations:

1. **Be Careful in Falling Markets:** Since the stock price has mostly gone down during the analyzed period, investors should be cautious when trading in a declining market. Using short-term moving average crossovers can help identify good buying opportunities.
2. **Trade within a Range:** Since the RSI mostly stayed between 30 and 70, investors can look for short-term opportunities. They could buy when the RSI is near 30 (oversold) and sell when it's close to 70 (overbought).
3. **Watch for Volume Spikes:** Big increases in trading volume can show times of high market activity, which might mean opportunities or risks. Keeping an eye on volume along with price changes can help investors better understand market trends and feelings.
4. **Think About Risk-Adjusted Returns:** Since the Sharpe Ratio is negative, investors should evaluate whether this stock is worth the risk. They might want to explore other investments or use strategies to reduce risk.
5. **Use ATR and VaR for Managing Risk:** Tools like ATR and VaR can help investors understand market ups and downs better. These tools can guide them in making safer decisions and protecting their money during unpredictable times.

## References:

Dataset: https://www.kaggle.com/datasets/zusmani/pakistan-stock-exchange-kse-100

# Index:


- **RSI (Relative Strength Index)**: A momentum indicator that measures the speed and change of price movements to identify overbought or oversold conditions.
- **Rolling Standard Deviation**: A measure of volatility that calculates how much the price deviates from its average over a specific moving time window.
- **Moving Average**: A trend-following indicator that smooths price data over a specified period to identify the direction of the trend.
- **Short SMA (Short-Term Simple Moving Average)**: The average of prices over a shorter time period, used to identify short-term trends.
- **Long SMA (Long-Term Simple Moving Average)**: The average of prices over a longer time period, used to identify long-term trends.
- **ATR (Average True Range)**: A measure of market volatility that calculates the average range between high and low prices over a specific time period.
- **VaR (Value at Risk):** A risk management tool that estimates the potential loss in an investment over a specified time period, given normal market conditions.
- **Sharpe Ratio**: A measure of risk-adjusted return, showing how much excess return is earned per unit of risk taken.
- **Volume Spikes**: Sudden increases in trading volume that indicate heightened market interest or activity.
- **Correlation**: A statistical measure that indicates the strength and direction of the relationship between two variables.
- **Distribution**: A statistical representation of the frequency of different values in a dataset, often visualized with a histogram.
- **Backtesting**: A method used to evaluate the performance of a trading strategy by applying it to historical data to see how it would have performed in the past.
- **Market Sentiment Analysis**: The process of assessing the overall mood or attitude of investors toward the market, often inferred from trading volume, price trends, and external factors.