FINAL PROJECT REPORT – STOCK MARKET ANALYSIS

Group 5

Jinal Darji   Siddharth Savla   Vaibhav Rai   Pranav Dalvi

FINAL PROJECT REPORT – STOCK MARKET ANALYSIS

FINAL PROJECT REPORT – STOCK MARKET ANALYSIS

Our project titled **Stock Market Analysis,** designed for a comprehensive exploration of stock market data. The project focuses on analyzing historical stock price movements, identifying trends, and gaining insights into the statistical properties of selected stocks. It leverages Python's robust data analysis libraries to retrieve, clean, visualize, and analyze time-series data from major technology companies such as Apple, Tesla, Google, and Meta. The analysis includes steps like data acquisition, preprocessing, visualization, and basic statistical computations, offering a hands-on approach to understanding stock market dynamics.

The data for this stock market analysis project is sourced from **Yahoo Finance**, a well-known platform that provides comprehensive financial market information. The **yfinance** Python library is utilized to programmatically download historical stock price data directly into the notebook. This enables the analysis of time-series data for major technology companies such as Apple, Tesla, Google, and Meta.

The process involves:

Data Acquisition:

Fetching historical stock data for selected companies using the yfinance library.
Setting a specific time range for the analysis to ensure relevance.

Data Integration:

1. Merging individual datasets for the selected companies into a unified DataFrame using **pd.concat()** to facilitate comparative and exploratory analysis.
2. This approach ensures that the data is both up-to-date and accessible for detailed analysis, including price trends, rolling averages, and statistical summaries.

Data Preprocessing

Data preprocessing is a critical step in this project to ensure the accuracy and reliability of the stock market analysis. The raw data retrieved from Yahoo Finance through the yfinance library required several modifications and checks before it could be used for analysis. Below are the key preprocessing steps undertaken and their importance:

Data Cleaning

In the data cleaning process, we ensured that the dataset was free from inconsistencies and errors. As part of this, we specifically checked for NaN (missing) values in the dataset and found that there were none, eliminating the need for imputation or deletion. This ensured that the data was complete and ready for analysis without any disruptions in statistical calculations or trend analysis. Additionally, we verified that all columns, especially those representing dates and prices, were in the correct data types to allow seamless processing and analysis throughout the project.

Data Inspection

Structural Overview: The **.info()** and **.describe()** methods were applied to inspect the structure and content of the dataset. This helped identify missing values, understand data distributions, and highlight potential anomalies.

Understanding Variability: Descriptive statistics provided insights into central tendencies, variability, and overall data distributions, setting a foundation for deeper analysis.

Time Series Preparation

Chronological Indexing: The stock price data was re-indexed based on the date to ensure proper sequencing, which is essential for accurate time-series analysis. This step also verified the continuity of time periods in the dataset.

Duplicate Handling

Removing Duplicates: Duplicate rows, if present, were identified and eliminated to avoid redundancy and ensure the integrity of the analysis.

Data Normalization and Transformation

Adjustments were made to historical stock prices to account for stock splits and dividend payouts. This ensured consistency and comparability of stock price data across different time periods.

Dataset Consolidation

Data for the selected companies—Apple, Tesla, Google, and Meta—was merged into a single DataFrame. This unified structure allowed for comparative analysis and exploration of trends across multiple stocks.

These preprocessing steps laid the foundation for robust data analysis, ensuring that the dataset was clean, well-structured, and free of inconsistencies. By addressing potential issues early in the workflow, we were able to perform reliable and meaningful analyses that directly contributed to the project objectives.

Methods of Analysis:

The stock market analysis project employs a variety of methods to transform raw data into meaningful insights. These methods include data manipulation, statistical computations, and visualizations, each aimed at answering key questions about stock behavior, trends, and relationships. This section provides a detailed description of the methods used, including the questions addressed, the fields utilized from the dataset, the key functions implemented in the code, and the results produced.

Key Questions Addressed:

What is the change in stock prices over time?

This question focuses on understanding how stock prices evolved over the analysis period. The objective was to detect trends, significant price movements, and periods of volatility or stability.

Approach:

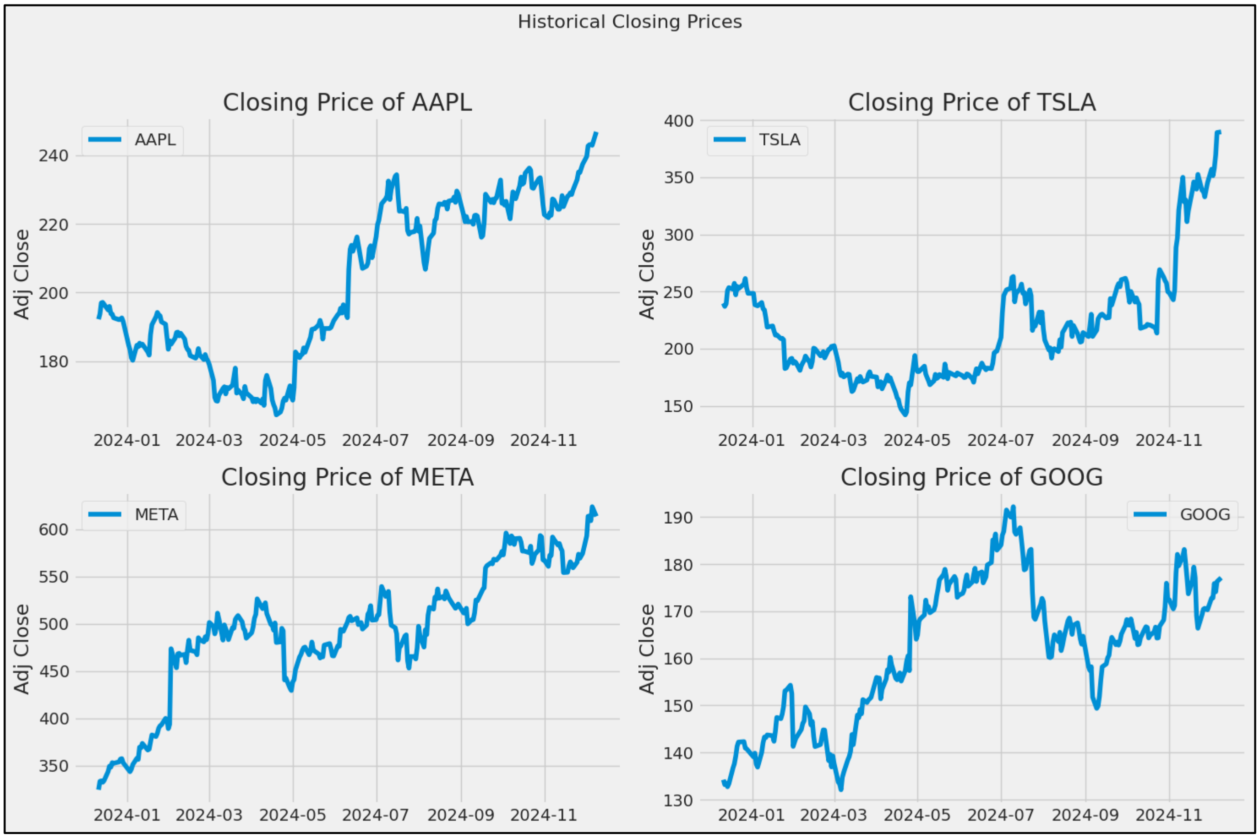
1. Extracted the adjusted closing prices (Adj Close) for time-series analysis.
2. Plotted line graphs of stock prices over time for each company to visualize trends and fluctuations.
3. Explored periods of price spikes or drops to identify market events or anomalies.

Code Implementation:

```python
# Creating a grid of subplots for visualizing the historical closing prices
fig, axes = plt.subplots(2, 2, figsize=(15,10)) # Create a 2x2 grid of subplots with a figure size of 15x10 inches.
fig.suptitle('Historical Closing Prices', fontsize=16) # Add a main title to the entire figure with larger font size.

# Loop through each subplot, company data, and ticker symbol
for ax, company, ticker in zip(axes.flat, company_list, tech_list):
    ax.plot(company['Adj Close'], label=ticker)# Plot the adjusted closing prices for the company on the current subplot.
    ax.set_title(f"Closing Price of {ticker}") # Set the title of the subplot using the ticker symbol.
    ax.set_xlabel('')  # Set an empty label for the x-axis (optional customization).
    ax.set_ylabel('Adj Close') # Label the y-axis to indicate the data represents adjusted closing prices.
    ax.legend() # Add a legend to the subplot to show the ticker symbol.

fig.tight_layout(rect=[0, 0, 1, 0.95])  # Leave space for the suptitle
plt.show() # Display the plot.
```

FINAL PROJECT REPORT – STOCK MARKET ANALYSIS

Results:



1. Line charts for each company, displaying how stock prices changed over the selected time period.
2. Insights into performance trends, such as steady growth in certain stocks or high volatility in others (e.g., Tesla).
3. Identification of specific dates where prices peaked or dipped significantly.

What are the rolling averages and daily returns for the stocks?

Rolling averages help smooth out daily price fluctuations, revealing underlying trends. Daily returns show short-term price movements, indicating the stock's volatility.

Approach:

1. Calculated 20-day rolling averages for adjusted closing prices, a common metric for trend identification in financial analysis.

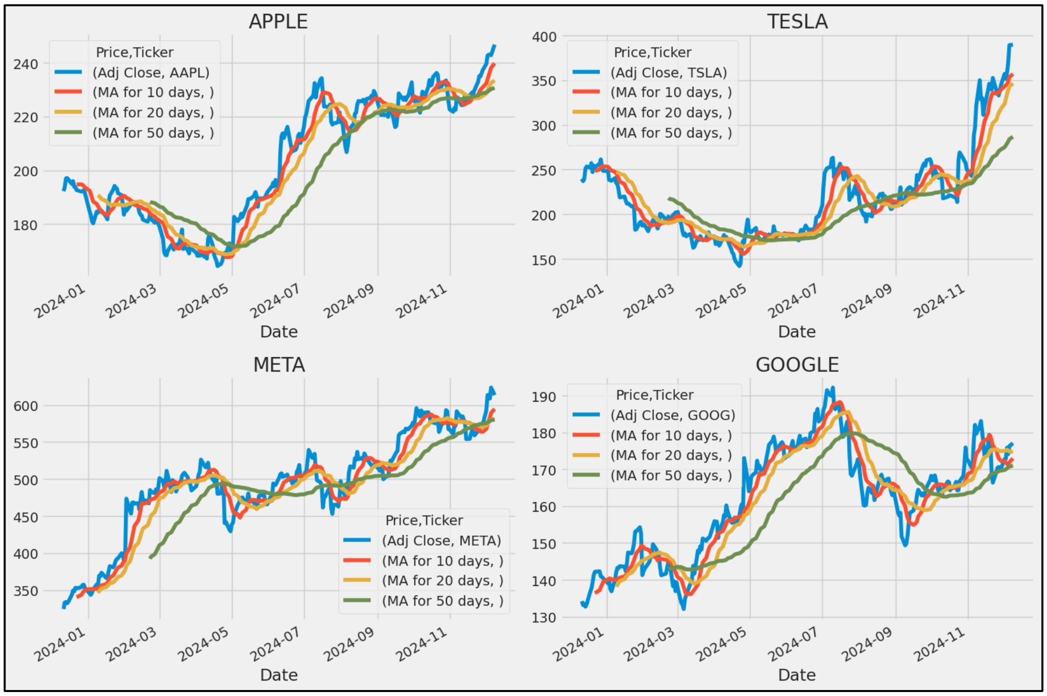
FINAL PROJECT REPORT – STOCK MARKET ANALYSIS

2. Computed daily returns as the percentage change in adjusted closing prices to observe short-term movements.
3. Visualized rolling averages alongside raw price data for comparison.
4. Summarized daily returns using descriptive statistics to understand volatility and average movement.

Code Implementation:

```
# Defining a list of moving average window sizes (10, 20, and 50 days).
ma_day = [10, 20, 50]


# Looping through each moving average window size and company DataFrame.
for ma in ma_day:
    for company in company_list:
      # Creating a new column name based on the moving average window size.
        column_name = f"MA for {ma} days"
        # Calculating the moving average of the adjusted closing price for the specified window size.
        company[column_name] = company['Adj Close'].rolling(ma).mean()

# Creating a 2x2 grid of subplots for visualizing the adjusted closing price and moving averages.

fig, axes = plt.subplots(nrows=2, ncols=2)
fig.set_figheight(10)
fig.set_figwidth(15)

# Plotting adjusted closing prices and moving averages for Apple.
AAPL[['Adj Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']].plot(ax=axes[0,0])
axes[0,0].set_title('APPLE')

# Plotting adjusted closing prices and moving averages for Tesla.
TSLA[['Adj Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']].plot(ax=axes[0,1])
axes[0,1].set_title('TESLA')

# Plotting adjusted closing prices and moving averages for Meta.
META[['Adj Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']].plot(ax=axes[1,0])
axes[1,0].set_title('META')

# Plotting adjusted closing prices and moving averages for Google.
GOOG[['Adj Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']].plot(ax=axes[1,1])
axes[1,1].set_title('GOOGLE')
```

Results:



1. Rolling averages revealed periods of consistent upward or downward trends, smoothing out daily noise.
2. Summary statistics of daily returns showed the average return, standard deviation (volatility), and distribution.
3. Volatile stocks (e.g., Tesla) displayed a wider range of daily returns compared to more stable ones (e.g., Apple).

What was the daily return of the stock on average?

Daily returns provide insights into the stock's performance over short time periods, helping to understand the average rate of return and the variability (volatility) of stock prices.

Approach:

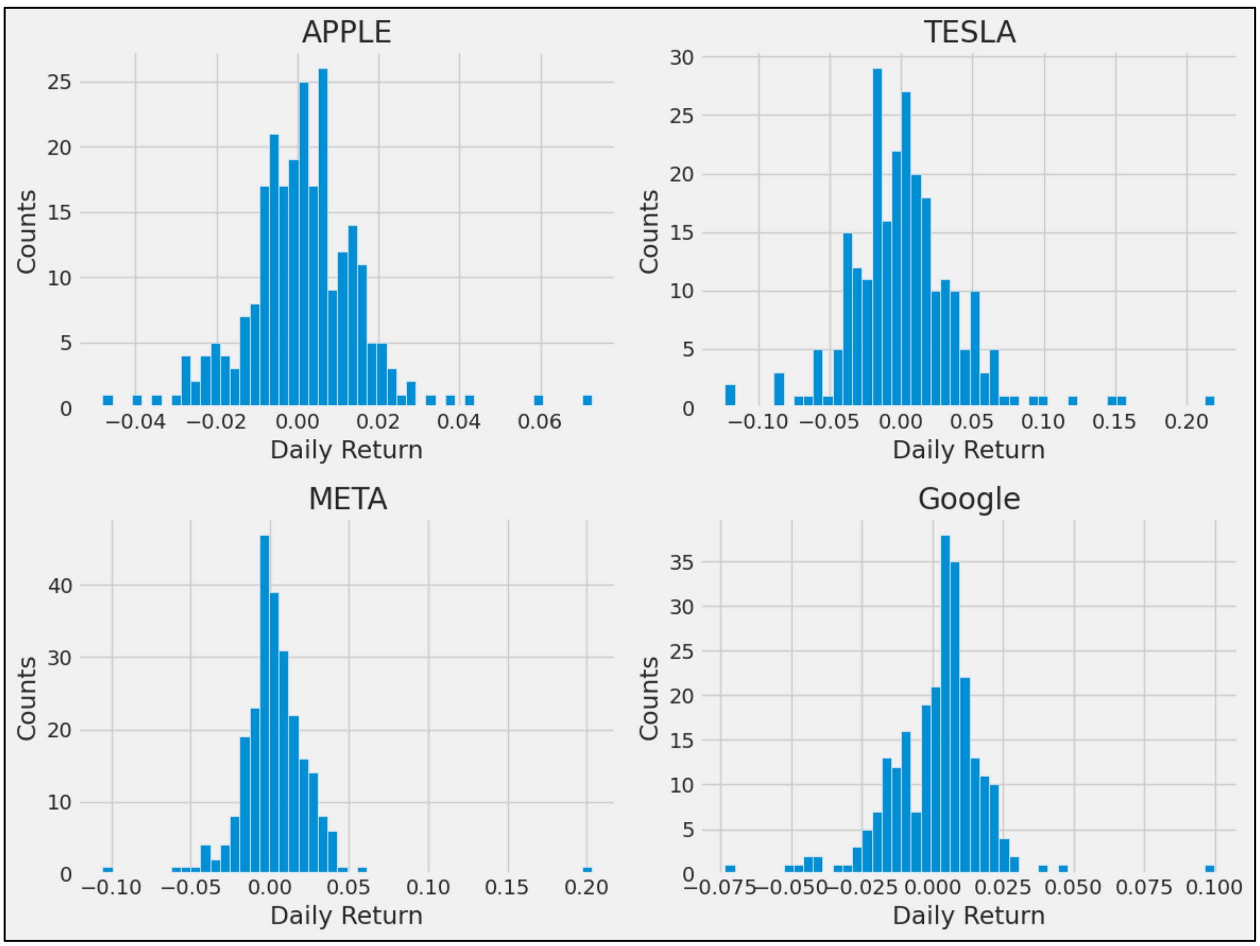
1. Computed daily returns as the percentage change in adjusted closing prices.
2. Analyzed the average daily return across the dataset using descriptive statistics.
3. Visualized daily returns through histograms to understand their distribution and identify patterns or outliers.

FINAL PROJECT REPORT – STOCK MARKET ANALYSIS

Code Implementation:

```
# Plotting histograms of daily return distributions for multiple companies in a grid layout.

plt.figure(figsize=(12, 9)) # Create a figure with dimensions 12x9 inches to accommodate the subplots.

for i, company in enumerate(company_list, 1): # Enumerate over the company list, starting index at 1.
    plt.subplot(2, 2, i) # Create a subplot grid of 2 rows and 2 columns, placing the current plot in the ith position.
    company['Daily Return'].hist(bins=50) # Plot a histogram of the 'Daily Return' column with 50 bins for the current company.
    plt.xlabel('Daily Return') # Label the x-axis to indicate it represents daily return values.
    plt.ylabel('Counts')
    plt.title(f'{company_name[i - 1]}') # Set the title for the current subplot using the company name.

# Adjust the layout to prevent overlapping of subplots
plt.tight_layout()
```

Results:



1. The average daily return for most stocks was close to zero, reflecting typical stock market behavior.
2. Tesla exhibited higher volatility, with wider distributions of daily returns, while Apple showed narrower distributions.
3. The standard deviation of daily returns highlighted the variability, with Tesla being the most volatile and Apple the least.

What was the correlation between different stocks' closing prices?

Correlation analysis reveals relationships between the movements of different stocks, indicating how closely their prices are tied to each other. This helps in understanding diversification benefits and identifying stocks influenced by similar factors.

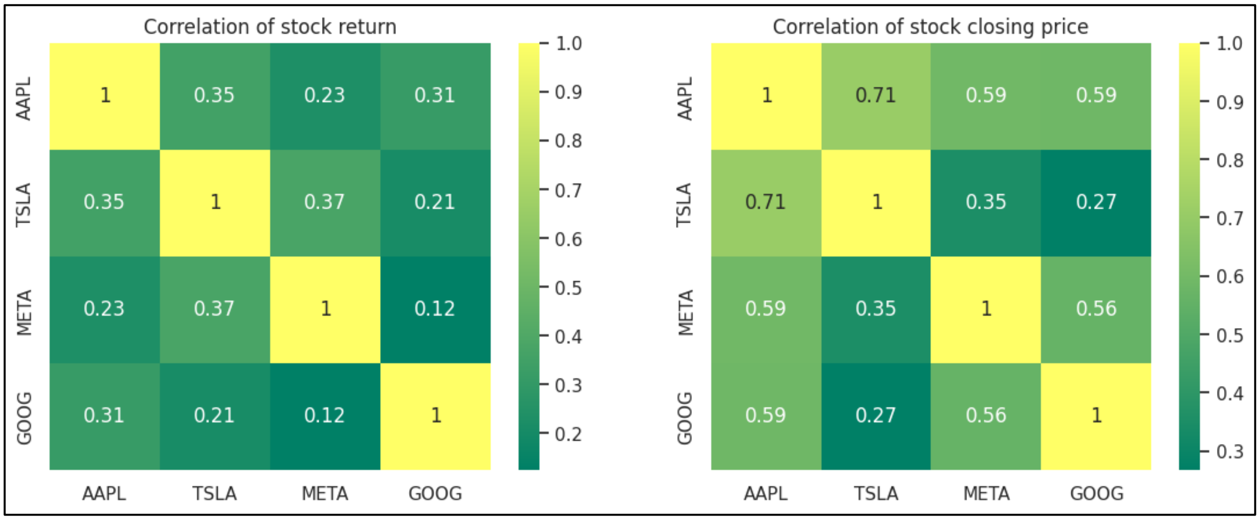
Approach:

1. Created a correlation matrix to quantify the relationships between the adjusted closing prices of the selected stocks.
2. Visualized the correlation matrix using a heatmap to highlight strong and weak relationships.
3. Analyzed patterns to determine sector-wide trends or unique stock behaviors.

Code Implementation:

```
# Create a figure to visualize the correlation heatmaps for stock returns and closing prices.
plt.figure(figsize=(12, 10))  # Set the figure size to 12x10 inches.

# Plot the heatmap for the correlation of stock returns.
plt.subplot(2, 2, 1)
sns.heatmap(tech_rets.corr(), annot=True, cmap='summer')# Plot the correlation matrix with annotations and a summer colormap.
plt.title('Correlation of stock return')

# Plot the heatmap for the correlation of stock closing prices.
plt.subplot(2, 2, 2)
sns.heatmap(closing_df.corr(), annot=True, cmap='summer')
plt.title('Correlation of stock closing price')
```

Results:



1. Strong correlations were observed between Google and Meta, suggesting they are influenced by similar market trends.
2. Tesla showed weaker correlations with other stocks, reflecting its unique market dynamics.
3. The heatmap provided a clear, visual representation of these relationships, aiding in portfolio diversification strategies.

How much value do we put at risk by investing in a particular stock?

Value at Risk (VaR) estimates the potential loss in value of a portfolio or investment over a defined time period at a given confidence level, helping to assess the risk involved in investing in specific stocks.

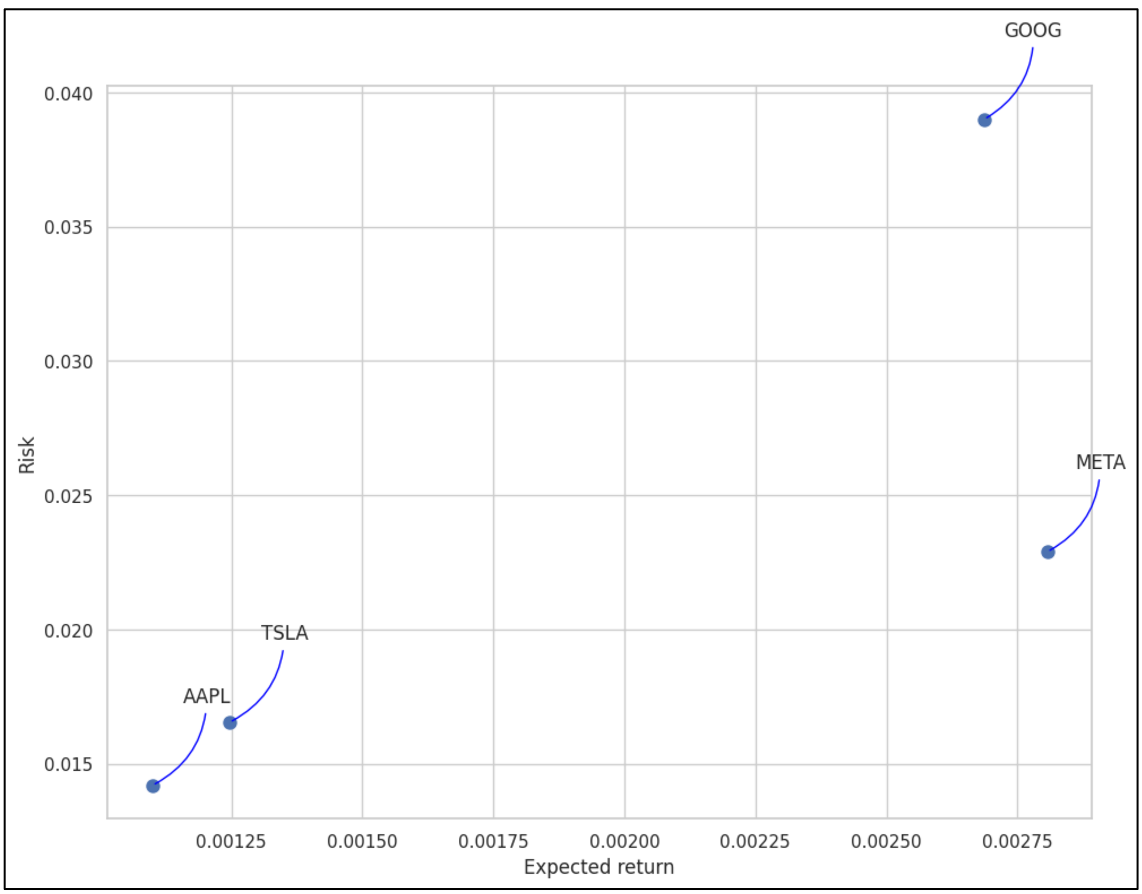
Approach:

1. Calculated daily returns for each stock and their standard deviations.
2. Applied the VaR formula for a 95% confidence level, considering normal distribution assumptions.
3. Summarized the maximum expected loss over a single day within the specified confidence level.

Code Implementation:

```
# Remove any missing values (NaN) from the returns DataFrame to prepare for analysis.

rets = tech_rets.dropna()

# Define the size of the scatter plot points.
area = np.pi * 20  # Set the area of each scatter plot point.

# Create a scatter plot to visualize the relationship between expected return and risk.
plt.figure(figsize=(10, 8))
plt.scatter(rets.mean(), rets.std(), s=area)
plt.xlabel('Expected return')
plt.ylabel('Risk')

# Annotate each point in the scatter plot with the corresponding stock label.
for label, x, y in zip(rets.columns, rets.mean(), rets.std()):
    plt.annotate(label, xy=(x, y), xytext=(50, 50), textcoords='offset points', ha='right', va='bottom',
                 arrowprops=dict(arrowstyle='-', color='blue', connectionstyle='arc3,rad=-0.3'))
```

Results:



1. Tesla had the highest Value at Risk, reflecting its high volatility, meaning investors face a larger potential single-day loss.
2. Apple and Google exhibited lower VaR values, making them relatively safer investments.
3. VaR provided investors with a quantitative measure of risk, crucial for risk management and portfolio construction.

Fields Used in the Analysis

The following fields from the dataset were instrumental in performing the analysis:

1. **Date**: The index for time-series data, ensuring chronological alignment for analysis.
2. **Adjusted Close (Adj Close)**: The key metric for price trends, adjusted for splits and dividends to ensure consistency.
3. **Open, High, Low, Close Prices**: Used to calculate descriptive statistics and understand daily price ranges.
4. **Volume**: Used to assess trading activity and its correlation with price movements.
5. **Daily Return**: Derived from adjusted closing prices to analyze short-term changes.
6. **Rolling Average 20**: Computed for trend analysis over a 20-day window.
7. **Company Name**: Distinguishes stocks for individual and comparative analysis.

Overall Description of Python Program:

The Python program is a robust framework designed for analyzing stock market data, focusing on historical trends, volatility, and inter-stock relationships. It uses the **yfinance** library to fetch historical stock data for major technology companies, including Apple, Tesla, Google, and Meta, and retrieves essential metrics such as `Date`, `Open`, `High`, `Low`, `Close`, `Adjusted Close`, and `Volume`. The program begins by cleaning and structuring the data through preprocessing steps like handling missing values, converting dates into the proper format, and merging datasets from different companies into a single DataFrame for unified analysis. Using Python's powerful data analysis libraries, such as **pandas**, **NumPy**, and visualization tools like **matplotlib** and **seaborn**, the program performs a series of analyses: time-series analysis of adjusted closing prices to identify trends, computation of 20-day rolling averages to smooth fluctuations, and calculation of daily percentage returns to assess short-term volatility. Additionally, it uses scatter plots, pair grids, and correlation matrices to analyze relationships between the daily returns of different stocks, highlighting interdependencies and correlations. The outputs include line charts showing trends over time, statistical summaries of daily returns, and heatmaps of correlations, providing insights into stock behavior and comparative performance. Overall, this Python program serves as a comprehensive tool for analyzing stock market data, combining statistical rigor with intuitive visualizations to deliver actionable insights for financial analysis and decision-making.

Output of the Python Program:

A. Descriptive Statistics about the data:

```
# Generating summary statistics for the TSLA (Tesla) stock DataFrame.
# This includes measures like count, mean, standard deviation, min, max, and quartiles for numerical columns.

TSLA.describe()
```

Price	Adj Close	Close	High	Low	Open	Volume
Ticker	TSLA	TSLA	TSLA	TSLA	TSLA	TSLA
count	251.000000	251.000000	251.000000	251.000000	251.000000	2.510000e+02
mean	219.515936	219.515936	223.997370	214.924581	219.463387	9.576062e+07
std	49.333650	49.333650	50.543751	47.772706	49.229288	3.342787e+07
min	142.050003	142.050003	144.440002	138.800003	140.559998	3.716760e+07
25%	180.000000	180.000000	184.419998	176.959999	181.900002	7.232360e+07
50%	210.660004	210.660004	215.880005	207.559998	211.880005	8.849100e+07
75%	246.385002	246.385002	250.480003	240.629997	244.619995	1.106870e+08
max	389.790009	389.790009	404.799988	378.010010	397.609985	2.438697e+08

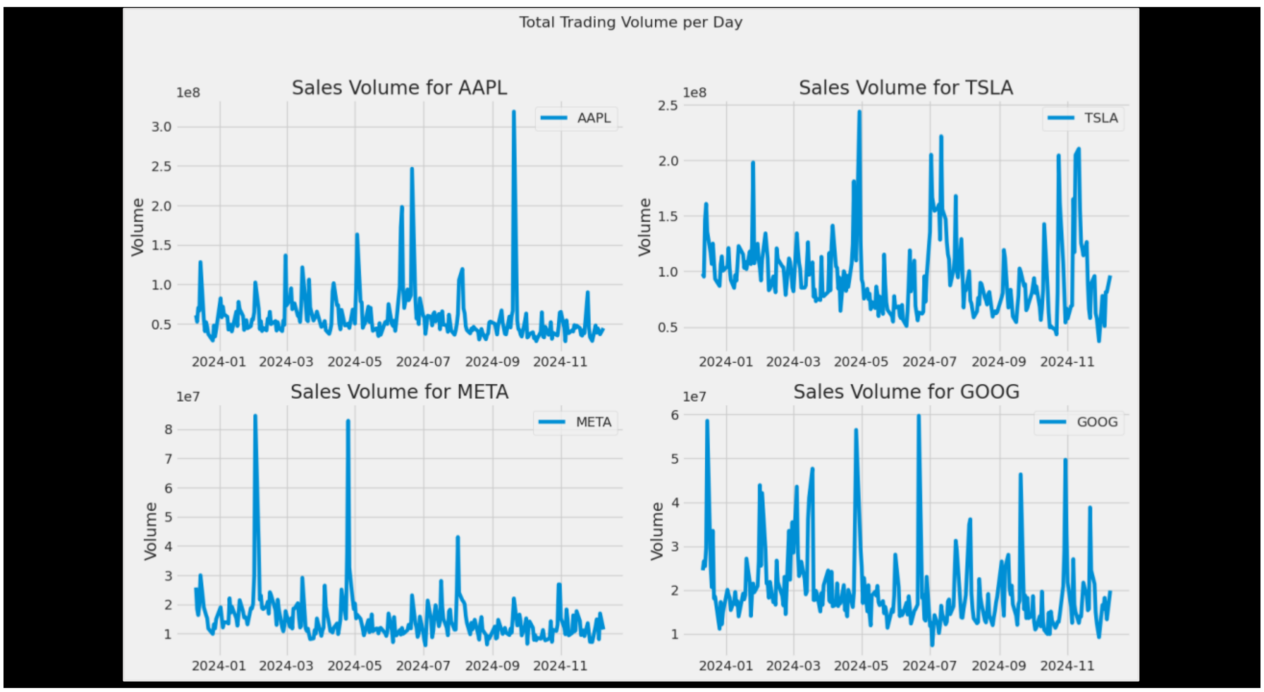
Output:

1. ***Count:*** 251 trading days of data were analyzed.
2. ***Mean:*** Average adjusted close price was $219.52, and average volume was ~9.576 million shares.
3. ***Standard Deviation:*** Price variability was $49.33, with trading volume variability of ~33.42 million shares.
4. ***Minimum Values:*** The lowest adjusted close price was $142.05, with a minimum volume of ~3.716 million shares.
5. ***Maximum Values:*** The highest adjusted close price was $389.79, with a maximum volume of ~243.87 million shares.
6. ***Quartiles:*** The 25th percentile adjusted close price was $180, and the 75th percentile was $246.39.

Insights:

1. ***Stock Price Behavior:*** Tesla's adjusted closing price averaged $219.52, with a range of $142.05 to $389.79, reflecting significant variability.
2. ***Trading Volume:*** Average trading volume was ~9.576 million shares, with a maximum of ~243.87 million shares on a single day.
3. ***Quartile Analysis:*** 50% of Tesla's adjusted closing prices were between $180 (Q1) and $246.39 (Q3).
4. ***High Volatility***: The wide range between the minimum and maximum prices underscores Tesla's dynamic and high-risk nature.

B. Volume of Sales Analysis:



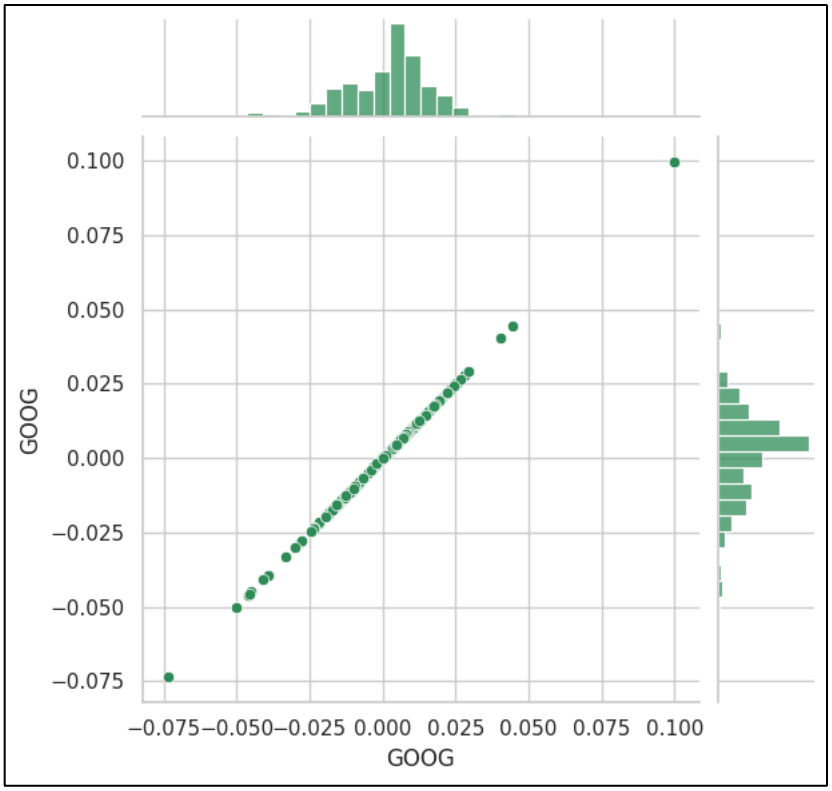
Outputs:

1. The code generates a grid of subplots displaying daily trading volumes for Apple, Tesla, Meta, and Google over the past year.
2. Each subplot shows a time-series plot of trading volume with labeled axes and legends for each stock.

Insights:

1. High trading volumes often coincide with significant price movements, indicating strong investor interest or market reactions.
2. Spikes in trading volume are linked to events like earnings releases or market news, reflecting bullish or bearish sentiment.
3. Comparing the stocks, Tesla showed the most frequent volume spikes, aligning with its high volatility and market activity.
4. Apple and Google exhibited steadier trading volumes, indicative of their consistent market liquidity and investor base.

C. Google's daily return against itself:



Output:

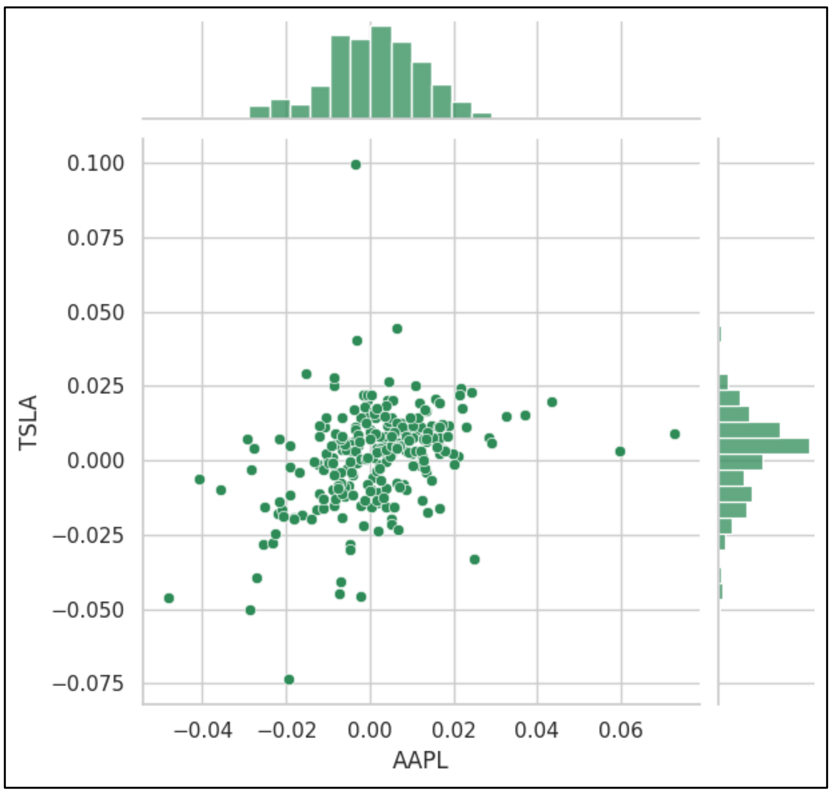
1. The plot is a pair grid or scatter plot matrix representing the daily returns of Google (GOOG) stock.
2. The main scatter plot shows a strong diagonal pattern, suggesting a linear correlation in daily returns within Google's data.

3. Marginal histograms on the axes show the distribution of daily returns for Google, revealing the concentration and spread.

Insights:

1. Correlation: The diagonal alignment of points indicates a strong positive relationship in Google's returns, consistent with expectations in daily returns data.
2. Distribution: The histograms suggest that most daily returns for Google are centered around zero, indicating stability with some degree of variability.
3. Outliers: Points farther from the cluster in the scatter plot could represent anomalous days with unusually high or low returns, potentially linked to specific market events.

D. Apple returns VS Tesla returns:



Output:

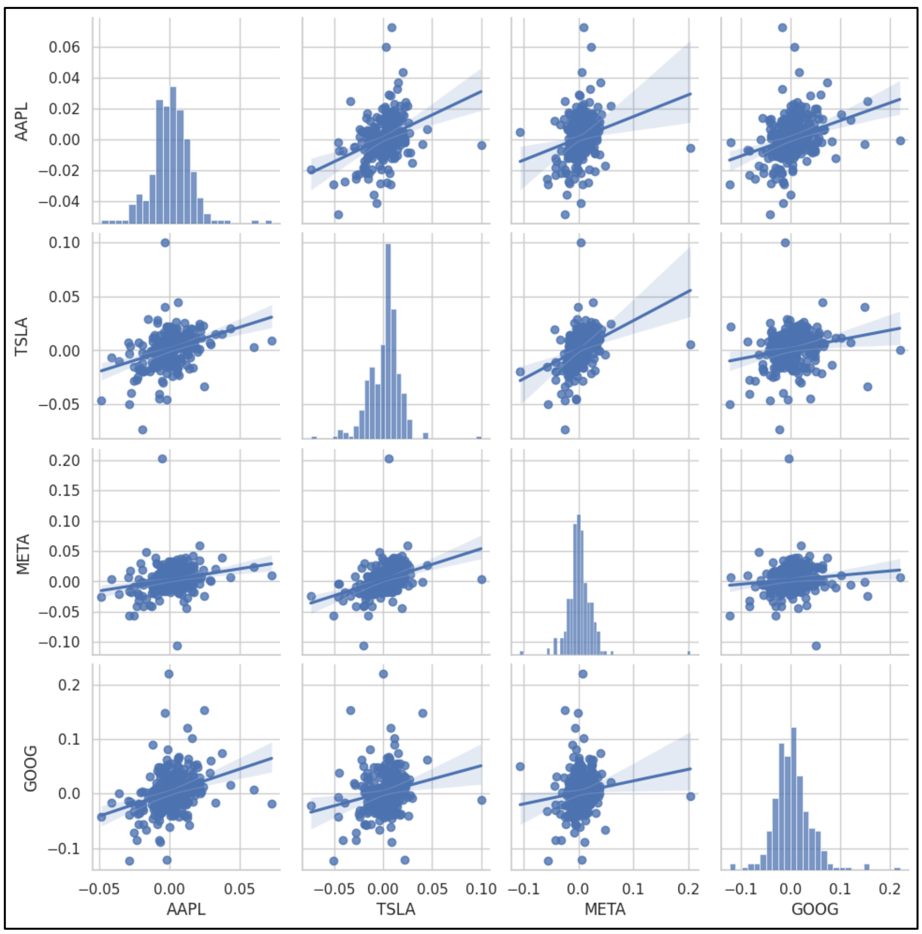
1. The pair plot visualizes the relationship between the daily returns of Apple (AAPL) and Tesla (TSLA) stocks.
2. The scatter plot at the center shows the distribution and correlation of daily returns between these two stocks.

3. Marginal histograms on the axes show the individual distributions of daily returns for Apple and Tesla.

Insights:

1. ***Correlation**:* The scattered points indicate a weak positive relationship between Apple and Tesla daily returns, suggesting limited co-movement.
2. ***Distribution**:* Tesla's returns (vertical axis) exhibit a wider spread, reflecting its higher volatility compared to Apple's narrower distribution.
3. ***Cluster Analysis**:* Most of the data points cluster near the center (around zero), highlighting that the majority of daily returns for both stocks are relatively small.
4. ***Outliers**:* A few points away from the cluster represent days with significant positive or negative returns, likely caused by specific market events.

E. Pairwise relationship between the daily returns of Apple, Tesla, Meta and Google:



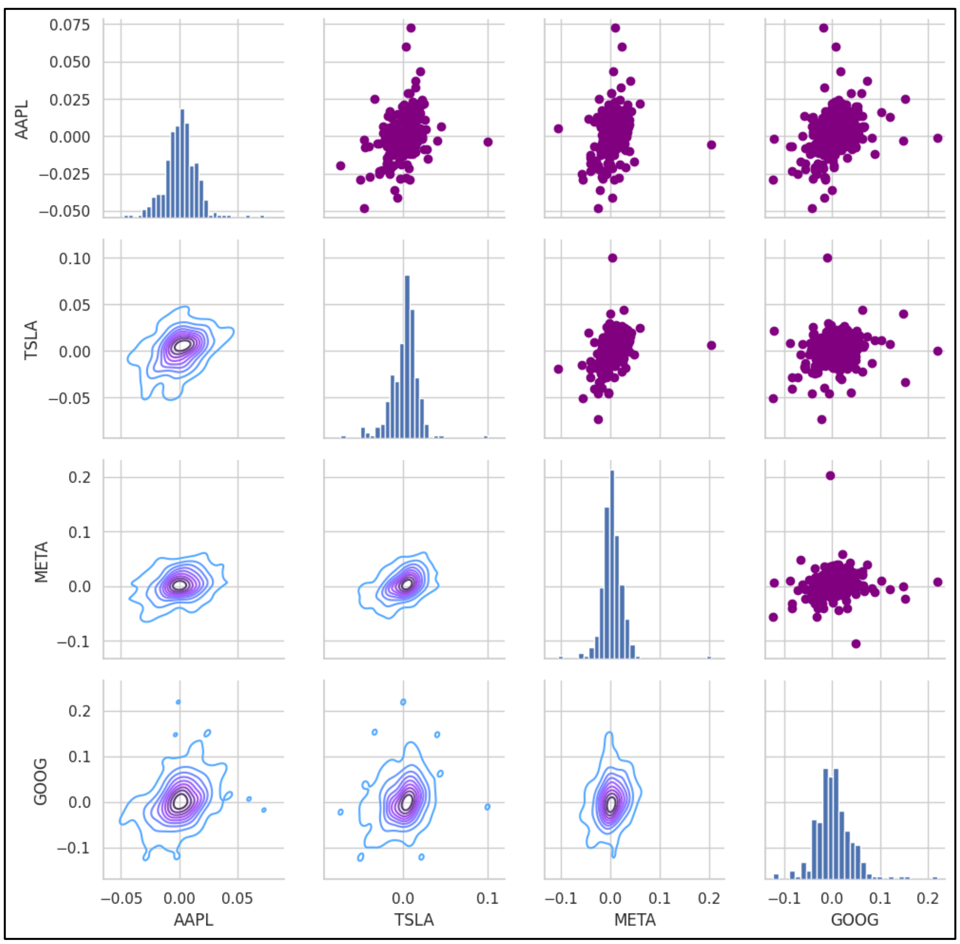
Output:

1. The scatter plots visualize correlations between the daily returns of Apple, Tesla, Meta, and Google.
2. Histograms on the diagonal show the distribution of daily returns for each stock.

Insights:

1. Google and Meta show strong positive correlations, moving similarly in the market.
2. Tesla has weaker correlations with other stocks, reflecting its unique market behavior.
3. Tesla's daily returns are the most volatile, as shown by the wider histogram spread.
4. Apple and Google have more stable daily returns, indicated by narrower distributions.

F. Pairwise relationship between tech_rets Dataframe:



Output:

1. The pair plot shows scatter plots (top triangle) and KDE (Kernel Density Estimate) plots (bottom triangle) for daily returns of Apple, Tesla, Meta, and Google.
2. Histograms on the diagonal illustrate the individual distributions of daily returns for each stock.

Insights:

1. Google and Meta exhibit strong positive correlations, evident from their aligned scatter plots and overlapping KDE contours.
2. Tesla's scatter plots show weaker correlations with other stocks, indicating its distinct market behavior.
3. Tesla's daily returns have the broadest spread, reflected in wider histograms and KDE plots, highlighting its volatility.
4. Apple and Google show narrower distributions, implying more stable and consistent daily returns.

Conclusions from the Stock Market Analysis:

The analysis of historical stock market data for Apple, Tesla, Meta, and Google revealed key insights into their performance and market behavior. Tesla exhibited the highest volatility with dramatic price fluctuations, reflecting its sensitivity to market events, while Apple and Google showed more stable growth, indicative of a mature market presence. Daily returns analysis highlighted Tesla as the most volatile, appealing to risk-tolerant investors, whereas Apple and Google displayed lower risks, making them suitable for conservative strategies. Rolling averages helped smooth fluctuations, uncovering sustained upward or downward trends critical for understanding market cycles. Comparative performance analysis showed strong correlations between Google and Meta, driven by similar market influences, while Tesla displayed weaker correlations due to its unique market dynamics. Overall, the technology sector demonstrated growth over the analysis period, though it remains vulnerable to broader economic conditions such as inflation, interest rate changes, and global events affecting tech demand.

Implications:

1. For Investors: This analysis highlights the importance of understanding a stock's volatility and market trends before making investment decisions. While Tesla offers high potential rewards with greater risks, Apple and Google may provide steadier returns for long-term investors.
2. For Analyst: The strong correlations within the tech sector suggest opportunities for diversification within related industries or sectors to mitigate risk.
3. For Future Analysis: Further exploration could incorporate macroeconomic indicators, earnings reports, or news sentiment analysis to enhance predictions and uncover deeper insights into stock price movements.

In conclusion, this analysis provided a detailed understanding of stock trends, volatility, and inter-company relationships, underscoring the dynamic nature of the tech stock market and its implications for various stakeholders.