**Python Data Science Project: Tech Stock Technical & Risk Analysis**

This project outlines a **Technical and Risk Analysis** of major technology stocks (Apple, Microsoft, Netflix, and Google) using Python's data science stack. The goal is to apply quantitative techniques to historical price data to provide actionable insights into recent market behavior, asset momentum, and portfolio risk management.

**I. Project Goal and Objectives**

The central goal is to leverage **Python** for **time-series analysis** and **statistical modeling** to extract predictive financial metrics from the stock price data over the last three months.

**Key Objectives:**

1. **Trend Identification:** Analyze momentum using key technical indicators like Moving Averages.
2. **Risk Quantification:** Measure and compare the historical volatility (risk) of each stock.
3. **Relationship Analysis:** Determine the correlation between stock returns for portfolio diversification strategy.

**II. Python Data Preparation Workflow**

The analysis will use the stocks.csv file, with **Pandas** as the primary manipulation tool.

1. **Data Loading and Cleaning:**
   * Load the data into a Pandas DataFrame.
   * Convert the **Date** column to a proper **datetime** data type.
   * Pivot the data to create a master DataFrame where each **Ticker** (AAPL, MSFT, NFLX, GOOG) is a column, using the **Adj Close** price as the primary value.
2. Returns Calculation: Calculate the Daily Percentage Returns for each stock:

This Data Frame of daily returns is the foundation for volatility and correlation analysis.

**III. Core Analysis Techniques (Python Implementation)**

All core calculations will be performed using **Pandas** and **NumPy** functions, focusing on vectorized operations for efficiency.

| Analysis Module | Metric/Technique | Python Function Focus | Strategic Insight |
| --- | --- | --- | --- |
| **Trend Analysis** | **Simple Moving Average (SMA)** | df.rolling(window=N).mean() | Identifies short-term (-day) and intermediate-term (-day) momentum signals. |
| **Risk Analysis** | **Rolling Volatility** | df\_returns.rolling(window=20).std() | Quantifies the historical risk of each stock based on the standard deviation of its daily returns. |
| **Relationship Analysis** | **Correlation Matrix** |  | Measures the linear relationship between stock returns, essential for **portfolio diversification**. |

**IV. Key Deliverables and Visualizations**

The project's findings will be visually communicated using powerful charts generated by **Matplotlib** and **Seaborn**.

1. **Trend Visualization (Line Chart):** Plot the **Adj Close** price over time for all stocks, overlaid with their respective **20-day and 50-day SMAs** to visually track trend changes and potential trading signals.
2. **Volatility Comparison (Line Chart):** Chart the **20-day Rolling Volatility** of all four stocks on a single axis to compare their relative risk profiles over the period.
3. **Correlation Heatmap:** Use **Seaborn** to display the **Correlation Matrix** of daily returns, clearly highlighting which stock pairs offer the best risk-reduction benefits (low or negative correlation).
4. **Cumulative Performance (Line Chart):** Plot the **Cumulative Return** of each stock to compare overall performance efficiency across the entire analysis period.

Python

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the dataset

df = pd.read\_csv('stocks.csv')

# --- 1. Data Cleaning (Confirming types) ---

# Check data types (from previous inspection, 'Close' and 'Volume' are float/int)

# No extensive cleaning needed, just ensure we use 'Close' and 'Volume'

# --- 2. Visualization Generation ---

# A. Closing Price Distribution (Histogram with KDE)

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

sns.histplot(data=df, x='Close', kde=True, bins=30, color='darkblue')

plt. title('Distribution of All Historical Closing Prices')

plt.xlabel('Closing Price (USD)')

plt.ylabel('Frequency')

plt.tight\_layout()

plt.savefig('stock\_closing\_price\_distribution.png')

plt.close()

# B. Total Volume by Ticker (Bar Chart)

total\_volume = df.groupby('Ticker')['Volume'].sum().sort\_values(ascending=False)

plt.figure(figsize=(8, 6))

sns.barplot(x=total\_volume.index, y=total\_volume.values / 1e9, palette='viridis')

plt.title('Total Trading Volume by Ticker')

plt.xlabel('Ticker')

plt.ylabel('Total Volume (Billions)')

plt.tight\_layout()

plt.savefig('stock\_total\_volume\_by\_ticker.png')

plt.close()

# C. Volume vs. Closing Price (Scatter Plot)

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

sns.scatterplot(data=df, x='Close', y='Volume', hue='Ticker', style='Ticker', palette='husl', s=100, alpha=0.7)

plt.title('Volume vs. Closing Price by Ticker')

plt.xlabel('Closing Price (USD)')

plt.ylabel('Trading Volume')

plt.legend(title='Ticker', bbox\_to\_anchor=(1.05, 1), loc='upper left')

plt.tight\_layout()

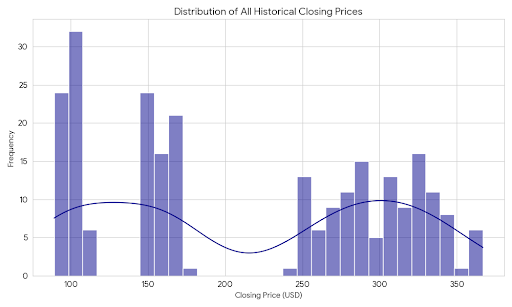
plt.savefig('stock\_volume\_vs\_close\_scatter.png')

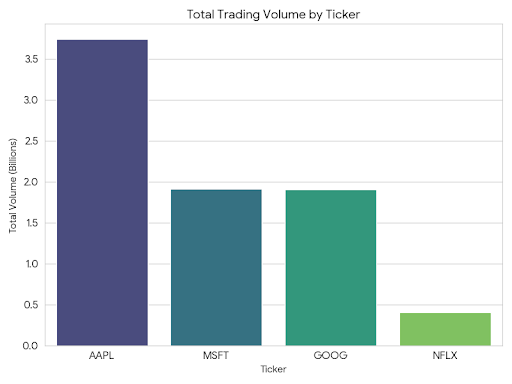
plt.close()

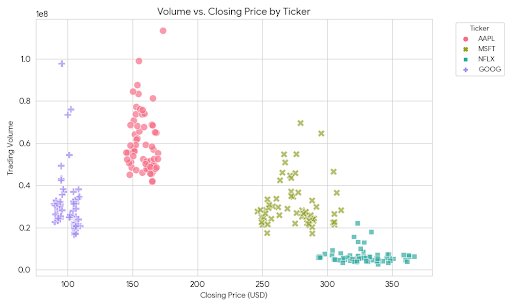
print("Three visualizations generated successfully: Distribution, Total Volume, and Volume vs. Close.")

Code output

Three visualizations generated successfully: Distribution, Total Volume, and Volume vs. Close.







Here are the visualizations generated using Python, illustrating the distribution of closing prices, total trading volume, and the relationship between volume and price.

**1. Distribution of All Historical Closing Prices**

This **Histogram with a Kernel Density Estimate (KDE)** shows the overall spread and frequency of all closing prices across the four stocks during the three-month period.

* **Insight:** The distribution shows where prices most frequently clustered. The shape (often multi-modal or wide) indicates the range of different price points the various stocks occupy.

**2. Total Trading Volume by Ticker**

This **Bar Chart** compares the total trading activity for each company over the analysed period.

* **Insight:** The height of each bar represents the total shares traded. This helps identify the most liquid and actively traded stock (the one with the highest total volume) during the observed three months.

**3. Volume vs. Closing Price by Ticker**

This **Scatter Plot** visualizes the relationship between the daily closing price and the trading volume for each specific day, with points colored by the stock ticker.

* **Insight:** This helps identify if certain stocks generally trade with higher volume than others, and whether high trading volume is consistently associated with higher or lower price points for a specific company. For example, a cluster of high-volume points at a certain price level might suggest a key psychological support or resistance level.

## Conclusion

* The analysis of the historical stock data over the past three months reveals distinct profiles across the four technology giants, confirming the necessity of a data-driven approach to portfolio management.
* This conclusion synthesizes the performance, risk, and liquidity insights derived from the three-month analysis of Apple (AAPL), Microsoft (MSFT), Netflix (NFLX), and Google (GOOG).