NVIDIA Stock Analysis (2015-2024) - EDA Project Guidance

Key Performance Indicators (KPIs) to Analyze

1. **Daily/Weekly/Monthly Returns**: Percentage change in closing prices
2. **Volatility**: Standard deviation of returns
3. **Trading Volume Trends**: Average volume changes over time
4. **Price Movement**: High-Low range as % of closing price
5. **Moving Averages**: 50-day and 200-day moving averages
6. **Relative Strength Index (RSI)**: Momentum indicator
7. **Maximum Drawdown**: Largest peak-to-trough decline
8. **Sharpe Ratio**: Risk-adjusted return

5 Stakeholder-Relevant Questions to Explore

1. **Investors**: "What were the best and worst periods to invest in NVIDIA, and what were the key drivers?"
2. **Traders**: "How does trading volume correlate with price volatility, and are there predictable patterns?"
3. **Company Management**: "How have major product launches or announcements impacted stock performance?"
4. **Market Analysts**: "How does NVIDIA's performance compare to industry benchmarks (e.g., NASDAQ, semiconductor index)?"
5. **Risk Managers**: "What were the maximum drawdown periods and recovery times for NVIDIA stock?"

Python EDA Code

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from datetime import datetime


# Load data (assuming your DataFrame is named 'gpu')
```

```
# gpu = pd.read_csv('nvidia_stock_2015_2024.csv')

# Basic EDA
print(gpu.info())
print(gpu.describe())

# Convert date to datetime and set as index
gpu['date'] = pd.to_datetime(gpu['date'])
gpu.set_index('date', inplace=True)

# 1. Calculate Daily Returns
gpu['daily_return'] = gpu['close'].pct_change() * 100

# 2. Calculate Volatility (rolling 30-day std)
gpu['volatility_30d'] = gpu['daily_return'].rolling(window=30).std()

# 3. Moving Averages
gpu['MA_50'] = gpu['close'].rolling(window=50).mean()
gpu['MA_200'] = gpu['close'].rolling(window=200).mean()

# 4. Relative Strength Index (RSI)
def calculate_rsi(data, window=14):
  delta = data['close'].diff()
  gain = (delta.where(delta > 0, 0)).rolling(window=window).mean()
  loss = (-delta.where(delta < 0, 0)).rolling(window=window).mean()
  rs = gain / loss
  return 100 - (100 / (1 + rs))

gpu['RSI'] = calculate_rsi(gpu)

# 5. High-Low Percentage
gpu['HL_Pct'] = (gpu['high'] - gpu['low']) / gpu['close'] * 100

# 6. Volume analysis
```

```
gpu['volume_ma_20'] = gpu['volume'].rolling(window=20).mean()

# 7. Maximum Drawdown
rolling_max = gpu['close'].rolling(window=252, min_periods=1).max()
daily_drawdown = gpu['close']/rolling_max - 1.0
max_daily_drawdown = daily_drawdown.rolling(window=252, min_periods=1).min()
gpu['max_drawdown'] = max_daily_drawdown

# Plotting some basic visualizations
plt.figure(figsize=(15, 10))

# Price and Moving Averages
plt.subplot(2, 2, 1)
gpu['close'].plot(label='Close Price')
gpu['MA_50'].plot(label='50-day MA')
gpu['MA_200'].plot(label='200-day MA')
plt.title('NVIDIA Stock Price with Moving Averages')
plt.legend()

# Daily Returns
plt.subplot(2, 2, 2)
gpu['daily_return'].plot()
plt.title('Daily Returns (%)')

# Volume
plt.subplot(2, 2, 3)
gpu['volume'].plot()
plt.title('Trading Volume')

# RSI
plt.subplot(2, 2, 4)
gpu['RSI'].plot()
plt.axhline(70, color='r', linestyle='--')
plt.axhline(30, color='g', linestyle='--')
```

```
plt.title('Relative Strength Index (RSI)')


plt.tight_layout()

plt.show()


# Correlation Heatmap

corr = gpu[['close', 'volume', 'daily_return', 'volatility_30d', 'RSI', 'HL_Pct']].corr()

plt.figure(figsize=(10, 8))

sns.heatmap(corr, annot=True, cmap='coolwarm')

plt.title('Correlation Heatmap')

plt.show()
```

Power BI Visualization Recommendations (Based on Andrew Abela's Chart Guide)

1. **Stock Price Trend**:
    - *Chart Type*: Line chart (date vs. close price)
    - *Purpose*: Show overall price movement over time
2. **Moving Averages Comparison**:
    - *Chart Type*: Line chart (date vs. close, MA_50, MA_200)
    - *Purpose*: Identify trends and potential crossover signals
3. **Daily Returns Distribution**:
    - *Chart Type*: Histogram (bins of daily return %)
    - *Purpose*: Understand return distribution and volatility
4. **Volume Analysis**:
    - *Chart Type*: Column chart (date vs. volume) with line for moving average
    - *Purpose*: Identify unusual trading activity
5. **RSI Indicator**:
    - *Chart Type*: Line chart with 30/70 thresholds (date vs. RSI)
    - *Purpose*: Identify overbought/oversold conditions
6. **High-Low Percentage**:

- *Chart Type*: Bar chart (date vs. HL_Pct)
- *Purpose*: Show daily price range as % of close

7. **Correlation Matrix**:
    - *Chart Type*: Matrix/heatmap
    - *Purpose*: Show relationships between different metrics
8. **Maximum Drawdown**:
    - *Chart Type*: Area chart (date vs. max_drawdown)
    - *Purpose*: Visualize risk periods for investors
9. **Monthly Returns Heatmap**:
    - *Chart Type*: Calendar heatmap (months vs. years, color by return)
    - *Purpose*: Identify seasonal patterns
10. **Volume vs. Price Change**:
    - *Chart Type*: Scatter plot (daily_return vs. volume)
    - *Purpose*: Examine relationship between volume and price movement

For Power BI implementation, you can export your processed DataFrame to CSV and connect it to Power BI, then create these visualizations using the appropriate chart types.