```python
import pandas as pd
import matplotlib.pyplot as plt
def load_stock_data(filepath):
    try:
        data = pd.read_csv(filepath, parse_dates=["Date"])
        data = data.sort_values("Date")  # Ensure data is sorted by
        return data
    except Exception as e:
        print(f"Error loading data: {e}")
        return None
def analyze_variability(data):
    closing_prices = data["Close"]
    daily_returns = closing_prices.pct_change().dropna()
    stats = {
        "Mean Closing Price": closing_prices.mean(),
        "Standard Deviation": closing_prices.std(),
        "Variance": closing_prices.var(),
        "Max Price": closing_prices.max(),
        "Min Price": closing_prices.min(),
        "Mean Daily Return (%)": daily_returns.mean() * 100,
        "Volatility (Daily Std Dev %)": daily_returns.std() * 100,
    }
    return stats, daily_returns
```

return stats, daily_returns

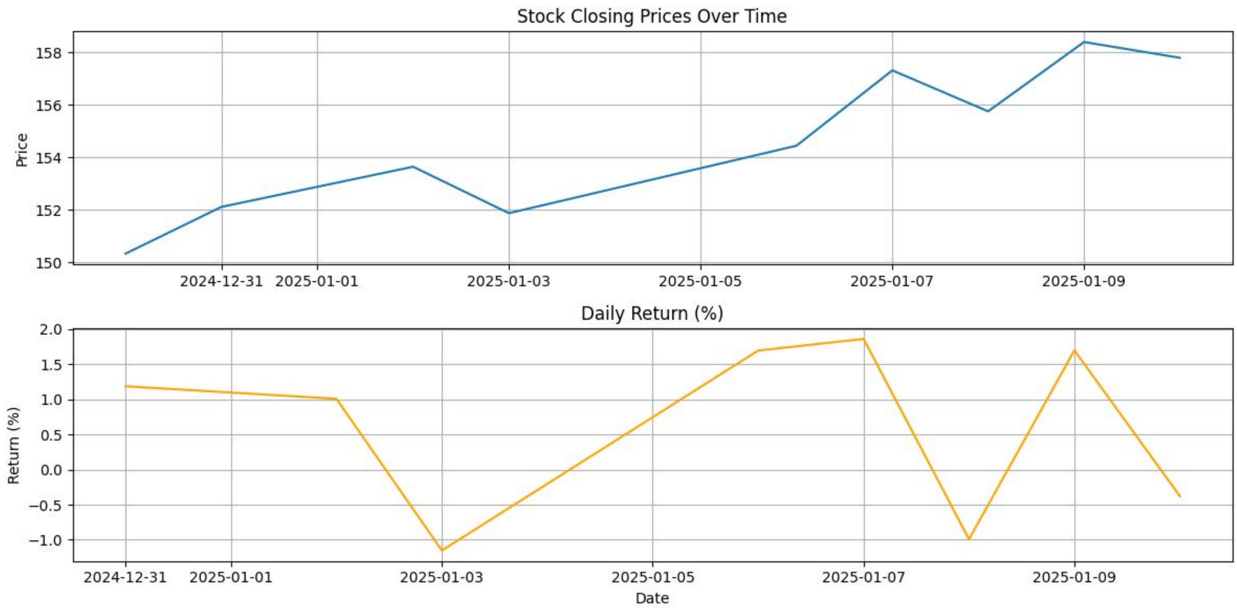
```python
def visualize_data(data, daily_returns):
    plt.figure(figsize=(12, 6))
    plt.subplot(2, 1, 1)
    plt.plot(data["Date"], data["Close"], label="Closing Price")
    plt.title("Stock Closing Prices Over Time")
    plt.ylabel("Price")
    plt.grid()
    plt.subplot(2, 1, 2)
    plt.plot(data["Date"].iloc[1:], daily_returns * 100, label="Daily Returns", color="orange")
    plt.title("Daily Return (%)")
    plt.ylabel("Return (%)")
    plt.xlabel("Date")
    plt.grid()
    plt.tight_layout()
    plt.show()
if __name__ == "__main__":
    file_path = r'C:\Users\91637\OneDrive\Desktop\sev\stock.csv'
    stock_data = load_stock_data(file_path)
    if stock_data is not None:
        stats, daily_returns = analyze_variability(stock_data)
        print("Stock Variability Analysis:")
        for key, value in stats.items():
            print(f"{key}: {value:.2f}")
        visualize_data(stock_data, daily_returns)
```

OUTPUT

Stock Variability Analysis:

Mean Closing Price: 154.64

Standard Deviation: 2.87
Variance: 8.25
Max Price: 158.40
Min Price: 150.34
Mean Daily Return (%): 0.61
Volatility (Daily Std Dev %): 1.26



[ ]: