Name: Balakrishna Mupparaju

Assignment: Project Milestone 4

In [12]:
```python
import requests
import pandas as pd

# API Endpoint
api_url = "https://www.alphavantage.co/query"

# Sample ticker list
tickers = ["AAPL", "MSFT", "GOOGL"]
api_key = "19IUCK3K3ZLSIV1G"

# Store API data
api_data = []

# Loop through tickers and fetch data
for ticker in tickers:
    params = {
        "function": "TIME_SERIES_DAILY",
        "symbol": ticker,
        "apikey": api_key
    }

    response = requests.get(api_url, params=params)
    data = response.json()

    if "Time Series (Daily)" in data:
        time_series = data["Time Series (Daily)"]
        for date, values in time_series.items():
            api_data.append({
                "Ticker": ticker,
                "Date": date,
                "Open": float(values.get("1. open", 0)),
                "High": float(values.get("2. high", 0)),
                "Low": float(values.get("3. low", 0)),
                "Close": float(values.get("4. close", 0)),
                "Volume": int(values.get("5. volume", 0))
            })
    else:
        print(f"No data for {ticker}: {data.get('Note', 'Unknown error')}")
```

In [14]:
```python
#Step #1: Replace Headers
#Description: Ensures consistency in naming conventions.
#Renames column headers to match naming conventions across datasets.

for record in api_data:
    record["Symbol"] = record.pop("Ticker")  # Rename Ticker to Ticker
    record["Transaction_Date"] = record.pop("Date")  # Rename Date
```

In [16]:
```
#Step #2: Fix Casing & Format Values
#Description: Standardizes text fields and ensures clean formatting.
#Removes trailing spaces and ensures uppercase consistency.


for record in api_data:
    record["Symbol"] = record["Symbol"].strip().upper()  # Trim spaces & set
```

In [18]:
```
#Step #3: Identify & Remove Duplicates
#Description: Eliminates repeated records.
#Filters out duplicate records based on Stock Symbol & Date

seen = set()
filtered_api_data = []
for record in api_data:
    identifier = (record["Symbol"], record["Transaction_Date"])  # Unique Ke
    if identifier not in seen:
        seen.add(identifier)
        filtered_api_data.append(record)
```

In [20]:
```
#Step #4: Convert Date Formats
#Description: Ensures proper datetime formatting.
#Converts Date field into proper datetime format for analysis.

for record in filtered_api_data:
    record["Transaction_Date"] = pd.to_datetime(record["Transaction_Date"],
```

In [22]:
```
#Step #5: Remove Outliers (Extreme Stock Prices)
#Description: Identifies and removes extreme anomalies in stock prices.
#Removes bad data (e.g., incorrect extreme values for stock price).
# Define acceptable stock price range

min_price, max_price = 1, 5000

cleaned_api_data = [record for record in filtered_api_data if min_price <= r
```

In [24]:
```
#All transformations applied within the API/JSON source.
#Ensures consistency with Wikipedia & Kaggle data. Maintains clean & structu
import pprint
pprint.pprint(cleaned_api_data[:5])  # Display a sample of cleaned data
```

```
[{'Close': 211.26,
  'High': 212.57,
  'Low': 209.77,
  'Open': 212.36,
  'Symbol': 'AAPL',
  'Transaction_Date': Timestamp('2025-05-16 00:00:00'),
  'Volume': 54737850},
 {'Close': 211.45,
  'High': 212.96,
  'Low': 209.54,
  'Open': 210.95,
  'Symbol': 'AAPL',
  'Transaction_Date': Timestamp('2025-05-15 00:00:00'),
  'Volume': 45029473},
 {'Close': 212.33,
  'High': 213.94,
  'Low': 210.5801,
  'Open': 212.43,
  'Symbol': 'AAPL',
  'Transaction_Date': Timestamp('2025-05-14 00:00:00'),
  'Volume': 49325825},
 {'Close': 212.93,
  'High': 213.4,
  'Low': 209.0,
  'Open': 210.43,
  'Symbol': 'AAPL',
  'Transaction_Date': Timestamp('2025-05-13 00:00:00'),
  'Volume': 51909332},
 {'Close': 210.79,
  'High': 211.2679,
  'Low': 206.75,
  'Open': 210.97,
  'Symbol': 'AAPL',
  'Transaction_Date': Timestamp('2025-05-12 00:00:00'),
  'Volume': 63775814}]
```

Ethical implications:

In this data wrangling process, I made several modifications to the raw website data: I replaced ambiguous headers for clarity, standardized textual values by enforcing title case, removed duplicate records to ensure uniqueness, converted date strings to proper datetime objects for accurate temporal analysis, and trimmed critical fields to eliminate extra spaces and missing values. Although the data—sourced directly from Wikipedia—is publicly available and generally reputable, legal and regulatory guidelines require proper attribution and compliance with Wikipedia's terms of use. The primary risks in these transformations include potential mismatches during fuzzy matching or inadvertent removal of valid data if duplicates are misidentified; additionally, assumptions such as converting all text to title case may not perfectly represent all company names. The data was sourced by scraping an official and publicly accessible page, and its credibility was verified through cross-referencing with other industry sources where possible. Data was

acquired ethically under the guidelines for public web scraping, and to mitigate ethical risks, I documented all transformation assumptions, applied conservative duplicate removal criteria, and ensured that no personally sensitive information was manipulated or exposed.

In [ ]: