Final Project Report

A.D.H.- G: Hannah Yoran, Dev Kunjadia, Andrew Baker

Git Hub Repo Link: https://github.com/devk03/SI206_FINAL/tree/main

Original Goals & Planning the Project

The overall objective of our final project was to explore potential correlations and trends between COVID-19 data, stock market performance, and employment statistics over time. To achieve this, we utilized three APIs: the COVID Tracking Project API (https://covidtracking.com/data/api/version-2), the Alpha Vantage Stock Market API (https://www.alphavantage.co/), and the Bureau of Labor Statistics API (https://www.bls.gov/bls/api_features.htm).

Using the COVID Tracking Project API, we planned to gather data on infection rates over time (2019–2024), regional infection patterns, and the severity of cases (e.g., hospitalization and ventilator use). From the Alpha Vantage API, we aimed to analyze stock market indicators such as index fund performance and growth trends, investigating potential connections to economic impacts. Lastly, through the Bureau of Labor Statistics API, we planned to collect employment data from 2017–2024, including unemployment rates, wage and income trends, and regional variations. This allowed us to assess pre- and post-COVID trends to identify potential positive or negative patterns.

Goals Achieved & Data Gathered

We successfully achieved our goals of gathering and analyzing data related to COVID-19 infection rates over time, with a focus on regional data for California. Additionally, we compared this information to stock market performance, specifically the S&P 500 (SPY). To accomplish these objectives, we used the following APIs:

- COVID-19 Transmission Data: COVID Act Now API (https://covidactnow.org/data-api) for detailed infection rates and trends.
- Stock Market Data: Alpha Vantage API (https://www.alphavantage.co/) to track S&P 500 performance over the same period.
- Weather Data: Open-Meteo API (https://open-meteo.com/) to analyze weather trends as an additional contextual factor.

`The data collected from these sources allowed us to explore the relationships between COVID-19 infection rates, stock market trends, and environmental factors, effectively meeting part of the original project goal. However, we decided to change part of the goal while working through the project to focus on including the weather and focusing on that new aspect instead of employment statistics.

Problems Faced

One significant challenge was finding a suitable COVID-19 API that was open and accessible. The original COVID-19 API we planned to use was no longer functional when we revisited it. Additionally, identifying APIs that were well-maintained and up-to-date proved difficult. We wanted to make sure that the data was current or atleast closer to the present date. We ran into difficulties when finding a COVID-19 API that had data past 2021.

Another issue arose with integrating labor and employment statistics into our analysis. Upon deeper exploration of the labor statistics API we initially intended to use, we found that the dataset was limited and lacked strong correlations with the other data we were analyzing. This made it challenging to draw meaningful conclusions. As a result, we decided to pivot and take a more engaging project by incorporating weather data into our analysis, which offered further insights and was more aligned with our project goals.

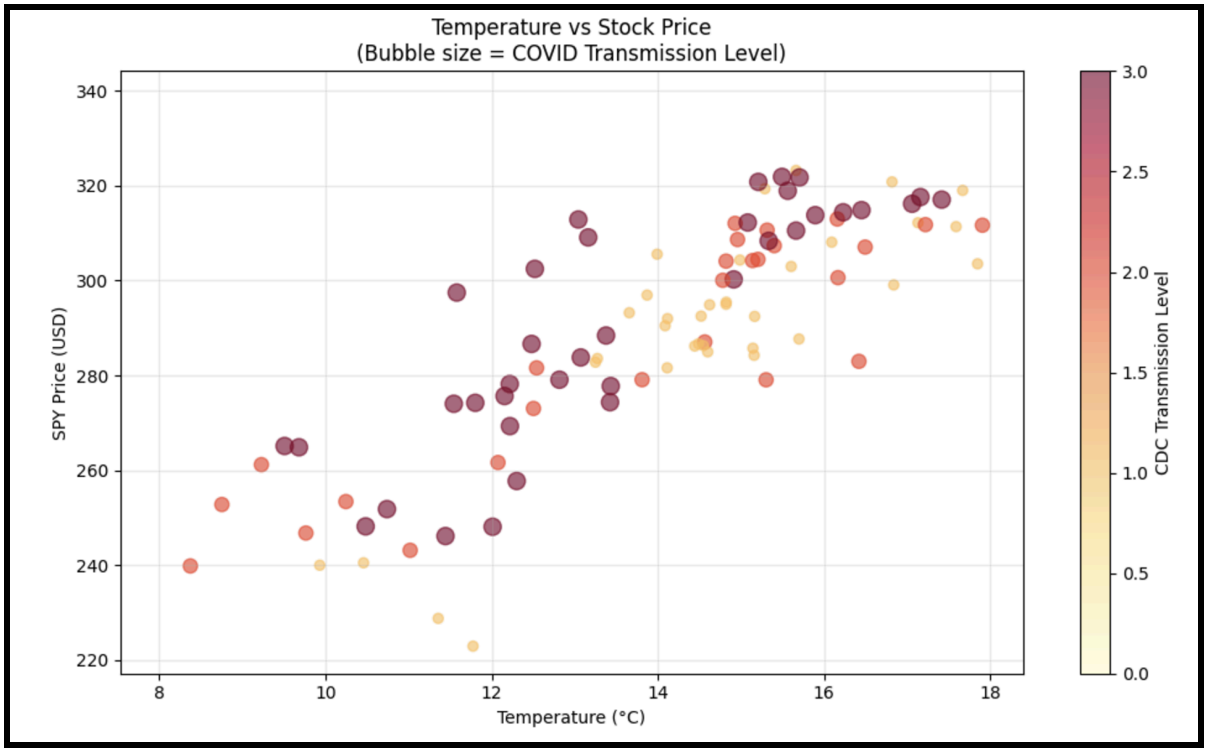
Finally, while trying to combine and analyze the data we wanted the common key to be the date. While this worked perfectly for the COVID-19 API and the Weather API, for the Stock Market API some on dates there was nothing reported. Due to this there were null values in our dataset so we had to clean our data by inserting some empty rows.

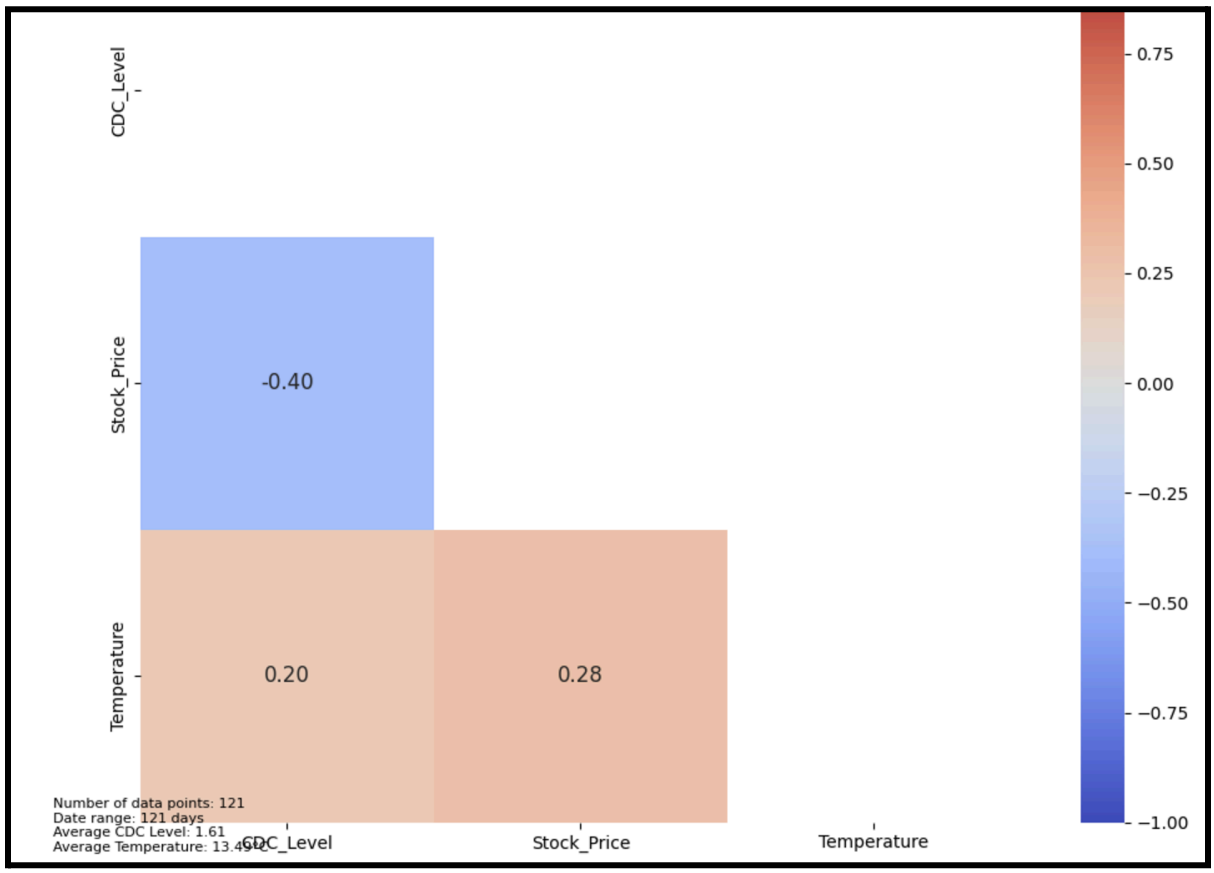
Calculations from the Data in the Database

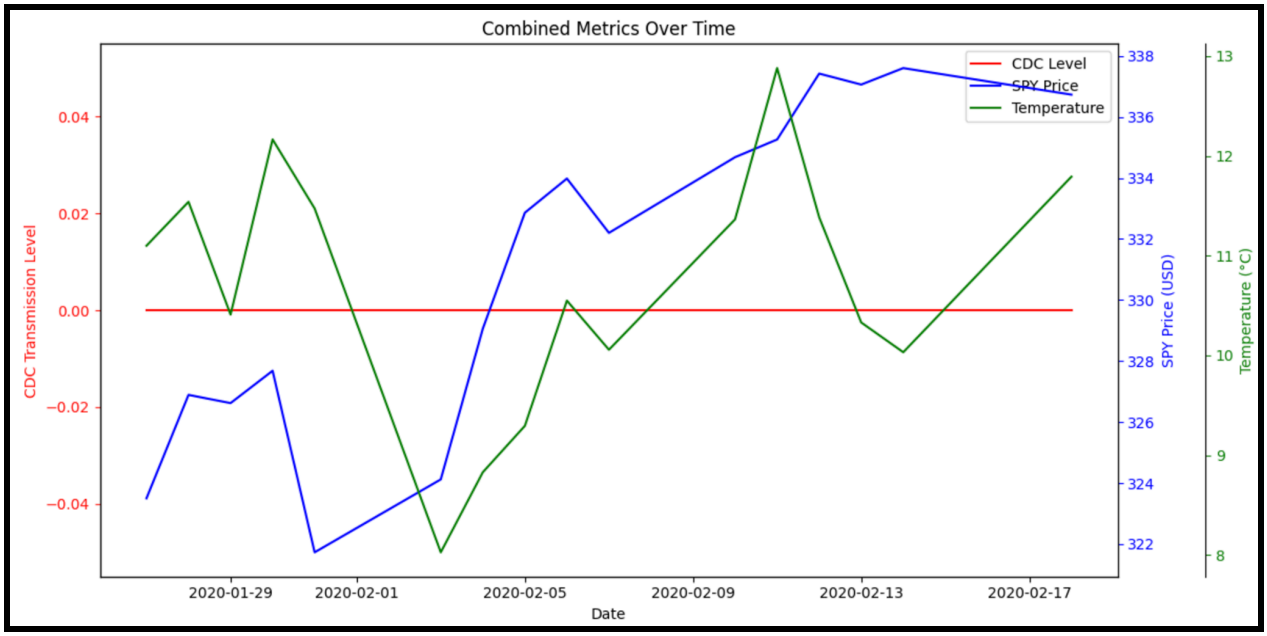
1. The calculations from the data in the database (i.e. a screenshot) (10 points)

```
Current record counts:
COVID records: 250
Stock records: 250
Weather records: 250

Performing statistical analysis...

Data Analysis Results:

1. Average Stock Prices by CDC Transmission Level:
Level 0: $326.94
Level 3: $309.66
Level 2: $310.08
Level 1: $290.12

2. Temperature Correlation with Transmission Levels:
Correlation coefficient: 0.328
Weak or moderate correlation

3. Market Volatility:
Daily returns standard deviation: 2.470%
High volatility period

4. Monthly Trends:

2020-01:
  Average Stock Price: $325.28
  Average CDC Level: 0.0
  Average Temperature: 11.3°C

2020-02:
  Average Stock Price: $327.37
  Average CDC Level: 0.0
  Average Temperature: 11.3°C

2020-03:
  Average Stock Price: $264.79
  Average CDC Level: 2.3
  Average Temperature: 11.5°C

2020-04:
  Average Stock Price: $275.41
  Average CDC Level: 2.4
  Average Temperature: 12.7°C

2020-05:
  Average Stock Price: $291.65
  Average CDC Level: 1.0
  Average Temperature: 14.8°C

2020-06:
  Average Stock Price: $310.15
  Average CDC Level: 1.7
  Average Temperature: 15.9°C

2020-07:
  Average Stock Price: $319.99
  Average CDC Level: 3.0
  Average Temperature: 15.3°C

2020-08:
  Average Stock Price: $338.75
  Average CDC Level: 2.9
  Average Temperature: 16.9°C

2020-09:
  Average Stock Price: $336.04
  Average CDC Level: 2.0
  Average Temperature: 17.0°C

Data has been exported to 'analysis_output.json'.
```

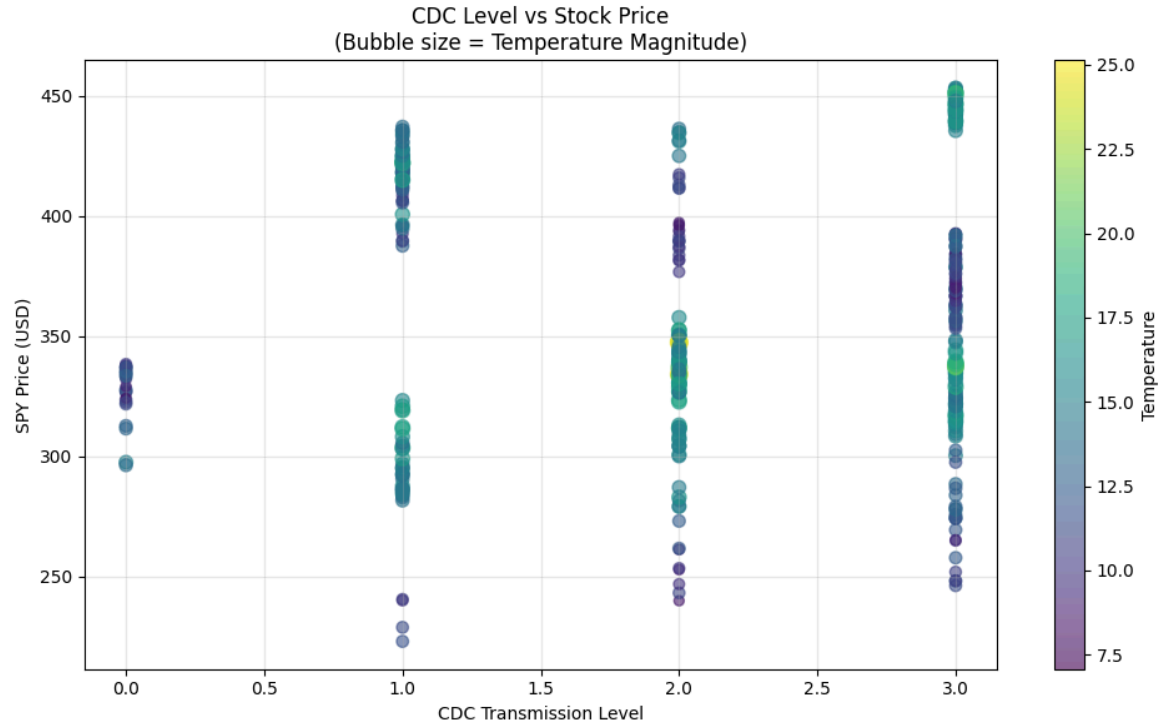
The Visualizations Created (3)

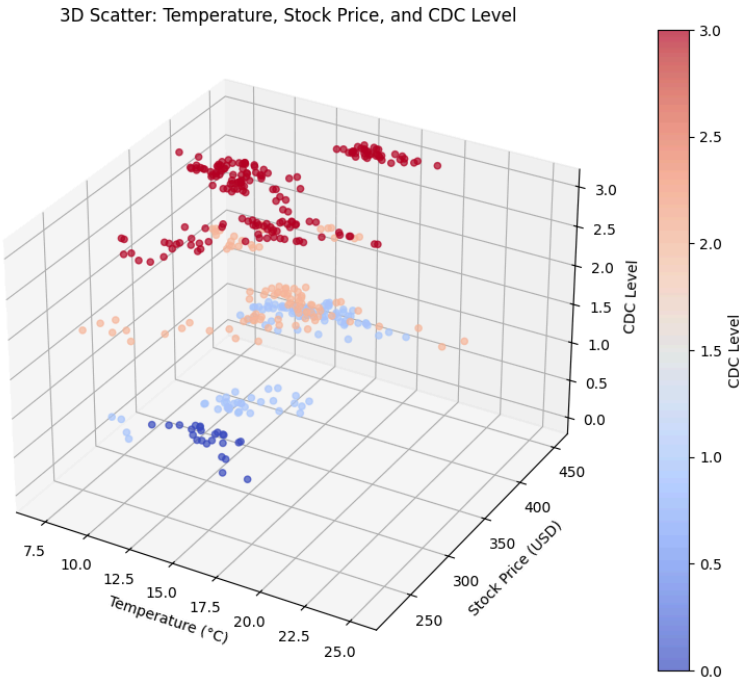

Temperature vs Stock Price
(Bubble size = COVID Transmission Level)
SPY Price (USD)
Temperature (°C)
CDC Transmission Level
340
320
300
280
260
240
220
8
10
12
14
16
18
3.0
2.5
2.0
1.5
1.0
0.5
0.0


CDC_Level
Stock_Price
Temperature
-0.40
0.20
0.28
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Number of data points: 121
Date range: 121 days
Average CDC Level: 1.61
Average Temperature: 13.45°C
CDC_Level
Stock_Price
Temperature



Extra Credit Visualizations Created (2)





Instructions for Running our Code

To run the COVID-19, Weather, and Stock Market Analysis Tool, start by cloning the repository using git clone [repository-url] and navigating to the project directory. Set up a virtual environment with python3 -m venv venv and activate it using source venv/bin/activate (or venv\Scripts\activate on Windows). Install the required dependencies by running pip install -r requirements.txt. Next, execute the main script with python main.py. Since the script processes 25 records per run, you'll need to run it four times to collect the full dataset of 100 records per table. During each run, the script will fetch and process new data for COVID-19 transmission, stock market, and weather metrics, updating the SQLite database and displaying the current record count in the terminal. After the fourth run, the script will automatically generate a combined dataset (analysis_output.json) in the project directory and save visualizations in the visualizations/ folder. These visualizations include visualization1_combined_metrics.png, visualization2_bubble_plot.png, and visualization3_enhanced_correlation.png, providing insights into the relationships between COVID-19, stock market performance, and weather patterns.

Documentation for Each Function (included - describing input & output for each funx)

- get_or_create_date_id(conn, date_str)
    - Purpose: Ensures a specific date exists in the dim_date table and returns its ID. If the date is not present, it inserts it.
    - Inputs:

        - conn: SQLite database connection object.
        - date_str: A string representing the date (e.g., "2023-12-01").
    - Outputs:
        - An integer date_id corresponding to the date in the dim_date table.
- get_latest_covid_date(conn)
    - Purpose: Retrieves the most recent date from the covid_transmission table for which data is available.
    - Inputs:
        - conn: SQLite database connection object.
    - Outputs:
        - A string representing the latest date (e.g., "2023-12-01") or a default date if no data is available.
- process_covid_data(conn)
    - Purpose: Fetches data from the COVID Act Now API and inserts up to 25 new records into the covid_transmission table.
    - Inputs:
        - conn: SQLite database connection object.
    - Outputs:
        - An integer representing the number of new records inserted.
- get_unprocessed_stock_dates(conn)
    - Purpose: Retrieves up to 25 dates for which stock market data has not yet been inserted.
    - Inputs:
        - conn: SQLite database connection object.
    - Outputs:
        - A list of strings, each representing an unprocessed date.
- process_stock_data(conn)
    - Purpose: Fetches stock market data from Alpha Vantage API and inserts up to 25 records into the stock_market table.
    - Inputs:
        - conn: SQLite database connection object.
    - Outputs:
        - An integer representing the number of new records inserted.
- Get_unprocessed_weather_dates(conn)
    - Purpose: Retrieves up to 25 dates for which weather data has not yet been inserted.
    - Inputs:
        - conn: SQLite database connection object.
    - Outputs:
        - A list of strings, each representing an unprocessed date.
- process_weather_data(conn)
    - Purpose: Fetches weather data from Open-Meteo API and inserts up to 25 records into the weather_data table.
    - Inputs:

        - conn: SQLite database connection object.
    - Outputs:
        - An integer representing the number of new records inserted.
- verify_data_counts(conn)
    - Purpose: Prints and returns the current counts of records in each database table.
    - Inputs:
        - conn: SQLite database connection object.
    - Outputs:
        - A tuple containing counts of records in the covid_transmission, stock_market, and weather_data tables.
- setup_visualization_directory()
    - Purpose: Ensures the existence of a visualizations directory for storing plots.
    - Inputs: None.
    - Outputs:
        - A string representing the directory name ("visualizations").
- create_visualizations(json_data)
    - Purpose: Generates three visualizations using combined JSON data:
        - A combined time series plot.
        - A bubble plot.
        - A correlation heatmap.
    - Inputs:
        - json_data: A list of dictionaries containing combined data.
    - Outputs:
        - Saves visualization files in the visualizations/ directory.
- export_json_data(conn)
    - Purpose: Joins data from all three tables and exports it to a JSON file (analysis_output.json).
    - Inputs:
        - conn: SQLite database connection object.
    - Outputs:
        - A Python list containing the combined data as dictionaries.
- analyze_data_statistics(conn)
    - Purpose: Performs statistical analysis on the collected data, including:
        - Average stock prices by CDC level.
        - Correlations between temperature and transmission levels.
        - Market volatility analysis.
        - Monthly trends.
    - Inputs:
        - conn: SQLite database connection object.
    - Outputs:
        - A dictionary containing analysis results.
- main()
    - Purpose: Orchestrates the execution of the entire script, including data collection, visualization, and analysis.

- Inputs: None.
- Outputs: None directly; manages the flow of the program and logs progress/results.

Documentation for Each Resource Used

2. You must also clearly document all resources you used. The documentation should be of the following form (20 points)

APIs
Lecture notes
Random online stuff?
Documentation - like method documentation

Date	Issue Description	Location of Resource	Result (did it solve the issue?)
11/25	Needed to retrieve historical COVID-19 transmission data for analysis.	COVID Act Now API Documentation	Solved the issue. The API provided detailed transmission data, which was successfully fetched and inserted into the database.
11/25	Needed accurate and reliable stock market data, particularly for the S&P 500 (SPY).	Alpha Vantage API Documentation	Solved the issue. The API provided the required historical stock market data, which was processed and stored.
11/26	Required weather data for San Francisco to analyze its relationship with COVID-19 transmission levels and stock prices.	Open-Meteo API Documentation	Solved the issue. The API provided weather data, including average, maximum, and minimum temperatures, which were used in the analysis.
11/27	Needed to verify the correct usage of SQLite foreign keys for linking data tables.	SQLite Documentation (https://sqlite.org/foreignkeys.html)	Solved the issue. The documentation clarified the syntax and rules for defining and enforcing foreign keys, ensuring efficient database design.

12/2	Struggled to implement visualizations using matplotlib, particularly with creating bubble plots and heatmaps.	Matplotlib Documentation (https://matplotlib.org/stable/contents.html) and Stack Overflow (https://stackoverflow.com)	Solved the issue. The documentation and examples from Stack Overflow provided the necessary guidance to create effective visualizations.