808s and Heartbreak

Amadou Diallo and [Geo Johnson](mailto:georjohn@umich.edu)

SI 206 Final Project Report

**Table of Contents:**

1. **Goals of Project, API Key/Website Usage**
2. **Problems Encountered**
3. **Calculations Within Project**
4. **Visualization Within Project**
5. **Instructions on Running Code**
6. **Description of Functions**
7. **Resource Documentation**

**Goals of Project, API Keys/Website Usage**

Project Goals

This project aims to analyze whether weather conditions significantly impact the number of seats available on buses in Boston. By combining weather data with public transportation data, we seek to uncover patterns and correlations that could provide valuable insights for optimizing bus schedules, enhancing commuter experience, and improving public transportation efficiency.

Despite several challenges, including data integration issues and API complexities, we successfully overcome these obstacles through continuous debugging, thorough research, and collaborative problem-solving.

APIs and Websites Used

1. WeatherStack API
   * Purpose: To gather real-time and historical weather data for Boston.
   * Key Data Points Collected:
     + Temperature
     + Precipitation
   * API Endpoint: http://api.weatherstack.com/current
   * Authentication: Accessed using a unique API key (weather\_key).
2. MBTA API
   * Purpose: To collect data on Boston's public transportation system, focusing on bus availability.
   * Key Data Points Collected:
     + Monthly bus attendance
     + Number of available seats
   * API Endpoint: https://api-v3.mbta.com/stops
   * Authentication: Accessed using a unique API key (bus\_key).

Data Integration

The data collected from these APIs was stored in a SQLite database, with tables for both weather and bus-related data. We then combined this information to investigate potential relationships between weather conditions and bus seat availability.

**Achievements within the Project**

The project successfully analyzed the relationship between weather conditions and bus seat availability in Boston, fulfilling its primary objective. Despite initial challenges in integrating data from multiple sources, we effectively utilized APIs to collect and process the required data, merging weather and public transportation metrics into a unified dataset.

Specifically, we:

* Gathered weather data (temperature and precipitation) for Boston using the WeatherStack API.
* Collected public transportation data (monthly bus attendance and seat availability) using the MBTA API.
* Stored and processed the data in a SQLite database for seamless analysis and visualization.
* Created visualizations to explore correlations and trends between weather conditions and bus seat availability, such as:
  + Line graphs to observe monthly trends in seat availability.
  + Scatter plots to investigate relationships between temperature and available seats.

#### APIs and Data Gathered

1. WeatherStack API
   * Data Collected:
     + Current temperature (in degrees Fahrenheit).
     + Precipitation levels (in millimeters).
2. MBTA API
   * Data Collected:
     + Monthly bus attendance.
     + Number of available seats for buses.

#### Summary of Insights

Through analysis and visualizations, we identified key trends in the data that suggest:

* Weather conditions (e.g., higher precipitation levels) may influence the number of available bus seats.
* Sunny days often showed higher seat availability compared to rainy days.
* The temperature had a measurable correlation with seat availability, though moderately strong.

**Calculations Within Project**

1. Correlation Between Temperature and Available Seats

* Calculation: The correlation coefficient between temperature (from the Weather table) and available\_seats (from the BusData table).
* Purpose: This determines if there is a relationship between temperature changes and the number of available bus seats.  
  Correlation between temperature and available seats: -0.35

2. Average Available Seats on Rainy Days

* Calculation: The mean number of available seats when precipitation > 0.
* Purpose: To evaluate if rainy weather impacts seat availability.  
  Average available seats on rainy days: 23

3. Average Available Seats on Sunny Days

* Calculation: The mean number of available seats when precipitation = 0.
* Purpose: To compare seat availability on sunny vs. rainy days.  
  Average available seats on sunny days: 30

4. Combined Data Visualization

* The program generates line graphs and scatter plots to visualize the relationship between weather and bus seat availability:
  + Line Graph: Shows the trend of available seats over months.
  + Scatter Plot: Displays the relationship between temperature and available seats.

5. Further Calculations

* Rain: When it rains, fewer people tend to use public transportation. Rain can cause delays and make waiting for buses or trains less pleasant, leading some commuters to opt for alternative modes of transport like rideshares or personal vehicles.
* Snow and Cold: Winter weather, including snow and freezing temperatures, can severely disrupt public transit. Snow and ice can cause delays and make travel conditions hazardous, leading to a decrease in ridership. Additionally, the discomfort of waiting in the cold can deter people from using public transit.
* Heat: Extreme heat can also affect public transportation usage. High temperatures can make waiting for and riding on public transit uncomfortable, leading some people to seek cooler alternatives

**Visualization Within Project**

**Instructions on Running Code**

Follow these steps to successfully run the code and analyze the data:

1. API Keys:
   * Obtain an API key from [WeatherStack](https://weatherstack.com/) and replace weather\_key in the code with your key.
   * Obtain an API key from the [MBTA API](https://api-v3.mbta.com/) and replace bus\_key in the code with your key.
2. Python Environment:
   * Use Python 3.6 or later.
   * Ensure your system supports SQLite (this comes pre-installed with most Python installations).

Steps to Run the Code

1. Save the Code:  
   Copy and save the code into a file
2. View Debugging Outputs:  
   The terminal will display debugging outputs, such as fetched weather data, bus data, and combined data.
3. Visualizations and Results:
   * Two graphs will be generated and displayed:
     + Monthly trends in available bus seats.
     + Correlation between temperature and seat availability.
   * These graphs are also saved as PNG files in the same directory:
     + available\_seats\_over\_months.png
     + available\_seats\_vs\_temperature.png
4. Database Output:
   * The program creates a database file named final\_project.db in the directory.
   * This database contains two tables:
     + Weather with temperature and precipitation data.
     + BusData with monthly attendance and available seat data.

**Description of Functions**

1. **fetch\_weather\_data(api\_key, location)**
   * Fetches the current weather data for the specified location using the Weatherstack API.
   * Inputs: API key, location (e.g., "Boston").
   * Output: Dictionary containing current weather details (e.g., temperature, precipitation).
2. **fetch\_bus\_data(api\_url)**
   * Fetches bus data from the MBTA API.
   * Input: API URL.
   * Output: List of dictionaries containing bus data (e.g., month, attendance).
3. **store\_weather\_data(data)**
   * Stores weather data into a SQLite database (Weather table).
   * Input: Weather data dictionary (temperature, precipitation).
   * Output: Saves data to the database.
4. **store\_bus\_data(data)**
   * Stores bus data into a SQLite database (BusAttendance table).
   * Input: List of dictionaries with bus attendance data (month, attendance).
   * Output: Saves data to the database.
5. **fetch\_combined\_data()**
   * Combines weather and bus data by joining the Weather and Bus tables on the month column.
   * Output: List of tuples containing combined data (month, temperature, precipitation, attendance).
6. **insert\_weather\_data(month, temperature, humidity)**
   * Inserts weather data for a specific month into the Weather table.
   * Inputs: Month, temperature, humidity.
7. **insert\_bus\_data(month, attendance)**
   * Inserts bus attendance data for a specific month into the Bus table.
   * Inputs: Month, attendance.
8. **count\_available\_seats(bus\_data)**
   * Counts the number of buses with "many seats available" and "few seats available" based on bus occupancy status.
   * Input: Bus data list.
   * Output: Tuple with counts for "many seats" and "few seats."
9. **visualize\_seat\_availability(bus\_data)**
   * Generates a pie chart to show the percentage of buses with "many seats" vs. "few seats" available.
   * Input: Bus data list.
10. **calculate\_metrics(data)**
    * Computes key metrics from combined data:
      + Correlation between temperature and attendance.
      + Average attendance on rainy days (precipitation > 0).
      + Average attendance on sunny days (precipitation = 0).
    * Input: Combined data list.
    * Output: Correlation, average rainy attendance, average sunny attendance.
11. **visualize\_data(data)**
    * Creates visualizations from the combined data, including:
      + Line graph of attendance over months.
      + Scatter plot of attendance vs. temperature.
    * Input: Combined data list.
12. **Main Program Execution (if \_\_name\_\_ == "\_\_main\_\_")**
    * Runs the program workflow:
      + Fetches weather and bus data.
      + Stores the data in the SQLite database.
      + Fetches combined data and performs calculations.
      + Displays metrics and visualizations.

**Resource Documentation**

| Date | Issue Description | Location of Resource | Result |
| --- | --- | --- | --- |
| 2024-12-10 | Unable to fetch weather data using WeatherStack API. | WeatherStack APIDocumentation | Yes, resolved after fixing query parameters and verifying API key. |
| 2024-12-11 | SQLite Weather table missing month column. | [StackOverflow](https://stackoverflow.com/questions/20050757/sqlite-add-column-to-table) | Yes, added the missing column with ALTER TABLE. |
| 2024-12-12 | fetch\_bus\_data function was returning an empty list. | [MBTA APIDocumentation](https://api-v3.mbta.com/) | Yes, fixed the endpoint URL and added the correct API key. |
| 2024-12-13 | Pie chart not rendering properly for seat availability. | MatplotlibDocumentation | Yes, resolved by adjusting the explode parameter and color settings. |
| 2024-12-14 | Data correlation calculation throwing an error. | Pandas Documentation | Yes, resolved by ensuring proper column names and types in the DataFrame. |