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 have a significant impact on 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 overcame 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**

**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**

**Resource Documentation**