Community Resilience Analysis Library

This document provides a compact overview of the Community Resilience Analysis Library — a Python-based toolkit designed to evaluate community recovery following disasters using mobility data. GitHub: <https://github.com/amarnath-reddy-0-9-1-2/Resilience/>

**1. Project Overview**

The library helps researchers and analysts quantify and visualize how communities recover from disasters using human mobility data. It supports two key models — the Resilience Triangle Model and the Area Under the Curve (AUC) Model — to measure the speed and quality of recovery over time.

**2. Disaster Context**

The primary dataset used is `portarthur\_sd\_df\_2019.rdata`, focusing on Port Arthur, Texas. The analysis centers around Tropical Storm Imelda, which impacted the region between September 17 and September 27, 2019.

**3. Key Models & Functionality**

* The library implements two core models:
* Resilience Triangle Model – Measures how quickly a community returns to pre-disaster conditions, calculating both recovery speed and depth.
* Area Under the Curve (AUC) Model – Quantifies total impact by calculating area-based loss and recovery over time.
* Main functionalities include:
* Data Preparation – Load and preprocess SafeGraph mobility data.
* Resilience Metrics Calculation – Automatically compute key recovery indicators.
* Visualization – Generate plots to illustrate resilience and recovery trends.
* Batch Processing – Run resilience analysis across all CBGs and export results to CSV.

**4. References**

### [1]Hong, H., Liu, L., Peng, Z.-R., & Li, W. (2021). *Measuring inequality in community resilience to natural disasters using large-scale mobility data*. Nature Communications, 12(1), 1870. <https://doi.org/10.1038/s41467-021-22177-2>

### [2] Chen, K., Hu, S., Hong, H., & Peng, Z.-R. (2024). *Community resilience to wildfires: A network analysis approach by utilizing human mobility data*. Computers, Environment and Urban Systems, 104, 102032. <https://doi.org/10.1016/j.compenvurbsys.2023.102032>

### **5. Folder Structure**

| **Folder/File** | **Detailed Description** |
| --- | --- |
| data/ | Contains the raw mobility data files required for analysis. Example:  • `portarthur\_sd\_df\_2019.rdata`, which stores SafeGraph mobility data for Port Arthur.  • `tl\_2019\_48\_bg.zip` has geographic data of the Texas region. |
| models/ | Includes core implementation of resilience models: • `resilience\_auc.py`: Implements the Area Under Curve (AUC) model to calculate the area loss and recovery after a disaster. • `resilience\_triangle.py`: Implements the Resilience Triangle model to measure loss and recovery using geometric triangle-based methods. |
| notebooks/ | Interactive Jupyter notebooks to test and debug the model logic: • `mobility\_patterns.ipynb`: Analyzes mobility data trends across various granularities. • `resilience\_auc.ipynb`: Runs and visualizes AUC model for a specific Census Block Group (CBG). • `resilience\_triangle.ipynb`: Executes and visualizes Triangle model for a specific CBG.  • `geographic\_patterns.ipynb`: This will generate geographic maps based on the data. |
| run\_examples/ | Standalone example scripts to run models easily: • `batch\_processing.py`: Applies the Triangle model to all CBGs and outputs a summary CSV. • `run\_auc\_example.py`: Demonstrates running the AUC model for one CBG. • `run\_triangle\_example.py`: Demonstrates running the Triangle model for one CBG. |
| visualization/ | Functions to create visual and textual output: • `graph\_visualization.py`: Plots mobility and resilience curves for each region. • `log\_visualizations.py`: Logs model metrics in a human-readable, well-formatted way. |
| utils.py | General-purpose utility functions used across the project. Includes helpers for smoothing, normalization, and date handling. |
| data\_processing.py | Responsible for loading, cleaning, and preprocessing SafeGraph mobility data. Prepares the dataset for modeling. |
| results/ | Stores all generated outputs: CSV results, resilience patterns, geo patterns, mobility patterns from model runs. Used to review or share analysis findings. More explanation in the results section. |

# **6. Results and Analysis**

The images and the CSV files are available in the results folder. Here are a few observations from the results:

1. All the CBG results are available in the cbg\_resilience\_summary.csv file.

2. Filtered CBGs that show a clear trend are stored in cbg\_resilience\_summary\_filtered.csv.

3. The ‘resilience\_patterns’ folder contains visualizations for one individual CBG using both models.

4. The ‘mobility\_patterns’ folder contains overall mobility patterns at different levels.

5. The ‘geo\_patterns folder’ contains the geographic plots based on the model data on the map.

**Key Findings**

1. Out of 309 CBGs, around 222 CBGs showed a clear trend for Hurricane Imelda.

2. The results are sensitive to hyperparameters such as:  
 - Baseline value  
 - Smoothing period  
 - Disaster timeline (some CBGs showed trends slightly after the disaster period)

3. Consistent Dip During Disaster (Sep 17–27, 2019):  
 All regions (CBGs/counties) show a sharp decline in inflow or in-degree, confirming disruption in daily mobility.

4. Varying Recovery Patterns Across Regions:  
 - Jefferson County shows faster recovery, indicating stronger resilience.  
 - Orange and Jeff Davis Counties show delayed/weaker recovery, indicating lower resilience.

5. Resilience Triangle Area Reflects Recovery Efficiency:  
 - Smaller triangle area → quicker mobility recovery and higher resilience.  
 - Larger triangle area → prolonged disruption and lower resilience.

6. Aggregated Trends Mask Local Disparities:  
 - Metro-level aggregation hides local variations.  
 - CBG-level plots reveal specific mobility dynamics and localized recovery trends.

7. Pre-Disaster Baseline is Crucial:  
 - Calculated as 2-month average before disaster.  
 - Enables identification of dip (tD) and recovery (t1) points.

8. Summary Statistics:  
 - Average Resilience: 15.43% , Average Robustness: 0.288, Average Vulnerability: 0.0049