Project Report

I modeled how criminal activity “spreads” across Los Angeles by treating each unique (calendar day, neighborhood) pair as its own graph node—yielding ~38 530 vertices from the LA Open Data “Crime Data from 2020–Present” (≈ 1 000 000 records). My goal was to ask: How connected are these day-area nodes on the same day? Which serve as critical bridges? And how fragmented is the network?

I began by writing scripts/0_preprocess.py in Python to deduplicate the raw CSV into data/day_area.csv. In Rust, I split the pipeline into three modules: graph.rs builds an UnGraph<(NaiveDate,String),()> and provides a bfs_distances function; analysis.rs implements degree_distribution, avg_shortest_path, closeness_centrality, and component_count; and main.rs ties everything together, prints summaries, and writes report/metrics.json and report/degree_counts.csv. I added unit tests for BFS on a 3-node chain, triangle degree checks, isolated-node component counts, and tiny-CSV graph building to ensure correctness.

My results show a heavy-tailed degree distribution, an average shortest-path length of 0.952 hops, five day-areas with maximal closeness (score = 38 529), and 1 904 disconnected components. To reproduce my work, run:

py -3 scripts/0_preprocess.py

cargo build --release && cargo run --release (~5 s)

py -3 report/python.py

I used ChatGPT to help structure code comments and draft this report; all suggestions were reviewed and adapted.