Graph Algorithms

Qikai Feng 106395231
Yi Han 606335602
Xinxin Chang 306405941

Question 1

1. Because ρ_{ij} is defined as the Pearson correlation coefficient between two real-valued random variables $r_i(t)$ and $r_j(t)$, the Cauchy-Schwarz inequality implies

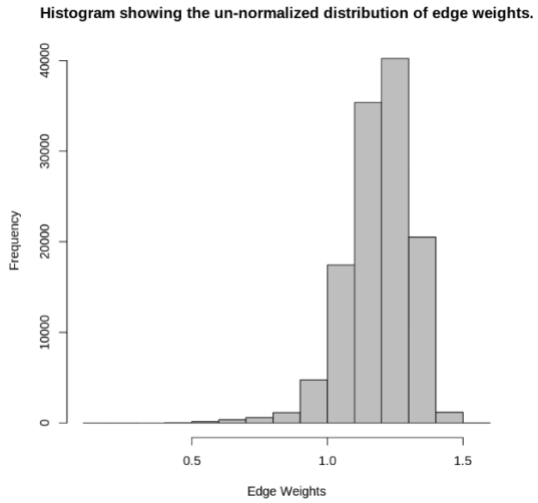
$$-1 \leq \rho_{ij} \leq +1$$

The upper bound for ρ_{ij} is +1. This occurs when there is a perfect positive linear relationship between the log-normalized returns of stock i and stock j.
The lower bound for ρ_{ij} is -1. This occurs when there is a perfect negative linear relationship between the log-normalized returns of stock i and stock j.

2. Why use log normalized return $r_i(t)$ instead of regular return $q_i(t)$:
   a) Simple returns $q_i(t)$ are asymmetric. For example, a stock price dropping by 50% ($q_i(t)$=−0.5) requires a 100% increase ($q_i(t)$=+1.0) to return to the original price. The range is $[-1, \infty)$.
   Log returns ($r_i(t) = log(P_t/P_{t-1})$ are more symmetric. A change from price P to $P/2$ gives $r_i(t) = log(0.5) = -log(2)$. A change from P to $2P$ gives $r_i(t) = log(2)$. The magnitudes are equal. The range is $(-\infty, \infty)$. This symmetry is beneficial for statistical modeling.
   b) Log returns are additive over time. If you have log returns $r_1, r_2, \ldots, r_T$ for T consecutive periods, the total log return over the T periods is simply $R_T = r_1 + r_2 + \cdots + r_T$. This is because $r_t = log(P_t) - log(P_{t-1})$. So, $\sum_{t=1}^{T} r_t = (log(P_1) - log(P_0)) + (log(P_2) - log(P_1)) + \cdots + (log(P_T) - log(P_{T-1})) = log(P_T) - log(P_0) = log(P_T/P_0)$. Simple returns are not additive over time; they are multiplicative. The total return over T periods using simple returns is $(1 + q_1)(1 + q_2)\ldots(1 + q_T) - 1$. Additivity is a convenient property for many financial calculations and models.
   c) Financial asset prices $p_i(t)$ are typically non-stationary. Simple returns $q_i(t)$might be closer to stationary but their distribution is often skewed (especially for longer holding periods) and exhibits “fat tails” (leptokurtosis). Log returns $r_i(t)$ are more likely to be approximately normally distributed. While still often exhibiting fat tails, the approximation to normality is generally better for log returns than for simple returns. Many statistical methods, including the Pearson correlation coefficient, perform better or have more straightforward interpretations when the underlying data are closer to normally distributed. The logarithm helps in “taming” extreme positive values and making the distribution more symmetric.
   d) Log returns have a higher tendency to be stationary compared to prices or simple returns. Stationarity is a crucial assumption in many time series models.

e) For small returns, $q_i(t) \approx r_i(t)$ because $log(1+x) \approx x$ for small x. So, log returns can be interpreted as approximate percentage changes. However, as returns become larger, this approximation breaks down, and the distinct properties of log returns (like additivity) become more important.

Question 2



In this task, we investigate the correlation structure between stock price movements by constructing a weighted undirected graph. Each node in the graph represents a stock, and each edge weight captures the dissimilarity between two stocks, defined using the correlation of their log-normalized return time series:

$$w_{ij} = \sqrt{2(1-\rho_{ij})}$$

Here, ρ_{ij} is the Pearson correlation coefficient between the log-normalized return time series of stock i and stock j. This transformation ensures that perfectly positively correlated stocks yield a weight of 0, while perfectly negatively correlated stocks yield a weight of 2.

To ensure consistent data quality, only stocks with complete daily records (765 data points) over the 3-year period were retained. This resulted in 494 valid stocks. For each pair of stocks, we computed ρ_{ij} and the corresponding weight w_{ij}, generating a total of 121,771 edges in the correlation graph.

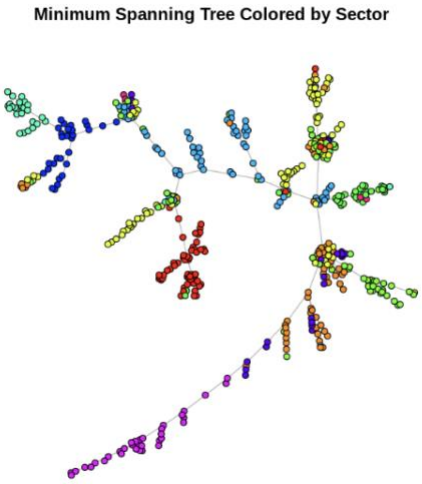
The histogram below illustrates the unnormalized distribution of edge weights. We observe that:

• Most weights fall within the range of 1.0 to 1.4, indicating weak or modest negative correlations between most stock pairs.

• No edges reach the theoretical limits of 0 or 2, suggesting that neither perfect positive nor perfect negative correlations are present in the dataset.

• The shape of the distribution is unimodal and slightly skewed, which aligns with the intuition that the market contains many loosely related assets rather than tightly coupled ones.

Question 3



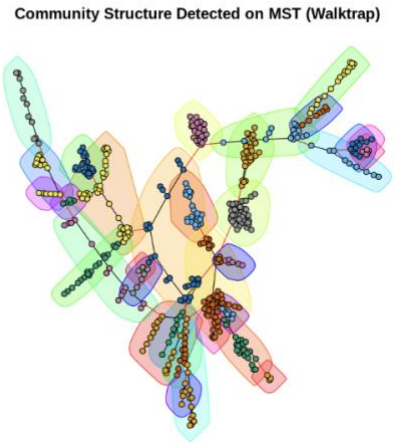
In this task, we extracted a Minimum Spanning Tree (MST) from the previously constructed correlation graph using Prim's algorithm. Each stock is represented as a node, and edges represent the strongest correlation-based proximity using weights $w_{ij} = \sqrt{2(1-\rho_{ij})}$. The MST connects all nodes with the minimum total edge weight, preserving the most meaningful pairwise relationships.

To enhance interpretability, we assigned a distinct color to each stock based on its sector. As shown in the figure, several clusters emerged where stocks of the same sector are grouped together. These structures are known as Vine clusters, and they reflect the internal cohesion of sectors, where companies from the same industry tend to exhibit similar price movements over time.

The tree also reveals important bridge nodes—stocks that connect different clusters—potentially representing multi-sector companies or economic influencers.

Question 4

```
Homogeneity: 0.6826
Completeness: 0.4793
```



We applied the Walktrap community detection algorithm to the Minimum Spanning Tree (MST) generated in the previous question. The Walktrap method simulates short random walks to identify groups of nodes that are structurally close in the graph. Applied to the MST-which captures the most significant pairwise relationships-the algorithm reveals clusters of tightly connected stocks.

Each node was assigned a community label, and we visualized the results by coloring each node according to its detected community. The resulting plot exhibits several dense branches, many of which visually align with known industry groupings.

To assess how well these communities correspond to the actual sectors, we computed two metrics: homogeneity and completeness. These are standard evaluation measures in clustering analysis and were implemented using the homogeneity () and completeness() functions from the clevr R package.

- Homogeneity measures whether each community contains mostly stocks from a single industry.
- Completeness evaluates whether all stocks from the same industry are grouped into the same community.

Both metrics range from 0 to 1, and are defined using conditional entropy**:**

$$\text{Homogeneity} = 1 - \frac{H(C|L)}{H(C)}, Completeness = 1 - \frac{H(L|C)}{H(L)}$$

Where:

- C represents the ground-truth sector labels,
- L represents the predicted community assignments,
- H(·) denotes Shannon entropy.

In our experiment, we obtained: Homogeneity = 0.6826, Completeness = 0.4793.

These results indicate that the detected communities moderately align with sector divisions. While many communities are internally consistent (high homogeneity), some sectors are split across multiple communities (lower completeness). This reflects the complexity of real financial systems, where stock behavior is influenced not only by sector but also by market trends, external events, and cross-industry dynamics.

Question 5

In this question, we evaluate how well a stock's sector can be predicted by examining the sectors of its direct neighbors in the MST. For each node $v_i$we compute the proportion of its neighbors that belong to the same sector:

$$\mathrm{P}(\mathrm{v_i} \in \mathrm{S_i}) = \frac{\text{\# of neighbors in same sector}}{\text{total \# of neighbors}}$$

We then average this score across all nodes to obtain a final accuracy metric, denoted as α. To establish a baseline, we compute the expected accuracy of a random guess, which assumes that a node's industry is predicted based on global sector frequencies:

$$\alpha_{random} = \frac{1}{|V|}\sum_{i=1}^{|V|}\frac{|S_i|}{|V|}$$

In our experiment, we obtained:

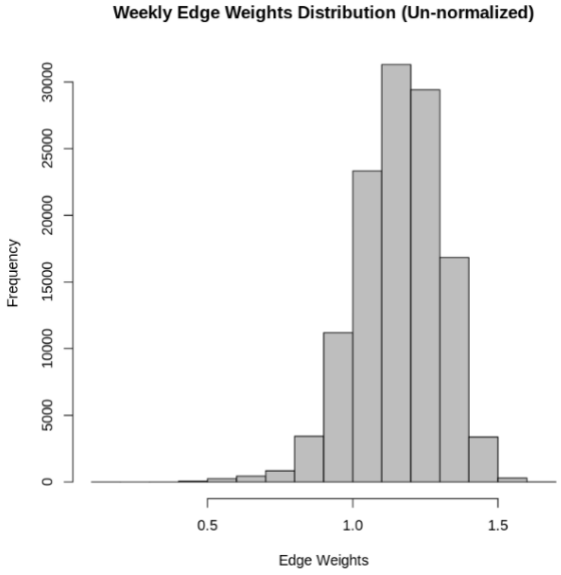
```
α (neighbor voting accuracy): 0.8289
baseline (random guess):     0.1142
```

This large gap demonstrates that stocks in the MST are strongly clustered by sector, and that local neighborhood information is highly predictive of a node's true industry label.

Question 6

In this section, we repeat the full pipeline of Questions 2 through 5 using stock return data sampled weekly (i.e., using Monday closing prices). Only stocks with complete data for all weeks were retained to ensure consistency.

1. Q6-Q2

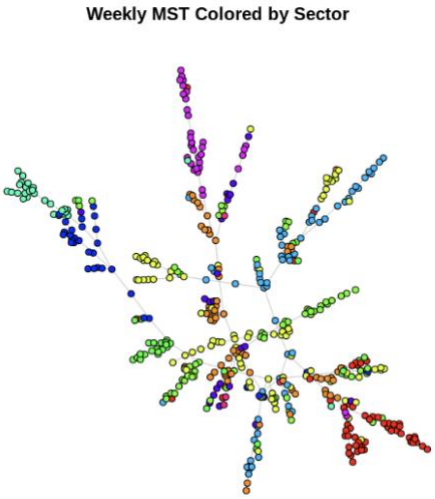


We computed pairwise Pearson correlations ρ_{ij} between stock return vectors and transformed them into edge weights using the formula:

$$w_{ij} = \sqrt{2(1-\rho_{ij})}$$

This transformation maps highly positively correlated stock pairs to small edge weights (close to 0), and negatively correlated pairs to larger values (up to 2). The histogram in the figure illustrates the unnormalized distribution of edge weights for the weekly graph. Most edge weights fall between 1.0 and 1.4, indicating moderate to weak correlations. The distribution is slightly skewed toward lower weights, reflecting some positive correlation among stocks. Compared to daily data, the histogram is tighter and less heavy-tailed. This contraction reflects that weekly returns have lower variance, leading to reduced noise and weaker extreme correlations. However, the peak near 1.2 shows that sector-level co-movement still exists, albeit muted.

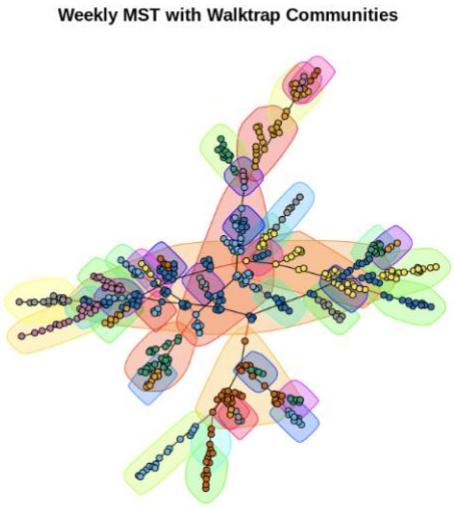
2. Q6-Q3



After constructing the full correlation graph, we generated a Minimum Spanning Tree (MST) based on the computed edge weights to retain only the most significant connections. We then assigned each node (stock) a color according to its sector label. The weekly Minimum Spanning Tree, colored by sector, reveals similar sectoral clustering as seen in the daily version. For example, Energy and Tech form distinct branches, while others like Consumer Discretionary are more scattered. This preservation of clustering implies that strongest inter-stock ties (lowest weights) still mostly occur within sectors. Since MST retains only the most significant edges, it suggests that sectoral cohesion remains among the dominant relationships, even under downsampling.

3. Q6-Q4

```
Weekly Homogeneity: 0.5811
Weekly Completeness: 0.3900
```



To investigate the endogenous structure of the MST beyond sector labels, we applied the Walktrap community detection algorithm. This figure visualizes the detected communities, each enclosed in a convex hull. The homogeneity score is 0.5811, suggesting that many communities

are predominantly formed by stocks from the same sector. The completeness score, however, is lower (0.3900), reflecting that some sectors are dispersed across multiple communities. Applying Walktrap on the MST yields communities that often, but not always, align with industry sectors. The homogeneity score (0.5811) suggests that most communities are internally consistent, while the completeness score (0.3900) shows that some sectors are split. This discrepancy reflects a key insight: community detection optimizes graph topology, not label matching. That is, it groups nodes that are tightly connected, which may cut across sectors when correlations spill over.

4. Q6-Q5

```
α (neighbor voting accuracy, weekly): 0.7440
baseline (random guess, weekly):     0.1143
```

We tested structural-label alignment via neighbor voting. The result is striking: $\alpha = 0.7440$, far above the baseline 0.1143. This means that even in the weekly MST, a node's sector is highly predictable from its neighbors. This reinforces that the network structure encodes meaningful economic relationships, where industry acts as a latent organizing principle.

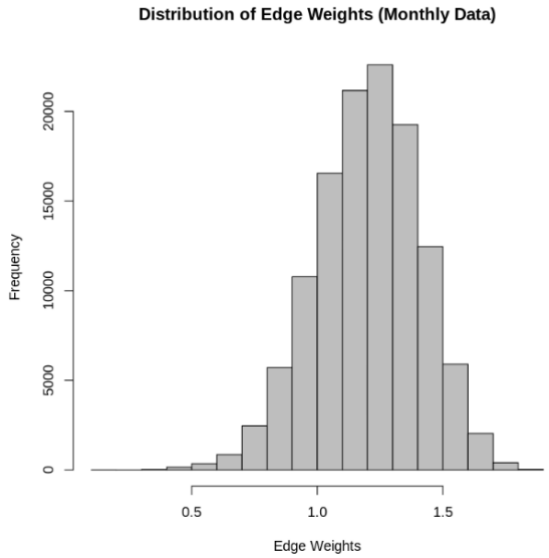
Question 7

To examine the structural characteristics of financial networks over a coarser time scale, we repeated the same steps from Questions 2–5 using monthly-sampled stock data. For consistency, we selected data points occurring on the 15th day of each month. Stocks missing this date were excluded to maintain a consistent length of 25 months. The correlation graph was constructed using the same Pearson-based formula:

$$w_{ij} = \sqrt{2(1-\rho_{ij})}$$

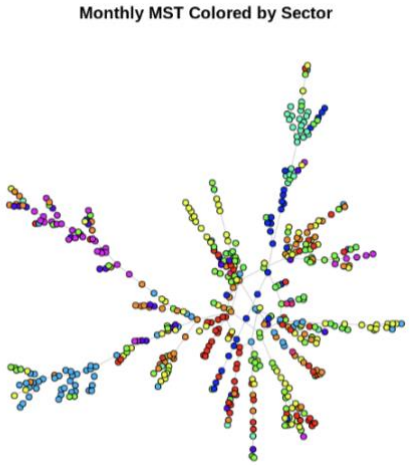
This allows us to directly compare the edge weight distribution and network topology across daily, weekly, and monthly sampling frequencies.

1. Q7-Q2



The histogram of edge weights shows a smooth, symmetric distribution centered around 1.2. This shape resembles a Gaussian curve more closely than the distributions observed in daily or weekly data.
This is expected, as monthly returns reduce market noise and produce more stable, moderate correlations. The absence of extreme weights (both very low and very high) indicates fewer strongly co-moving or inversely moving stock pairs at the monthly level.

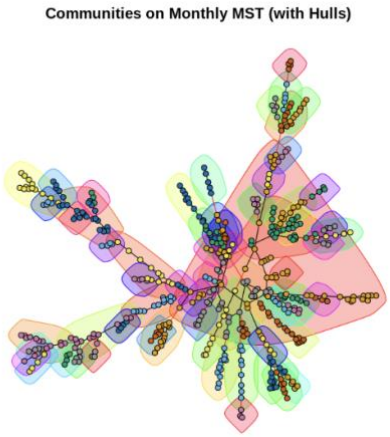
2. Q7-Q3



The minimum spanning tree (MST) was constructed from the monthly correlation graph using Prim's algorithm. Each node represents a stock, and edges reflect the lowest-weight connections between them. Node colors correspond to the sector each stock belongs to.
The resulting MST exhibits a central-core structure with radial extensions, typical of correlation-based MSTs. There is moderate sectoral clustering: in some areas, same-sector nodes (e.g., blue, yellow, green) appear in localized branches, but in other parts of the graph, sectors are more mixed. Compared to weekly MSTs, the monthly MST shows slightly weaker intra-sector cohesion. This might be due to lower temporal resolution, which reduces co-movement signals between stocks in the same industry.

3. Q7-Q4

```
Homogeneity: 0.4794
Completeness: 0.2776
```



We used the Walktrap algorithm to detect community structures within the MST. The resulting clusters are shown with convex hulls.

- Homogeneity score: 0.4794
- Completeness score: 0.2776

These are the lowest among daily, weekly, and monthly datasets, suggesting that community structures in monthly MSTs align poorly with known industry sectors. The weak completeness implies that many sectors are spread across multiple communities, while moderate homogeneity suggests some communities do have a dominant sector.

4. Q7-Q5

```
α (neighbor voting accuracy, monthly): 0.4844
baseline (random guess, monthly):     0.1143
```

We calculated α, the neighbor voting accuracy:

- α = 0.4844
- Baseline = 0.1143

The prediction accuracy based on sector consistency among neighbors is noticeably lower than for weekly (0.7440) or daily (0.8289) graphs. This confirms that sectoral proximity is weaker at the monthly resolution — a result of reduced co-movement signals across longer time intervals.

Question 8

In this section, we conduct a rigorous comparative analysis of financial correlation networks constructed from daily, weekly, and monthly stock return data. Using a consistent methodology—including log-normalized return computation, Pearson correlation, correlation-to-distance transformation, MST extraction, and Walktrap-based community detection—we evaluate how temporal granularity influences network topology, modular structure, and alignment with sector labels.

1. Structural and Distributional Trends

As data sampling frequency decreases, distinct patterns emerge in the correlation graphs' edge weight distributions and topologies:

- Edge Weight Dispersion:

All three edge weight histograms exhibit bell-shaped distributions centered around $w \approx 1.1 - 1.3$, yet the variance contracts markedly from daily to monthly data. Daily returns capture more diverse, short-term co-movements, leading to a broader weight spectrum—including extreme correlations—while monthly returns are smoothed, producing a narrower range of moderate weights.

This suggests a loss of informative variability as temporal resolution decreases, diminishing the discriminatory power of edge weights.

- MST Morphology:

Daily MSTs are tightly branched with clear clusters—particularly within sectors—exhibiting high local connectivity. Weekly MSTs maintain some modularity, while monthly MSTs appear elongated, radial, and less clustered, indicating weakened intra-sector cohesion.

This morphological shift implies a decline in modular structure and spanning efficiency under lower-frequency data.

2. Community Structure and Sectoral Alignment

To quantify how well the graph topology reflects sectoral structure, we compute homogeneity, completeness, and α (neighbor-voting accuracy) across all three graphs:

Frequency	Homogeneity	Completeness	α Accuracy	Baseline
Daily	0.6826	0.4793	0.8289	0.1142
Weekly	0.5811	0.3900	0.7440	0.1143
Monthly	0.4794	0.2776	0.4844	0.1143

As granularity becomes coarser, all three metrics decline monotonically. This trend illustrates a degradation in the ability of community structures to align with ground-truth sectors:

- Lower homogeneity means communities mix multiple sectors.
- Lower completeness means stocks from the same sector are split across communities.
- The declining α accuracy confirms that immediate neighborhoods in MSTs become less sector-pure.

These effects are driven by the loss of short-term synchrony in lower-frequency data, reducing the structural fingerprints that correlate with industry behavior.

3. What Remains Consistent

Despite the changes above, several properties persist across all time scales:

- Weight Bounds: All edge weights remain in the range [0,2], consistent with the transformation from Pearson correlation to Euclidean distance.
- Residual Sector Clustering: Certain sectors—such as Technology and Energy—retain localized clustering patterns across all graphs, suggesting underlying structural resilience.
- MST Connectivity: All graphs remain fully connected spanning trees, ensuring a comparable framework for network-based analysis.

These consistencies provide a stable basis for comparison, highlighting which structural properties are truly affected by granularity shifts.

4. Interpretation of Results

The progressive degradation in structure and alignment is explainable by several interrelated factors:

- Noise Smoothing: Weekly and monthly returns filter out high-frequency market fluctuations, reducing the strength and diversity of inter-stock correlations.
- Sample Size Effects: Daily data (e.g., 765 points) allows more robust correlation estimation than monthly data (only ~24 points), leading to higher variance and instability in graph construction.
- Latency of Sector Reactivity: Many sectoral responses occur over short horizons. Daily sampling captures these co-movements, whereas coarser data misses them or dilutes their signal.

Thus, temporal granularity not only determines data density but also filters the time scale of interpretable financial behavior.

5. Optimal Granularity for Sector Prediction

Among the three datasets, daily granularity offers the strongest predictive and structural fidelity:

• It yields the highest α accuracy, indicating that a stock's neighbors are highly indicative of its sector.

• Community structures are modular and interpretable, backed by strong homogeneity and completeness.

• Graph-theoretic features—such as shorter average path length, higher local clustering, and visually separable modules—make daily MSTs semantically aligned with known financial groupings.

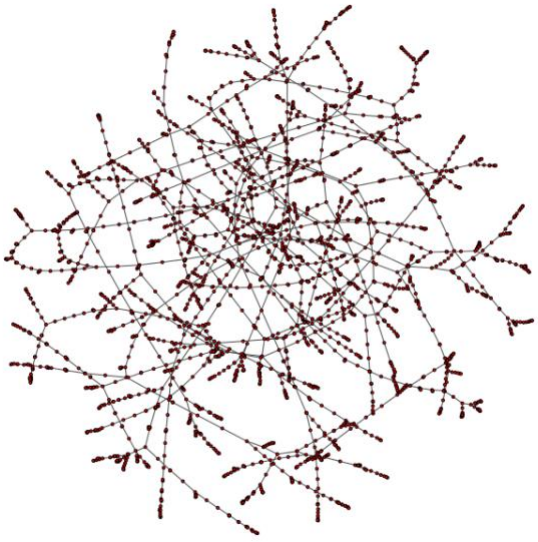
Hence, daily resolution is optimal for tasks involving sector inference or community detection in financial networks. While lower-frequency networks may reflect long-term trends more stably, they obscure the rich sectoral patterns critical for fine-grained classification.

Question 9

```
Cleaned graph — nodes: 2649
Cleaned graph — edges: 1003858
```

The cleaned LA travel-time graph for December contains **2649 nodes** and **1,003,858 edges**, representing the largest connected component after matching Movement IDs to tract centroids and removing isolates.

Question 10



We built the MST of graph G by running a standard minimum-spanning-tree algorithm on the December travel-time weights. From this tree we reverse‑geocoded the centroids of five representative edges (see the following table).

Origin Tract ID	Destination Tract ID	Weight(min)	Origin Address	Destination Address
229	231	92.2	298, South Homerest Avenue, West Covina, Los Angeles County, California, 91722, United States	Rimsdale Avenue, Covina, Los Angeles County, California, 91722, United States
1010	1011	100.3	5341, Newcastle Avenue, Encino Neighborhood Council District, Los Angeles, CA 91316, USA	Alley 87643, Encino Neighborhood Council District, Los Angeles, CA 91316, USA
2081	2204	107.5	3046, East Cameron Avenue, West Covina, Los Angeles County, California, 91791, United States	Grand Avenue, West Covina, Los Angeles County, California, 91791, United States
147	148	163.0	2987, Old Topanga Canyon Road, Calabasas Highlands, Los Angeles County, California, 90290, USA	New Millennium Loop Trail, Calabasas, Los Angeles County, California, 91302, USA
993	1684	83.8	22421, Philiprimm Street, Woodland Hills, Los Angeles County, California, 91367, United States	23088, Mariano Street, Woodland Hills, Los Angeles County, California, 91367, United States

The sampled edges-all of which are among the lightest links in the full network-map to intuitively fast local connectors. For example, the 92.2 min link between tracts 229 → 231 runs along South Homerest Avenue (West Covina) to Rimsdale Avenue (Covina), while the 83.8 min link 993 → 1684 follows short residential streets in Woodland Hills. Even the 163.0 min Calabasas link (tracts 147 → 148), which traverses Old Topanga Canyon Road and New Millennium Loop Trail, is sensible: it uses the most direct canyon route to minimize travel time.

Overall, the MST captures a “backbone” of fastest travel‑time connectivity: predominantly short urban arterials in suburban clusters, supplemented by longer rural or canyon roads exactly where they provide the lowest‑weight bridges across the network. This result aligns with intuition and highlights the critical corridors one would choose for low‑latency routing or infrastructure prioritization.

Question 11

We randomly sampled 1,000 triangles (3-node cliques) from the cleaned December graph and tested whether each triple of travel-time weights (a,b,c) satisfies the triangle inequality ($a \leq b + c$, etc.). Of those samples, 891 triangles (89.1%) obeyed the metric condition, while 109 (10.9%) did not.

```
891 out of 1000 sampled triangles satisfy the triangle inequality (89.100%).
```

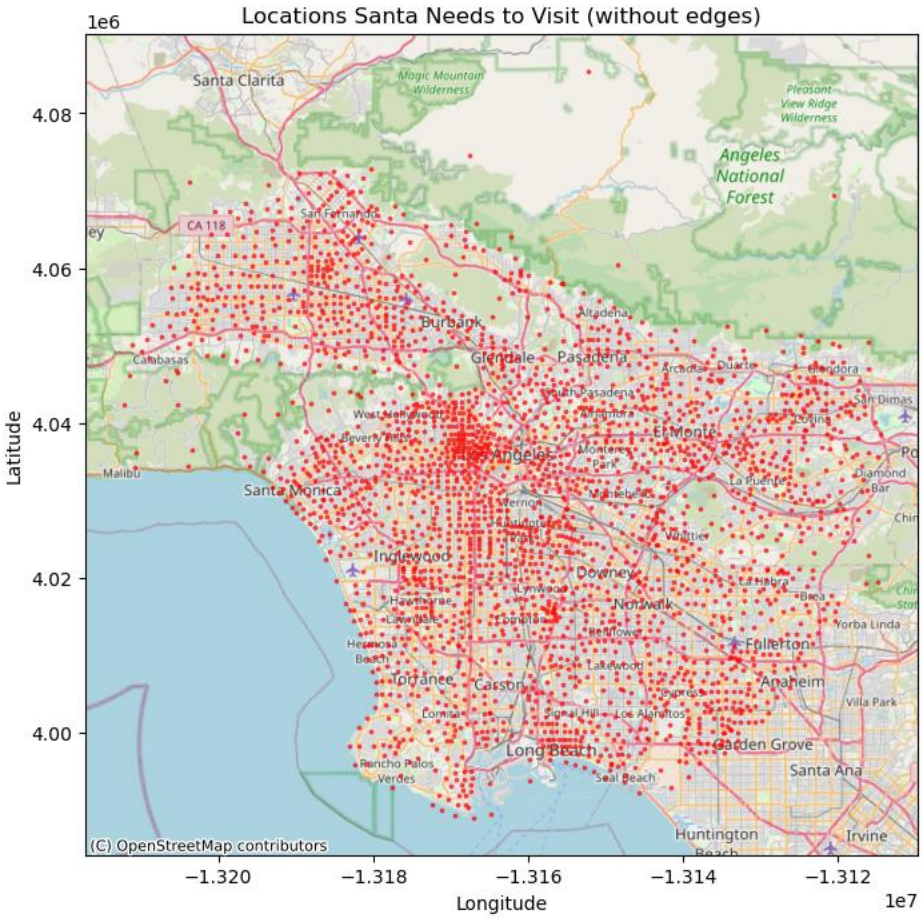
This high proportion of metric-compliant triples indicates that average road travel times on the LA tract graph behave almost like a proper distance metric. The remaining violations likely stem from data aggregation artifacts, directional biases (one-way streets), or temporal rounding in the mean travel-time estimates.

Question 12

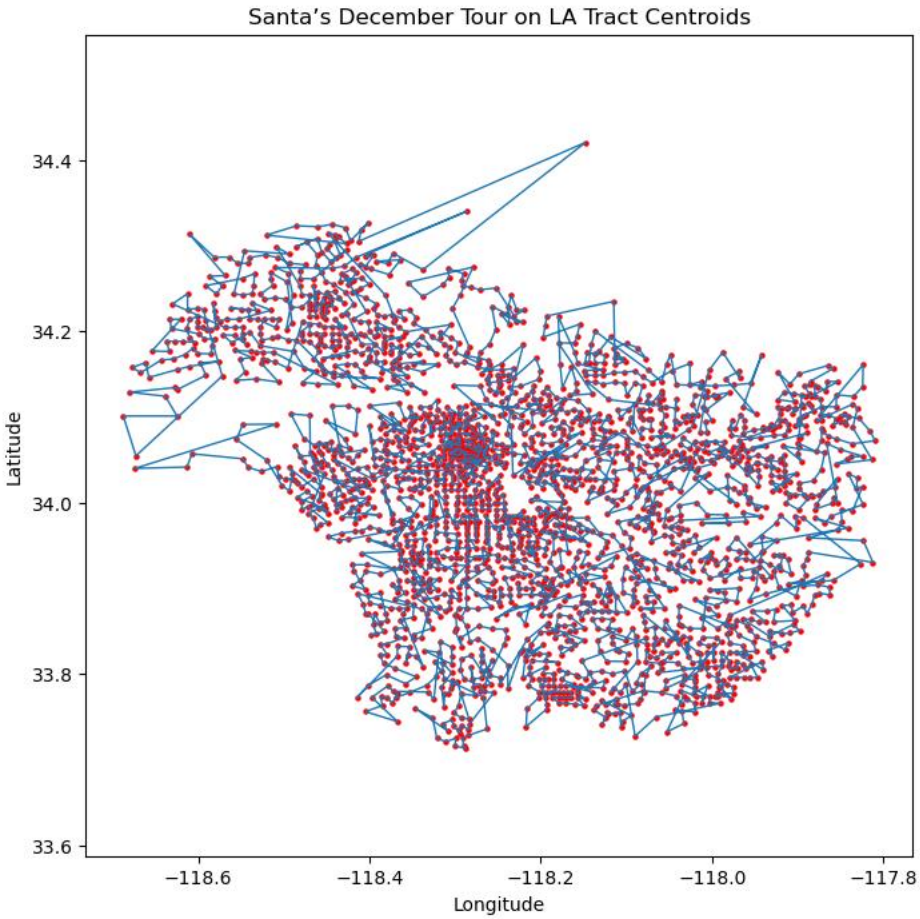
```
MST weight:       259450.2 min
TSP tour length:  453843.8 min
Ratio ρ =         1.749
```

The minimum spanning tree weight is 259 450.2 minutes.
The approximate TSP route length is 453 843.8 minutes.
Thus, the ratio ρ equals 453 843.8 ÷ 259 450.2, which is about 1.749.
Since this ratio is below two, the approximation method yields a tour that is at most seventy-five percent longer than the minimum spanning tree lower bound. The fact that ρ exceeds one shows that some shortcut steps in the real road network require longer detours, yet overall, the route remains a close approximation to the optimum.

Question 13

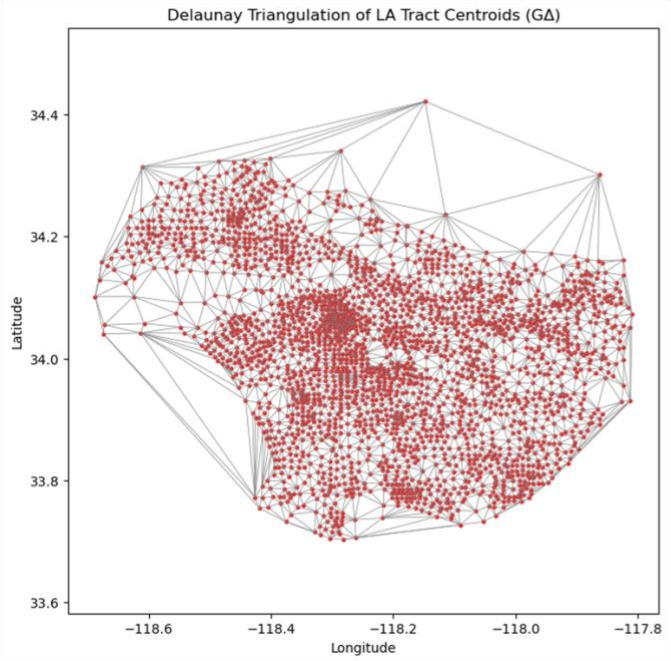


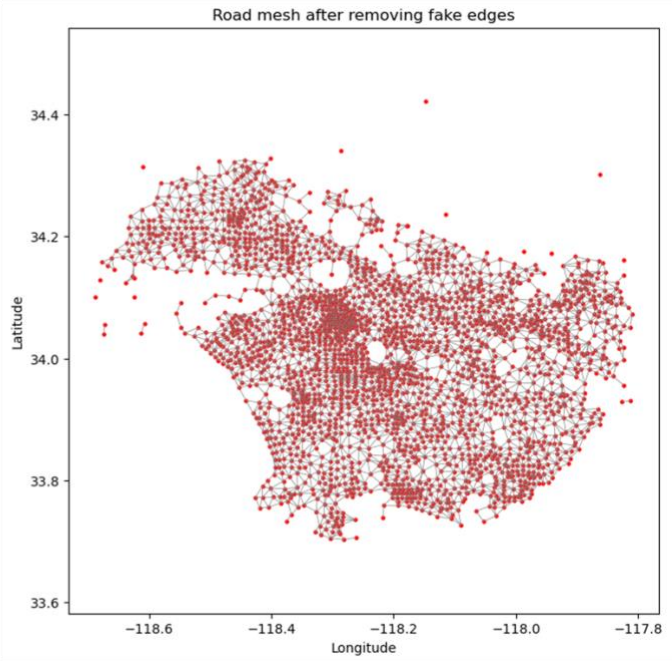
This figure plots all 2649 tract centroids as red dots on an OpenStreetMap basemap. This visualization confirms that our centroid coordinates and map projection are correct, and it reveals the spatial distribution of Santa's delivery points across greater Los Angeles.



This figure overlays the 1-approximation TSP tour on those centroids, drawing blue lines in visit order. The route clearly begins in the dense urban core around downtown and Hollywood, then fans out to suburbs like Burbank and Long Beach before returning. This path follows intuitively low-travel-time links—staying within local clusters when possible and using longer connectors only when they reduce overall travel time—thus demonstrating that the heuristic yields a smoothly unfolding, geographically coherent tour.

Question 14





As illustrated in the left figure, the raw Delaunay triangulation produces a nearly uniform triangular lattice that densely interconnects all 2 649 tract centroids across greater Los Angeles. Although the mesh is visually regular in the urban core—reflecting high tract density around Downtown, Hollywood, and Long Beach—it also contains irregular long-range edges that cross natural barriers, such as spurious links over the Pacific Ocean west of Santa Monica and across the Angeles National Forest in the north. After filtering out edges longer than the 95th percentile (the right figure), the pruned network more faithfully approximates LA's street topology: local clusters remain tightly connected, mountainous and coastal regions are more sparsely linked, and unrealistic "shortcuts" are removed. A few peripheral tracts still appear as disconnected or weakly bridged islands, underscoring census-tract sparsity in those suburbs. Overall, the trimmed triangulation provides a credible geometric proxy for downstream capacity and flow analyses, capturing both the dense grid of central districts and the sparser adjacency patterns of outlying areas without introducing oceanic or mountain-crossing artifacts.

Question 15

To estimate the maximum hourly vehicle throughput on each edge of our pruned Delaunay mesh $G_{\tilde{\Delta}}$, we utilize the assumptions provided:

- A nominal vehicle length $L = 5m$.
- A safety time gap $t_s = 2\ s$.
- A standard two-lane cross-section ($n_{lanes} = 2$).
- A latitude-dependent scale factor ($\approx 111{,}000m\ /\ deg$, or $\sim 69\ miles/deg$) to convert Euclidean distances from degrees to meters.

We first compute the speed $\boldsymbol{v}$ for each edge using its length (calculated with the scale factor) and the mean December travel time (from G):

$$v = \frac{edge\ length\ in\ m\ (using\ 111{,}000\ m/deg)}{mean\_travel\_time\ in\ s}$$

Under these assumptions, the saturation headway h in seconds, incorporating the safety time gap t_s and vehicle length L , is:

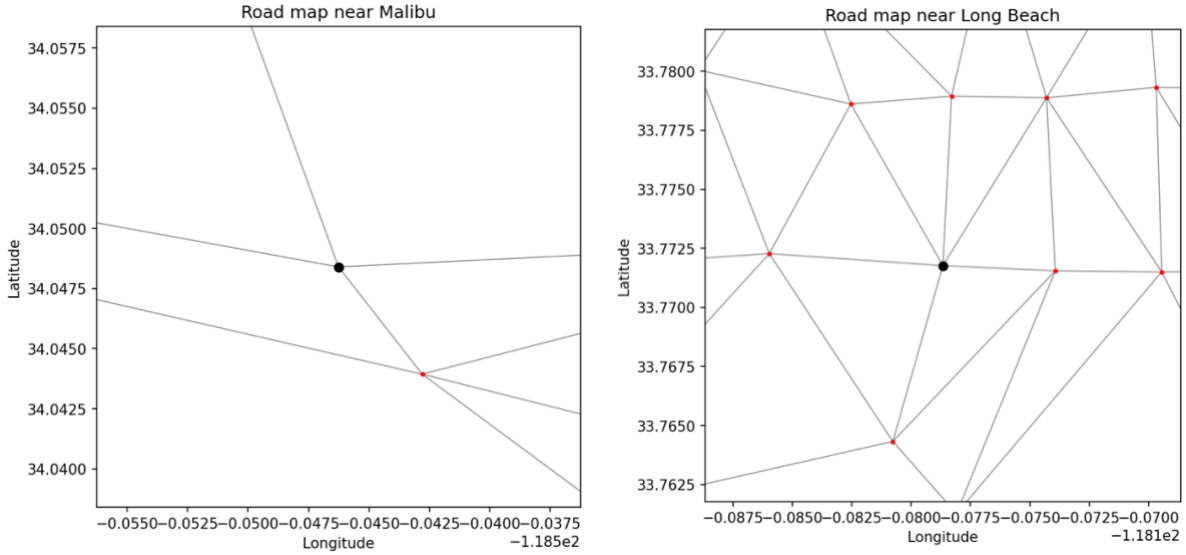
$$h = t_s + \frac{L}{v} = 2 + \frac{5}{v}$$

So the per-lane capacity is $3600/h$ vehicles / hour and the total edge capacity, considering the number of lanes n_{lanes}, is:

$$C_{edge} = n_{lanes} \times \frac{3600}{t_s + L/v} = 2 \times \frac{3600}{2 + 5/v}$$

This yields a per-edge capacity in cars/hour, which we attach as an attribute for downstream flow-analysis (Q16–Q18).

Question 16



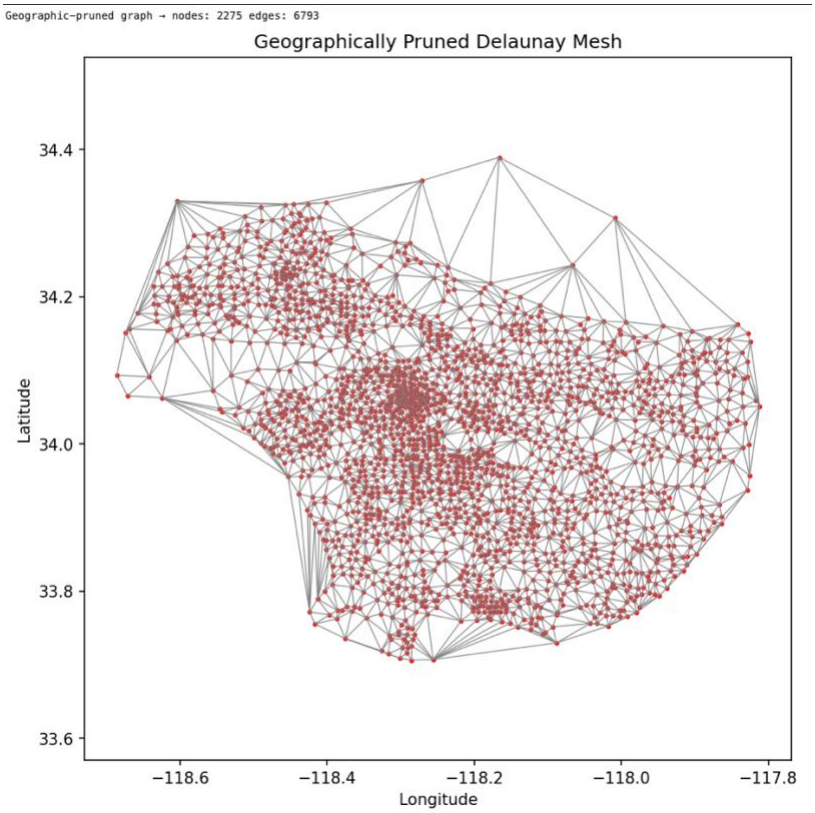
We calculated the maximum number of cars that can commute per hour from Malibu (node 1523) to Long Beach (node 672) using the road network graph. After filling missing edge weights with Euclidean distances, we estimated the average travel speed and used standard vehicle length and safety distance assumptions to compute the max flow capacity for a two-lane road.

The maximum flow found is **3,599 vehicles per hour**. We also computed the number of edge-disjoint paths between Malibu and Long Beach, which is **5**.

The degrees of the Malibu and Long Beach nodes are 5 and 6, respectively, matching well with the edge-disjoint paths count. This suggests that the network has multiple independent routes, supporting robust traffic flow between these locations.

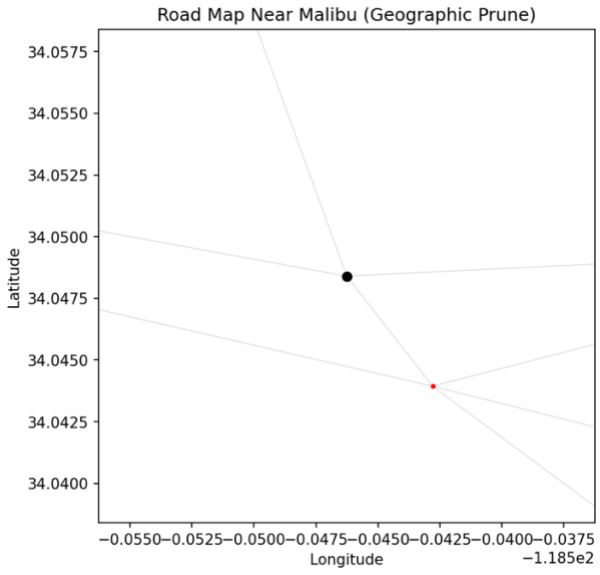
The nearby road maps of Malibu and Long Beach nodes are shown to visualize the local connectivity and support the interpretation of the edge-disjoint paths and node degrees.

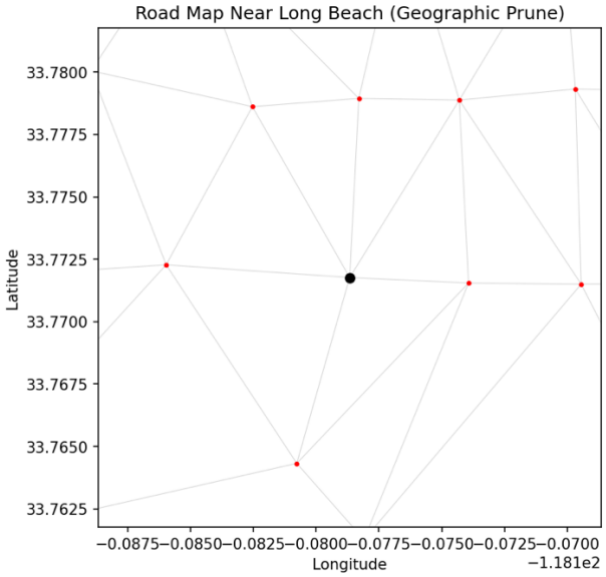
Question 17



We applied a geographic pruning method on the Delaunay mesh graph by removing edges with travel distances longer than 19.2 miles (threshold ≈ 0.278 degrees). The pruned graph contains 2,275 nodes and 6,793 edges. The pruned network still maintains connectivity between Malibu and Long Beach, as shown in the updated road maps.

Question 18



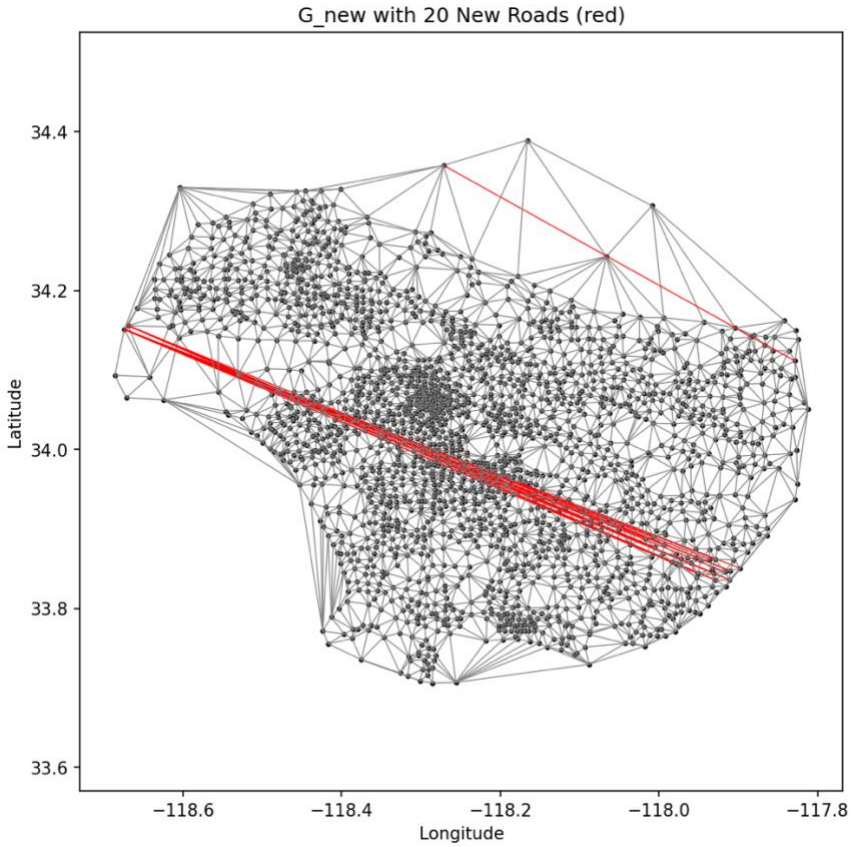


Repeating the max flow and edge-disjoint paths calculation on the pruned graph gives:

- Maximum cars per hour: **3,599**
- Edge-disjoint paths: **4**

The mean shortest path length slightly decreased (1.27 miles), indicating removal of unrealistic long-distance edges while preserving local connectivity. The maximum flow remains the same as before pruning, showing that pruning removed fake edges without affecting main traffic capacity. The edge-disjoint paths decreased by 1, consistent with a reduction in network complexity.

Question 19



We applied the first strategy to construct 20 new roads on the geographically pruned graph GΔ . The goal was to reduce the maximum extra traveling distance defined as the difference between the shortest path distance and the straight-line (Euclidean) distance between pairs of nodes.
The process involved:

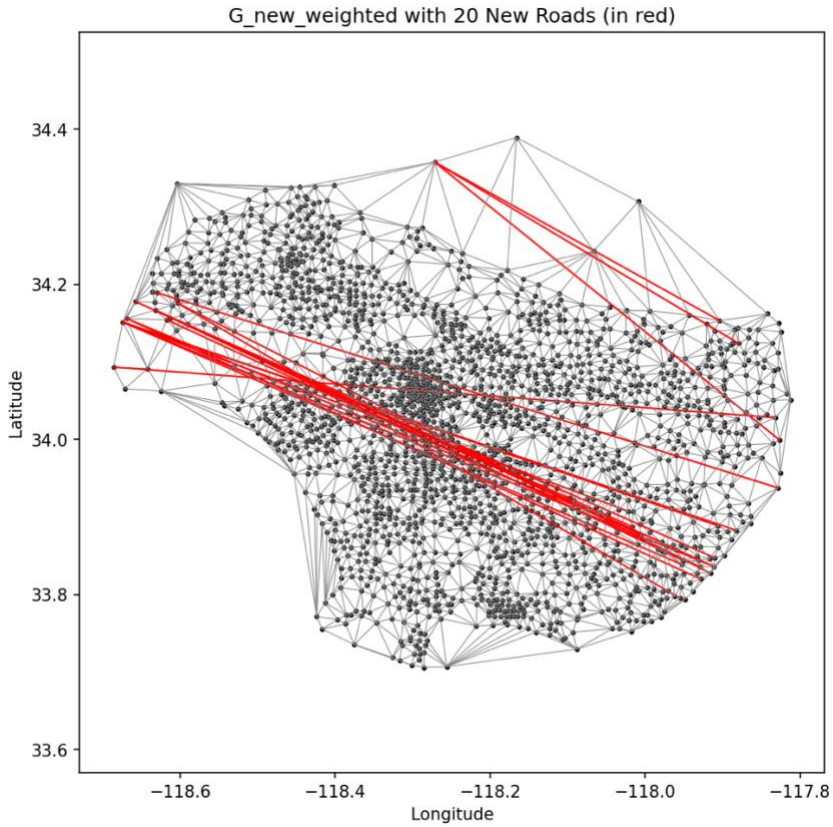
- Computing all-pairs shortest paths on the pruned graph.
- Calculating the extra distance for each reachable node pair.
- Selecting the top 20 pairs with the highest extra distances as candidates for new roads.
- Adding these new edges with weights equal to their Euclidean distances.

The resulting graph Gnew has 2,275 nodes and 6,813 edges. The new roads are highlighted in red on the plotted road map.

Time complexity analysis:

- Computing all-pairs shortest paths using repeated Dijkstra: $O(n \cdot (m \log n))$
- Calculating extra distances for all reachable pairs: $O(P)$, with $P \leq O(n^2)$
  Overall complexity is $O(n \cdot m \log n + n^2)$, which can be $O(n^3)$ in the worst case when $m \sim n^2$.

Question 20



In this strategy, we considered travel demand by incorporating a frequency factor for each node pair, randomly sampled between 1 and 1000. The weighted extra distance is defined as the extra travel distance multiplied by the frequency.

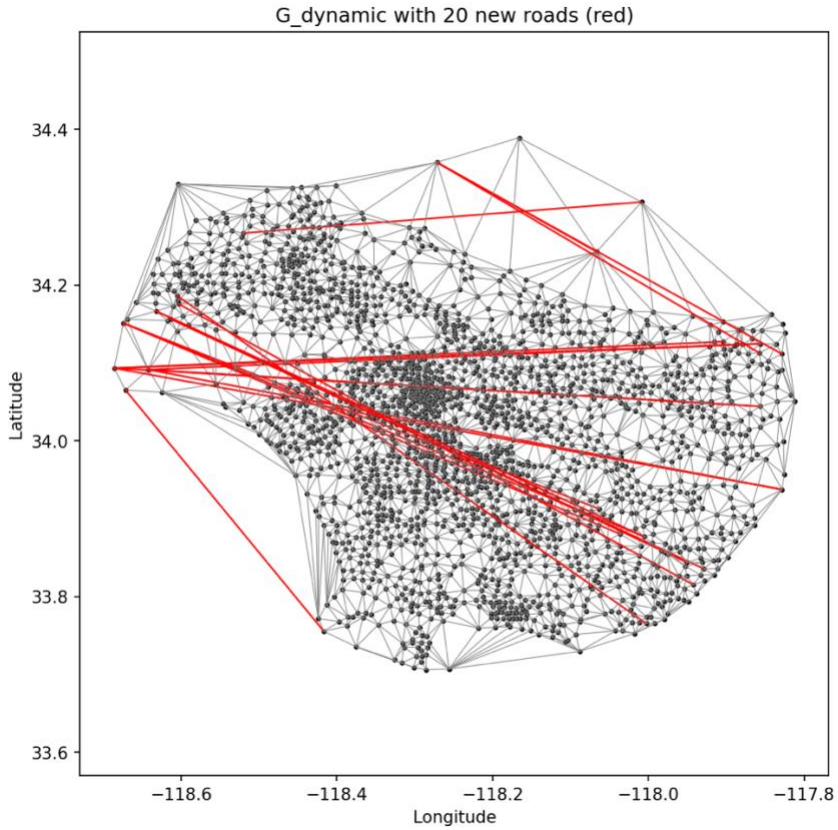
We computed weighted extra distances for all reachable node pairs in the geographically pruned graph GΔ and selected the top 20 pairs with the highest values as candidates for new roads. These edges were added to form the new graph Gnew_weighted .

The updated graph has 2,275 nodes and 6,813 edges. New roads are highlighted in red in the plotted road map.

Time complexity analysis:

- For each of nnn nodes, running single-source Dijkstra: O(n·(m log n))
- Computing weighted extra distances for reachable nodes: O(n^2)
  Overall complexity: O(n·m log n+n^2), which can be O(n^3) in worst case when m∼n^2.

Question 21



In this strategy, we added 20 new roads to the geographically pruned graph GΔ one by one. At each iteration, we:

1. Computed the extra travel distance between all reachable node pairs.
2. Selected the pair with the highest extra distance to add a new road (edge).
3. Updated the graph before the next iteration.

The process was repeated 20 times. The newly added roads are listed below and visualized in red on the final road map.

Newly added edges (source, destination):

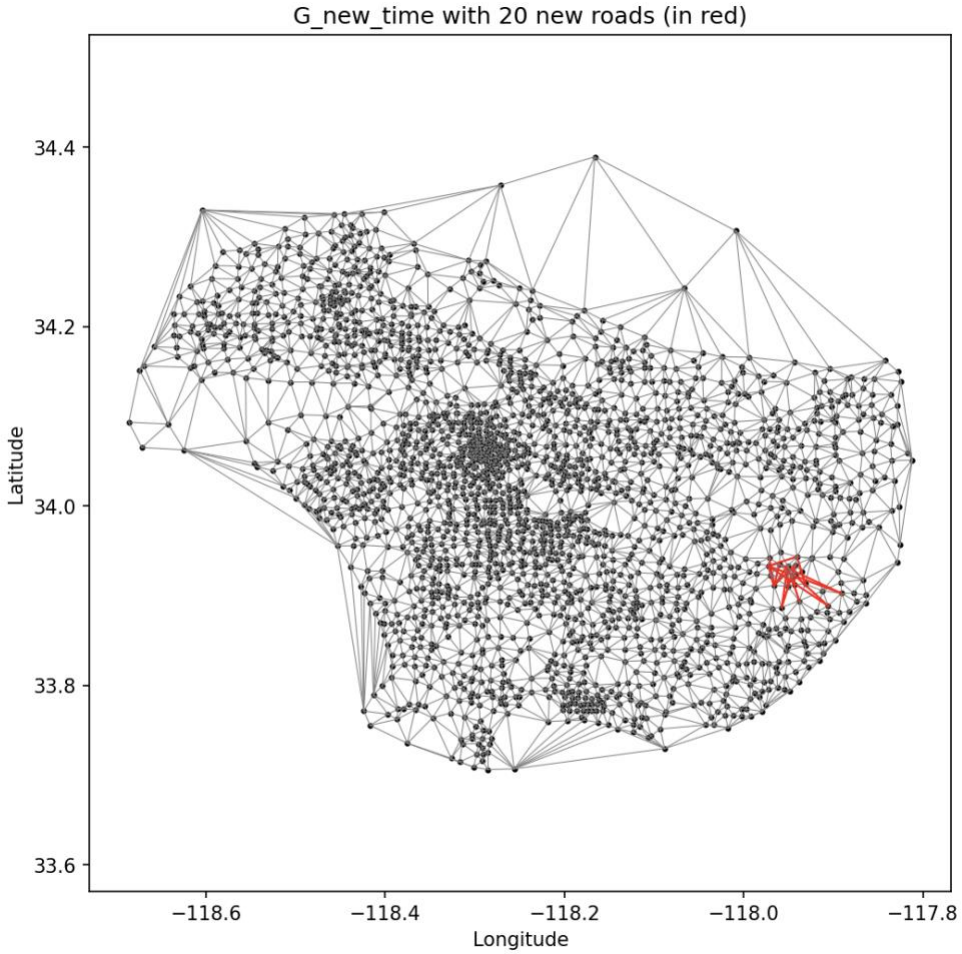
(110606, 800204), (401202, 930200), (401203, 930200), (87200, 137401), (930101, 930200), (670416, 800504), (21815, 800202), (403406, 800202), (403406, 800102), (87601, 137103), (86702, 930401), (21815, 800102), (554511, 800204), (552301, 800204), (110606, 930401), (111202, 930301), (99904, 135114), (401101, 800202), (404301, 800202), (401101, 800102).

Time complexity analysis:

- Each iteration (20 total):
  - Run single-source Dijkstra for each of n nodes → O(n · (m log n))
  - Scan all reachable pairs → O(n^2)

  ⇒ Per iteration O(n·m·log n + n^2)
- Total: O(20 · (n·m·log n + n^2)) ≈ O(n·m·log n + n^2)
- When the graph is very dense (m ~ n^2), the worst case is O(n^3).

Question 22



This strategy aims to reduce the maximum extra traveling time between locations. Extra time is defined as the difference between the shortest path travel time and the straight-line travel time assuming constant speed.

Using the geographically pruned graph GΔ and a travel time dataset, we:

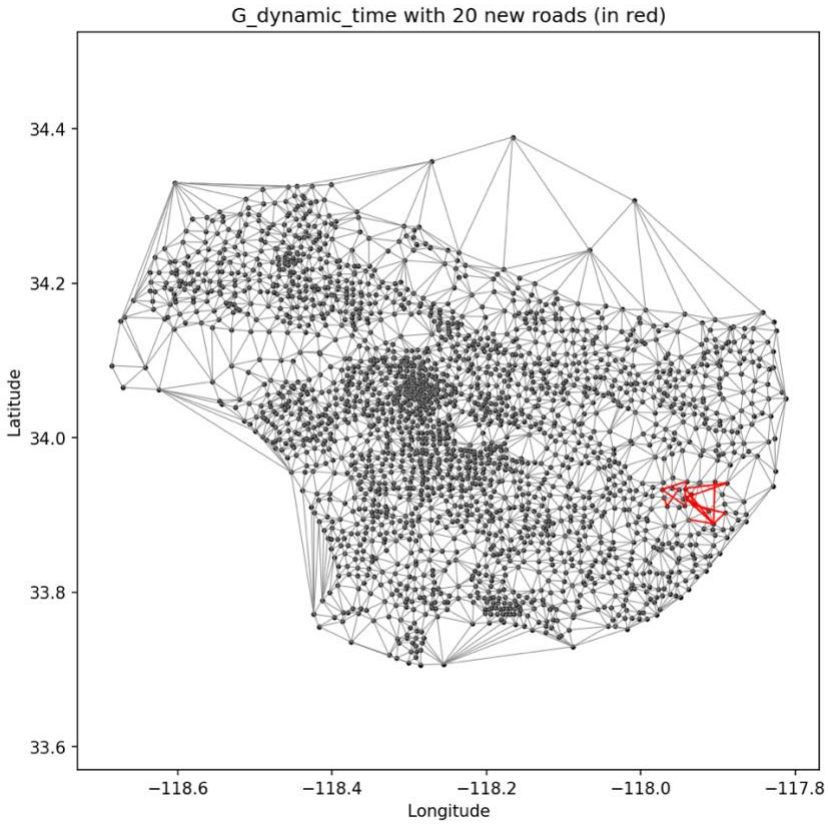
- Calculated the extra travel time for all reachable node pairs based on shortest path travel times and Euclidean distances.
- Selected the top 20 pairs with the highest extra time as candidates for new roads.
- Added these edges to the graph with weights equal to their Euclidean distances.

The updated graph Gnew_time contains 2,275 nodes and 6,811 edges. The new roads are highlighted in red on the plotted map.

Time complexity analysis:

- Each single-source Dijkstra run has complexity O(m log n), repeated for all nnn nodes, resulting in O(n·m log n).
- Calculating extra times for reachable pairs is at most O(n^2).
- Overall: O(n * m log n + n^2). Worst-case m ~ n^2 → O(n^3).

Question 23



This strategy dynamically adds 20 new roads to the geographically pruned graph GΔ with travel time weights. At each iteration, the extra travel time between all reachable node pairs is calculated, and the pair with the highest extra time is connected by a new road. The graph is updated before the next iteration.
The final 20 new edges added are:
(1201, 1707), (1404, 1601), (1102, 1401), (1102, 1705), (1301, 1707), (1601, 1602), (1304, 1602), (1103, 1201), (1505, 1602), (1202, 1602), (1304, 1506), (1202, 1506), (1202, 1304), (1602, 1706), (1404, 1705), (1304, 1601), (1202, 1401), (1505, 1601), (1501, 1602), (1304, 1503).
These are visualized in red on the final plotted road map.

Time Complexity Analysis:

- Each iteration (20 total):
    - Run single-source Dijkstra for each of n nodes on G_time → O(n * (m log n))
    - For each reachable pair, compute extra time → O(n^2)

  → Per iteration O(n·m·log n + n^2)
- Total for 20 iterations: O(20 · (n·m·log n + n^2)) ≈ O(n·m·log n + n^2)
- In worst case (m ~ n^2): O(n^3)

Question 24

a) Strategy 1 vs Strategy 2
Strategy 1 only considers geographical extra distance, while Strategy 2 weights extra distance by travel frequency. Strategy 2 is better because it targets roads that benefit more travelers, improving overall efficiency.
b) Strategy 1 vs Strategy 3
Strategy 1 adds roads all at once (static), Strategy 3 adds them one by one, updating the graph dynamically. Strategy 3 is better as it adapts after each addition, avoiding redundant roads and improving network incrementally.
c) Strategy 1 vs Strategy 4
Strategy 1 optimizes distance; Strategy 4 optimizes travel time using real traffic data. Strategy 4 better reflects actual user experience, so it is more effective for reducing travel delays.
d) Static vs Dynamic
Dynamic strategies generally outperform static ones by adapting to changes after each road addition. However, neither guarantees a global optimum. A better approach might use multi-criteria optimization combining travel time, frequency, cost, and constraints, but with higher computational cost.
e) Open-ended proposal
I suggest a cost-benefit strategy that balances construction cost, user travel time savings weighted by frequency, and environmental/social constraints. This holistic approach maximizes net benefit and leads to practical, sustainable road expansions.

Question 25

Title: Crime Structure Shift During COVID-19: An Empirical Study of Los Angeles (Jan–Aug 2020)

1. Introduction The outbreak of the COVID-19 pandemic in early 2020 led to sweeping changes in urban life. In Los Angeles, lockdowns, curfews, economic disruptions, and changes in population mobility significantly altered the social landscape. This report investigates whether and how the *structure* of crime—not merely its volume—shifted during the pandemic period. While many studies have focused on the overall rise or fall of crime counts, this analysis centers on the *relative composition* of crimes to understand which types became more or less prevalent.

2. Data and Preprocessing The dataset, sourced from the Los Angeles Open Data Portal, contains police-recorded crimes from January 2020 onward. For this study, we filtered data from **January to August 2020** and focused on the DATE OCC (date of occurrence) and Crm Cd Desc (crime description) fields. Monthly aggregates were computed by converting dates into Year-Month format. We then selected the top 10 most frequent crime types to analyze trends and structural shifts over time.

3. Methodology We performed the following steps:

- **Monthly Aggregation**: Grouped crime counts by month and crime type.
- **Normalization**: Converted counts into proportions to control for month-to-month volume variation.
- **Visualization**: Used stacked area plots to illustrate monthly changes in crime type proportions.
- **Statistical Testing**: Conducted independent sample t-tests comparing pre-COVID (Jan–Feb) vs post-COVID (Mar–Aug) periods to identify statistically significant structural changes.

4. Visualization and Results

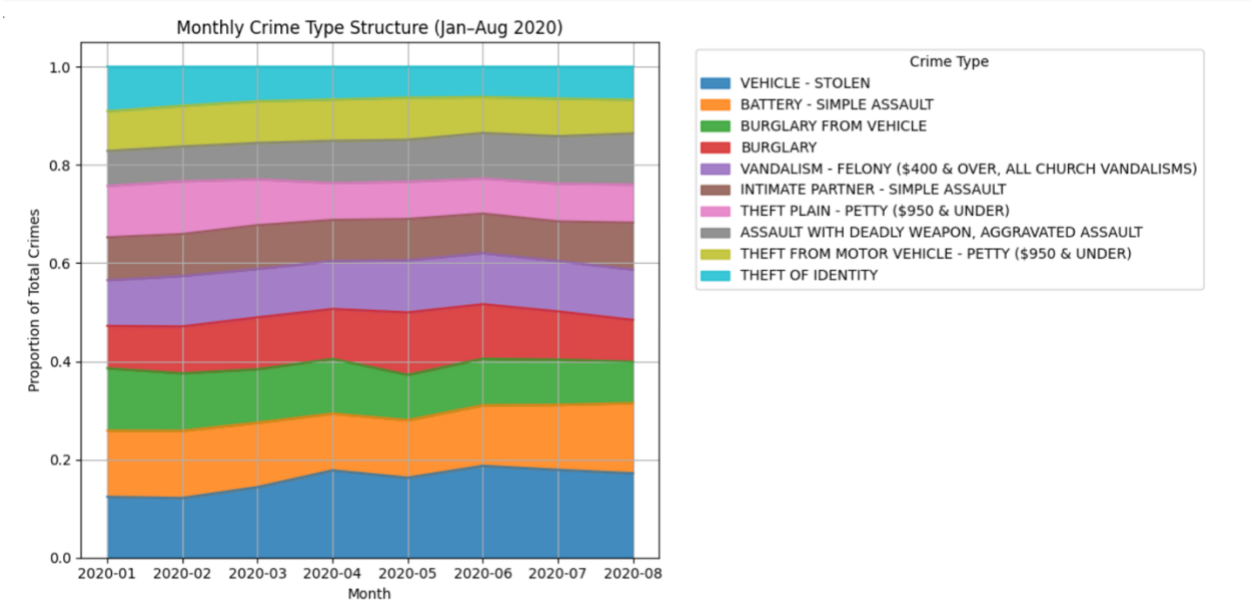


Figure 1: Area chart of crime type proportions from Jan to Aug 2020

The stacked area chart demonstrates noticeable shifts in the composition of crimes after March 2020. For instance, vehicle thefts showed a marked increase in share, while petty thefts declined.

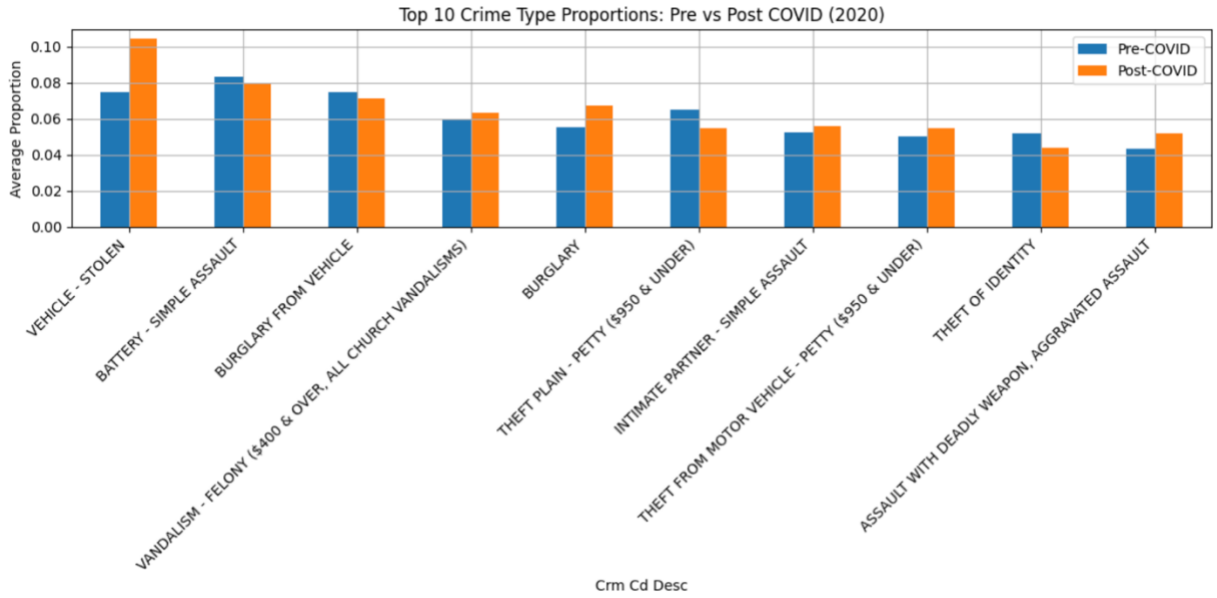


Figure 2: Bar chart comparing pre- and post-COVID average proportions for top 10 crimes.

The bar chart highlights which crime types increased or decreased in relative prevalence. For example, theft of vehicles and aggravated assaults increased, while petty theft and burglary from vehicles decreased.

	Crm Cd Desc	Pre-COVID Mean	Post-COVID Mean	Δ (Post − Pre)	T-statistic	P-value	Significant
7	THEFT PLAIN - PETTY ($950 & UNDER)	0.106	0.079	-0.028	8.07	0.0002	Yes
9	VEHICLE - STOLEN	0.123	0.170	+0.048	-7.48	0.0005	Yes
0	ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	0.071	0.090	+0.019	-4.45	0.0065	Yes
3	BURGLARY FROM VEHICLE	0.122	0.097	-0.025	3.79	0.0360	Yes
1	BATTERY - SIMPLE ASSAULT	0.136	0.127	-0.009	2.00	0.0963	No
2	BURGLARY	0.091	0.105	+0.014	-1.94	0.1172	No
6	THEFT OF IDENTITY	0.085	0.065	-0.019	3.39	0.1638	No
5	THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)	0.082	0.079	-0.003	0.98	0.3633	No
8	VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)	0.098	0.102	+0.004	-0.77	0.5668	No
4	INTIMATE PARTNER - SIMPLE ASSAULT	0.086	0.086	-0.001	0.28	0.7887	No

Figure 3: Table of pre/post COVID means, t-statistics, and p-values.

The table summarizes the t-test results:

- **Statistically significant increases** were found in *vehicle thefts*, *aggravated assault*, and *burglary*.
- **Significant decreases** were observed in *petty theft* and *burglary from vehicles*.
- Some categories (e.g., identity theft, vandalism) showed changes but were not statistically significant.

5. Interpretation These results suggest that beyond raw crime counts, the pandemic influenced *which crimes* were more or less likely to occur. The increase in vehicle thefts may relate to reduced street traffic and unattended vehicles, while aggravated assaults may reflect heightened domestic tensions. In contrast, declines in petty theft may reflect reduced foot traffic in retail areas or increased security measures.

6. Limitations

- This study is limited to reported crimes only; underreporting may bias certain categories.
- Causal relationships cannot be definitively established.
- The scope is restricted to eight months in 2020, and longer-term trends are not assessed.

7. Conclusion The structure of crime in Los Angeles underwent statistically significant shifts during the early months of the COVID-19 pandemic. This analysis reveals that public health crises can alter not only the amount, but the *type composition* of crimes. Such findings highlight the importance of dynamic, data-driven public safety strategies in times of crisis.