ECE232E Project 5

Graph Algorithms

Team Members:

Hongyan Gu – 205025476

Jiawei Du - 404943853

Xin Liu – 505037053

Aoxuan (Douglas) Li - 905027231

Part 1 Stock Market

In this part, we studied the correlation structure the among fluctuation patterns of stock prices using tools from graph theory.

Question 1

To use log-normalized return, we could have the following advantages:

(1) Sometimes, we could assume the prices are distributed log normally, and as a result, $\log(1 + q_i(t))$ is conveniently normally distributed, because:

$$1 + q_i(t) = \frac{p_i(t)}{p_i(t-1)}$$

This is handy given much of classic statistics presumes normality.

(2) When the return is very small (and indeed it does in our project), the log-normalized one ensures that it is very close to the raw return:

$$\log\big(1 + q_i(t)\big) \approx q_i(t)$$

(3) The log-normalized return holds with time-additivity. For calculating compounding return, we need to calculate the sequence over time. If we use unnormalized one, we got it not smooth and clear:

$$(1 + r_1)(1 + r_2) \cdots (1 + r_n) = \prod_{i=1}^{n} r_i$$

But when we turn it into log-normalized one, one could get a clear summation:

$$\sum_{i=1}^{n} \log(1 + r_i) = \ log(1 + r_1) + log(1 + r_2) + \cdots + log(1 + r_n) = log\left(\prod_{i=1}^{n}(1 + r_i)\right)$$

(4) It is mathematically convenient regrading to calculus. As we know,

$$e^x = \int e^x dx = \frac{d}{dx} e^x$$

This identity is tremendously useful, as much of financial mathematics is built upon continuous time stochastic processes which rely heavily upon integration and differentiation.

(5) It is numerically steady. Addition of small numbers is numerically safe, while multiplying small numbers is not as it is subject to arithmetic underflow. For many interesting problems, this is a serious

potential problem. To solve this, either the algorithm must be modified to be numerically robust or it can be transformed into a numerically safe summation via logs.

The upper and lower bound for ρ_{ij} is 0.9884238 and -0.1985732.

Question 2

In this question, we constructed a graph using edge weights correlation given by the equation listed below.

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

We derived a correlation list containing 494 nodes, and an edge list with cardinality of 121771. The distribution of degree od the graph is plotted in Fig. 1.1. From the plot we can see that all the nodes in the graph have a degree of 493, which means the graph is a fully connected graph, and each node connect to each other in the graph.

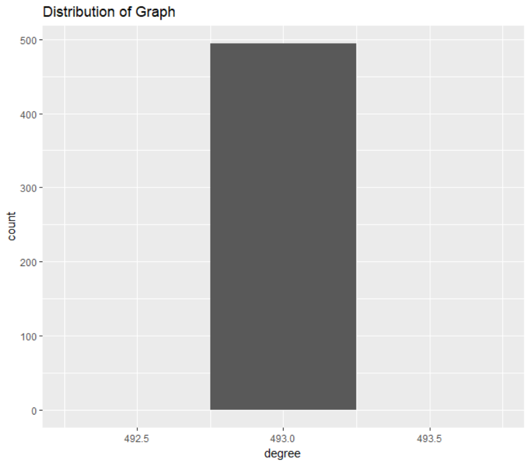

Fig. 1.1 Degree distribution of the graph

Fig. 2 shows the distribution of unnormalized weights. First, we would like to justify whether the data matches normal distribution. We used Shapiro-Wilk normality test, and the result shows that W = 0.96451, p-value < 2.2e-16. Although W → 1, p is not significant enough, so there is not sufficient proof to show the data match normal distribution. (Another interesting finding is that the log-normalized data don’t match normal distribution either.)

Fig.1. 2 shows that there’s a long tail in the left of the distribution, and the majority of the data lie between (1.0, 1.4). The distribution may show that, in the real stock market, the weight correlation would be mostly likely lies in (1.0, 1.4). If not, they are more likely to have a smaller weight than a larger one.

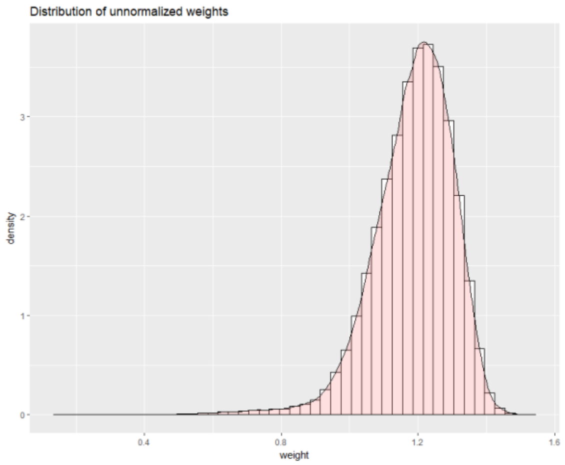

Fig. 1.2 Distribution of unnormalized weights

Question 3

In this question, we searched the MST of the graph and drew the plots. There are 11 different sectors in the MST, and we drew them with different colors on the nodes. We first drew it with a cluster layout in the first picture, and with a tree layout in the second picture. The two plots are shown in Fig 1.3.

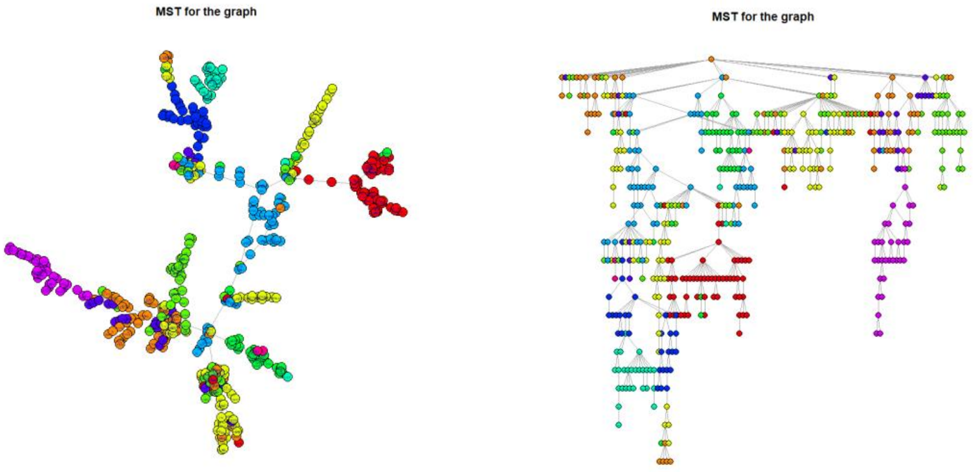

Fig. 1.3 Cluster Layout for MST (left) and Tree layout for MST

From the plot, we can see that the parent node and child node is likely to be in the same sector (painted in the same color in Fig. 1.3). If we extract sub-trees in the graph, chances are great that all the nodes in this subgraph is in the same category. From the equation $w_{ij} = \sqrt{2(1-\rho_{ij})}$, we can note that w_{ij} is decreasing as ρ_{ij} is increasing, which means two nodes that have higher correlation ρ_{ij} would have a smaller weight w_{ij}, and accordingly, two nodes that are in the same sector would have a smaller weight w_{ij}. Thus, MST of the weighted graph would have clustering effect on node by sectors.

Question 4

In this question, we calculated the alpha for the two cases for different ways of calculating possibilities for a node belonging to a sector. The results are shown in Table. 1.1

Table.1.1 Alpha values for two cases

	Alpha for first case	**Alpha for second case**
Value	0.8289301	0.1141881

From table 1.1, we can see that, the α for the first case is much larger than the second case. This difference is majorly devoted by the calculation difference of $P(v_i \in S_i)$. For method 2, P is calculated by dividing the cardinality of V, which makes $P(v_i \in S_i)$ smaller. Consider the first case, the α is 0.83, which is a good result indicating that the majority of the neighbors of a node in the MST belongs to the same sector of a node, and that MST has a good clustering effect.

Question 5

In this question, unlike the previous parts, we only used the data for Monday of each data file, as we wanted to figure out the features in a week sense. Similar to what we did in question 4, we drew the MST in two kinds of layout by only using the data of Monday. We also calculated α using the method mentioned in Question 4. The result is shown in Table 1.2. Compared with both parameters and MST structures, the α using the first method is smaller in this question and the MST clustering in Fig. 1.5 is not as obvious as than in Fig. 1.6. A potential explanation is that, the trading of stock market can be done as "t+0", which means a broker can buy and sell a stock in the same day, and the stock market can be influenced very quickly. A weekly summary may be a bit long for observing the fluctuation in the stock market, and thus the clustering effect from weekly data is not as good as that from daily.

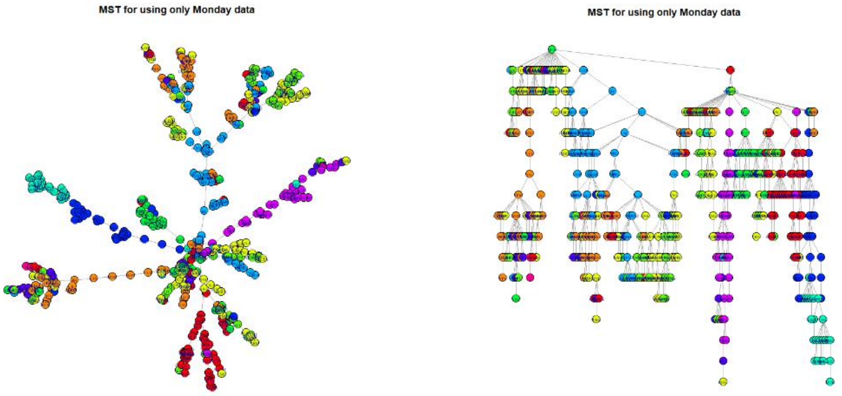
MST for using only Monday data
MST for using only Monday data

Fig. 1.5 MST with cluster layout for data only using Monday (left) and MST with tree layout for data only using Monday (right)

Table.1.2 Alpha values for two cases

	Alpha for first case	Alpha for second case
Value	0.6954122	0.1141881

Part 2 Let's Help Santa!

In this part, we explored the 2-approximation algorithm of classical NP-Hard Problem---Travelling Salesman Problem(TSP). The dataset is real statistical data in 2017 Quarter 4 for San Francisco area, which is provided by Uber.

Question 6

The remaining graph after cleaning is called G.

The number of nodes in G is **1880**.

The number of edges in G is **311802**.

Question 7

The information regarding the endpoints of five edges selected from total 1879 edges in Minimum Spanning Tree of G is described in Table 2.1.

Table 2.1 Street Address of Five Edges in MST of G

Edge Number	Street Address-1	Street Address-2
1	400 Northumberland Avenue, Redwood Oaks, Redwood City	1500 Oxford Street, Palm Park, Redwood City
2	3200 Nightingale Drive, Modesto	2300 Pamela Lane, Modesto
3	900 Sutter Street, Lower Nob Hill, San Francisco	200 Myrtle Street, Tenderloin, San Francisco
4	2200 Central Avenue, Alameda	2000 Clement Avenue, Alameda
5	300 Ruth Avenue, Monta Loma,	300 Tioga Court, Greenmeadow,

Mountain View	Palo Alto

We can tell the results are fairly reasonable and intuitive, for the addresses of endpoints are basically located in the same or nearby region.

In Figure 2.1, the MST is also visualized on Google Map for better illustration.

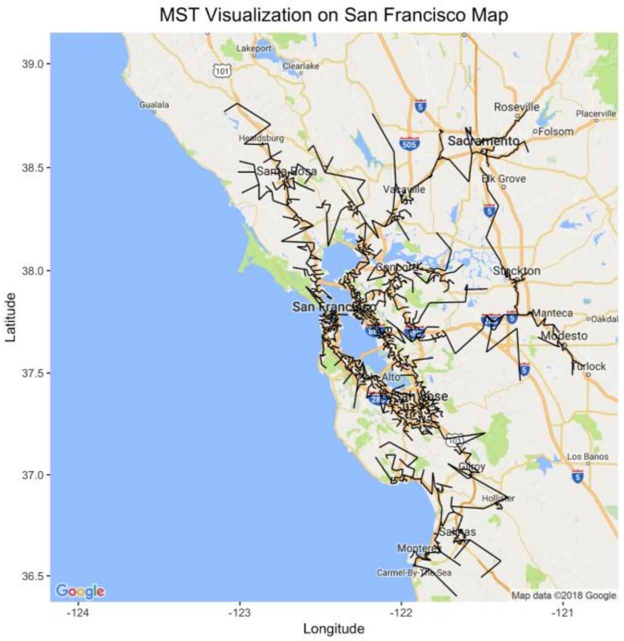

Figure 2.1 MST Visualization on San Francisco Map

As is clearly depicted in the figure, most of travels do not involve long distances. Thus, the departure point and destination point are usually within the same city or adjacent ones. The picture enables the result to appear more intuitive.

Question 8

93.2% of triangles in the graph satisfy the triangle inequality, ensuring the effectiveness of 2-approximation algorithm of TSP problem, which is based on the preorder sequence of MST.

Question 9

In this question, we are asked to find the upper bound of empirical performance of the approximate algorithm:

$$\rho = \frac{Approximate\ TSP\ Cost}{Optimal\ TSP\ Cost}$$

Since TSP problem belongs to NP-Complete, we cannot acquire the cost of optimal TSP-tour efficiently and it is impossible to get at such a large scale. However, we can use the mathematical relation of MST cost and optimal TSP-tour cost to bound the value of performance above.

$$\rho = \frac{Approximate\ TSP\ Cost}{Optimal\ TSP\ Cost} < \frac{Approximate\ TSP\ Cost}{MST\ Cost}$$

The approximate TSP tour sequence varies with the start point of DFS on MST. We ran DFS many different times to find a maximum resulting TSP-tour cost as the numerator of ratio, which is 464916.5. Combined with MST cost 279408.2, the upper bound of performance of approximate algorithm we found is **1.664**.

As is learned in the lecture, the cost of approximate TSP-tour should be no greater than two times of optimal TSP-tour cost if the graph's cost function meets the requirement of "triangle inequality".The actual result is quite expected because the graph almost satisfies the triangle inequality according to our prior exploration in Question 8.

Question 10

The final TSP-tour trajectory is shown in Figure 2.2. We hope that Santa finds it helpful for his gift delivery for the next Christmas.

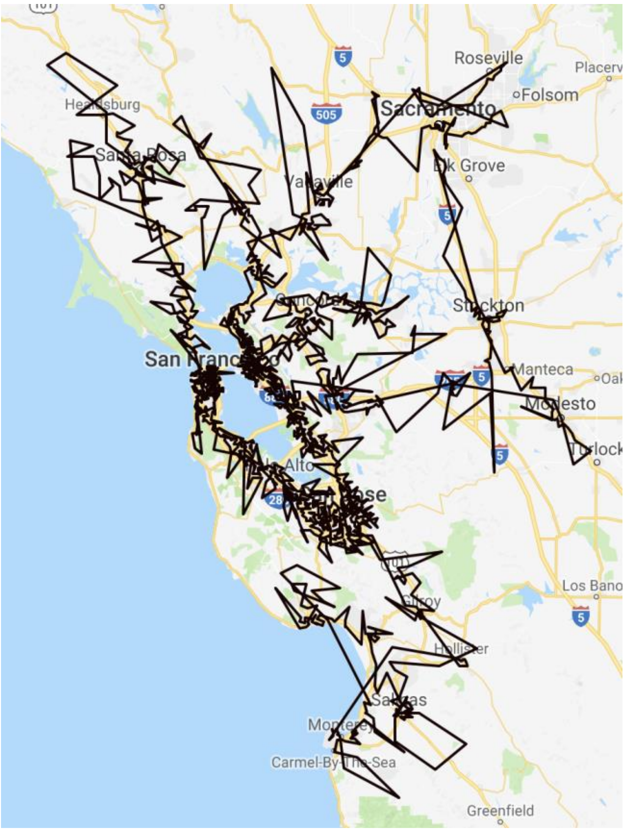

Figure 2.2 TSP trajectory

With the assistance of additional number markers in Figure 2.3, we can see that many collection points are located around popular cities in San Francisco Bay area, such as San Francisco, Palo Alto and San Jose.

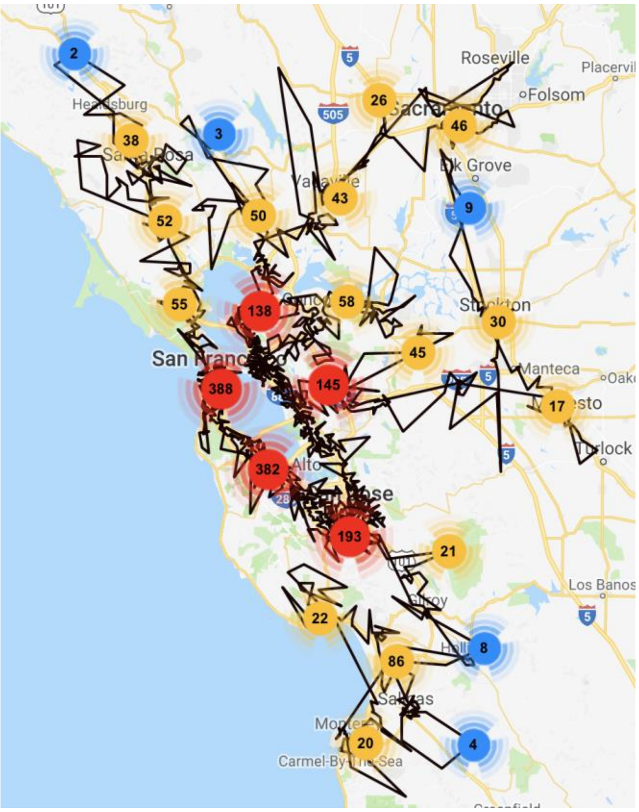

Figure 2.3 TSP trajectory with Number Markers

Part 3 Analyzing the Traffic Flow

Question 11

In this section, a graph is generated by Delaunay triangulation on the nodes coordinates. Delaunay triangulation is a triangulation on discrete points P such that no point of P is inside the circumcircle of any triangle; in other words, this algorithm maximizes the minimum angles of the triangles in this triangulation. The Delaunay triangulation graph G_{Δ} has 1880 nodes with 5627 edges. This graph is plotted as Fig 3.1, and intuitively, it is fairly sparse but highly connected. However, the Delaunay triangulation algorithm produces some extremely long edges which are impossible in real world.

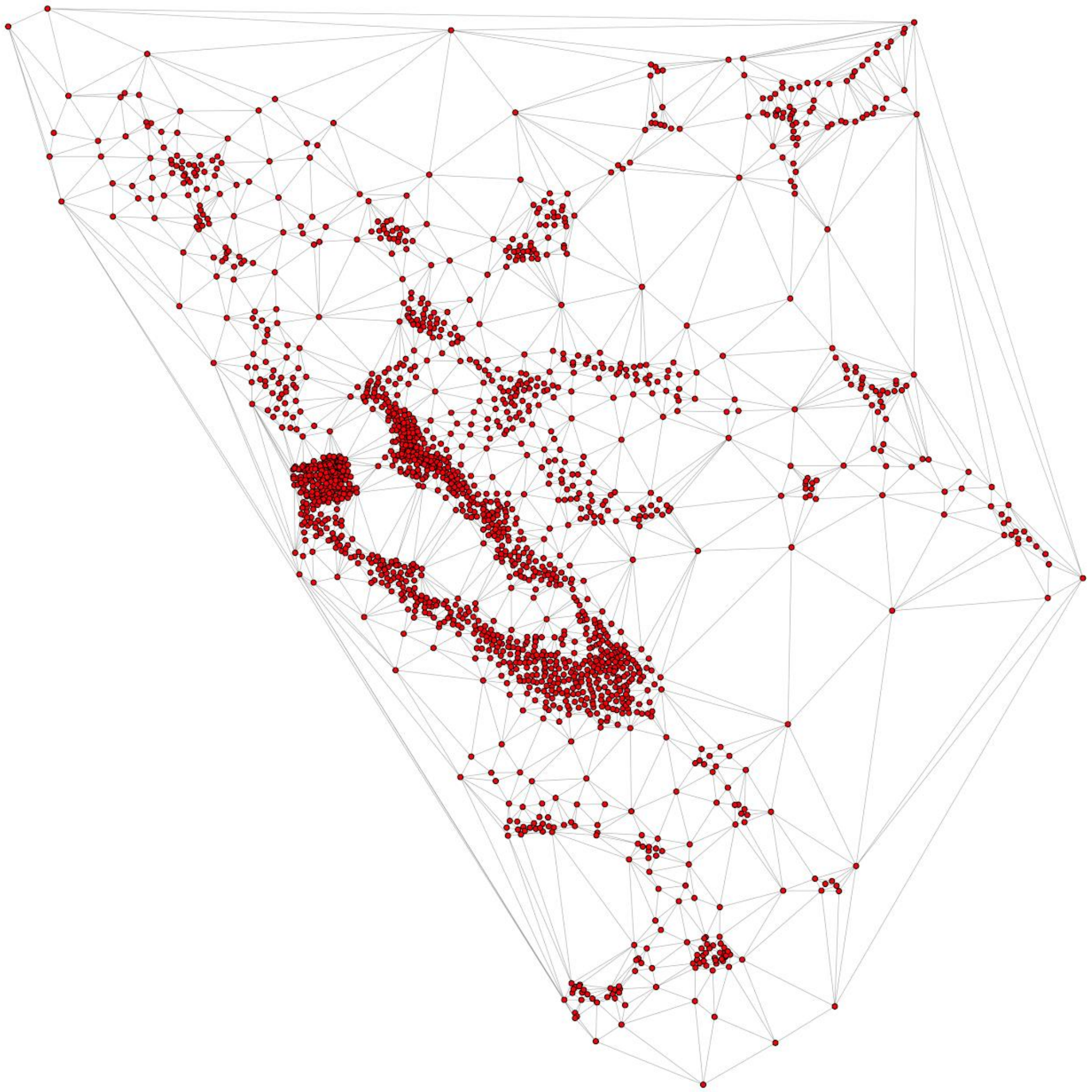

Fig 3.1 Road Mesh from Delaunay Triangulation

Question 12

In this section, the capacity of each road in G_{Δ} in terms of cars/hour. The process is shown as following:

1. Calculate length of each road. Given ends coordinates of a road as (x_1, y_1) and (x_2, y_2), the length is $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \times 69 miles.$
2. Calculate the mean travel time of each road. If this road exists in G, the mean time is given by the mean travel time in G, otherwise given by the short path in G on the mean travel time.
3. Calculate the mean speed of each road by length/time.
4. The capacity of each road is given by $\frac{1\ hour}{\frac{car\ length}{speed} + safe\_time} \times 2.$

Question 13

The maximum number of cars is given by the max flow from Stanford to UCSC. The result is 298.926 ≈ 299 cars per hour. The number of edge-disjoint paths is 5. In Fig 3.2, it is depicted that both nodes corresponding to Stanford and UCSC has degree greater or equal to 5. Since this graph is highly connected, the result is reasonable.

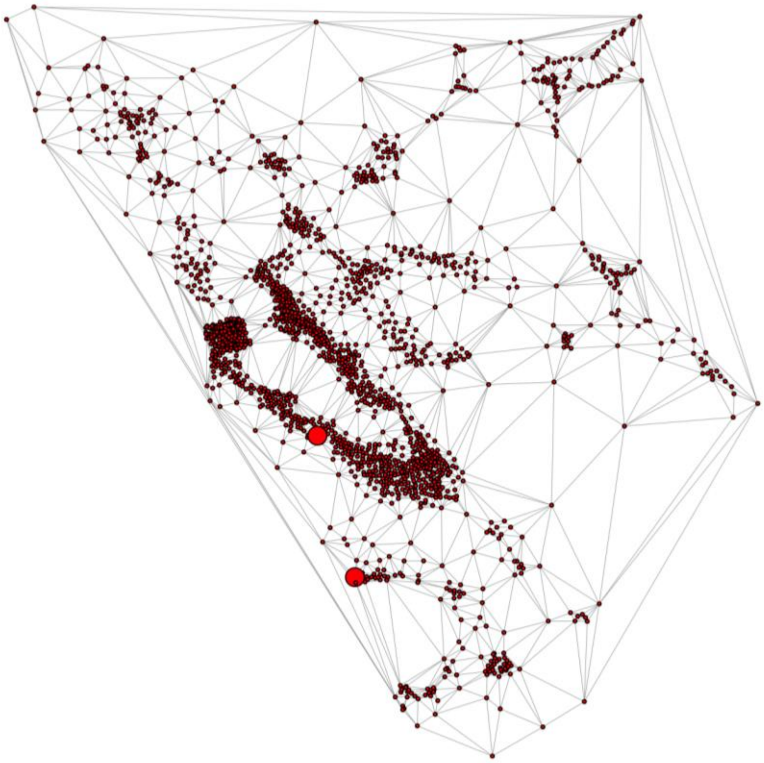

Fig 3.2 Road Mesh from Delaunay Triangulation (highlight Stanford and UCSC)

Question 14

In this section, G_{Δ} is trimmed with a threshold on both travel time and road length. The first try is shown in Fig 3.3 and Fig 3.4. In Fig 3.3 the road length is bounded to 20 miles, and in Fig 3.4 the travel time is bounded to 27 min, the mean travel time in G. However, both results are not coordinated well to the real world model, i.e. too many fake bridges and roads on the graphs. Bounded with the road length has immanent disadvantage in that road generated by Delaunay triangulation may be short but difficult to construct e.g. cross a mountain or a river. Therefore, the graph is further trimmed while the travel time is bounded to 13 min. The new graph $\tilde{G}_{\Delta}$ plotted in Fig 3.5, a satellite map, and Fig 3.6, a road map. In $\tilde{G}_{\Delta}$, only San Mateo Bridge, the longest and busiest bridge is not preserved. This problem could be solved by special selected threshold.

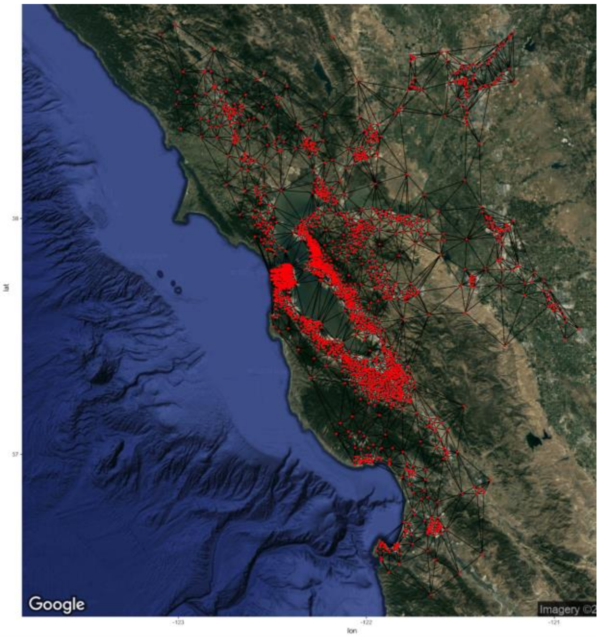

Fig 3.3 Trimmed Road Mesh (Road Length < 20 miles)

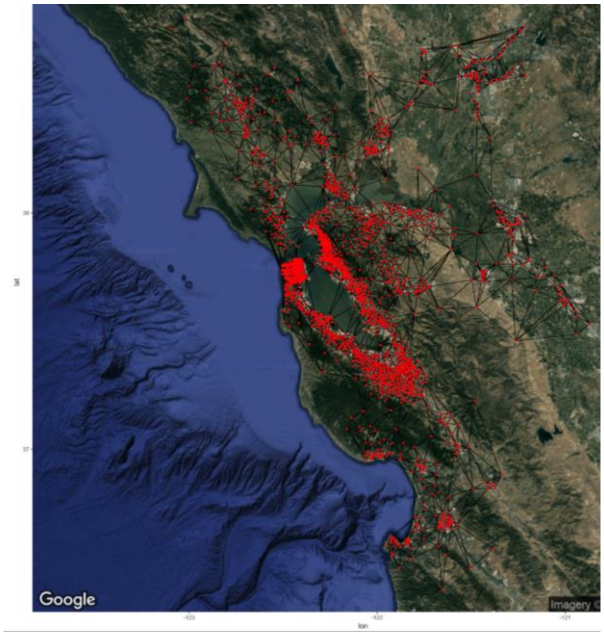

Fig 3.4 Trimmed Road Mesh (Travel Time < 27 min)

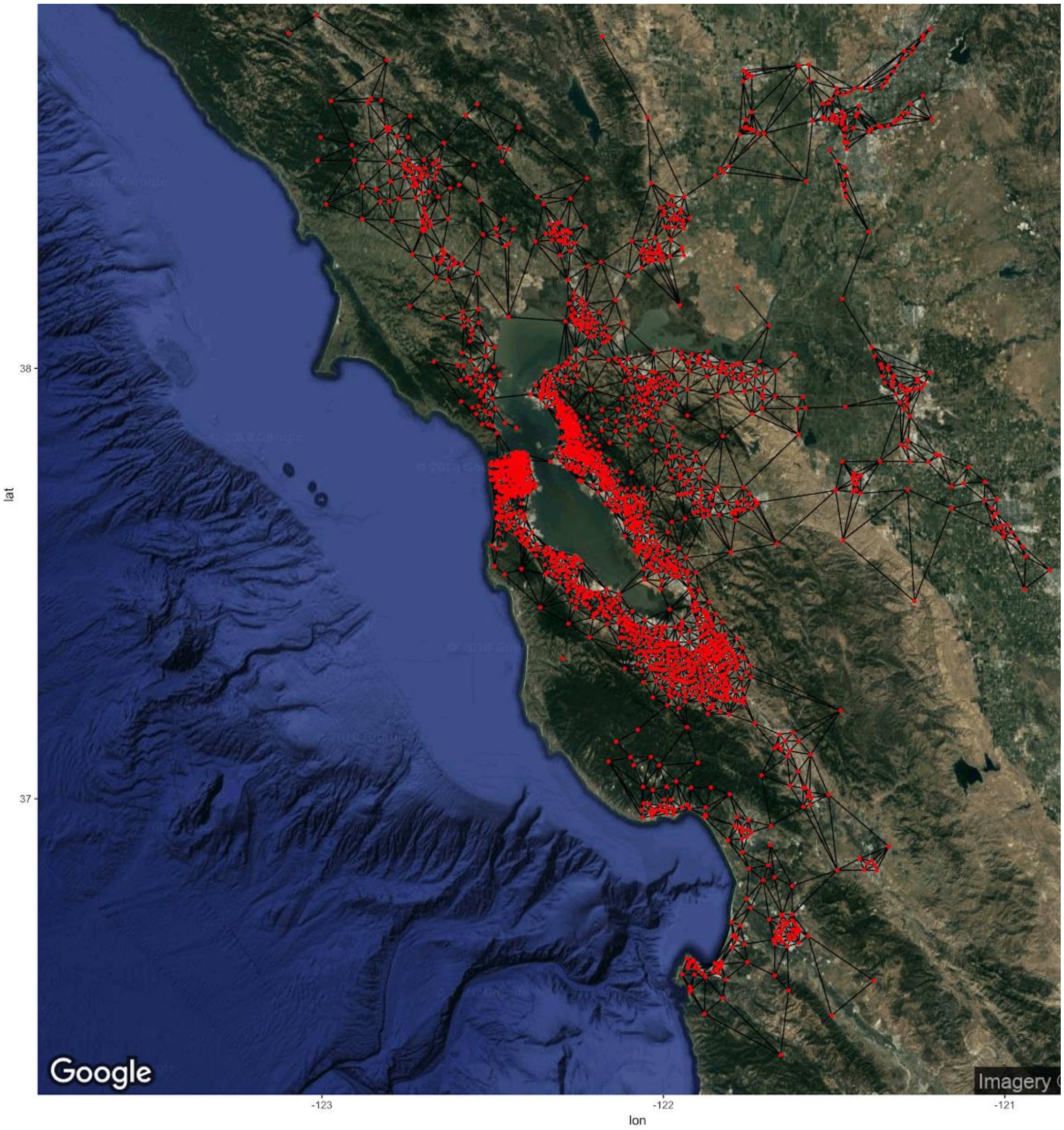

Fig 3.5 Trimmed Road Mesh (Travel Time < 13 min)

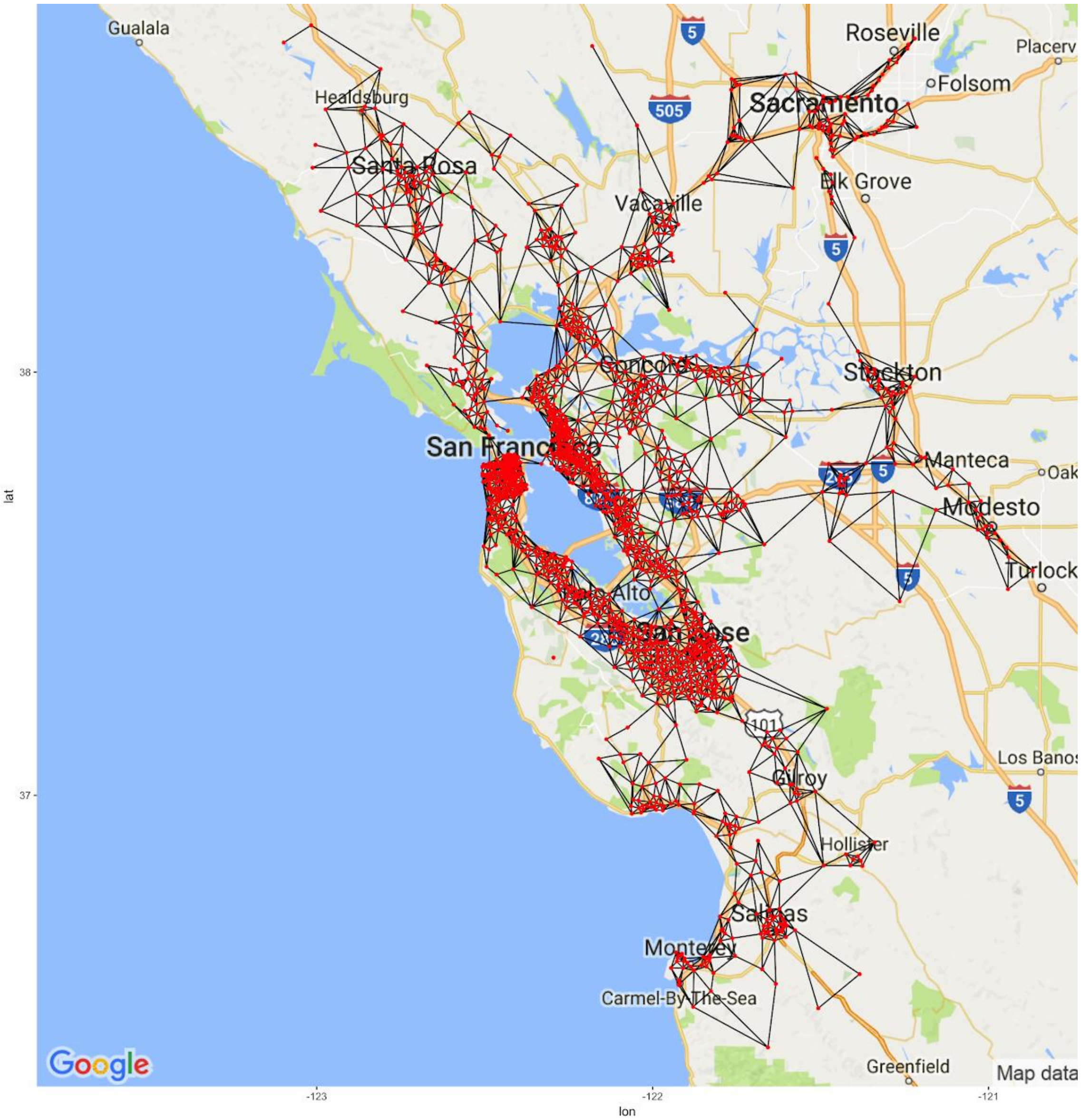

Fig 3.6 Trimmed Road Mesh on Real Road Map (Travel Time < 13 min)

Question 15

In this section, percentage of triangles in the graph $\tilde{G}_\Delta$ satisfy the triangle inequality is determined. Following the same process as in Question 8, the result of $\tilde{G}_\Delta$ is 86.8%. This significant change owns to the fact that many edges in $\tilde{G}_\Delta$ have short length but big mean travel time as the end points of a fake road in $\tilde{G}_\Delta$ are connected by several roads in real world.