# Project 5: Graph Algorithms

Duan Li 005026839
Di Ma 004945175
Yuan Shao 504880181
Weijie Tang 305029285

## Part 1: Stock Market

### 1.1 Return correlation

Q1:
The range of $\rho_{ij}$ is from -1 to 1.

The reasons to use log-normalized return instead of regular return are as following:

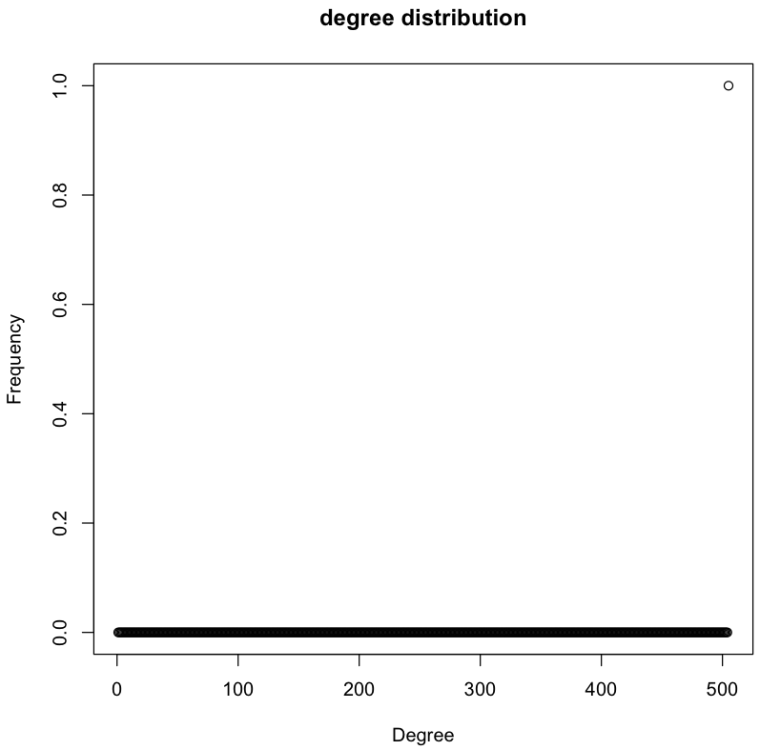
- Log-normalized return transform the non-numerical-safe multiplication of regular return into the **numerical safe** summation via logs.
- Log-normalized return has the property of **time addictivity**. The compound return over n periods can be calculated by the difference in log between period of 1 and period of n. Also, it will reduce the time complexity from O(n) multiplications to O(1) additions.
- Log-normalized return ensures the **log-normality**. Assume that stock prices are log normally distributed, then r (t) is normally distributed due to the following equation:

  $$r_i(t) = logp_i(t) - logp_i(t-\tau) = log\frac{p_i(t)}{p_i(t-\tau)}$$

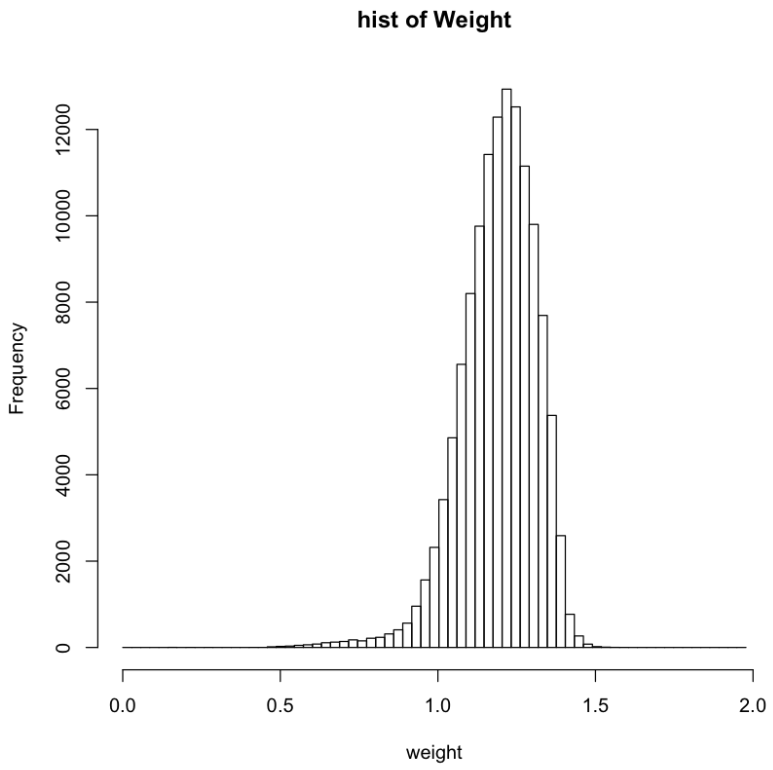
  So exponential problem can be converted to linear one since stock prices have normal distribution and stock returns are close to normal distribution.
- Log-normalized return **makes the residual more manageable** to analyze further without the need to have the auto-regression be normalized in order to obtain meaningful result.
- Log-normalized return is an **autoregressive coefficient with significance,** which indicates the potential stock that can be traded on, given that deviations from the random walk theory is able to trade profitably.

### 1.2 Constructing correlation graphs

Q2:
The graph below is the degree distribution of the correlation graph



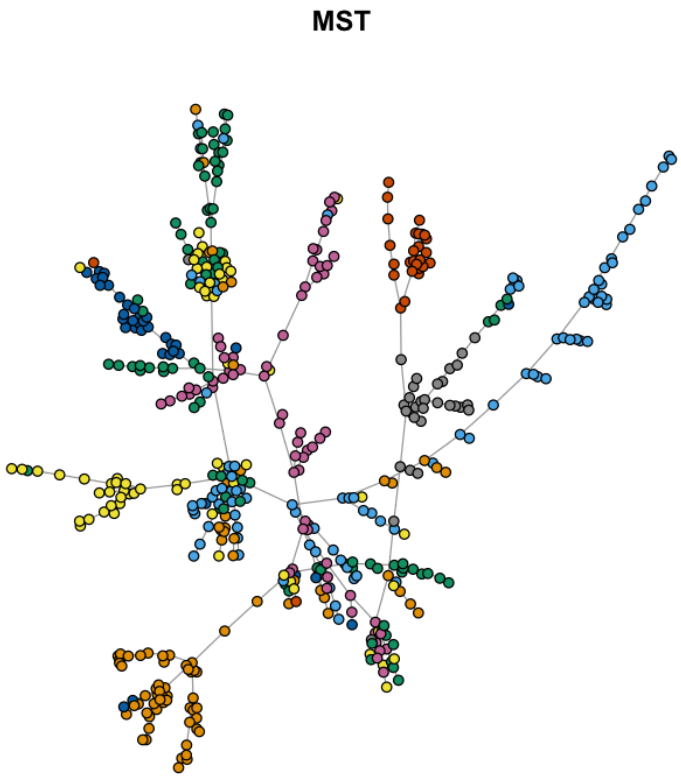

The graph below is the histogram showing the un-normalized distribution of edge weights. The mean is around 1.2 and the variance is around 0.016. This result supports the normal distribution assumed in question 1.




## 1.3 Minimum spanning tree (MST)

Q3:

The graph below is the MST plot.




From the graph above, we observe several different clusters. The nodes on each MST branch almost have the same color, which indicates that stocks in the same sector form the vine cluster in MST. If the weight is the correlation value, then the nodes in same sector will be clustered together.

The weight of correlation graphis calculated by $w_{ij} = \sqrt{2(1-\rho_{ij})}$ . Once two stocks have positive correlation base on the log-normalized return for the price, the weight between them will be small. According to the definition of the MST, a MST is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight. Thus, the MST will cluster the pair of stocks with positive correlation and small weight, in order to minimize total edge weight.

## 1.4 Sector clustering in MST's

Q4:

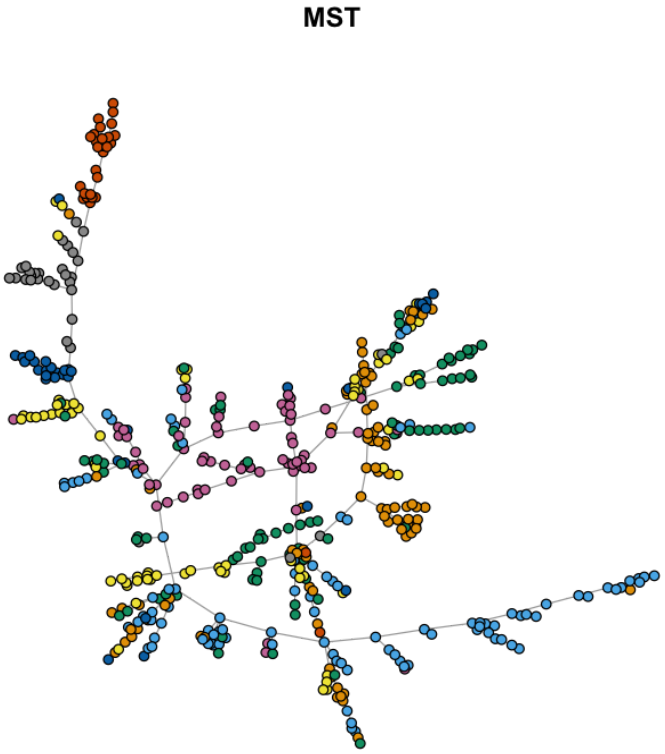
| | $P(v_i \in S_i) = \frac{\lvert Q_i \rvert}{\lvert N_i \rvert}$ based on neighbors | $P(v_i \in S_i) = \frac{\lvert S_i \rvert}{\lvert V \rvert}$ random assignment |
|---|---|---|
| alpha | 0.821755109099061 | 0.121351133202188 |

To predict the market sector of an unknown stock, there are two methods: (1) based on the immediate neighbors of the stock in the MST (2) based on the random sector assigned in the MST.

We observe that the alpha value of neighbor assignment is around 0.82, which is much higher than the alpha value of random assignment, which is around 0.12. This indicates that neighbor assignment haves better performance than random assignment on sector prediction because stocks connected together are more likely to be in the same sector (color), which obeys the observation of question 3. Therefore, we can easily predict the sector by neighbor assignment since it has the tendency to form clusters.

## 1.5 Correlation graphs for weekly data

Q5:
In question 3, we used daily closing prices for stocks to compute returns. Now, we sample the stock data weekly on Mondays. The plot below is the MST based on weekly data.




Recall what we observe from the daily MST in question 3. There is a clear tendency that nodes in the same color are close to each other and on the same branch. This pattern shows the vine-cluster of MST, where the stocks in the same sector are clustered.

However, in the weekly MST, this vine-cluster pattern is not as clear as it does in daily MST. The nodes with the same color seem to be somewhat distributed through the MST graph. This is to say that the correlation is not so close between stocks in the same sector. This observation makes sense because in the weekly MST, trading days become 5 rather than 1 in the daily MST, which leads to more unpredictable price change of the stock, and less correlation than the daily MST.

# Part 2: Let’s Help Santa!

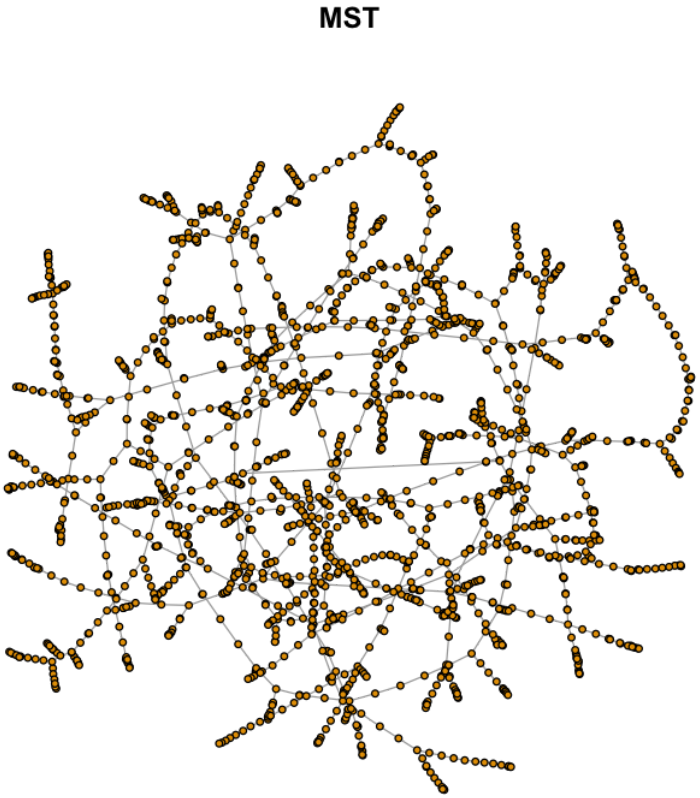
## 2.1 Download the Data

## 2.2 Build Your Graph

Q6:
Number of nodes:1880
Number of edges: 311802

## 2.3 Traveling Salesman Problem

Q7:
The plot below is the MST of G




Here are the street address of 5 edges. The results are intuitive because they addresses are all in San Jose and are very close to each other in real map. So it makes sense that they are connected together in the graph.

| id | endpoint 1 address | id | endpoint 2 address | map distance |
|---|---|---|---|---|
| 19 | 3300 Brodie Drive, South San | 1872 | 1200 Ashcroft Lane, Cambrian, | 6.5 miles |

| | | | | |
|---|---|---|---|---|
| 38 | Jose, San Jose | | San Jose | |
| 1938 | 3300 Brodie Drive, South San Jose, San Jose | 1873 | 400 Piercy Road, Edenvale, San Jose | 5.0 miles |
| 1938 | 3300 Brodie Drive, South San Jose, San Jose | 1891 | 800 Jury Court, North San Jose, San Jose | 8.7 miles |
| 1938 | 3300 Brodie Drive, South San Jose, San Jose | 550 | 2100 Naida Avenue, Alum Rock, San Jose | 6.3 miles |
| 1938 | 3300 Brodie Drive, South San Jose, San Jose | 176 | 500 Hull Avenue, Central San Jose, San Jose | 8.2 miles |

Q8:
Among the 1000 triangles sampled from the graph,  92.6% of them satisfy the triangle inequality. Since the triangle inequality principle holds for 90% of the sampled triangles, we would get a good performance.

Q9:
We know that:

MST cost < optimal TSP cost < approximate TSP cost < double MST cost

The lower bound of TSP cost is MST cost = 279408.18 and the upper bound bound of TSP cost is double MST cost = 558816.36. Now we need to find the approximated TSP cost as follow:

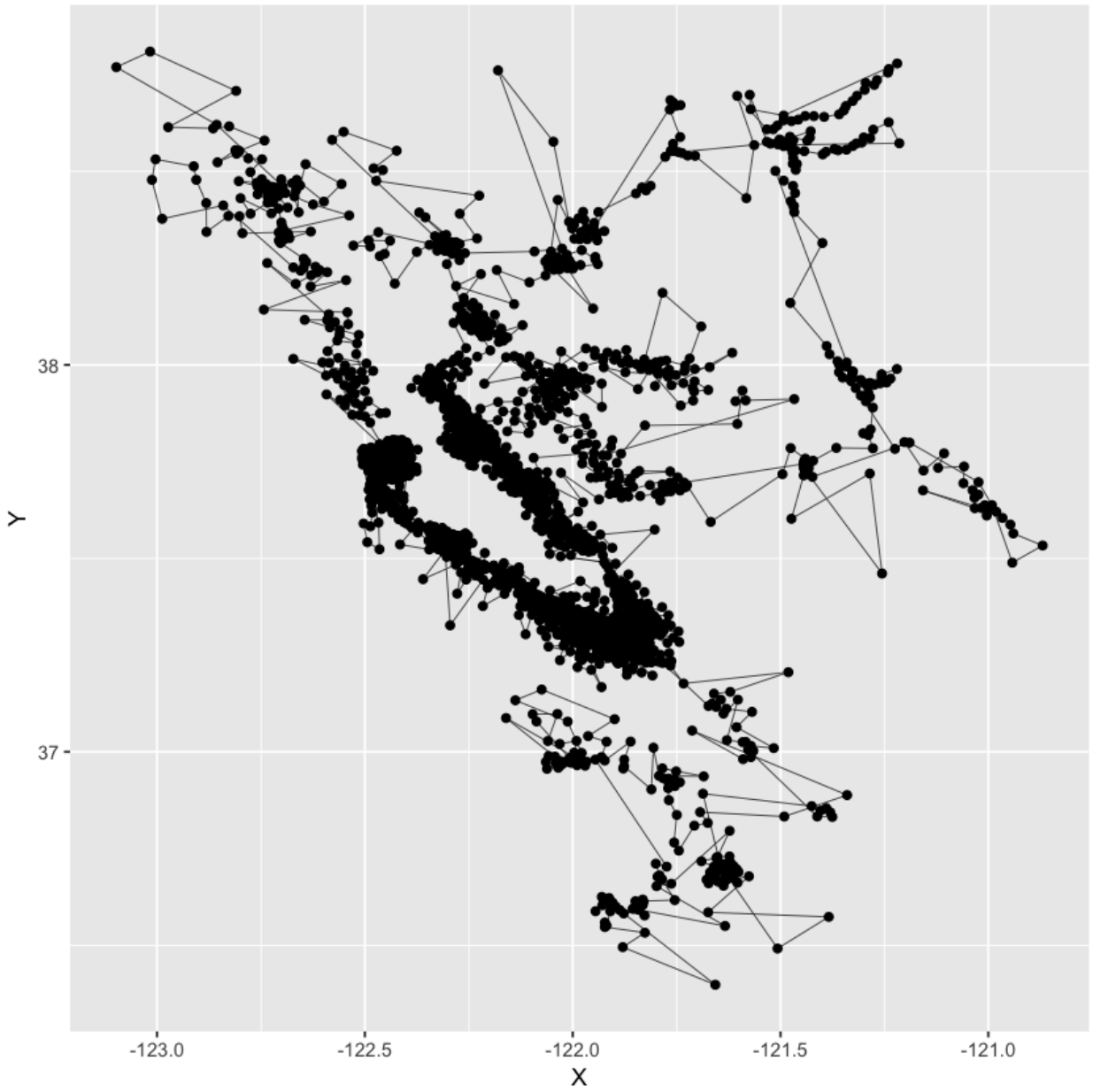
- Duplicate the edges in MST to create the multigraph in R
- Find the Eular tour in Python and save it to tsp_tour.txt
- Compute the sum of edges of TSP solution based on the adjacency matrix of correlation graph in R

It is impossible to obtain the real optimal TSP cost because it is an NP-Complete problem. The current solution is the best choice at each step, every node but is not guaranteed to be globally optimal. We found that the approximate TSP cost is 464500.5 and it obeys the principle mentioned above: MST cost < approximate TSP cost < double MST cost.

| MST cost | approximate TSP cost | Double MST cost |
|---|---|---|
| 279408.18 | 464500.5 | 558816.36 |

The empirical performance is $\rho = \frac{\text{Approximate TSP Cost}}{\text{Optimal TSP Cost}} = \frac{464500.5}{558816.36} = 0.8312221$ , which indicates that the performance is not bad.

Q10:
The plot below is the trajectory. From the plot, we can find the start point is around (-121.2, 37.6)

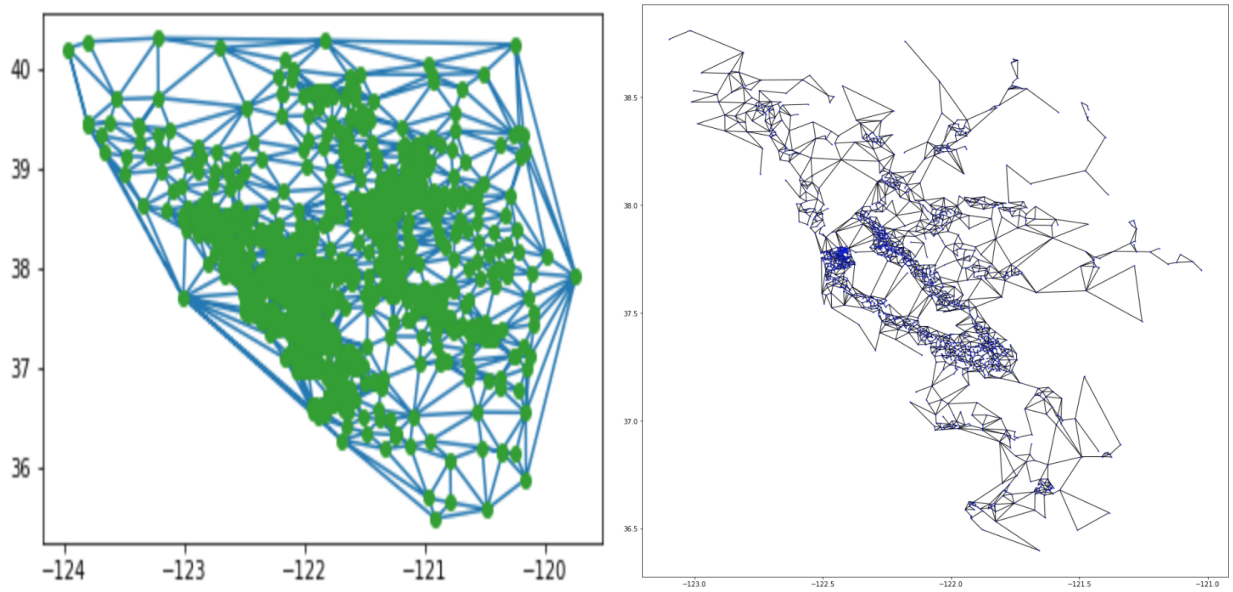



# Part 3: Analysing the Traffic Flow

## 3.1 Estimate the Roads

Q11:

Below are the graph before (left one) and after (right one) applying the Delaunay triangulation algorithm. We can see that after the Delaunay triangulation algorithm and removing the edges without weight, there are 10,810 edges left. The subgraph GΔ maximizes the minimum angle of all the angles of the triangles in the triangulation. As a result, there are some fake routes generated in this subgraph, like those fake bridges over the bay.




## 3.2 Calculate Road Traffic Flows

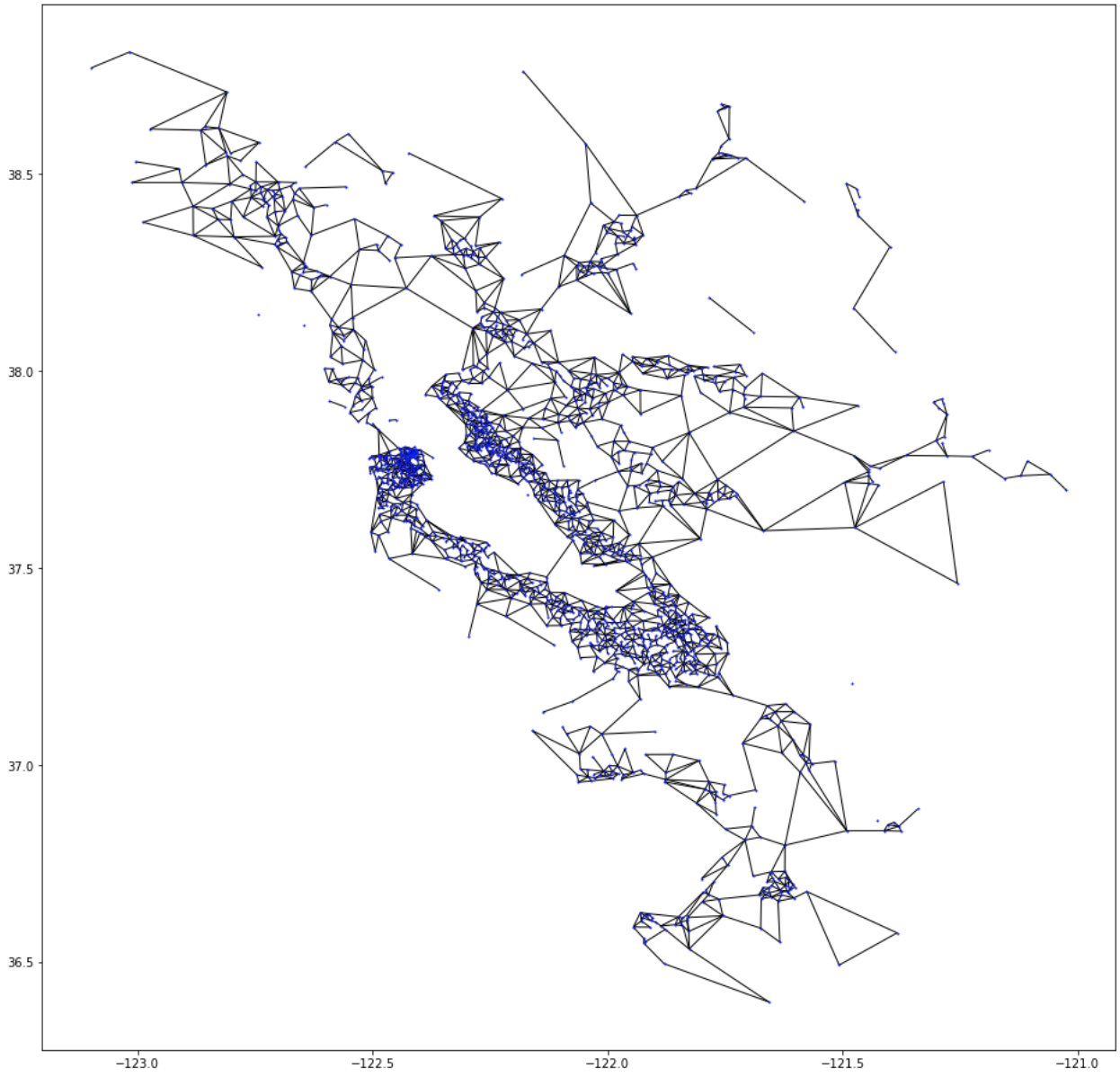
Q12:
For each road, we first calculate its length using both ends' coordinates. Then we divide the road length by the travel time to get the speed. Finally, the traffic flow equals `speed * (1 / (0.003 + (speed * 2) / 3600.0))`. The traffic flow of each road is shown in the output of the code, and it is saved as the capacity of each road.

## 3.3 Calculate the Max Flow

Q13:
After finding the corresponding nodes representing Stanford and UCSC in our graph, we can get the max flow from Stanford to UCSC is about **3184.24** cars per hour, and there are **2** edge disjoint paths between the two spots. From a real map, we can see there might be 4 edge disjoint paths between the two spots: CA-1 from west, CA-1 from east, CA-9 and CA-17. The graph contains fewer edge disjoint paths than the real map.

## 3.4 Defoliate Your Graph

Q14:
To trim the fake bridges crossing the bay, we defoliate the graph and remove the ones having larger than 870 as weight. Below is the defoliated graph plotted on real map coordinates.
We can see that out of five bridges, only the Golden Gate Bridge and the Dambarton Bridge are preserved, and there are still two fake bridges near the Golden Gate Bridge which are too difficult to be defoliated in terms of edge weights.




Q15:

For the defoliated graph, we get the max flow from Stanford to UCSC is about **3047.13** cars per hour, and there are **2** edge disjoint paths between the two spots. The max flow is a little bit smaller than the one in Q13 and the number of edge disjoint paths remains the same. The results make sense because since there are fewer edges in the graph, the max flow is likely to decrease.