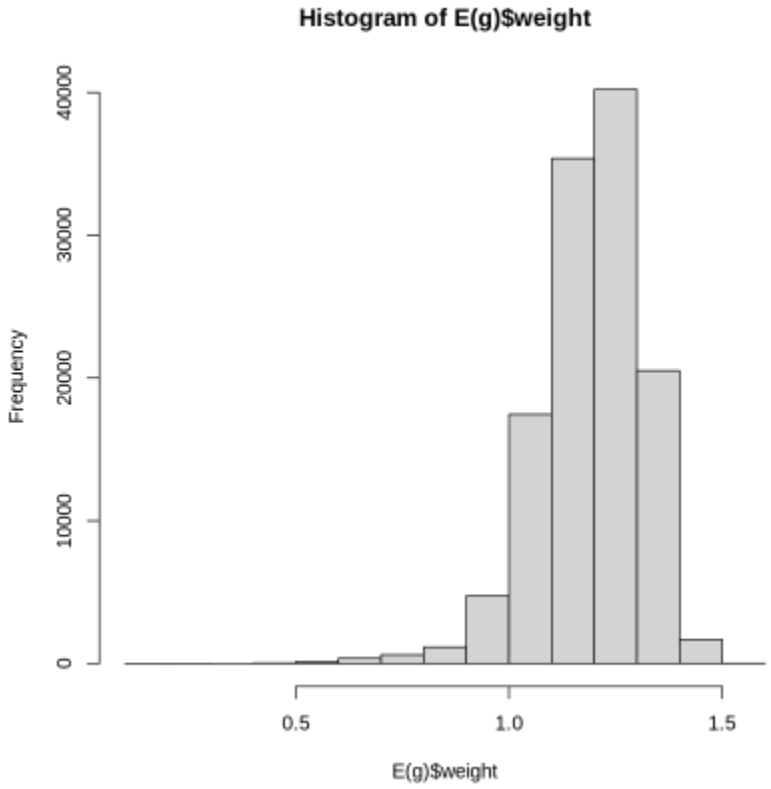
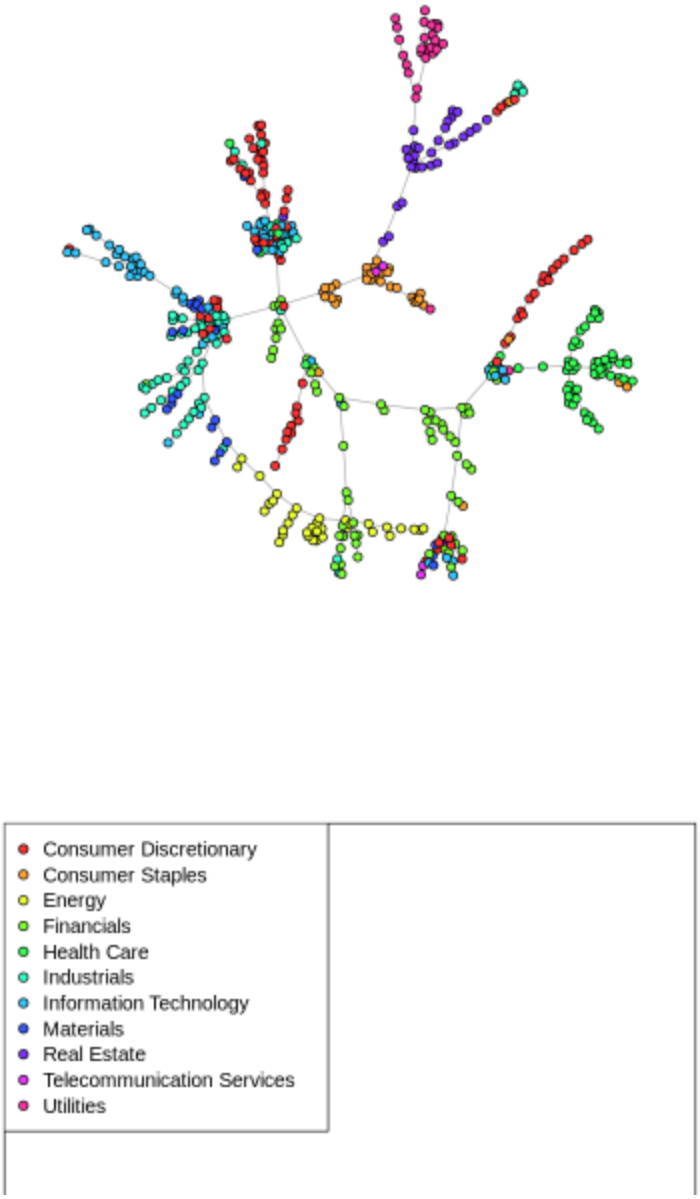
Project 4:
Haoting Ni (905545789), Yikai Wang (905522085), Yuanxuan Fang (005949389)
Question 1:
According to the formula, pij shows the linear relationship between stock i and j. We can clearly see that this formula is in the format of Pearson correlation. By definition, the upper bound is 1 and the lower bound is -1. Using log-normalized returns will have more stable data, which will help reduce the variance. In contrast, other volatility might affect regular returns and result in an incorrect correlation. Also, the log-normalized returns treat positive and negative results equally while the regular return is more sensitive to positive data, and result in a positive skewness.

Question 2:

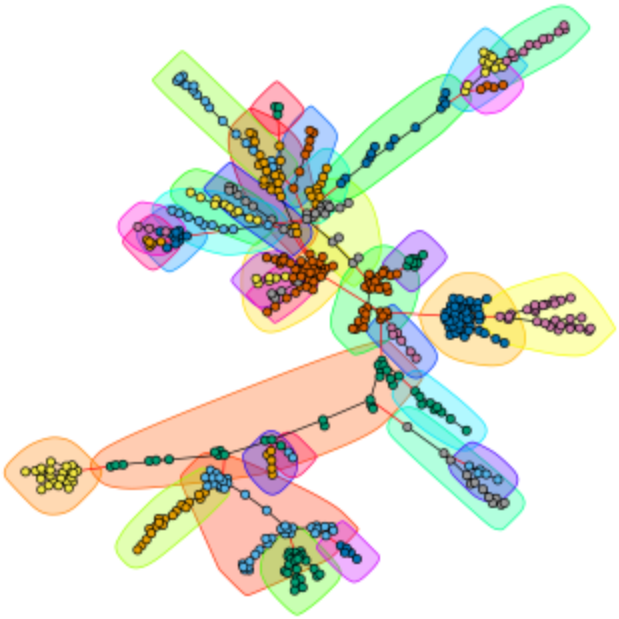


Question 3:



We can see that stocks within the same sector are forming the cluster in the MST. These vine clusters indicate the clusters of strokes are highly correlated, which means stocks within the same sector have the same behaviors. But you can also see some sectors are interconnected. Nodes with different colors may also have connections. For example, Consumer Discretionary, Industrials, and Materials are in the same joint and may share the same behaviors.

Question 4:



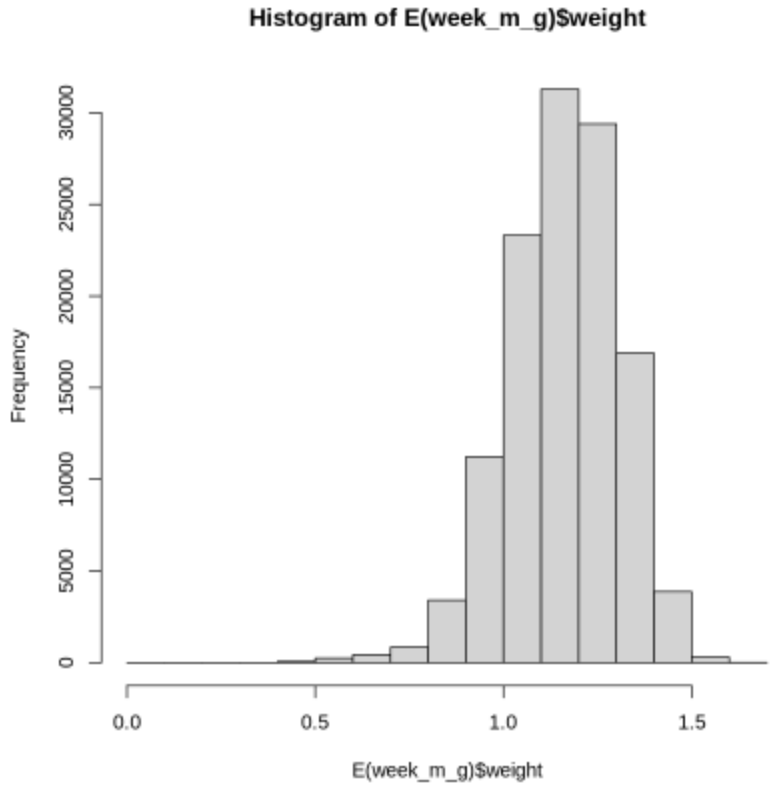
Homogeneity: 0.698800771048459
Completeness: 0.479770924836185

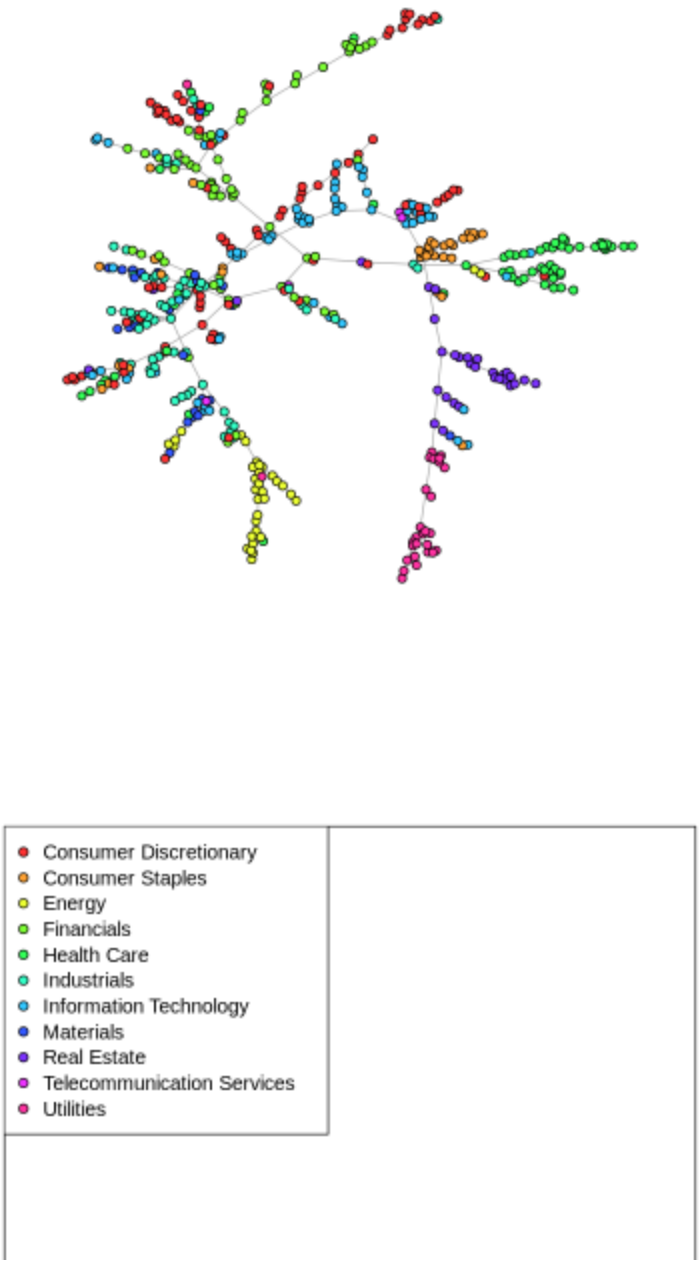
Question 5:

```
[1] "Alpha value for method 1 is: 0.824671864347978"
[1] "Alpha value for method 2 is: 0.114188070612533"
```

Method 1 indicates the probability of a node belonging to the same sector as its neighbor nodes. Method 2 indicates the probability of a node belonging to a sector based on the overall proportion of stocks. The difference suggests that Method 1 is a better method to analyze the MST that has a vine cluster structure. Because Method 1 considers the local neighborhood while Method 2 considers the whole MST. The difference also tells us that we should use Method 1 while predicting market sectors for stocks.

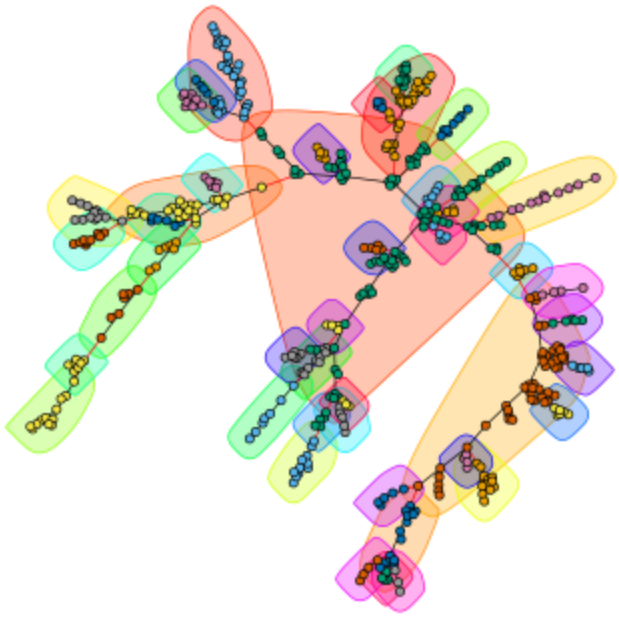
Question 6:


Histogram of E(week_m_g)$weight
Frequency
0
5000
10000
15000
20000
25000
30000
0.0
0.5
1.0
1.5
E(week_m_g)$weight



This is the MST obtained by Monday's stock price only. Compared with the MST from question 3, we can see that the nodes in this MST are less clustered. But they still form the vine cluster structure. We can also see the connections between different sectors from the MST because different colors of nodes are stacked on each other. Because weekly data may have lower volatility than daily data, less correlation may result in a less clustered MST. Weekly data capture a broader market trend, which may result in a high diversification and more dispersed result.

Community:

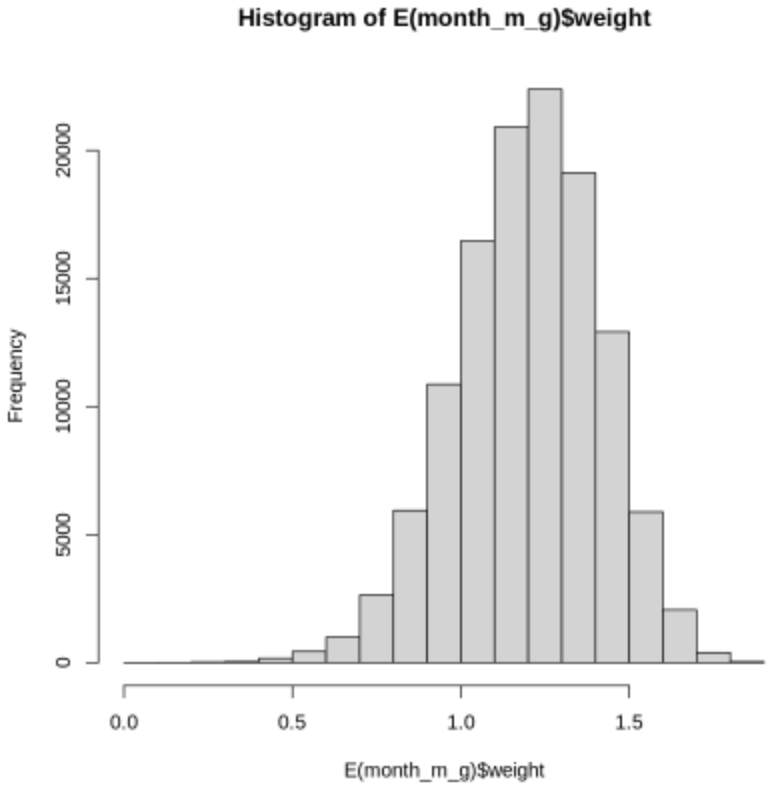


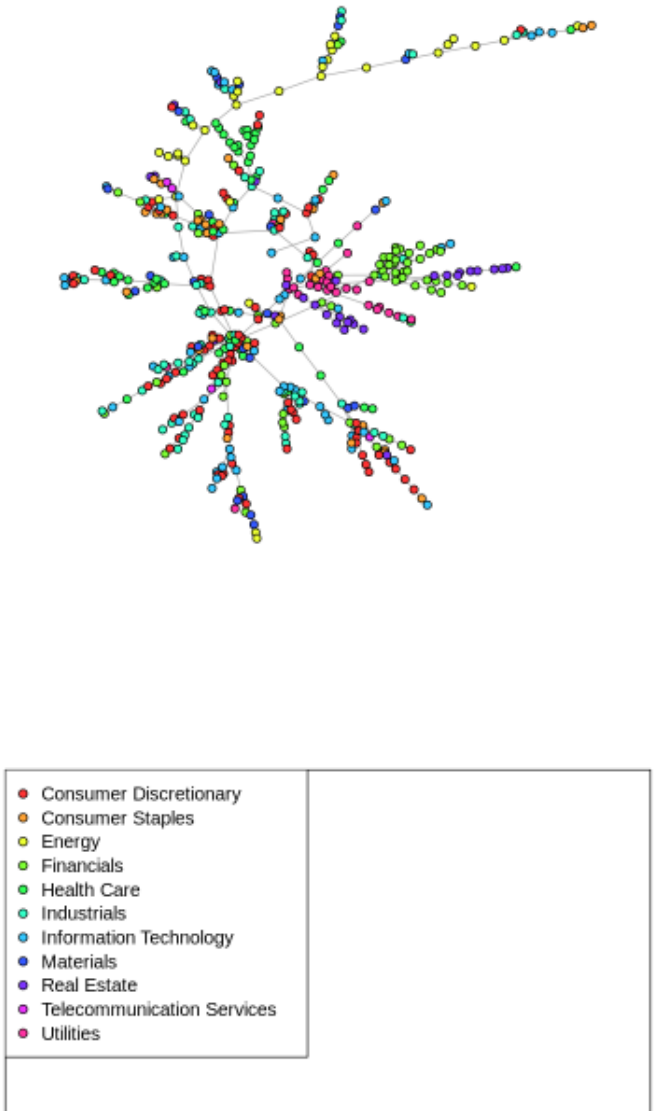
Homogeneity: 0.593106108661072
Completeness: 0.395297839518033

```
[1] "Alpha value for method 1 is: 0.744384573195549"
[1] "Alpha value for method 2 is: 0.114308612598321"
```

We can see the alpha value for method 2 is almost the same for weekly and daily data. Because the calculation of method 2 is based on the entire graph, weekly or daily data will have little effect. Also, the weekly data is less clustered than the daily data, so the alpha value for method 1 is less. Because Method 1 indicates the probability of a node belonging to the same sector as its neighbor nodes. Method 1 focus on local cluster and neighborhood.

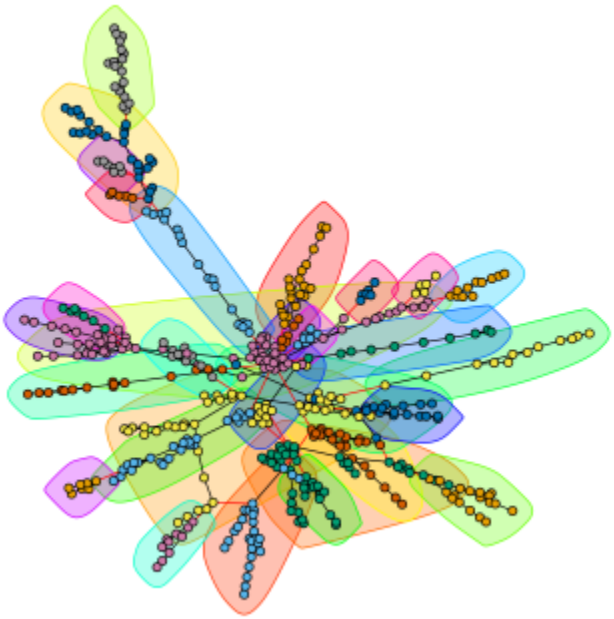
Question 7:


Histogram of E(month_m_g)$weight
Frequency
0
5000
10000
15000
20000
0.0
0.5
1.0
1.5
E(month_m_g)$weight



This is the MST obtained by stock prices on the 15th of every month. Compared with the MST from question 3, we can see that the nodes in this MST are obviously less clustered. But they still form some kind of vine cluster structure. We can also see the connections between different sectors from the MST because different colors of nodes are stacked on each other. There are more connections because there are cycles formed compared to MST in question 3. Because monthly data may have lower volatility than daily data, less correlation may result in a less clustered and complexed MST. Monthly data capture a broader market trend, which may result in a high diversification and more dispersed graph.

Community:



Homogeneity: 0.360621299806933
Completeness: 0.257163641262016

```
[1] "Alpha value for method 1 is: 0.455305754543559"
[1] "Alpha value for method 2 is: 0.114308612598321"
```

We can see the alpha value doesn't change a lot. Because the calculation of method 2 is based on the entire graph. Also, the monthly data is scattered. Because Method 1 indicates the probability of a node belonging to the same sector as its neighbor nodes. Method 1 focus on local cluster and neighborhood. The value will drop if the graph is scattered.

Question 8:

I find that the less data we use will generate a less clustered graph based on the MST we get. This means daily data will have the most clustered graph and fewer cycles, while the monthly data will have the least clustered graph and more cycles formed. Also, a less clustered graph will also have a less value of Homogeneity and Completeness. The alpha value for method 1 will also drop if the graph is less clustered. But the alpha value for method 2 remains the same because the calculation of method 2 is based on the entire graph. However, Method 1 indicates the probability of a node belonging to the same sector as its neighbor nodes. Method 1 focus on local cluster and

neighborhood. The value will drop if the graph is scattered. The homogeneity and completeness scores are used to evaluate the performance of graph clustering. Because there is less distinct clustering in the graph to be captured, the value of Homogeneity and Completeness will drop.

Based on the vine clusters on MST, the value of alpha for method 1, and other indicators mentioned above, I think the daily data will give the best results. Because a more clustered graph gives a better and clear idea of the sector of an unknown stock and provides higher accuracy. Also, from the data, the probability of predicting a node is in the same sector as its neighbor nodes are the highest for daily data.

Question 9:
Number of nodes: 2649
Number of edges: 1003858

Question 10:

```
Edge: [-118.12053321   34.10309557] -> [-118.13138209   34.09626386]
Edge: [-118.12053321   34.10309557] -> [-118.11656383   34.09585388]
Edge: [-118.13785063   34.09645121] -> [-118.13138209   34.09626386]
Edge: [-118.13785063   34.09645121] -> [-118.13224544   34.10349303]
Edge: [-118.13785063   34.09645121] -> [-118.14184446   34.08538654]

823 E Grand Ave, Alhambra, CA 91801 -> 300 N 3rd St, Alhambra, CA 91801 (3
min 0.8 mile)

823 E Grand Ave, Alhambra, CA 91801 ->  308 S Cordova St, Alhambra, CA
91801 (3 min 0.6 mile)

400 N Marguerita Ave, Alhambra, CA 91801 -> 300 N 3rd St, Alhambra, CA
91801 (3 min 0.7 mile)

400 N Marguerita Ave, Alhambra, CA 91801 -> 830 N Garfield Ave, Alhambra,
CA 91801 (3 min 0.7 mile)

400 N Marguerita Ave, Alhambra, CA 91801 -> 3VP5+572 Alhambra, CA 91801 (3
min 1.0 mile)

The results are intuitive. The minimum spanning tree consists of edges
that connect all the nodes with the least possible weight. 5 edges
averaged 0.7 miles away, which are all only 3 minutes away. They satisfy
the condition of being the edge of a minimum-spanning tree.
```

Question 11:

Triangle inequality holds by 91.2%.

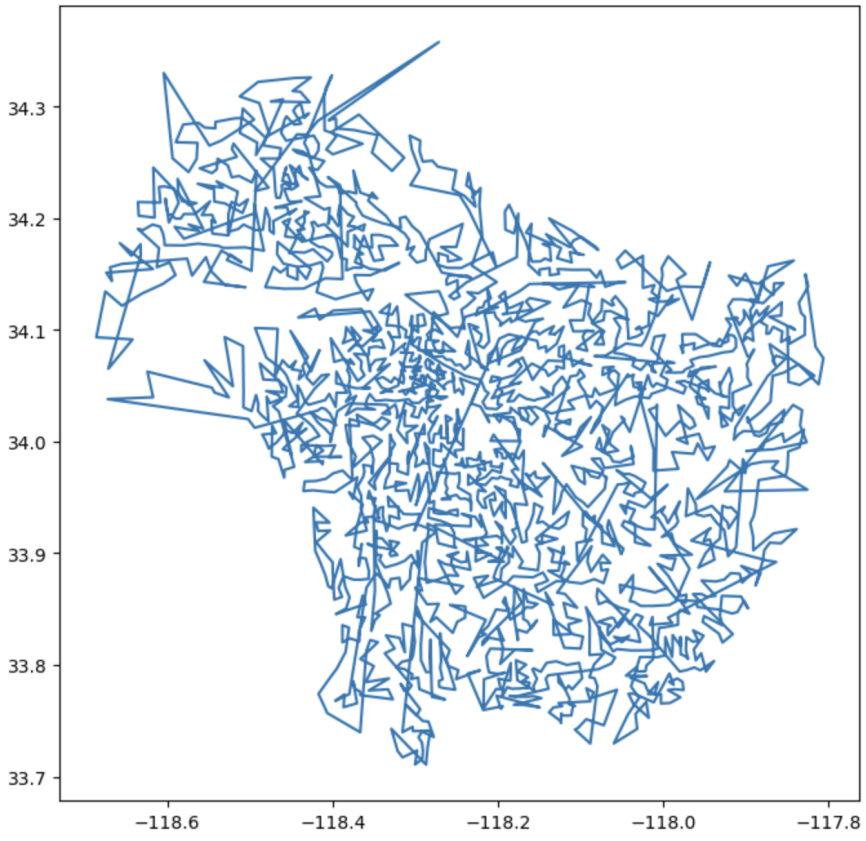
Question 12:

```
MST Cost: 266472
TSP Cost: 454867
Ratio: 1.71
```

By following the steps from the slides to calculate the TSP cost. MST cost is the optimal tsp cost from MST.TSP cost is the total weight of edges from the Eulerian graph.

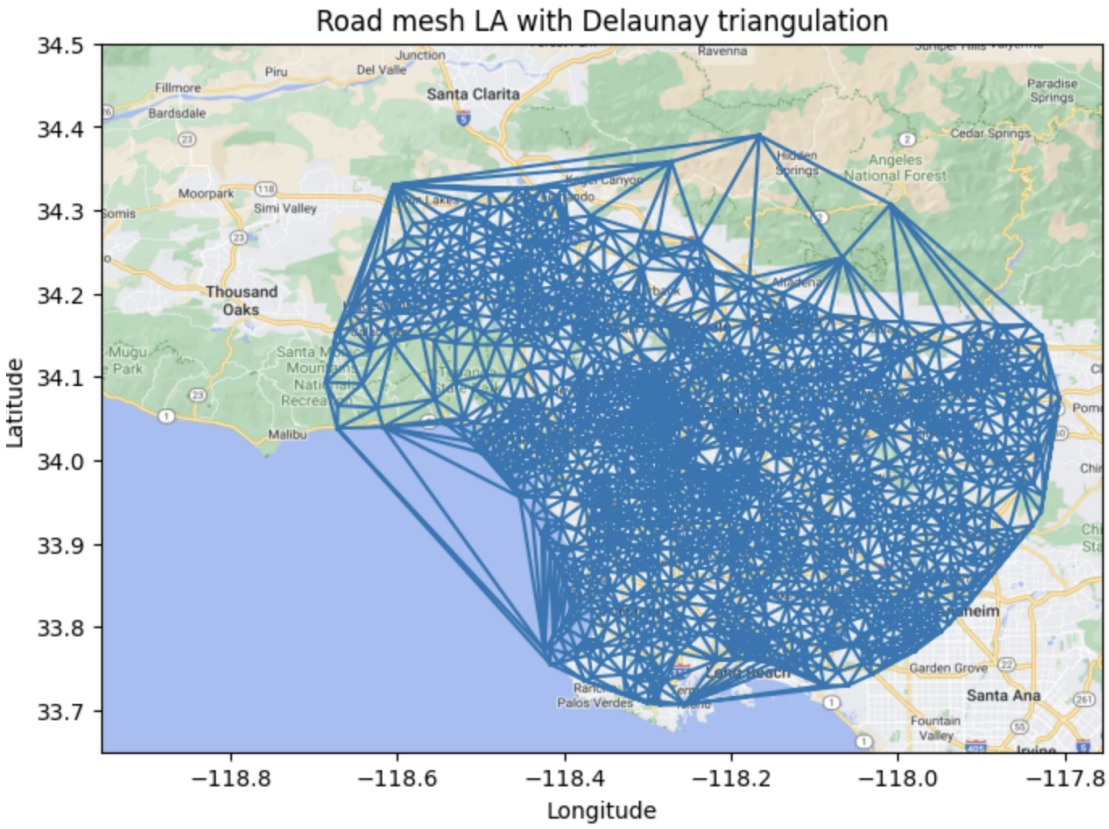
After running on the approximate algorithm, we can get the Ratio would be 1.71

Question 13:
The trajectory that Santa has to travel is shown as blow



Question 14:

The road mesh that listed below has shown that traffic is denser than the mountain area. However, there are also couple edges cross the ocean which are not the real road in real time becuase Delanunay's triangulation method forms additional edges that fits the situation.



Question 15:

Total distance = $(V_{car} * \text{mean travel time}) / 3600$

Car length + gap length = $(0.003 + (V_{car} * 2)/3600)$

Number of cars on road = $[\ 2 * ((V_{car} * \text{mean travel time}) / 3600) / (0.003 + V_{car} / 1800)\ ]$ cars

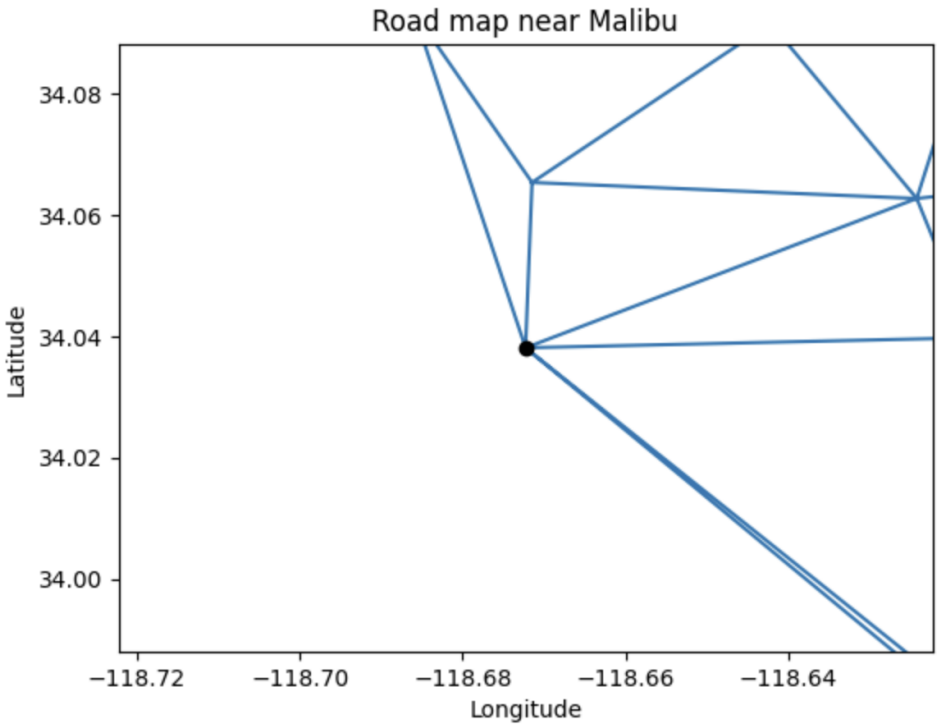
$= [\ (V_{car} * \text{mean travel time}) / (5.4 + V_{car})\ ]$ cars

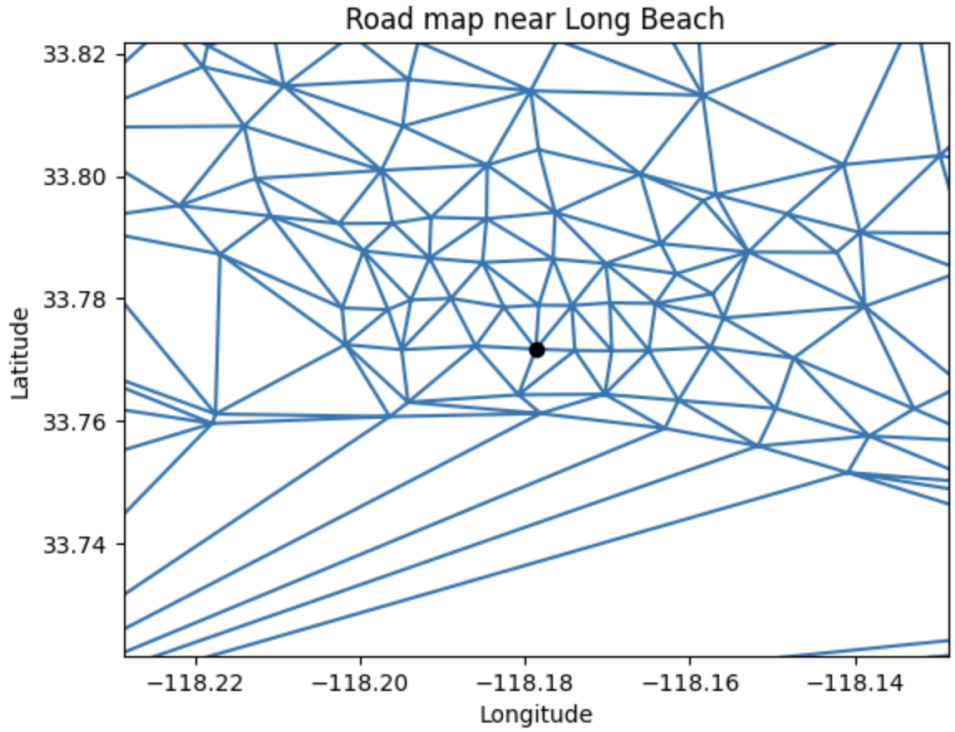
Traffic flow = $3600 / \text{mean travel time} * (V_{car} * \text{mean travel time}) / (5.4 + V_{car})$

$= [\ (3600 * V_{car}) / (5.4 + V_{car})\ ]$ cars/hr

Question 16:

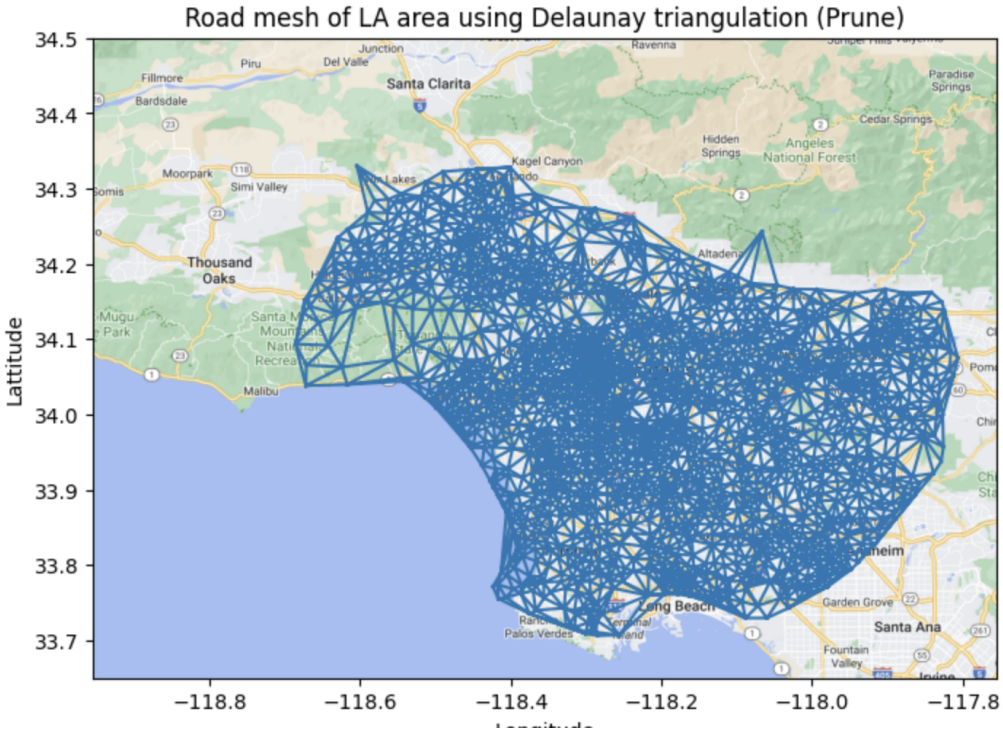
As the source shows the coordinates of Malibu and Long beach [34.04, -118.56] to [33.77, -118.18], applying the maxflow function from thoes two cities. We get value of 13089 cars/hour. The number of edge-disjoint paths between two spots is 6. The number of independent road is 4.





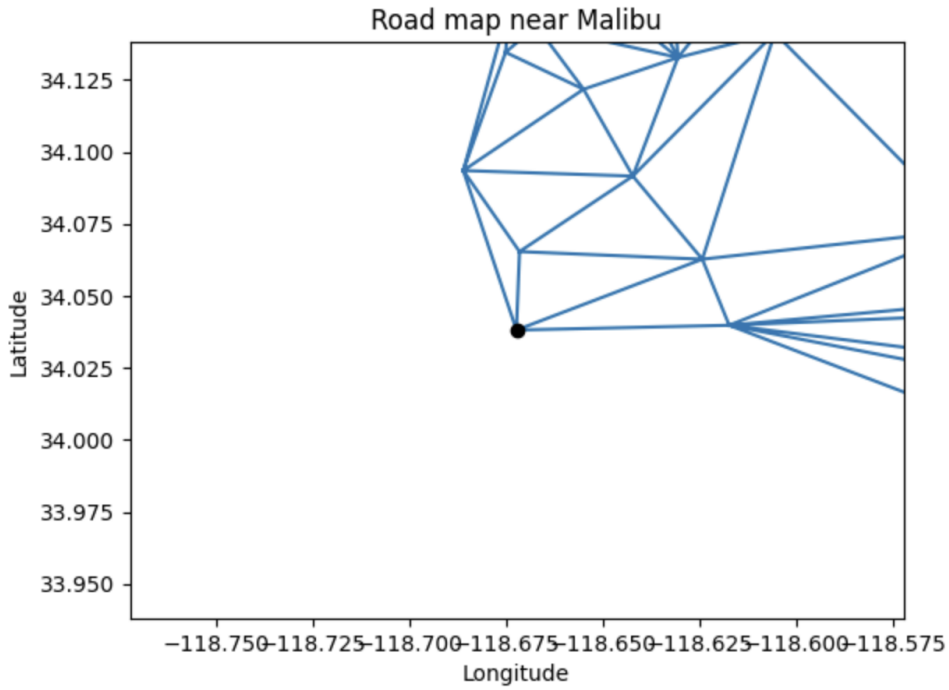
From the graph, we can see that both figure has the 6 edge that connected to the city which matches the result from what we calculate earlier.

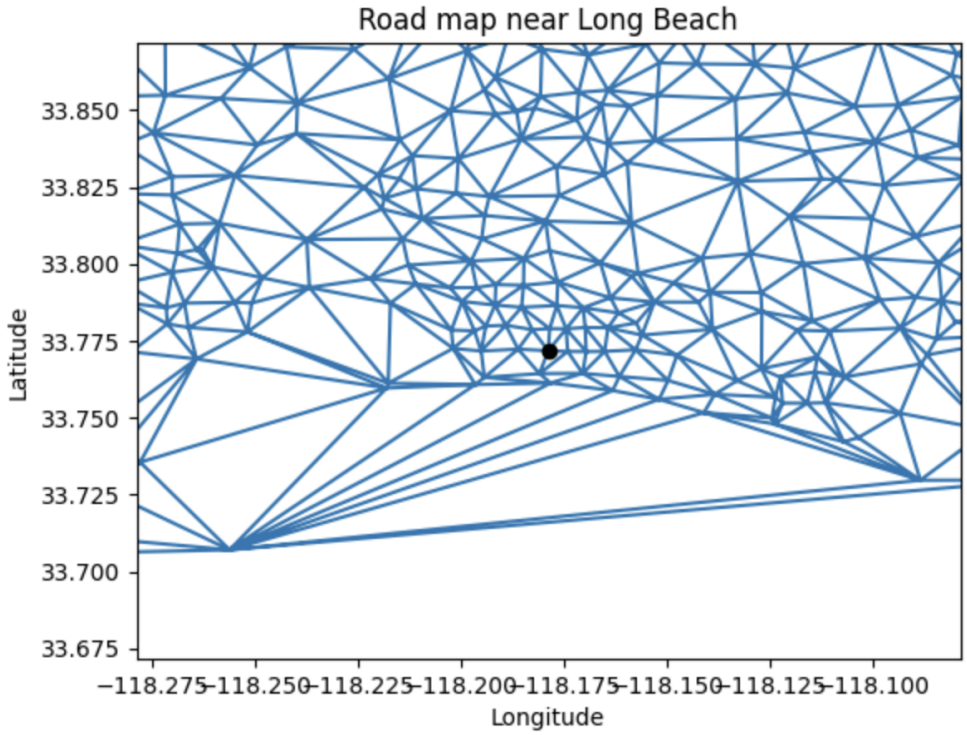
Question 17:



In the G_{Δ} , there are some unreal roads that can be removed, such as some connections that along the concavities of the beach and mountains. Then, we will apply threshold method on the travel, to remove this roads. We apply threshold on the travel time in In the G_{Δ}. This will remove the edge that Delaunay triangulation makes fit the situation. We also add information that points are close but have large travel time needed to be removed to make it precise. The graph above shows the $\tilde{G}_{\Delta}$.

Question 18:





After the pruning, we can see the road map near Malibu the connection decrease from 6 to 4. The result comes from the cut of 2 unreal deges. However, the road map near Long Beach remains the same. So, we take the minimum of two degrees is the number of edge disjoint path between those two cities. Thus, the number of edge disjoint path is 4. The maximum flow is still 13095 cars/hour. The car flow remains the same.

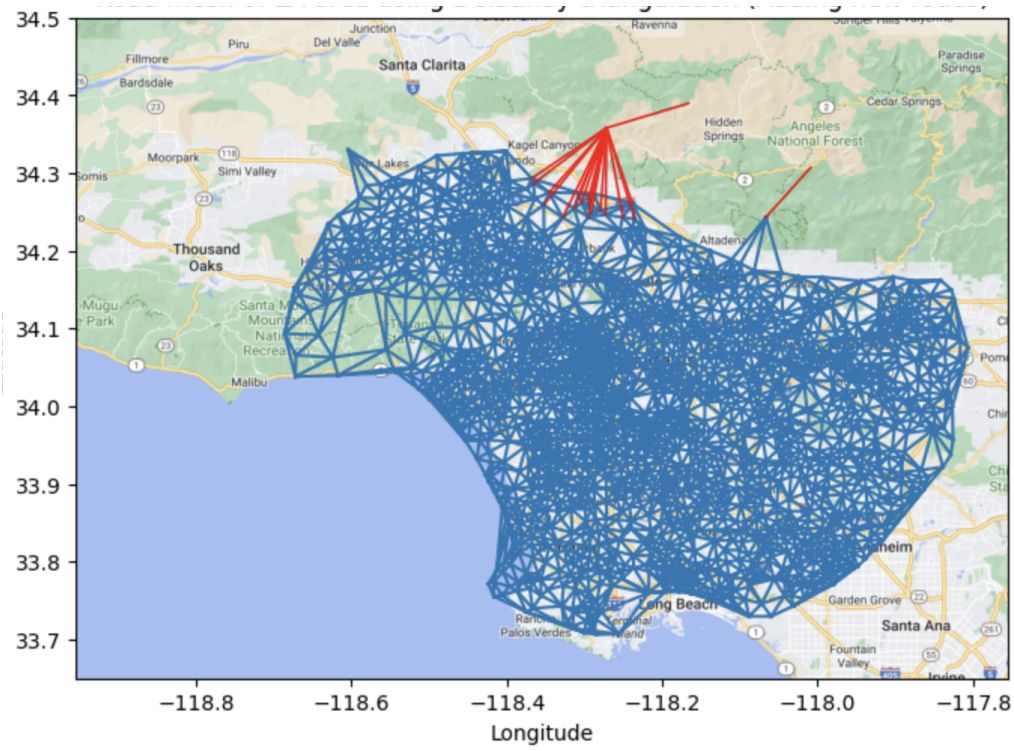
Question 19:

We do this strategy by using euclidean distance from two points in the 2-dimension map. We found the shortest path distance by dijkstra algorithm. Extra distance is subtracting euclidean distance from the the shortest path distance. We sort it with the top 20 paths then added it to the pruning graph in question 17, then we get the graph.

New edges are shown as below:

```
(-118.06647650215331, 34.24297495004306) - (-118.00882743874644, 34.30687687852888)
(-118.28809007499999, 34.2733324) - (-118.2715898179769, 34.357824822485206)
(-118.3069377237569, 34.2741446519337) - (-118.2715898179769, 34.357824822485206)
(-118.29409608227849, 34.26960164556962) - (-118.2715898179769, 34.357824822485206)
(-118.294821125, 34.26008425) - (-118.2715898179769, 34.357824822485206)
(-118.30412230000002, 34.2614321) - (-118.2715898179769, 34.357824822485206)
(-118.23864555555555, 34.26111577777778) - (-118.2715898179769, 34.357824822485206)
(-118.27952048148147, 34.25166522222222) - (-118.2715898179769, 34.357824822485206)
(-118.29105503125, 34.250763375000005) - (-118.2715898179769, 34.357824822485206)
 (-118.16620295644891, 34.38948489307649) - (-118.2715898179769, 34.357824822485206)
 (-118.34566161825725, 34.27462273858921) - (-118.2715898179769, 34.357824822485206)
 (-118.27287258260868, 34.246263686956524) - (-118.2715898179769, 34.357824822485206)
 (-118.31342583544303, 34.25429406329114) - (-118.2715898179769, 34.357824822485206)
 (-118.36819810798123, 34.2928966971831) - (-118.2715898179769, 34.357824822485206)
 (-118.25063870103092, 34.24278522680412) - (-118.2715898179769, 34.357824822485206)
 (-118.29481985915491, 34.24310354460094) - (-118.2715898179769, 34.357824822485206)
 (-118.36580732142855, 34.28615801190476) - (-118.2715898179769, 34.357824822485206)
 (-118.23560427067669, 34.24066394736842) - (-118.2715898179769, 34.357824822485206)
 (-118.35250637979095, 34.258879146341464) - (-118.2715898179769, 34.357824822485206)
 (-118.32820281690141, 34.24247783450704) - (-118.2715898179769, 34.357824822485206)
```

The new edges added in the graph is shown below:



The time complexity of this strategy is O(n^3), since for dijkstra is O(n^2) and goes over all the node O(n).

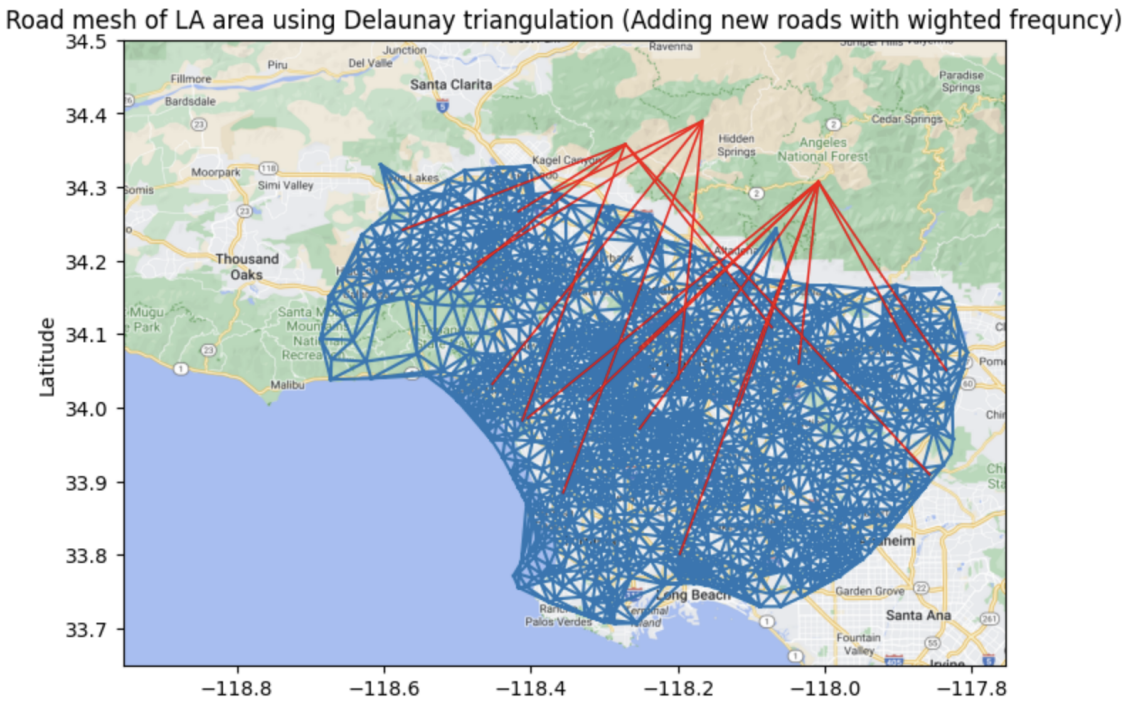
Question 20:

Now, we consider the frequency in the real world problem. We compute the extra distance as the same method as question 19, then we time a random number as a frequency, then we find the top 20 new edges.

New edges are shown as below:

```
(-118.4168469, 34.2665133) - (-118.2715898179769, 34.357824822485206)
(-118.03553834375, 34.059989625) - (-118.00882743874644, 34.30687687852888)
(-118.19966358333335, 34.03817975) - (-118.16620295644891, 34.38948489307649)
(-118.47187526666669, 34.197441533333325) - (-118.16620295644891, 34.38948489307649)
(-118.41160800000002, 33.982572) - (-118.2715898179769, 34.357824822485206)
(-118.32179651724137, 34.010221620689656) - (-118.00882743874644, 34.30687687852888)
(-118.35617900000001, 33.884189551724134) - (-118.16620295644891, 34.38948489307649)
(-117.89096249122807, 34.090259157894735) - (-118.00882743874644, 34.30687687852888)
(-118.11723058490568, 34.00257496226415) - (-118.00882743874644, 34.30687687852888)
 (-118.25204867999999, 33.971206239999994) - (-118.00882743874644, 34.30687687852888)
 (-118.45259354166667, 34.031775875) - (-118.16620295644891, 34.38948489307649)
 (-118.40441693650791, 33.98575915873016) - (-118.00882743874644, 34.30687687852888)
 (-118.19738804545455, 33.80092936363637) - (-118.00882743874644, 34.30687687852888)
 (-118.2715898179769, 34.357824822485206) - (-117.8573798627451, 33.90897262745098)
 (-118.36819810798123, 34.2928966971831) - (-118.16620295644891, 34.38948489307649)
 (-117.8348265, 34.0507507) - (-118.00882743874644, 34.30687687852888)
 (-118.51037320754718, 34.16061260377358) - (-118.2715898179769, 34.357824822485206)
 (-118.07216047272726, 34.10883718181818) - (-118.2715898179769, 34.357824822485206)
 (-118.57437450724638, 34.24140673913044) - (-118.2715898179769, 34.357824822485206)
 (-118.25184176056335, 34.0809844084507) - (-118.00882743874644, 34.30687687852888)
```

The new edges added in the graph is shown below:



The time complexity remain the same as question 19 O(n^3), since we only times frequency overall, which is a constant, it will not effect the time complexity.

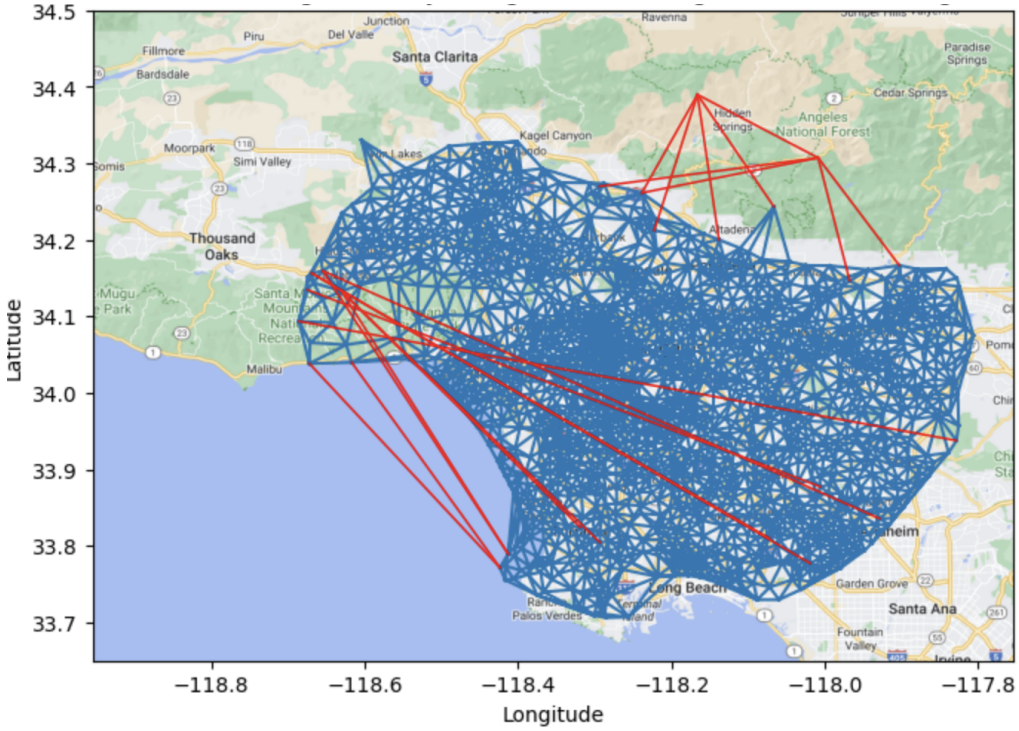
Question 21:

In this question, we create road one by one to avoid biulding unnecessary roads. We calculate each distance, then select the largest value of extra distance. In this strategy when we add the second edge, the prune graph is used by the previous edge 1 added, so it is sort of efficient in some way.

New edges are shown as below:

```
(-118.16620295644891, 34.38948489307649) - (-118.00882743874644, 34.30687687852888)
(-118.13817560000001, 34.19991316666667) - (-118.16620295644891, 34.38948489307649)
(-117.90283844444444, 34.16626599999999) - (-118.00882743874644, 34.30687687852888)
(-118.67225393333334, 34.03809095384616) - (-118.4237928111111, 33.77153057777778)
(-118.16620295644891, 34.38948489307649) - (-118.06647650215331, 34.24297495004306)
(-118.22248939694657, 34.21204154198474) - (-118.16620295644891, 34.38948489307649)
(-117.967588325, 34.14610455) - (-118.00882743874644, 34.30687687852888)
(-118.29409608227849, 34.26960164556962) - (-118.00882743874644, 34.30687687852888)
(-118.654899815534, 34.15918811650486) - (-118.413078828125, 33.789391734375)
 (-118.63048660074628, 34.132715593283585) - (-118.413078828125, 33.789391734375)
 (-118.6172417031802, 34.0397577491166) - (-118.4237928111111, 33.77153057777778)
 (-118.23864555555555, 34.26111577777778) - (-118.00882743874644, 34.30687687852888)
 (-118.66929280689655, 34.156163006896556) - (-118.01895590000002, 33.77716147)
 (-118.23864555555555, 34.26111577777778) - (-118.16620295644891, 34.38948489307649)
 (-118.66929280689655, 34.156163006896556) - (-118.07238421052632, 33.80988281578947)
 (-118.63048660074628, 34.132715593283585) - (-118.29324107547168, 33.804365245283016)
 (-118.63048660074628, 34.132715593283585) - (-118.31912093243241, 33.824276337837844)
 (-118.654899815534, 34.15918811650486) - (-117.92849649275362, 33.83554923188406)
 (-118.68599468727274, 34.093475498181824) - (-117.82852631818182, 33.93756910909091)
 (-118.67506014705882, 34.13454434491979) - (-118.00566880281691, 33.87708309859155)
```

The new edges added in the graph is shown below:



The time complexity of this strategy is O(n^3)

Question 22:

Now, we consider travel time. We still consider the euclidean distance, and calculate the extra time between all pair points. We select top 20 pairs with highest extra time which will be the new edge we will create.

New edges are shown as below:

(-118.16620295644891, 34.38948489307649) - (-118.2715898179769, 34.357824822485206)

(-118.23864555555555, 34.26111577777778) - (-118.16620295644891, 34.38948489307649)

(-118.23560427067669, 34.24066394736842) - (-118.16620295644891, 34.38948489307649)

(-118.28809007499999, 34.2733324) - (-118.16620295644891, 34.38948489307649)

(-118.25063870103092, 34.24278522680412) - (-118.16620295644891, 34.38948489307649)

(-118.227015, 34.230051875) - (-118.16620295644891, 34.38948489307649)

(-118.22109104455444, 34.227206539603955) - (-118.16620295644891, 34.38948489307649)

(-118.17811522222223, 34.21819922222222) - (-118.16620295644891, 34.38948489307649)

(-118.22744216666668, 34.22804516666667) - (-118.16620295644891, 34.38948489307649)

(-118.29409608227849, 34.26960164556962) - (-118.16620295644891, 34.38948489307649)

(-118.23833125, 34.22961990277778) - (-118.16620295644891, 34.38948489307649)

(-118.22793875555558, 34.223894977777775) - (-118.16620295644891, 34.38948489307649)

(-118.16620295644891, 34.38948489307649) - (-118.06647650215331, 34.24297495004306)

(-118.16620295644891, 34.38948489307649) - (-118.00882743874644, 34.30687687852888)

(-118.27952048148147, 34.25166522222222) - (-118.16620295644891, 34.38948489307649)

(-118.18650922371968, 34.21215386522911) - (-118.16620295644891, 34.38948489307649)

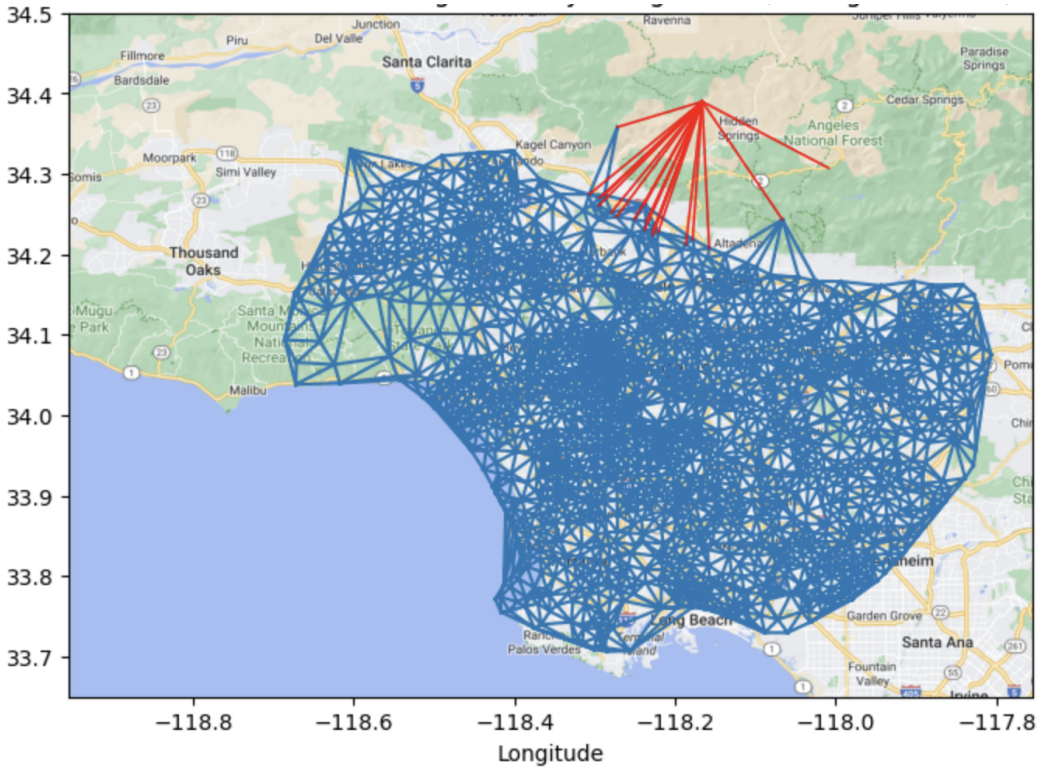
(-118.27287258260868, 34.246263686956524) - (-118.16620295644891, 34.38948489307649)

(-118.3069377237569, 34.2741446519337) - (-118.16620295644891, 34.38948489307649)

(-118.15749682068964, 34.20758387586206) - (-118.16620295644891, 34.38948489307649)

(-118.294821125, 34.26008425) - (-118.16620295644891, 34.38948489307649)

The new edges added in the graph is shown below:



The time complexity is O(n^3)

Question 23:

This strategy focus on extra time and shortest distance.

New edges are shown as below:

(-118.44574180681819, 34.325703420454545) - (-118.16620295644891, 34.38948489307649)

(-117.93646313636363, 34.067291465909086) - (-118.16620295644891, 34.38948489307649)

(-118.48561034042555, 33.99744193617021) - (-118.16620295644891, 34.38948489307649)

(-118.22735769101122, 34.12860570786516) - (-118.16620295644891, 34.38948489307649)

(-118.3204679142857, 33.935302057142856) - (-118.16620295644891, 34.38948489307649)

(-118.19409419607845, 33.815737078431376) - (-118.16620295644891, 34.38948489307649)

(-118.20220349132948, 34.156902803468206) - (-118.16620295644891, 34.38948489307649)

(-118.30410033333334, 34.093657) - (-118.16620295644891, 34.38948489307649)

(-118.07620971428571, 33.98477895238096) - (-118.16620295644891, 34.38948489307649)

(-118.02783257142855, 34.04611904761905) - (-118.16620295644891, 34.38948489307649)

(-118.4342153, 34.28989852857143) - (-118.16620295644891, 34.38948489307649)

(-118.16620295644891, 34.38948489307649) - (-117.94735429545455, 33.86110629545454)

(-118.36653712500001, 34.22304028125) - (-118.16620295644891, 34.38948489307649)

(-118.47325324705885, 34.014270894117644) - (-118.16620295644891, 34.38948489307649)

(-118.29658078, 33.87776949999999) - (-118.16620295644891, 34.38948489307649)

(-118.28007700000002, 34.025107000000006) - (-118.16620295644891, 34.38948489307649)

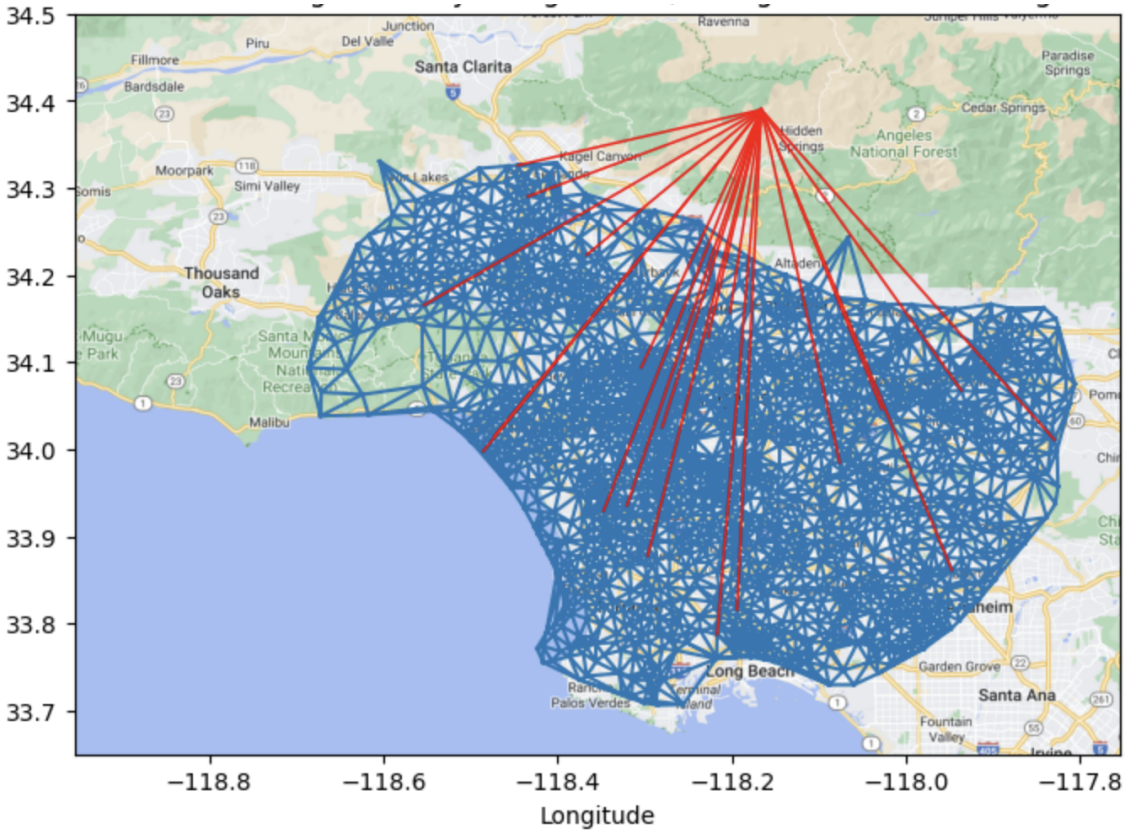
(-118.55339779069767, 34.16520172868217) - (-118.16620295644891, 34.38948489307649)

(-118.21696424999999, 33.787265562500004) - (-118.16620295644891, 34.38948489307649)

(-118.34736896226414, 33.92900001886793) - (-118.16620295644891, 34.38948489307649)

(-117.8296952037037, 34.01084702314815) - (-118.16620295644891, 34.38948489307649)

The new edges added in the graph is shown below:



Time complexity is O(n^3)

Question 24:

a) The difference between strategy 1 and 2 is that strategy 2 is that 2 times frequency which is more likely approached to the real life situation because we have to consider the road using status. Strategy 1 is also saving a lot distance if we do not consider the frequency which will be benefit to people who are living around the new road area. However, strategy 2 will be benefit to more people since the frequency. Thus, considering in the real life, strategy 2 might be better.

b) The difference between strategy 1 and 3 is that strategy 3 is like a dynamic method. When we build the second new road, it will consider the first added earlier new road as previous prune map as a whole, then calculate the top extra distance. From the graph we implemented we can see that, the new road is not focused in one area like strategy 1, it is more diversed. However, considering time consuming, strategy 3 might take a lot time. If we wanna make a better road map without considering time consuming, strategy 3 might be better.

c) The difference between strategy 1 and 4 is that strategy 4's travel time as a consideration which is important for most traveller. In this strategy, new edge is added by most time consuming during travelling. In real life, if we consider time travelling as a key parameter, then strategy 4 is making more sense that only considering eclidean distance. Thus, strategy 4 might be better.

d) From the graph we get from thoes strategies, we can see that dynamic method get some ocean edge and static get more like mountain area. But in real life, only one or two road will be selected because it will cost so much not only in money but time. So we might use a combination of those two method to optimize the options. The time complexity will be O(n^3)
e) We definitely need to consider time travelling as a key parameter, thus we need to do what strategy 4 or 5 do. We see that strategy 3 got diversed road coverage than strategy 1 so we might combined strategy 3 and 4 as our new method, we both consider time travelling and road coverage. The time complexity for this method will be O(n^3)

Question 25: Define My Own Task: Parking Recommender

In the plan to assist Santa, we have already aided him in expediting his deliveries. However, during the delivery process, an important consideration is how Santa can select parking for his vehicle. Therefore, we need to help Santa design a parking recommender. In this model, Santa can input the delivery address, and the model will recommend specific parking spot locations, fees, and maximum allowed parking duration. Moreover, since Santa Claus is delivering gifts to us without charge, in addition to the distance from the destination, this model also recommends the lowest-cost parking spots to Santa.

In this context, we obtain parking lot information about the LA area from the LA Express Park API (https://data.lacity.org/Transportation/LADOT-Metered-Parking-Inventory-Policies/s49e-q6j2)and extract and preprocess necessary features. The features include the parking spot's address, longitude and latitude, hourly fee, and allowable parking time. Then, based on these data, we designed two models. The first one is a recommendation system based on the KNN (K-Nearest Neighbors) algorithm. This model uses longitude and latitude coordinates to predict the unique identifier of a parking spot (spaceid). We chose k=5 in this model, and in the end, we obtained the five parking spots closest to the delivery location. The second model aims to find the nearest and lowest-cost parking spot based on the delivery location and parking time needs. We use geodesic from geopy.distance in Python to calculate the distance from the delivery location to each parking spot (geodesic is an algorithm to calculate the shortest distance between two points on Earth using their longitude and latitude coordinates). Then, we filter parking spots based on time constraints and calculate a score for each parking spot based on distance and fee. Finally, we sort parking spots by this score and choose the one with the lowest score as the recommendation. The scoring takes distance and fee into equal consideration, assigning them equal weights. If Santa Claus thinks one factor is more important, he can adjust their weights. For example, if Santa values distance more, he can assign a higher weight to distance, like 0.7, and a lower weight to fee, like 0.3.

Let's use my home address as an example. Suppose Santa is coming to deliver gifts at my home. I live in a flat called Apex The One, an apartment in Downtown LA. The longitude and latitude coordinates of the apartment are (34.04571, -118.26264). Using these coordinates as the model's input, we get the addresses and coordinates of five parking spots from the first KNN model.

```
Recommended parking spots:
    spaceid           blockface     metertype ratetype      raterange  \
473  CB1619   900 S FIGUEROA ST  Single-Space      TOD  $5.00 - $6.00
682   CB101   800 S FIGUEROA ST  Single-Space     FLAT          $2.00
76   CB3375    601 W OLYMPIC BL  Single-Space      TOD  $0.50 - $2.00
819  CB1605  801 W OLYMPIC BLVD  Single-Space      TOD  $0.50 - $3.00
113  CB1374      1000 S HOPE ST  Single-Space      TOD  $0.50 - $2.00

    timelimit                                              latlng  \
473       4HR  {'latitude': '34.045558', 'longitude': '-118.2...
682       2HR  {'latitude': '34.046801', 'longitude': '-118.2...
76        4HR  {'latitude': '34.044039', 'longitude': '-118.2...
819       4HR  {'latitude': '34.045275', 'longitude': '-118.2...
113       4HR  {'latitude': '34.043305', 'longitude': '-118.2...
```

```
    :@computed_region_qz3q_ghft :@computed_region_k96s_3jcv  \
473                        23076                         547
682                        23078                         546
76                         23076                         547
819                        23076                         546
113                        23076                         547

    :@computed_region_tatf_ua23 :@computed_region_kqwf_mjcx  \
473                        1050                           9
682                        1050                           9
76                         1050                           9
819                        1050                           9
113                         773                           9

    :@computed_region_2dna_qi2s :@computed_region_ur2y_g4cx   latitude  \
473                          76                         NaN  34.045558
682                          76                         NaN  34.046801
76                           76                         NaN  34.044039
819                          76                         NaN  34.045275
113                          76                         NaN  34.043305

      longitude
473 -118.263285
682 -118.262100
76  -118.262023
819 -118.264731
113 -118.261947
```

Then, we mark these five parking spots on the map, as shown in Figure 1. Figure 2 is a zoomed-in version of Figure 1 and provides a more intuitive understanding of the location relationship between the five parking spots and my apartment.

Figure 1:

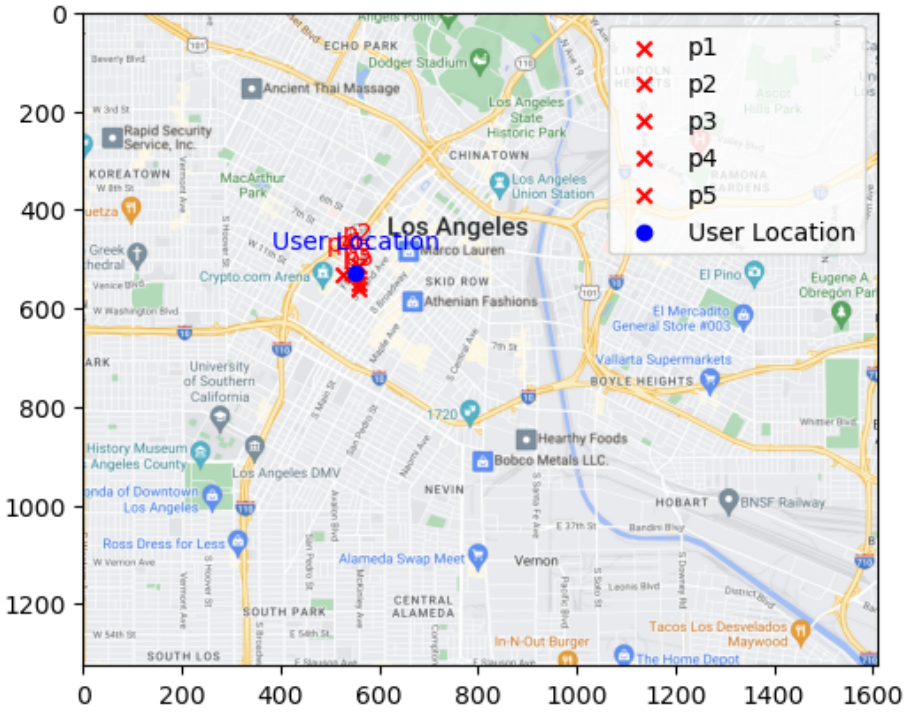
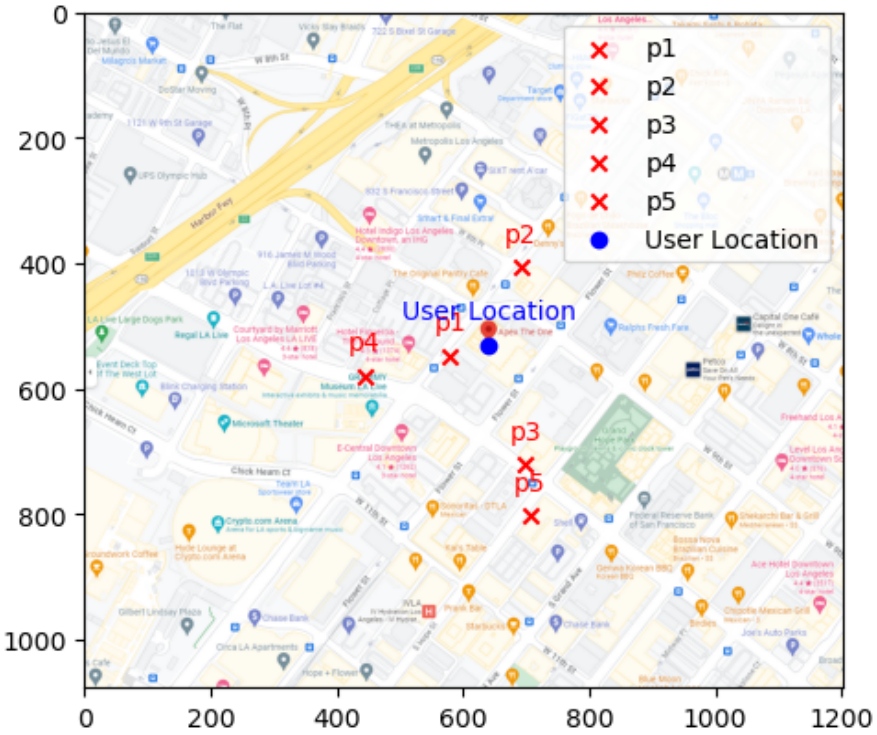


Figure 2:



Using the same apartment longitude and latitude coordinates as inputs for the second model, we find a parking spot that is the closest, has the lowest cost, and allows parking for more than two hours.

```
spaceid                                                          CB3375
blockface                                                601 W OLYMPIC BL
metertype                                                    Single-Space
ratetype                                                              TOD
raterange                                                  $0.50 - $2.00
timelimit                                                           240
latlng                    {'latitude': '34.044039', 'longitude': '-118.2...
:@computed_region_qz3q_ghft                                         23076
:@computed_region_k96s_3jcv                                           547
:@computed_region_tatf_ua23                                          1050
:@computed_region_kqwf_mjcx                                             9
:@computed_region_2dna_qi2s                                            76
:@computed_region_ur2y_g4cx                                           NaN
latitude                                                       34.044039
longitude                                                    -118.262023
rate                                                                 0.5
distance                                                        0.120491
score                                                           0.310245
Name: 76, dtype: object
```

We also mark these two locations on the map, as shown in Figures 3 and 4. This is the parking spot recommendation model we designed, hoping it can assist Santa.

Figure 3:

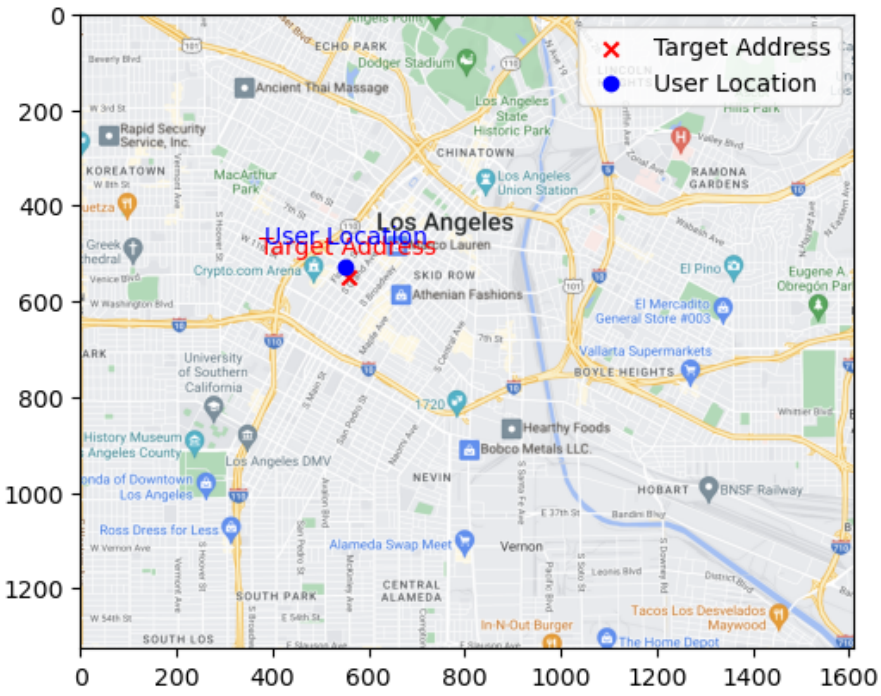


Figure 4: